

9.3

Przegląd techniczny produktu IBM MQ

IBM

Uwaga

Przed skorzystaniem z niniejszych informacji oraz produktu, którego one dotyczą, należy zapoznać się z informacjami zamieszczonymi w sekcji [“Uwagi” na stronie 319](#).

Niniejsze wydanie publikacji dotyczy wersji 9, wydania 3 produktu IBM® MQ oraz wszystkich jego późniejszych wydań i modyfikacji, aż do odwołania w nowych wydaniach publikacji.

Wysyłając informacje do IBM, użytkownik przyznaje IBM niewyłączne prawo do używania i rozpowszechniania informacji w dowolny sposób, jaki uzna za właściwy, bez żadnych zobowiązań wobec ich autora.

© **Copyright International Business Machines Corporation 2007, 2024.**

Spis treści

Przegląd techniczny.....	5
Wprowadzenie do kolejkowania komunikatów.....	5
Główne cechy i korzyści związane z kolejkowaniem komunikatów.....	7
Terminologia kolejkowania komunikatów.....	9
Komunikaty i kolejki.....	13
Obiekty IBM MQ.....	14
Typy obiektów.....	16
Nadawanie nazw obiektom IBM MQ.....	37
Rozproszone kolejkowanie i klastry.....	43
Komponenty kolejkowania rozproszonego.....	47
Komponenty klastra.....	57
Przesyłanie komunikatów w trybie publikowania/subskrypcji.....	63
Komponenty publikowania/subskrybowania.....	64
Przykład konfiguracji publikowania/subskrybowania pojedynczego menedżera kolejek.....	90
Rozproszone sieci publikowania/subskrypcji.....	91
IBM MQ Rozsyłanie grupowe.....	108
Początkowe pojęcia dotyczące rozsyłania grupowego.....	109
MQ TelemetryPrzegląd.....	110
MQ Telemetry - wprowadzenie.....	111
Przypadki użycia danych telemetrycznych.....	113
Łączenie urządzeń pomiarowych z menedżerem kolejek.....	119
Protokoły połączeń telemetrycznych.....	120
Usługa telemetryczna (MQXR).....	120
Kanały pomiarowe.....	121
IBM MQ Telemetry Transport :NONE.....	121
MQTT klienci.....	121
Wysyłanie komunikatu do klienta MQTT.....	122
Wysyłanie komunikatu do aplikacji IBM MQ z klienta MQTT.....	131
Aplikacje publikowania/subskrybowania MQTT.....	132
Aplikacje telemetryczne.....	133
Integracja programu MQ Telemetry z menedżerami kolejek.....	133
Sesje bezstanowe i stanowe MQTT.....	136
Gdy klient MQTT nie jest połączony.....	137
Luźne powiązanie między klientami MQTT i aplikacjami IBM MQ.....	137
Zabezpieczenia produktu MQ Telemetry.....	138
MQ Telemetry globalizacja.....	139
Wydajność i skalowalność produktu MQ Telemetry.....	139
Urządzenia obsługiwane przez MQ Telemetry.....	141
Zabezpieczenia w produkcie IBM MQ.....	142
Obsługa protokołu TLS zarządzanego klienta IBM MQ.NET.....	143
IBM MQ MQI clients.....	144
Dlaczego warto korzystać z klientów IBM MQ ?.....	146
Co to jest rozszerzony klient transakcyjny?.....	148
W jaki sposób klient łączy się z serwerem.....	149
Zarządzanie transakcjami i wsparcie.....	150
Rozszerzanie narzędzi menedżera kolejek.....	152
Interfejsy języka IBM MQ Java.....	153
IBM MQ classes for JMS/Jakarta Messaging.....	154
IBM MQ Dostawca komunikatów.....	165
IBM MQ for z/OS pojęcia.....	165
Menedżer kolejek w systemie z/OS.....	167
Inicjator kanału w systemie z/OS.....	168

Terminy i zadania związane z zarządzaniem produktem IBM MQ for z/OS.....	170
Kolejki współużytkowane i grupy współużytkowania kolejek.....	173
Kolejkowanie wewnątrz grupy.....	221
Zarządzanie pamięcią masową w systemie z/OS.....	234
logowanieIBM MQ for z/OS.....	239
Definicja systemu w systemie z/OS.....	250
Odtwarzanie i restartowanie w systemie z/OS.....	262
Pojęcia związane z bezpieczeństwem w produkcie IBM MQ for z/OS.....	280
Dostępność w systemie z/OS.....	287
Monitorowanie i statystyki w systemie IBM MQ for z/OS.....	290
Dyspozycja jednostki odzyskiwania w systemie z/OS.....	291
IBM MQ i inne produkty z/OS.....	294
IBM MQ i CICS.....	294
IBM MQ i IMS.....	296
IBM MQ i adaptery z/OS Batch, TSO i RRS.....	300
IBM MQ for z/OS i WebSphere Application Server.....	301
Managed File Transfer.....	302
W jaki sposób produkt MFT działa z produktem IBM MQ?.....	304
MFT przegląd topologii.....	304
Przegląd produktu MFTREST API.....	306
IBM MQ Internet Pass-Thru.....	306
Użycie MQIPT.....	307
Jak działa MQIPT.....	309
Możliwe konfiguracje produktu MQIPT.....	310
Zgodne konfiguracje.....	312
Obsługiwane konfiguracje kanałów.....	314
Warunki zakończenia i awarii kanału.....	315
Bezpieczeństwo komunikatów.....	315
Menedżery kolejek z wieloma instancjami i wysoka dostępność.....	315
IBM MQ Console i REST API.....	316
Uwagi.....	319
Informacje dotyczące interfejsu programistycznego.....	320
Znaki towarowe.....	321

IBM MQ Przegląd techniczny

Produkt IBM MQ umożliwia łączenie aplikacji i zarządzanie dystrybucją informacji w organizacji.

IBM MQ umożliwia programom komunikowanie się ze sobą przez sieć o różnych komponentach (procesorach, systemach operacyjnych, podsystemach i protokołach komunikacyjnych) za pomocą spójnego aplikacyjnego interfejsu programistycznego. Aplikacje zaprojektowane i napisane za pomocą tego interfejsu są nazywane aplikacjami kolejkowania komunikatów.

Poniższe podtematy zawierają informacje o kolejkowaniu komunikatów i innych funkcjach udostępnianych przez produkt IBM MQ.

Pojęcia pokrewne

[IBM MQ - wprowadzenie](#)

[Gdzie znaleźć wymagania produktu i informacje o wsparciu](#)

Zadania pokrewne

[Planowanie architektury IBM MQ](#)

Odsyłacze pokrewne

[“Główne cechy i korzyści związane z kolejkowaniem komunikatów” na stronie 7](#)

W tej sekcji przedstawiono niektóre funkcje i korzyści związane z kolejkowaniem komunikatów. Opisano w nim takie funkcje, jak bezpieczeństwo i integralność danych w kolejkowaniu komunikatów.

Wprowadzenie do kolejkowania komunikatów

Produkty IBM MQ umożliwiają programom komunikowanie się między sobą przez sieć różnych komponentów (procesorów, systemów operacyjnych, podsystemów i protokołów komunikacyjnych) za pomocą spójnego aplikacyjnego interfejsu programistycznego.

Aplikacje zaprojektowane i napisane przy użyciu tego interfejsu są nazywane aplikacjami *kolejkowania komunikatów*, ponieważ używają stylu *przesyłania komunikatów* i *kolejkowania*:

- Przesyłanie komunikatów oznacza, że programy komunikują się, wysyłając do siebie dane w postaci komunikatów, a nie wywołując się bezpośrednio.
- Kolejkowanie oznacza, że komunikaty są umieszczane w kolejkach w pamięci masowej, co umożliwia uruchamianie programów niezależnie od siebie, z różnymi szybkością i czasem, w różnych miejscach i bez logicznego połączenia między nimi.

Kolejkowanie komunikatów jest używane w przetwarzaniu danych od wielu lat. Jest najczęściej używany dzisiaj w poczcie elektronicznej. Bez kolejkowania wysyłanie komunikatu elektronicznego na duże odległości wymaga, aby każdy węzeł na trasie był dostępny do przesyłania komunikatów, a adresaci byli zalogowani i świadomi faktu, że użytkownik próbuje wysłać im komunikat. W systemie kolejkowania komunikaty są przechowywane w węzłach pośrednich, dopóki system nie będzie gotowy do ich przekazania. W miejscu przeznaczenia są one przechowywane w elektronicznej skrzynce pocztowej, dopóki adresat nie będzie gotowy do ich odczytania.

Mimo to wiele złożonych transakcji biznesowych jest przetwarzanych dzisiaj bez kolejkowania. W dużej sieci system może utrzymywać wiele tysięcy połączeń w stanie gotowym do użycia. Jeśli w jednej części systemu występuje problem, wiele części systemu staje się nieużytecznych.

Kolejkowanie komunikatów można traktować jako pocztę elektroniczną dla programów. W środowisku kolejkowania komunikatów każdy program, który stanowi część pakietu aplikacji, wykonuje dobrze zdefiniowaną, samodzielną funkcję w odpowiedzi na konkretne żądanie. Aby komunikować się z innym programem, program musi umieścić komunikat w predefiniowanej kolejce. Drugi program pobiera komunikat z kolejki i przetwarza żądania i informacje zawarte w komunikacie. Kolejkowanie komunikatów jest więc stylem komunikacji program-program.

Kolejkowanie jest mechanizmem, za pomocą którego komunikaty są wstrzymywane do momentu, gdy aplikacja będzie gotowa do ich przetwarzania. Kolejkowanie umożliwia:

- Komunikacja między programami (które mogą działać w różnych środowiskach) bez konieczności pisania kodu komunikacyjnego.
- Wybierz kolejność, w jakiej program przetwarza komunikaty.
- Równoważenie obciążeń w systemie przez zorganizowanie więcej niż jednego programu do obsługi kolejki, gdy liczba komunikatów przekroczy wartość progową.
- Zwiększ dostępność aplikacji, organizując alternatywny system do obsługi kolejek, jeśli system podstawowy jest niedostępny.

Co to jest kolejka komunikatów?

Kolejka komunikatów, zwana po prostu kolejką, jest nazwanym miejscem docelowym, do którego mogą być wysyłane komunikaty. Komunikaty są gromadzone w kolejkach, dopóki nie zostaną wczytane przez programy obsługujące te kolejki.

Kolejki rezydują w menedżerze kolejek i są przez niego zarządzane (patrz sekcja [“Terminologia kolejkowania komunikatów”](#) na stronie 9). Fizyczna natura kolejki zależy od systemu operacyjnego, w którym działa menedżer kolejek. Kolejka może być ulotnym obszarem buforu w pamięci komputera lub zestawem danych w urządzeniu pamięci trwałej (takim jak dysk). Fizyczne zarządzanie kolejkami należy do obowiązków menedżera kolejek i nie jest widoczne dla uczestniczących aplikacji.

Programy uzyskują dostęp do kolejek tylko za pośrednictwem usług zewnętrznych menedżera kolejek. Mogą oni otworzyć kolejkę, umieścić w niej komunikaty, pobrać z niej komunikaty i zamknąć kolejkę. Mogą również ustawiać i pytać o atrybuty kolejek.

Różne style kolejkowania komunikatów

Punkt-punkt

Jeden komunikat jest umieszczany w kolejce i jedna aplikacja odbiera ten komunikat.

W przypadku przesyłania komunikatów w trybie punkt z punktem aplikacja wysyłająca musi znać informacje o aplikacji odbierającej, zanim będzie mogła wysłać komunikat do tej aplikacji. Na przykład aplikacja wysyłająca może potrzebować nazwy kolejki, do której mają zostać wysłane informacje, a także nazwy menedżera kolejek.

Publikowanie/subskrypcja

Kopia każdego komunikatu publikowanego przez aplikację publikującą jest dostarczana do każdej zainteresowanej aplikacji. Może istnieć wiele, jedna lub żadna z zainteresowanych aplikacji. W przypadku publikowania/subskrybowania dana aplikacja jest nazywana subskrybentem, a komunikaty są umieszczane w kolejce identyfikowanej przez subskrypcję.

Przesyłanie komunikatów w trybie publikowania/subskrypcji umożliwia oddzielenie dostawcy informacji od konsumentów tych informacji. Aplikacja wysyłająca i aplikacja odbierająca nie muszą wiedzieć o sobie tak wiele, aby informacje były wysyłane i odbierane. Więcej informacji na ten temat zawiera sekcja [“Przesyłanie komunikatów w trybie publikowania/subskrypcji”](#) na stronie 63.

Zalety kolejkowania komunikatów do projektanta i programisty aplikacji

IBM MQ umożliwia aplikacjom korzystanie z *kolejkowania komunikatów* w celu uczestniczenia w przetwarzaniu sterowanym komunikatami. Aplikacje mogą komunikować się między różnymi platformami za pomocą odpowiednich produktów oprogramowania do kolejkowania komunikatów. Na przykład aplikacje z/OS mogą komunikować się za pośrednictwem serwera IBM MQ for z/OS. Aplikacje są osłonięte przed mechanikami bazowej komunikacji. Inne zalety kolejkowania komunikatów to:

- Aplikacje można projektować przy użyciu małych programów, które mogą być współużytkowane przez wiele aplikacji.
- Można szybko budować nowe aplikacje, ponownie wykorzystując te elementy składowe.
- Zmiany w sposobie działania menedżerów kolejek nie mają wpływu na aplikacje napisane w celu użycia technik kolejkowania komunikatów.

- Nie trzeba używać żadnych protokołów komunikacyjnych. Menedżer kolejek zajmuje się wszystkimi aspektami komunikacji.
- Programy, które odbierają komunikaty, nie muszą być uruchomione w momencie wysyłania do nich komunikatów. Komunikaty są zachowywane w kolejkach.

Projektanci mogą obniżyć koszty swoich aplikacji, ponieważ programowanie jest szybsze, potrzeba mniej programistów, a wymagania dotyczące umiejętności programowania są niższe niż w przypadku aplikacji, które nie korzystają z kolejkowania komunikatów.

Produkt IBM MQ implementuje wspólny aplikacyjny interfejs programistyczny znany jako *interfejs kolejki komunikatów* (lub MQI) wszędzie tam, gdzie działają aplikacje. Ułatwia to przeniesienie aplikacji z jednej platformy na inną.

Szczegółowe informacje na temat interfejsu MQI zawiera sekcja [Przegląd interfejsu kolejki komunikatów](#).

Główne cechy i korzyści związane z kolejkowaniem komunikatów

W tej sekcji przedstawiono niektóre funkcje i korzyści związane z kolejkowaniem komunikatów. Opisano w nim takie funkcje, jak bezpieczeństwo i integralność danych w kolejkowaniu komunikatów.

Główne funkcje aplikacji, które używają technik kolejkowania komunikatów, to:

- [“Brak bezpośrednich połączeń między programami” na stronie 7](#)
- [“Komunikacja niezależna od czasu” na stronie 8](#)
- [“Małe programy” na stronie 8](#)
- [“Przetwarzanie sterowane komunikatami” na stronie 8](#)
- [“Przetwarzanie sterowane zdarzeniami” na stronie 8](#)
- [“Priorytet komunikatu” na stronie 9](#)
- [“Zabezpieczenia” na stronie 9](#)
- [“Integralność danych” na stronie 9](#)
- [“Obsługa odtwarzania” na stronie 9](#)

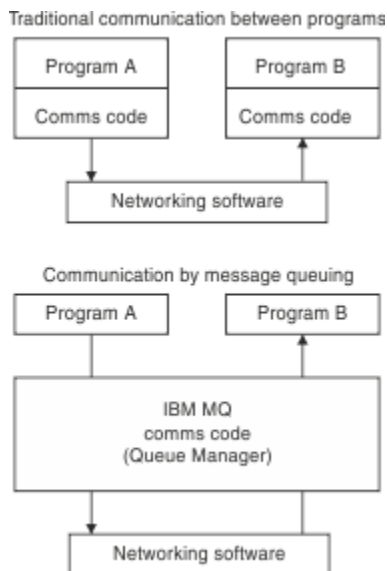
Uwaga: Podczas rozważania klientów i serwerów IBM MQ nie trzeba zmieniać aplikacji serwera w celu obsługi dodatkowego systemu IBM MQ MQI clients na nowych platformach. Podobnie IBM MQ MQI client może, bez zmian, działać z dodatkowymi typami serwerów.

Brak bezpośrednich połączeń między programami

Kolejkowanie komunikatów jest techniką pośredniej komunikacji program-program. Można go używać w dowolnej aplikacji, w której programy komunikują się ze sobą. Komunikacja odbywa się przez jeden program umieszczający komunikaty w kolejce (należy do menedżera kolejek) i inny program pobierający komunikaty z kolejki.

Programy mogą pobrać komunikaty, które zostały umieszczone w kolejce przez inne programy. Inne programy mogą być połączone z tym samym menedżerem kolejek co program odbierający lub z innym menedżerem kolejek. Ten inny menedżer kolejek może znajdować się w innym systemie, innym systemie komputerowym lub nawet w innej firmie lub przedsiębiorstwie.

Nie ma fizycznych połączeń między programami, które komunikują się za pomocą kolejek komunikatów. Program wysyła komunikaty do kolejki, której właścicielem jest menedżer kolejek, a inny program pobiera komunikaty z kolejki (patrz sekcja [Rysunek 1 na stronie 8](#)).



Rysunek 1. Kolejowanie komunikatów w porównaniu z tradycyjną komunikacją

Podobnie jak w przypadku poczty elektronicznej, pojedyncze wiadomości, które są częścią transakcji, są przesyłane przez sieć w ramach przechowywania i przekazywania. podstawy. Jeśli połączenie między węzłami nie powiedzie się, komunikat jest zachowywany do czasu odtworzenia połączenia lub do czasu przekierowania komunikatu przez operatora lub program.

Mechanizm, za pomocą którego komunikat jest przenoszony z kolejki do kolejki, jest ukryty przed programami. Dlatego programy są prostsze.

Komunikacja niezależna od czasu

Programy żądające wykonania pracy przez inne osoby nie muszą czekać na odpowiedź na żądanie. Mogą wykonywać inną pracę i przetwarzać odpowiedź po jej nadejściu lub później. Podczas pisania aplikacji do przesyłania komunikatów nie trzeba wiedzieć (lub być zaniepokojonym), kiedy program wysyła komunikat lub kiedy cel może odebrać komunikat. Komunikat nie zostanie utracony. Jest on zachowywany przez menedżer kolejek do czasu, aż element docelowy będzie gotowy do przetworzenia. Komunikat pozostaje w kolejce, dopóki nie zostanie usunięty przez program. Oznacza to, że aplikacje wysyłające i odbierające są oddzielone; nadawca może kontynuować przetwarzanie bez oczekiwania na potwierdzenie odbioru komunikatu przez odbiorcę. Aplikacja docelowa nie musi być nawet uruchomiona po wysłaniu komunikatu. Może on pobrać komunikat po jego uruchomieniu.

Małe programy

Kolejkowanie komunikatów umożliwia korzystanie z zalet używania małych, samodzielnych programów. Zamiast pojedynczego, dużego programu wykonującego wszystkie części zadania sekwencyjnie, można rozłożyć zadanie na kilka mniejszych, niezależnych programów. Program wysyłający żądanie wysyła komunikaty do każdego z oddzielnych programów, prosząc o wykonanie ich funkcji; po zakończeniu każdego programu wyniki są wysyłane z powrotem jako jeden lub więcej komunikatów.

Przetwarzanie sterowane komunikatami

Po nadejściu komunikatów do kolejki mogą one automatycznie uruchomić aplikację za pomocą *wyzwalania*. W razie potrzeby aplikacje mogą zostać zatrzymane po przetworzeniu komunikatu (lub komunikatów).

Przetwarzanie sterowane zdarzeniami

Programy mogą być kontrolowane w zależności od stanu kolejek. Na przykład można określić, że program ma zostać uruchomiony natychmiast po pojawieniu się komunikatu w kolejce, lub można określić, że

program nie zostanie uruchomiony, dopóki nie będzie na przykład 10 komunikatów powyżej określonego priorytetu w kolejce lub 10 komunikatów o dowolnym priorytecie w kolejce.

Priorytet komunikatu

Program może przypisać priorytet do komunikatu, gdy umieszcza go w kolejce. Określa to pozycję w kolejce, do której dodawany jest nowy komunikat.

Programy mogą pobrać komunikaty z kolejki w kolejności, w jakiej znajdują się one w kolejce, lub pobierając konkretny komunikat. (Program może chcieć otrzymać konkretny komunikat, jeśli szuka odpowiedzi na wystane wcześniej żądanie).

Zabezpieczenia

Dostępne są funkcje zabezpieczeń, w tym uwierzytelnianie aplikacji podczas korzystania z menedżera kolejek, sprawdzanie autoryzacji podczas korzystania z zasobów, takich jak kolejka w menedżerze kolejek oraz szyfrowanie danych komunikatów podczas ich przesyłania przez sieć, a także podczas rezydowania w kolejkach. Więcej informacji na temat zabezpieczeń zawiera sekcja [Przegląd zabezpieczeń](#).

Integralność danych

Integralność danych jest zapewniana przez jednostki pracy. Synchronizacja początku i końca jednostek pracy jest w pełni obsługiwana jako opcja dla każdej operacji MQGET lub MQPUT, co umożliwia zatwierdzenie lub wycofanie wyników jednostki pracy. Obsługa punktów synchronizacji działa wewnętrznie lub zewnętrznie względem produktu IBM MQ w zależności od formy koordynacji punktów synchronizacji wybranej dla aplikacji.

Obsługa odtwarzania

Aby odtwarzanie było możliwe, wszystkie trwałe aktualizacje IBM MQ są rejestrowane. Jeśli odtwarzanie jest konieczne, wszystkie komunikaty trwałe są odtwarzane, wszystkie transakcje w trakcie przetwarzania są wycofywane, a wszelkie zatwierdzenia i wycofania punktu synchronizacji są obsługiwane w normalny sposób przez menedżera punktów synchronizacji, który sprawuje kontrolę. Więcej informacji na temat trwałych komunikatów zawiera sekcja [Trwałość komunikatu](#).

Terminologia kolejkowania komunikatów

Informacje te umożliwiają zapoznanie się z terminami używanymi w kolejkowaniu komunikatów.

Obejmują:

- [Kanały](#)
- [Klaster](#)
- [IBM MQ MQI client](#)
-  [Kolejkowanie wewnątrz grupy](#)
- [Komunikat](#)
- [Agent kanału komunikatów](#)
- [deskryptor komunikatu](#)
- [Punkt-punkt](#)
- [Publikowanie/subskrypcja](#)
- [Kolejka](#)
- [menedżer kolejek](#)
-  [Grupa współużytkowania kolejek](#)
-  [Kolejka współużytkowana](#)

- [Subskrypcja](#)
- [Temat](#)

Kanały

Kanały są używane do przenoszenia komunikatów z jednego menedżera kolejek do innego i chronią aplikacje przed bazowymi protokołami komunikacyjnymi. Menedżery kolejek mogą istnieć w tym samym systemie, w różnych systemach na tej samej platformie lub na różnych platformach. Wysyłane wiadomości mogą pochodzić z wielu miejsc:

- Aplikacje napisane przez użytkownika, które przesyłają dane z jednego węzła do drugiego.
- Napisane przez użytkownika aplikacje administracyjne, które używają komend PCF lub interfejsu MQAI.
- IBM MQ Explorer.
- Menedżery kolejek, które wysyłają komunikaty zdarzeń instrumentacji do innego menedżera kolejek.
- Menedżery kolejek, które wysyłają zdalne komendy administracyjne do innego menedżera kolejek. Na przykład za pomocą komend MQSC lub administrative REST API.

Więcej informacji na temat kanałów zawiera sekcja [“Definicje kanałów”](#) na stronie 32.

Klaster

Klaster jest siecią menedżerów kolejek, które są w pewien sposób logicznie powiązane.

W sieci produktu IBM MQ korzystającej z rozproszonego kolejkowania bez klastrowania każdy menedżer kolejek jest niezależny. Jeśli jeden menedżer kolejek musi wysłać komunikaty do innego menedżera kolejek, musi mieć zdefiniowaną kolejkę transmisji i kanał do zdalnego menedżera kolejek.

Istnieją dwie różne przyczyny używania klastrów: ograniczenie administrowania systemem i zwiększenie dostępności oraz zrównoważenie obciążenia.





Po ustanowieniu nawet najmniejszego skupienia można skorzystać z uproszczonego administrowania systemem. Menedżery kolejek będące częścią klastra wymagają mniejszej liczby definicji, więc ryzyko popełnienia błędu w definicjach jest mniejsze.

Więcej informacji na temat grupowania zawiera sekcja [Klastry](#).

IBM MQ MQI client

IBM MQ Klienci *MQI* są niezależnie instalowanymi komponentami produktu IBM MQ. Klient MQI umożliwia uruchamianie aplikacji systemu IBM MQ z protokołem komunikacyjnym, interakcję z jednym lub większą liczbą serwerów interfejsu kolejki komunikatów (Message Queue Interface-MQI) na innych platformach oraz nawiązywanie połączeń z ich menedżerami kolejek.

Szczegółowe informacje na temat instalowania i używania komponentów produktu IBM MQ MQI client można znaleźć w następujących tematach:

-  [Instalowanie klienta IBM MQ w systemie AIX](#)
-  [Instalowanie klienta IBM MQ w systemie Linux®](#)
-  [Instalowanie klienta IBM MQ w systemie Windows](#)
-  [Instalowanie klienta IBM MQ w systemie IBM i](#)

i [Konfigurowanie połączeń między serwerem i klientem](#).

Kolejkowanie wewnątrz grupy



Menedżery kolejek w grupie współużytkowania kolejek mogą komunikować się przy użyciu zwykłych kanałów lub można użyć techniki nazywanej *kolejkowaniem wewnątrz grupy* (IGQ), która umożliwia szybkie przesyłanie komunikatów bez definiowania kanałów. Dotyczy to tylko systemu IBM MQ for z/OS.

Więcej informacji na temat kolejkowania wewnątrz grupy zawiera sekcja [“Kolejkowanie wewnątrz grupy” na stronie 221](#).

Komunikat

W kolejkowaniu komunikatów komunikat jest kolekcją danych wysłanych przez jeden program i przeznaczonych dla innego programu. Patrz [Komunikaty produktu IBM MQ](#).

Więcej informacji na temat typów komunikatów zawiera sekcja [Typy komunikatów](#).

Agent kanału komunikatów

Jednym z końców kanału jest agent kanału komunikatów. Para agentów kanału komunikatów, jeden wysyłający i jeden odbierający, tworzą kanał i przenoszą komunikaty z jednego menedżera kolejek do innego.

Informacje na temat sposobu używania agentów kanałów komunikatów zawiera sekcja [Wprowadzenie do rozproszonego zarządzania kolejkami](#).

deskryptor komunikatu

Komunikat IBM MQ składa się z informacji sterujących i danych aplikacji.

Informacje sterujące są definiowane w strukturze deskryptora komunikatu (MQMD) i zawierają takie elementy, jak:

- Typ komunikatu
- Identyfikator komunikatu
- Priorytet dostarczania komunikatu

Struktura i treść danych aplikacji są określane przez programy uczestniczące, a nie przez program IBM MQ.

Więcej informacji na ten temat zawiera sekcja [MQMD](#).

Przesyłanie komunikatów w modelu punkt-punkt

W przesyłaniu komunikatów w trybie punkt z punktem każdy komunikat jest przesyłanym z jednej aplikacji generującej komunikat do jednej aplikacji konsumującej. Komunikaty są przesyłane za pośrednictwem aplikacji generującej komunikaty umieszczającej je w kolejce, a aplikacja konsumująca pobiera je z tej kolejki.

Przesyłanie komunikatów w trybie publikowania/subskrypcji

W przypadku przesyłania komunikatów w trybie publikowania/subskrypcji kopia każdego komunikatu publikowanego przez aplikację publikującą jest dostarczana do każdej zainteresowanej aplikacji. Może istnieć wiele aplikacji, jedna lub żadna z nich. W przypadku publikowania/subskrybowania dana aplikacja jest nazywana subskrybentem, a komunikaty są umieszczane w kolejce identyfikowanej przez subskrypcję.

Więcej informacji na ten temat zawiera [“Przesyłanie komunikatów w trybie publikowania/subskrypcji” na stronie 63](#).

Kolejka

Nazwane miejsce docelowe, do którego mogą być wysyłane komunikaty. Komunikaty są gromadzone w kolejkach, dopóki nie zostaną wczytane przez programy obsługujące te kolejki.

Więcej informacji na ten temat zawiera [“Kolejki” na stronie 20](#).

Menedżer kolejek

Menedżer kolejek to program systemowy udostępniający aplikacjom usługi kolejkowania.

Udostępnia on aplikacyjny interfejs programistyczny, który umożliwia umieszczanie komunikatów w kolejkach i pobieranie komunikatów z kolejek. Menedżer kolejek udostępnia dodatkowe funkcje, dzięki którym administratorzy mogą tworzyć nowe kolejki, zmieniać właściwości istniejących kolejek i sterować działaniem menedżera kolejek.

Aby usługi kolejkowania komunikatów IBM MQ były dostępne w systemie, musi być uruchomiony menedżer kolejek. W jednym systemie może działać więcej niż jeden menedżer kolejek (na przykład w celu oddzielenia systemu testowego od *aktywnego* systemu). W przypadku aplikacji każdy menedżer kolejek jest identyfikowany przez *uchwyt połączenia* (*Hconn*).

Wiele różnych aplikacji może korzystać z usług menedżera kolejek w tym samym czasie i te aplikacje mogą być całkowicie niepowiązane. Aby program mógł korzystać z usług menedżera kolejek, musi nawiązać połączenie z tym menedżerem kolejek.

Aby aplikacje mogły wysyłać komunikaty do aplikacji połączonych z innymi menedżerami kolejek, muszą być w stanie komunikować się między sobą. IBM MQ implementuje protokół *store-and-forward* w celu zapewnienia bezpiecznego dostarczania komunikatów między takimi aplikacjami.

Więcej informacji na ten temat zawiera [“Menedżery kolejek” na stronie 29](#).

Grupa współużytkowania kolejek



Menedżery kolejek, które mogą uzyskać dostęp do tego samego zestawu kolejek współużytkowanych, tworzą grupę nazywaną *grupą współużytkowania kolejek* (QSG). Komunikują się one ze sobą za pomocą narzędzia CF, które przechowuje kolejki współużytkowane. Dotyczy to tylko systemu IBM MQ for z/OS.

Więcej informacji na ten temat zawiera [“Kolejki współużytkowane i grupy współużytkowania kolejek” na stronie 173](#).

Kolejka współużytkowana



Kolejka współużytkowana to typ kolejki lokalnej z komunikatami, do której może uzyskać dostęp co najmniej jeden menedżer kolejek znajdujący się w syspleksie. Nie jest to taka sama kolejka, jak kolejka współużytkowana przez więcej niż jedną aplikację używającą tego samego menedżera kolejek. Dotyczy to tylko systemu IBM MQ for z/OS.

Subskrypcja

Aplikacja publikowania/subskrypcji może zarejestrować zainteresowanie komunikatami dotyczącymi konkretnych tematów. Gdy aplikacja wykonuje tę czynność, jest ona nazywana subskrybentem, a termin subskrypcja definiuje sposób, w jaki zgodne komunikaty są umieszczane w kolejce do przetwarzania.

Subskrypcja zawiera informacje o tożsamości subskrybenta oraz o tożsamości kolejki docelowej, w której mają zostać umieszczone publikacje. Zawiera również informacje o sposobie umieszczania publikacji w kolejce docelowej.

Więcej informacji na ten temat zawiera [“Subskrybenci i subskrypcje” na stronie 66](#).

Temat

Temat to łańcuch znaków służący do opisywania tematyki informacji, które zostaną opublikowane w ramach komunikatu publikowania/subskrypcji.

Tematy są kluczowym czynnikiem zapewniającym pomyślne dostarczanie komunikatów w ramach systemu publikowania/subskrypcji. Publikator nie wprowadza konkretnego adresu docelowego w przypadku każdego komunikatu, lecz przypisuje temat do komunikatu. Menedżer kolejek dopasowuje temat do listy subskrybentów tego tematu i dostarcza ten komunikat do każdego z nich.

Więcej informacji na ten temat zawiera [“Tematy” na stronie 69](#).

Komunikaty i kolejki

Komunikaty i kolejki są podstawowymi komponentami systemu kolejkowania komunikatów.

Co to jest wiadomość?

Komunikat to łańcuch bajtów zrozumiały dla aplikacji, które z niego korzystają. Komunikaty są używane do przesyłania informacji z jednej aplikacji do innej (lub między różnymi częściami tej samej aplikacji). Aplikacje mogą działać na tej samej platformie lub na różnych platformach.

Komunikat IBM MQ składa się z:

- *Dane aplikacji*. Treść i struktura danych aplikacji są definiowane przez aplikacje, które z nich korzystają.
- *Deskryptor komunikatu*. Deskryptor komunikatu identyfikuje komunikat i zawiera dodatkowe informacje sterujące, takie jak typ komunikatu i priorytet przypisany do komunikatu przez aplikację wysyłającą.

Format deskryptora komunikatu jest definiowany przez właściwość IBM MQ. Pełny opis deskryptora komunikatu zawiera sekcja [MQMD-deskryptor komunikatu](#).

- *Właściwości komunikatu*. Metadane dotyczące komunikatu. Treść właściwości komunikatu jest definiowana przez aplikacje, które z nich korzystają. Więcej informacji na ten temat zawiera sekcja [Właściwości komunikatu](#).

Długości komunikatów

Domyślna maksymalna długość komunikatu to 4 MB, ale można ją zwiększyć do maksymalnej długości 100 MB (gdzie 1 MB jest równe 1 048 576 bajtów). W praktyce długość komunikatu może być ograniczona przez:

- Maksymalna długość komunikatu zdefiniowana dla kolejki odbiorczej
- Maksymalna długość komunikatu zdefiniowana dla menedżera kolejek
- Maksymalna długość komunikatu zdefiniowana przez kolejkę
- Maksymalna długość komunikatu zdefiniowana przez aplikację wysyłającą lub odbierającą
- Ilość pamięci dostępnej dla komunikatu

Wysłanie wszystkich informacji wymaganych przez aplikację może wymagać kilku komunikatów.

W jaki sposób aplikacje wysyłają i odbierają komunikaty?

Aplikacje wysyłają i odbierają komunikaty przy użyciu **wywołań MQI**.

Na przykład, aby umieścić komunikat w kolejce, aplikacja:

1. Otwiera wymaganą kolejkę, wywołując wywołanie MQI MQOPEN .
2. Wysyła wywołanie MQI MQPUT w celu umieszczenia komunikatu w kolejce.

Inna aplikacja może pobrać komunikat z tej samej kolejki, wywołując wywołanie MQI MQGET

Więcej informacji na temat wywołań MQI zawiera sekcja [Wywołania MQI](#).

Co to jest kolejka?

Kolejka jest strukturą danych używaną do przechowywania komunikatów.

Każda kolejka należy do *menedżera kolejek*. Menedżer kolejek obsługuje należące do niego kolejki i zapisuje wszystkie odebrane komunikaty w odpowiednich kolejkach. Komunikaty mogą być umieszczane w kolejce przez aplikacje lub przez menedżera kolejek w ramach normalnego działania.

Predefiniowane kolejki i kolejki dynamiczne

Kolejki można scharakteryzować w następujący sposób:

- **Predefiniowane kolejki** są tworzone przez administratora przy użyciu odpowiednich komend MQSC lub PCF. Predefiniowane kolejki są trwałe i istnieją niezależnie od aplikacji, które z nich korzystają, i pozostają po restarcie systemu IBM MQ.
- **Kolejki dynamiczne** są tworzone, gdy aplikacja wysyła żądanie MQOPEN, określając nazwę *kolejki modelowej*. Utworzona kolejka jest oparta na *definicji kolejki szablonu*, która jest nazywana kolejką modelową. Kolejkę modelową można utworzyć za pomocą komendy MQSC DEFINE QMODEL. Atrybuty kolejki modelowej (na przykład maksymalna liczba komunikatów, które mogą być w niej przechowywane) są dziedziczone przez dowolną kolejkę dynamiczną, która jest z niej tworzona.

Kolejki modelowe mają atrybut określający, czy kolejka dynamiczna ma być trwałą, czy tymczasową. Trwałe kolejki są zachowywane po restarcie aplikacji i menedżera kolejek; tymczasowe kolejki są tracone po restarcie.

Pobieranie komunikatów z kolejek

Odpowiednio autoryzowane aplikacje mogą pobierać komunikaty z kolejki zgodnie z następującymi algorytmami pobierania:

- Pierwszy przyszedł-pierwszy wyszedł (FIFO).
- Priorytet komunikatu zdefiniowany w deskrytorze komunikatu. Komunikaty o takim samym priorytecie są pobierane w oparciu o FIFO.
- Żądanie programu dla konkretnego komunikatu.

Używana metoda jest określana przez żądanie MQGET z aplikacji.

Obiekty IBM MQ

Menedżery kolejek definiują właściwości obiektów produktu IBM MQ. Wartości tych właściwości wpływają na sposób przetwarzania tych obiektów przez program IBM MQ. Do tworzenia obiektów i zarządzania nimi służą komendy i interfejsy IBM MQ. W aplikacjach do sterowania obiektami używany jest interfejs kolejki komunikatów (Message Queue Interface-MQI). Obiekty są identyfikowane przez IBM MQ *deskrytor obiektu* (MQOD), gdy są adresowane z programu.

Administrowanie obiektami




Administrowanie obiektami obejmuje następujące zadania:

- Uruchamianie i zatrzymywanie menedżerów kolejek.
- Tworzenie obiektów, w szczególności kolejek, dla aplikacji.
- Wyświetlanie lub zmienianie atrybutów obiektów.
- Usuwanie obiektów.
- Praca z kanałami w celu utworzenia ścieżek komunikacyjnych do menedżerów kolejek w innych (zdalnych) systemach.
- Tworzenie *klastrów* menedżerów kolejek w celu uproszczenia ogólnego procesu administracyjnego i zrównoważenia obciążenia.

Z wyjątkiem kolejek dynamicznych, przed rozpoczęciem pracy z nimi należy zdefiniować obiekty w menedżerze kolejek.


Jeśli do wykonania operacji administrowania obiektem używana jest komenda IBM MQ , menedżer kolejek sprawdza, czy użytkownik ma poziom uprawnień wymagany do wykonania tej operacji. Podobnie, gdy aplikacja używa wywołania MQOPEN do otwarcia obiektu, menedżer kolejek sprawdza, czy aplikacja ma wymagany poziom uprawnień, zanim zezwoli na dostęp do tego obiektu. Sprawdzana jest nazwa otwieranego obiektu.


Obiekty można definiować i zarządzać nimi za pomocą następujących metod:


- Komendy PCF opisane w sekcji [Programmable command formats reference](#) (Skorowidz formatów komend programowalnych) i [Automating administration tasks](#) (Automatyzacja zadań administracyjnych).
- Komendy MQSC opisane w sekcji [Komendy MQSC](#)
-  Operacje i panele sterowania IBM MQ for z/OS opisane w sekcji [Administrowanie IBM MQ for z/OS](#) .
-   IBM MQ Explorer (Windows i Linux tylko dla systemów Intel). Więcej informacji na ten temat zawiera sekcja [Wprowadzenie do programu MQ Explorer](#) .

Obiektami można również zarządzać za pomocą następujących metod:

- Komendy sterujące, które są wpisywane z klawiatury. Patrz sekcja [Administrowanie produktem IBM MQ for Multiplatforms za pomocą komend sterujących](#) .
- Wywołania interfejsu administracyjnego IBM MQ (MQAI) w programie. Patrz [IBM MQ Administration Interface \(MQAI\)](#) .

 W przypadku sekwencji komend IBM MQ w systemie AIX, Linux, and Windows można użyć narzędzia MQSC do uruchamiania serii komend przechowywanych w pliku. Więcej informacji na ten temat zawiera sekcja [Administrowanie produktem IBM MQ przy użyciu komend MQSC](#) .

 W przypadku sekwencji komend IBM MQ for IBM i , które są regularnie używane, można pisać programy w języku CL. Więcej informacji na ten temat zawiera sekcja [Zarządzanie systemem IBM MQ for IBM i za pomocą komend CL](#) .

 W przypadku sekwencji komend IBM MQ for z/OS , które są regularnie używane, można pisać programy administracyjne, które tworzą komunikaty zawierające komendy i umieszczają te komunikaty w kolejce wejściowej komend systemowych. Menedżer kolejek przetwarza komunikaty w tej kolejce w taki sam sposób, w jaki przetwarza komendy wprowadzone z wiersza komend lub z paneli sterowania i operacji. Ta technika jest opisana w sekcji [Pisanie programów do administrowania produktem IBM MQi zademonstrowana w przykładowej aplikacji Mail Manager dostarczanej z produktem IBM MQ for z/OS](#) . Opis tego przykładu zawiera sekcja [Przykładowe programy dla produktu IBM MQ for z/OS](#) .

Atrybuty obiektu

Właściwości obiektu są definiowane przez jego atrybuty. Niektóre można określić, a inne można tylko wyświetlić.

Na przykład maksymalna długość komunikatu, jaką może pomieścić kolejka, jest definiowana przez jej atrybut **MaxMsgLength** . Atrybut ten można określić podczas tworzenia kolejki. Atrybut **DefinitionType** określa sposób tworzenia kolejki; można tylko wyświetlić ten atrybut.

W programie IBM MQ istnieją dwa sposoby odwoływania się do atrybutu:

- Przy użyciu nazwy PCF, na przykład **MaxMsgLength** .
- Przy użyciu nazwy komendy MQSC, na przykład MAXMSGL.

Grupy współużytkownia kolejek



Menedżery kolejek, które mogą uzyskać dostęp do tego samego zestawu kolejek współużytkowanych, tworzą grupę nazywaną *grupą współużytkowania kolejek* (QSG) i komunikują się ze sobą za pomocą narzędzia CF, które przechowuje kolejki współużytkowane. Należy zauważyć, że QSG nie jest ściśle obiektem.

Kolejka współużytkowana to typ kolejki lokalnej z komunikatami, do której może uzyskać dostęp co najmniej jeden menedżer kolejek należący do grupy współużytkowania kolejek. Nie jest to taka sama kolejka, jak kolejka współużytkowana przez więcej niż jedną aplikację używającą tego samego menedżera kolejek.

Nazwy grup współużytkujących kolejki składają się z maksymalnie czterech znaków. Nazwa taka musi być unikalna w danej sieci i nie może być identyczna z nazwą menedżera kolejek.

Ważne: Kolejki współużytkowane i grupy współużytkowania kolejek są obsługiwane tylko w systemie IBM MQ for z/OS.

Więcej informacji zawiera sekcja [“Kolejki współużytkowane i grupy współużytkowania kolejek”](#) na stronie 173.

Domyślne obiekty systemowe

Domyślne obiekty systemowe to zestaw definicji obiektów, które są tworzone automatycznie podczas tworzenia menedżera kolejek. Wszystkie te definicje obiektów można kopiować i modyfikować w celu ich użycia w aplikacjach podczas instalacji.

Domyślne nazwy obiektów mają rdzeń SYSTEM; na przykład domyślną kolejką lokalną jest SYSTEM.DEFAULT.LOCAL.QUEUE, a domyślnym kanałem odbiorczym jest SYSTEM.DEF.RECEIVER. Nie można zmienić nazw tych obiektów; wymagane są domyślne obiekty o tych nazwach.

Podczas definiowania obiektu wszystkie atrybuty, które nie zostały jawnie określone, są kopiowane z odpowiedniego obiektu domyślnego. Na przykład, jeśli definiowana jest kolejka lokalna, nieokreślone atrybuty są pobierane z domyślnej kolejki SYSTEM.DEFAULT.LOCAL.QUEUE.

Więcej informacji na ten temat zawiera sekcja [Obiekty systemowe i domyślne](#).

Typy obiektów

Wiele zadań administracyjnych obejmuje manipulowanie różnymi typami produktu IBM MQ *obiekty*.

Informacje na temat nadawania nazw obiektom IBM MQ zawiera sekcja [“Nadawanie nazw obiektom IBM MQ”](#) na stronie 37.

Informacje na temat domyślnych obiektów utworzonych w menedżerze kolejek zawiera sekcja [“Domyślne obiekty systemowe”](#) na stronie 16.

Informacje na temat różnych typów obiektów IBM MQ można znaleźć w następujących sekcjach:

Obiekty informacji uwierzytelniającej

Obiekt informacji uwierzytelniającej udostępnia definicje wymagane do sprawdzania odwołań certyfikatów.

Obiekt informacji uwierzytelniającej menedżera kolejek stanowi część obsługi protokołu TLS (Transport Layer Security) w produkcie IBM MQ. Udostępnia on definicje niezbędne do sprawdzania unieważnionych certyfikatów. Ośrodki certyfikacji odbierają certyfikaty, które nie mogą być już zaufane.

Do zdefiniowania obiektu informacji uwierzytelniającej można użyć komendy MQSC **DEFINE AUTHINFO**. Więcej informacji na temat atrybutów obiektów informacji uwierzytelniającej zawiera sekcja [DEFINE AUTHINFO](#).

Z obiektem informacji uwierzytelniających można używać następujących komend sterujących systemem IBM MQ:

- [setmqaut](#) (nadawanie lub odbieranie uprawnień)

- **dspmqaut** (wyświetlenie autoryzacji obiektu)
- **dmpmqaut** (autoryzacje zrzutu)
- **rcrmqobj** (ponowne tworzenie obiektu)
- **rcdmqing** (zapis obrazu nośnika)
- **dspmqfls** (nazwy plików wyświetlanych)

Przegląd protokołu TLS i korzystanie z obiektów informacji uwierzytelniającej zawiera sekcja [Pojęcia związane z protokołem TLS \(Transport Layer Security\)](#) oraz sekcja [Protokoły TLS \(TLS\) w produkcie IBM MQ](#).

Kanały

Kanały są obiektami, które udostępniają ścieżkę komunikacji między menedżerami kolejek.

Więcej informacji zawiera sekcja [“Kanały” na stronie 30](#).

Obiekty informacji o komunikacji

Funkcja rozsyłania grupowego produktu IBM MQ umożliwia niezawodne rozsyłanie grupowe komunikatów z zachowaniem małych opóźnień i wysokiego zwielokrotnienia. Obiekt informacji o komunikacji (COMMINFO) jest wymagany dla transmisji w trybie rozsyłania grupowego.

Więcej informacji zawiera sekcja [“IBM MQ Rozsyłanie grupowe” na stronie 108](#).

Obiekt COMMINFO jest obiektem IBM MQ, który zawiera atrybuty powiązane z transmisją rozsyłania grupowego. Więcej informacji na temat tych atrybutów zawiera sekcja [DEFINE COMMINFO](#). Więcej informacji na temat tworzenia obiektu COMMINFO zawiera sekcja [Pierwsze kroki z rozsyłaniem grupowym](#).


Procesy nasłuchujące

Programy nasłuchujące to procesy, które akceptują żądania sieciowe od innych menedżerów kolejek lub aplikacji klienckich i uruchamiają powiązane kanały.

Procesy nasłuchiwanie można uruchomić za pomocą komendy sterującej **`runmq1sr`**.

Obiekty nasłuchiwanie są obiektami systemu IBM MQ, które umożliwiają zarządzanie uruchamianiem i zatrzymywaniem procesów nasłuchiwanie z poziomu menedżera kolejek. Definiując atrybuty obiektu nasłuchiwanie, należy wykonać następujące czynności:

- Skonfiguruj proces nasłuchiwanie.
- Określ, czy proces nasłuchiwanie ma być automatycznie uruchamiany i zatrzymywany podczas uruchamiania i zatrzymywania menedżera kolejek.

Ważne:  Obiekty nasłuchiwanie nie są obsługiwane w systemie IBM MQ for z/OS. Więcej informacji na temat implementowania nasłuchiwanie przez produkt IBM MQ for z/OS przy użyciu inicjatora kanału zawiera sekcja [“Inicjator kanału w systemie z/OS” na stronie 168](#).


Listy nazw

Lista nazw jest obiektem IBM MQ zawierającym listę nazw klastrów, nazw kolejek lub nazw obiektów informacji uwierzytelniającej. W klastrze może on być używany do identyfikowania listy klastrów, dla których menedżer kolejek przechowuje repozytoria.

Lista nazw jest obiektem IBM MQ zawierającym listę innych obiektów IBM MQ. Zazwyczaj listy nazw są używane przez aplikacje, na przykład monitory wyzwalacza, w przypadku których służą do identyfikowania grupy kolejek. Zaletą korzystania z listy nazw jest to, że jest ona obsługiwana niezależnie od aplikacji. Można ją aktualizować bez zatrzymywania aplikacji, które z niej korzystają. Ponadto, jeśli jedna aplikacja ulegnie awarii, lista nazw nie zostanie naruszona, a inne aplikacje będą mogły jej nadal używać.

Listy nazw są również używane z klastrami menedżerów kolejek do przechowywania listy klastrów, do których odwołuje się więcej niż jeden obiekt IBM MQ .

Listy nazw można definiować i modyfikować za pomocą komend MQSC [DEFINE NAMELIST](#) i [ALTER NAMELIST](#) .

Uwaga:  Alternatywnie w systemie z/OS można użyć operacji IBM MQ for z/OS i paneli sterujących.

Program może użyć interfejsu MQI, aby sprawdzić, które kolejki znajdują się na tych listach nazw. Organizacja list nazw należy do obowiązków projektanta aplikacji i administratora systemu.

Listę dostępnych do użycia atrybutów listy nazw zawiera sekcja [Atrybuty list nazw](#).


Definicje procesów

Obiekty definicji procesu umożliwiają uruchamianie aplikacji bez konieczności interwencji operatora przez zdefiniowanie atrybutów aplikacji, które mają być używane przez menedżer kolejek.

Obiekt definicji procesu definiuje aplikację, która jest uruchamiana w odpowiedzi na zdarzenie wyzwalające w menedżerze kolejek systemu IBM MQ . Atrybuty definicji procesu obejmują identyfikator aplikacji, typ aplikacji i dane specyficzne dla aplikacji. Więcej informacji na ten temat zawiera sekcja *Kolejki inicjowania* w podręczniku [“Kolejki używane do konkretnych celów przez produkt IBM MQ”](#) na stronie 27.

Aby umożliwić uruchamianie aplikacji bez konieczności interwencji operatora, zgodnie z opisem w sekcji [Uruchamianie aplikacji IBM MQ przy użyciu wyzwalaczy](#), atrybuty aplikacji muszą być znane menedżerowi kolejek. Atrybuty te są definiowane w *obiekcie definicji procesu*.

Atrybut **ProcessName** jest ustalany podczas tworzenia obiektu. Inne atrybuty można jednak zmienić za pomocą komend IBM MQ .

Uwaga:  Alternatywnie w systemie z/OS można użyć operacji IBM MQ for z/OS i paneli sterujących.

Aby uzyskać informacje o wartościach wszystkich atrybutów, należy użyć opcji [MQINQ-Zapytanie o atrybuty obiektu](#).

Lista dostępnych do użycia atrybutów definicji procesów znajduje się w sekcji [Atrybuty definicji procesów](#).

Kolejki

IBM MQ *kolejka* jest nazwanym obiektem, w którym aplikacje mogą umieszczać komunikaty i z którego mogą je otrzymywać.

Więcej informacji zawiera sekcja [“Kolejki”](#) na stronie 20.

Menedżery kolejek

Menedżery kolejek systemu IBM MQ udostępniają aplikacjom usługi kolejkowania i zarządzają należącymi do nich kolejkami.

Więcej informacji zawiera sekcja [“Menedżery kolejek”](#) na stronie 29.

Usługi

Obiekty *usługi* to sposób definiowania programów, które mają być uruchamiane podczas uruchamiania lub zatrzymywania menedżera kolejek.

Programy mogą być jednego z następujących typów:


Serwery

Serwer jest obiektem usługi, którego parametr SERVTYPE ma wartość SERVER. Obiekt usługi serwera jest definicją programu, który zostanie uruchomiony po uruchomieniu określonego

menedżera kolejek. Jednocześnie może być wykonywana tylko jedna instancja procesu serwera. Podczas działania status procesu serwera można monitorować za pomocą komendy MQSC DISPLAY SVSTATUS. Zwykle obiekty usług serwera są definicjami programów, takich jak programy obsługi niedostarczonych wiadomości lub monitory wyzwalaczy, jednak programy, które mogą być uruchamiane, nie są ograniczone do tych, które są dostarczane z produktem IBM MQ. Dodatkowo można zdefiniować obiekt usługi serwera, aby zawierał komendę, która zostanie uruchomiona po zamknięciu określonego menedżera kolejek w celu zakończenia programu.

Komendy

Komenda jest obiektem usługi, który ma parametr SERVTYPE określony jako COMMAND. Obiekt usługi komendy jest definicją programu, który zostanie wykonany po uruchomieniu lub zatrzymaniu określonego menedżera kolejek. Wiele instancji procesu komendy może być wykonywanych współbieżnie. Obiekty usług komend różnią się od obiektów usług serwera tym, że po wykonaniu programu menedżer kolejek nie będzie monitorować programu. Zwykle obiekty usług komend są definicjami programów, które są krótkotrwałe i będą wykonywać określone zadania, takie jak uruchamianie jednego lub większej liczby innych zadań.

Ważne:  Obiekty usług nie są obsługiwane w systemie IBM MQ for z/OS.

Więcej informacji na ten temat zawiera sekcja [Praca z usługami](#).

Klasy pamięci masowej



Klasa pamięci odwzorowuje jedną lub więcej kolejek do zbioru stron.

Oznacza to, że komunikaty dla tej kolejki są przechowywane (podlegające buforowaniu) w tym zestawie stron.

Klasy pamięci masowej są obsługiwane tylko w systemie IBM MQ for z/OS.

Więcej informacji na temat klas pamięci masowej zawiera sekcja [Planowanie środowiska IBM MQ w systemie z/OS](#).

Obiekty tematu

Obiekt tematu jest obiektem IBM MQ, który umożliwia przypisywanie konkretnych, innych niż domyślne atrybutów do tematów.

Temat jest definiowany przez aplikację publikującą lub subskrybującą konkretny *łańcuch tematu*. Łańcuch tematu może określać hierarchię tematów, oddzielając je znakiem ukośnika (/). Można to zwizualizować za pomocą *drzewa tematów*. Jeśli na przykład aplikacja publikuje w łańcuchach tematów /Sport/American Football i /Sport/Soccer, tworzone jest drzewo tematów, które ma węzeł nadrzędny Sport z dwoma elementami potomnymi: American Football i Soccer.

Tematy dziedziczą swoje atrybuty z pierwszego nadrzędnego węzła administracyjnego znalezione w drzewie tematów. Jeśli w danym drzewie tematów nie ma węzłów tematów administracyjnych, wszystkie tematy dziedziczą swoje atrybuty z podstawowego obiektu tematu SYSTEM.BASE.TOPIC.

Obiekt tematu można utworzyć w dowolnym węźle drzewa tematów, określając łańcuch tematu tego węzła w atrybucie TOPICSTR obiektu tematu. Można również zdefiniować inne atrybuty dla węzła tematu administracyjnego. Więcej informacji na temat tych atrybutów zawiera sekcja [Komendy MQSClub Automatyzacja administrowania za pomocą komend PCF](#). Każdy obiekt tematu domyślnie dziedziczy atrybuty z najbliższego nadrzędnego węzła administracyjnego tematu.

Obiekty tematów mogą być również używane do ukrywania pełnego drzewa tematów przed twórcami aplikacji. Jeśli dla tematu /Sport/American Football zostanie utworzony obiekt tematu o nazwie FOOTBALL.US, aplikacja może opublikować lub subskrybować obiekt o nazwie FOOTBALL.US zamiast łańcucha /Sport/American Football z takim samym wynikiem.

W przypadku wprowadzenia znaku #, +,/lub * w łańcuchu tematu w obiekcie tematu znak ten jest traktowany jako zwykły znak w łańcuchu i jest traktowany jako część łańcucha tematu powiązanego z obiektem tematu.

Więcej informacji na temat obiektów tematu zawiera sekcja [“Przesyłanie komunikatów w trybie publikowania/subskrypcji”](#) na stronie 63.

Pojęcia pokrewne

[“Wprowadzenie do kolejkowania komunikatów”](#) na stronie 5

Produkty IBM MQ umożliwiają programom komunikowanie się między sobą przez sieć różnych komponentów (procesorów, systemów operacyjnych, podsystemów i protokołów komunikacyjnych) za pomocą spójnego aplikacyjnego interfejsu programistycznego.

Odsyłacze pokrewne

[Komendy MQSC](#)

Kolejki

Wprowadzenie do kolejek i atrybutów kolejek systemu IBM MQ .

Komunikaty są przechowywane w kolejce, więc jeśli aplikacja umieszczająca oczekuje odpowiedzi na komunikat, może wykonywać inne czynności podczas oczekiwania na tę odpowiedź. Aplikacje uzyskują dostęp do kolejki za pomocą interfejsu kolejki komunikatów (Message Queue Interface-MQI) opisanego w sekcji [Przegląd interfejsu kolejki komunikatów](#).

Aby można było umieścić komunikat w kolejce, kolejka musi być już utworzona. Właścicielem kolejki jest menedżer kolejek, który może być właścicielem wielu kolejek. Jednak każda kolejka musi mieć unikalną nazwę w ramach tego menedżera kolejek.

Kolejka jest obsługiwana przez menedżer kolejek. W większości przypadków każda kolejka jest fizycznie zarządzana przez menedżera kolejek, ale nie jest to widoczne dla aplikacji. IBM MQ for z/OS Kolejki współużytkowane mogą być zarządzane przez dowolny menedżer kolejek w grupie współużytkowania kolejek.

Aby utworzyć kolejkę, można użyć komend IBM MQ (MQSC), komend PCF lub interfejsów specyficznych dla platformy. Na przykład operacje IBM MQ for z/OS i panele sterowania są zależne od platformy.

Kolejki lokalne dla zadań tymczasowych można tworzyć *dynamicznie* z poziomu aplikacji. Na przykład można utworzyć kolejki *reply-to* (które nie są potrzebne po zakończeniu działania aplikacji). Więcej informacji na ten temat zawiera sekcja [“Kolejki dynamiczne i modelowe”](#) na stronie 25.

Przed użyciem kolejki należy ją otworzyć i określić, co ma z nią zrobić. Na przykład można otworzyć kolejkę dla:

- Tylko przeglądanie komunikatów (bez ich pobierania)
- Pobieranie komunikatów (i współużytkowanie dostępu z innymi programami lub dostęp na wyłączność)
- Umieszczanie komunikatów w kolejce
- Uzyskiwanie informacji o atrybutach kolejki
- Ustawianie atrybutów kolejki

Pełną listę opcji, które można określić podczas otwierania kolejki, zawiera sekcja [MQOPEN-Open object](#)(Obiekt MQOPEN-Open).

Atrybuty kolejek

Niektóre atrybuty kolejki są określane podczas definiowania kolejki i nie można ich później zmienić (na przykład typ kolejki). Inne atrybuty kolejek można pogrupować w atrybuty, które można zmieniać:

- Przez menedżera kolejek podczas przetwarzania kolejki (na przykład bieżące zapełnienie kolejki).
- Tylko za pomocą komend (na przykład opisu tekstowego kolejki)
- Przez aplikacje, za pomocą wywołania MQSET (na przykład, czy operacje umieszczania są dozwolone w kolejce)

Wartości wszystkich atrybutów można znaleźć za pomocą wywołania MQINQ.

Atrybuty wspólne dla więcej niż jednego typu kolejki to:

Nazwa QName

Nazwa kolejki.

QTYPE

Typ kolejki.

Opis kolejki

Tekst opisu kolejki.

InhibitGet

Określa, czy programy mogą pobrać komunikaty z kolejki. Jednak nigdy nie można pobrać komunikatów ze zdalnych kolejek.

InhibitPut

Określa, czy programy mogą umieszczać komunikaty w kolejce.

DefPriority


Domyślny priorytet komunikatów umieszczanych w kolejce.

DefPersistence

Domyślna trwałość komunikatów umieszczanych w kolejce

Zasięg

Określa, czy pozycja dla tej kolejki istnieje również w usłudze nazw.

 Atrybut **Scope** nie jest obsługiwany w systemie z/OS

Pełny opis tych atrybutów zawiera sekcja [Atrybuty kolejek](#).

Sposoby definiowania kolejek

Kolejki można zdefiniować w programie IBM MQ za pomocą komendy MQSC [DEFINE](#) lub komendy PCF [Create Queue](#). Komendy określają typ kolejki i jej atrybuty. Na przykład obiekt kolejki lokalnej ma atrybuty określające, co się dzieje, gdy aplikacje odwołują się do tej kolejki w wywołaniach MQI. Przykłady atrybutów:

- Określa, czy aplikacje mogą pobierać komunikaty z kolejki (włączona metoda GET)
- Określa, czy aplikacje mogą umieszczać komunikaty w kolejce (włączone PUT)
- Określa, czy dostęp do kolejki jest wyłączony dla jednej aplikacji, czy współużytkowany przez aplikacje.
- Maksymalna liczba komunikatów, które mogą być przechowywane w kolejce w tym samym czasie (maksymalne zapętnienie kolejki)
- Maksymalna długość komunikatów, które można umieścić w kolejce

Istnieją również różne interfejsy specyficzne dla platformy, których można używać do definiowania kolejek.

Pojęcia pokrewne

[“Kolejki klastra” na stronie 59](#)

Kolejka klastra to kolejka udostępniana przez menedżer kolejek klastra innym menedżerom kolejek w klastrze.

[“Kolejki niedostarczonych komunikatów” na stronie 50](#)

Kolejka niedostarczonych komunikatów (lub kolejka niedostarczonych komunikatów) jest kolejką, do której wysyłane są komunikaty, jeśli nie mogą być kierowane do właściwego miejsca docelowego. Każdy menedżer kolejek zwykle ma kolejkę niedostarczonych komunikatów.

[Automatyzowanie administrowania za pomocą komend PCF](#)

[Praca z kolejkami w programie IBM MQ Console](#)

Zadania pokrewne

[Administrowanie programem IBM MQ za pomocą komend MQSC](#)

[Tworzenie i konfigurowanie menedżerów kolejek i obiektów za pomocą programu MQ Explorer](#)

Odsyłacze pokrewne

[“Porównanie kolejek współużytkowanych i kolejek klastra” na stronie 59](#)

Informacje te mają na celu pomóc w porównaniu kolejek współużytkowanych i kolejek klastra oraz w podjęciu decyzji, które z nich mogą być bardziej odpowiednie dla danego systemu.

Informacje pokrewne

[“Co to jest kolejka współużytkowana?” na stronie 173](#)

Kolejki lokalne

Kolejki transmisji, inicjowania, niedostarczonych komunikatów, komend, kolejek domyślnych, kanałów i kolejek zdarzeń są typami kolejek lokalnych.

Kolejka jest znana w programie jako *lokalna*, jeśli jej właścicielem jest menedżer kolejek, z którym połączony jest program. Komunikaty można pobierać z kolejek lokalnych oraz umieszczać je w nich.

Obiekt definicji kolejki przechowuje informacje o definicji kolejki, a także komunikaty fizyczne umieszczone w kolejce.

Każdy menedżer kolejek może mieć kilka kolejek lokalnych, które są używane do specjalnych celów:

Kolejki transmisji

Gdy aplikacja wysyła komunikat do kolejki zdalnej, menedżer kolejek lokalnych zapisuje komunikat w specjalnej kolejce lokalnej, nazywanej *kolejką transmisji*. Aplikacje mogą umieszczać komunikaty bezpośrednio w kolejce transmisji lub pośrednio poprzez definicję kolejki zdalnej.

Gdy menedżer kolejek wysyła komunikaty do zdalnego menedżera kolejek, identyfikuje kolejkę transmisji w następującej kolejności:

1. Kolejka transmisji o nazwie określonej w atrybucie XMITQ lokalnej definicji kolejki zdalnej.
2. Kolejka transmisji o takiej samej nazwie jak zdalny menedżer kolejek. Ta wartość jest wartością domyślną w XMITQ lokalnej definicji kolejki zdalnej.
3. Kolejka transmisji o nazwie określonej w atrybucie DEFXMITQ lokalnego menedżera kolejek.

Agent kanału komunikatów jest programem kanału powiązany z kolejką transmisji i dostarcza komunikat do następnego miejsca docelowego. Następnym miejscem docelowym jest menedżer kolejek, z którym połączony jest kanał komunikatów. Nie musi to być ten sam menedżer kolejek, co końcowe miejsce docelowe komunikatu. Po dostarczeniu komunikatu do następnego miejsca docelowego jest on usuwany z kolejki transmisji. Komunikat może wymagać przejścia przez wiele menedżerów kolejek w drodze do miejsca docelowego. Należy zdefiniować kolejkę transmisji w każdym menedżerze kolejek na trasie, z których każdy wstrzymuje komunikaty oczekujące na przesłanie do następnego miejsca docelowego. Normalna kolejka transmisji przechowuje komunikaty dla następnego miejsca docelowego, chociaż komunikaty mogą mieć różne miejsca docelowe. Kolejka transmisji klastra zawiera komunikaty dla wielu miejsc docelowych. Parametr `correlID` dla każdego komunikatu identyfikuje kanał, w którym komunikat jest umieszczany w celu przesłania go do następnego miejsca docelowego.

W menedżerze kolejek można zdefiniować kilka kolejek transmisji. Można zdefiniować kilka kolejek transmisji dla tego samego miejsca docelowego, z których każda będzie używana dla innej klasy usług. Na przykład można utworzyć różne kolejki transmisji dla małych komunikatów i dużych komunikatów, które będą kierowane do tego samego miejsca docelowego. Następnie można przesłać komunikaty przy użyciu różnych kanałów komunikatów, aby duże komunikaty nie przechowywały mniejszych komunikatów. Wszystkie komunikaty do kolejek klastra lub tematów klastra są domyślnie umieszczane w pojedynczej kolejce transmisji klastra `SYSTEM.CLUSTER.TRANSMIT.QUEUE`. Opcjonalnie można zmienić wartość domyślną i rozdzielić ruch komunikatów do różnych menedżerów kolejek klastra na różne kolejki transmisji klastra. Jeśli atrybut menedżera kolejek `DEFCLXQ` ma wartość `CHANNEL`, każdy kanał nadawczy klastra tworzy oddzielną kolejkę transmisji klastra.

Alternatywnie można ręcznie zdefiniować kolejki transmisji klastra, które mają być używane przez kanały nadawcze klastra.

Kolejki transmisji mogą wyzwolić agenta kanału komunikatów w celu wysyłania komunikatów dalej; patrz sekcja [Uruchamianie aplikacji IBM MQ za pomocą wyzwalaczy](#).

z/OS W systemie IBM MQ for z/OS, jeśli używane jest kolejkowanie wewnątrz grupy, kolejka transmisji jest obsługiwana przez *agenta kolejkowania wewnątrz grupy*. Współużytkowana kolejka transmisji jest używana w przypadku używania kolejkowania wewnątrz grupy w systemie IBM MQ for z/OS.

Kolejki inicjująca

Kolejka inicjująca jest kolejką lokalną, w której menedżer kolejek umieszcza komunikat wyzwalacza po wystąpieniu zdarzenia wyzwalającego w kolejce aplikacji.

Zdarzenie wyzwalające jest zdarzeniem, które ma spowodować rozpoczęcie przetwarzania kolejki przez program. Na przykład zdarzenie może mieć więcej niż 10 nadchodzących komunikatów. Więcej informacji na temat działania wyzwalania zawiera sekcja [Uruchamianie aplikacji IBM MQ za pomocą wyzwalaczy](#).

Kolejka niedostarczonych komunikatów

Kolejka niedostarczonych komunikatów to kolejka lokalna, w której menedżer kolejek umieszcza komunikaty, których nie może dostarczyć.

Gdy menedżer kolejek umieszcza komunikat w kolejce niedostarczonych komunikatów, dodaje nagłówek do komunikatu. Informacje w nagłówku zawierają przyczynę umieszczenia komunikatu przez menedżera kolejek w kolejce niedostarczonych komunikatów. Zawiera również miejsce docelowe oryginalnego komunikatu, datę i godzinę umieszczenia komunikatu przez menedżera kolejek w kolejce niedostarczonych komunikatów.

Aplikacje mogą również używać kolejki dla komunikatów, których nie mogą dostarczyć. Więcej informacji na ten temat zawiera sekcja [Korzystanie z kolejki niedostarczonych komunikatów](#).

Kolejka komend systemowych

Kolejka komend systemowych jest kolejką, do której aplikacje z odpowiednią autoryzacją mogą wysyłać komendy IBM MQ . Kolejki te odbierają komendy PCF, MQSC i CL, zgodnie z ich obsługiwany na platformie, w gotowości do ich działania przez menedżera kolejek.

z/OS W systemie IBM MQ for z/OS kolejka ma nazwę SYSTEM . COMMAND . INPUT ; na innych platformach nosi nazwę SYSTEM . ADMIN . COMMAND . QUEUE. Akceptowane komendy różnią się w zależności od platformy. Szczegółowe informacje można znaleźć w sekcji [Skorowidz formatów komend programowalnych](#).

Domyślne kolejki systemowe

Domyślne kolejki systemowe zawierają początkowe definicje kolejek systemu. Podczas tworzenia definicji kolejki menedżer kolejek kopiuje definicję z odpowiedniej domyślnej kolejki systemowej. Tworzenie definicji kolejki różni się od tworzenia kolejki dynamicznej. Definicja kolejki dynamicznej jest oparta na kolejce modelowej wybranej jako szablon dla kolejki dynamicznej.

kolejki zdarzeń

Kolejki zdarzeń przechowują komunikaty zdarzeń. Te komunikaty są raportowane przez menedżera kolejek lub kanał.

Kolejki zdalne

W przypadku programu kolejka jest *zdalna* , jeśli należy do innego menedżera kolejek niż ten, z którym jest połączony program.


Po ustanowieniu tarczy komunikacyjnego program może wysłać komunikat do kolejki zdalnej. Program nigdy nie może pobrać komunikatu ze zdalnej kolejki.

Obiekt definicji kolejki, utworzony podczas definiowania kolejki zdalnej, zawiera tylko informacje niezbędne lokalnemu menedżerowi kolejek do znalezienia kolejki, do której ma zostać wysłany komunikat. Ten obiekt jest nazywany *lokalną definicją kolejki zdalnej*. Wszystkie atrybuty kolejki zdalnej są wstrzymane przez menedżera kolejek, który jest jej właścicielem, ponieważ jest to kolejka lokalna dla tego menedżera kolejek.

Podczas otwierania kolejki zdalnej, aby zidentyfikować kolejkę, należy określić jedną z następujących opcji:

- Nazwa definicji lokalnej, która definiuje kolejkę zdalną. Z punktu widzenia aplikacji jest to takie samo, jak otwieranie kolejki lokalnej. Aplikacja nie musi wiedzieć, czy kolejka jest lokalna, czy zdalna.

Aby utworzyć lokalną definicję kolejki zdalnej na wszystkich platformach z wyjątkiem IBM i, należy użyć komendy `DEFINE QREMOTE`.

 W systemie IBM należy użyć komendy `CRTMQMQ`.

- Nazwa zdalnego menedżera kolejek i nazwa kolejki znana temu zdalnemu menedżerowi kolejek.

Definicje lokalne kolejek zdalnych mają trzy atrybuty oprócz wspólnych atrybutów opisanych w sekcji "Atrybuty kolejek" na stronie 20. Są to następujące trzy atrybuty:

RemoteQName

Nazwa, pod którą menedżer kolejek jest właścicielem kolejki.

RemoteQMgrNazwa

Nazwa menedżera kolejek będącego właścicielem.

XmitQName

Nazwa lokalnej kolejki transmisji, która jest używana podczas przekazywania komunikatów do innych menedżerów kolejek.

Więcej informacji na temat tych atrybutów zawiera sekcja [Atrybuty kolejek](#).


Jeśli wywołanie `MQINQ` jest używane względem lokalnej definicji kolejki zdalnej, menedżer kolejek zwraca tylko atrybuty definicji lokalnej, tj. nazwę kolejki zdalnej, nazwę menedżera kolejek zdalnych i nazwę kolejki transmisji, a nie atrybuty zgodnej kolejki lokalnej w systemie zdalnym.

Patrz także [Kolejki transmisji](#).

Kolejki aliasowe

Kolejka aliasowa jest obiektem IBM MQ, którego można użyć w celu uzyskania dostępu do innej kolejki lub tematu. Oznacza to, że więcej niż jeden program może pracować z tą samą kolejką, uzyskując do niej dostęp przy użyciu różnych nazw.

Kolejka będąca wynikiem rozstrzygnięcia nazwy aliasu, zwanej kolejką podstawową, może być dowolnym z następujących typów kolejek obsługiwanych przez platformę:

- Kolejka lokalna
- Lokalna definicja kolejki zdalnej.
-  Kolejka współużytkowana będąca typem kolejki lokalnej dostępnej tylko w systemie IBM MQ for z/OS.
- Predefiniowana kolejka
- Kolejka dynamiczna

Alias może również zostać przetłumaczony na temat. Jeśli aplikacja aktualnie umieszcza komunikaty w kolejce, można ją opublikować w temacie, tworząc nazwę kolejki aliasem tematu. Zmiana kodu aplikacji nie jest konieczna.

Uwaga: Alias nie może zostać bezpośrednio przetłumaczony na inny alias w tym samym menedżerze kolejek.

Przykładem użycia kolejek aliasowych jest nadanie przez administratora systemu różnych uprawnień dostępu do nazwy kolejki podstawowej (czyli kolejki, na którą alias jest tłumaczony) i do nazwy kolejki aliasowej. Oznacza to, że program lub użytkownik może być uprawniony do korzystania z kolejki aliasowej, ale nie z kolejki podstawowej.

Alternatywnie można ustawić autoryzację, aby nie zezwalać na operacje umieszczania dla nazwy aliasu, ale zezwalać na nie dla kolejki podstawowej.

W niektórych aplikacjach użycie kolejek aliasowych oznacza, że administratorzy systemu mogą łatwo zmienić definicję obiektu kolejki aliasowej bez konieczności zmiany aplikacji.

Gdy program IBM MQ próbuje użyć tej nazwy, sprawdza autoryzację względem nazwy aliasu. Nie sprawdza, czy program ma uprawnienia dostępu do nazwy, na którą alias jest tłumaczony. Program może być zatem uprawniony do uzyskania dostępu do nazwy kolejki aliasowej, ale nie do przetłumaczonych nazw kolejek.

Oprócz ogólnych atrybutów kolejki opisanych w sekcji [“Kolejki”](#) na stronie 20, kolejki aliasowe mają atrybut **BaseQName**. Jest to nazwa kolejki podstawowej, na którą tłumaczona jest nazwa aliasu. Pełniejszy opis tego atrybutu zawiera sekcja [BaseQName \(MQCHAR48\)](#).

Atrybuty *InhibitGet* i **InhibitPut** (patrz sekcja [“Kolejki”](#) na stronie 20) Kolejki aliasowe należą do nazwy aliasu. Na przykład, jeśli alias-nazwa kolejki ALIAS1 zostanie przetłumaczona na podstawową nazwę kolejki BASE, to zablokowanie opcji ALIAS1 będzie miało wpływ tylko na opcję ALIAS1, a opcja BASE nie będzie zablokowana. Jednak blokady na BASE mają również wpływ na ALIAS1.

Atrybuty *DefPriority* i **DefPersistence** również należą do nazwy aliasu. Na przykład można przypisać różne priorytety domyślne do różnych aliasów tej samej kolejki podstawowej. Ponadto można zmienić te priorytety bez konieczności zmiany aplikacji, które używają aliasów.


Kolejki dynamiczne i modelowe

Te informacje umożliwiają wgląd w kolejki dynamiczne, właściwości tymczasowych i trwałych kolejek dynamicznych, użycie kolejek dynamicznych, niektóre zagadnienia związane z używaniem kolejek dynamicznych i kolejek modelowych.

Gdy aplikacja wysyła wywołanie MQOPEN w celu otwarcia kolejki modelowej, menedżer kolejek dynamicznie tworzy instancję kolejki lokalnej z takimi samymi atrybutami, jak kolejka modelowa. W zależności od wartości pola *DefinitionType* kolejki modelowej menedżer kolejek tworzy tymczasową lub trwałą kolejkę dynamiczną (patrz sekcja [Tworzenie kolejek dynamicznych](#)).

Właściwości tymczasowych kolejek dynamicznych

Tymczasowe kolejki dynamiczne mają następujące właściwości:

-  Nie mogą to być kolejki współużytkowane dostępne z poziomu menedżerów kolejek w grupie współużytkowania kolejek.
Należy zauważyć, że grupy współużytkowania kolejek są dostępne tylko w systemie IBM MQ for z/OS.
- Przechowują tylko komunikaty nietrwale.
- Są one nienaprawialne.
- Są one usuwane po uruchomieniu menedżera kolejek.
- Są one usuwane, gdy aplikacja, która wywołała wywołanie MQOPEN, które utworzyło kolejkę, zamknie ją lub zakończy działanie.
 - Jeśli w kolejce znajdują się jakiegokolwiek zatwierdzone komunikaty, są one usuwane.
 - Jeśli w tym momencie istnieją niezatwierdzone wywołania MQGET, MQPUT lub MQPUT1 dla kolejki, kolejka jest oznaczana jako usunięta logicznie i jest usuwana fizycznie (po zatwierdzeniu tych wywołań) w ramach przetwarzania zamknięcia lub po zakończeniu działania aplikacji.
 - Jeśli kolejka jest obecnie używana (przez utworzenie lub inną aplikację), jest ona oznaczana jako logicznie usunięta i jest fizycznie usuwana tylko wtedy, gdy jest zamykana przez ostatnią aplikację używającą tej kolejki.

- Próby uzyskania dostępu do logicznie usuniętej kolejki (innej niż zamknięcie kolejki) nie powiodły się z kodem przyczyny MQRC_Q_DELETED.
- MQCO_NONE, MQCO_DELETE i MQCO_DELETE_PURGE są traktowane jako MQCO_NONE, gdy są określone w wywołaniu MQCLOSE dla odpowiedniego wywołania MQOPEN, które utworzyło kolejkę.

Właściwości trwałych kolejek dynamicznych

Trwałe kolejki dynamiczne mają następujące właściwości:

- Przechowują one trwałe lub nietrwałe komunikaty.
- Są one odtwarzalne w przypadku awarii systemu.
- Są one usuwane, gdy aplikacja (niekoniecznie ta, która wywołała wywołanie MQOPEN, które utworzyło kolejkę) pomyślnie zamknie kolejkę za pomocą opcji MQCO_DELETE lub MQCO_DELETE_PURGE.
 - Żądanie zamknięcia z opcją MQCO_DELETE kończy się niepowodzeniem, jeśli w kolejce nadal znajdują się jakiegokolwiek komunikaty (zatwierdzone lub niezatwierdzone). Żądanie zamknięcia z opcją MQCO_DELETE_PURGE powiedzie się nawet wtedy, gdy w kolejce znajdują się zatwierdzone komunikaty (komunikaty usuwane w ramach zamykania), ale kończy się niepowodzeniem, jeśli dla kolejki istnieją niezatwierdzone wywołania MQGET, MQPUT lub MQPUT1.
 - Jeśli żądanie usunięcia powiedzie się, ale kolejka jest w użyciu (przez utworzenie lub inną aplikację), kolejka zostanie oznaczona jako usunięta logicznie i zostanie fizycznie usunięta tylko wtedy, gdy zostanie zamknięta przez ostatnią aplikację używającą tej kolejki.
- Nie są one usuwane, jeśli zostały zamknięte przez aplikację, która nie ma uprawnień do usunięcia kolejki, chyba że aplikacja zamykająca wysłała wywołanie MQOPEN, które utworzyło kolejkę. Sprawdzenia autoryzacji są wykonywane względem identyfikatora użytkownika (lub alternatywnego identyfikatora użytkownika, jeśli określono opcję MQOO_ALTERNATE_USER_AUTHORITY), który został użyty do sprawdzenia poprawności odpowiedniego wywołania MQOPEN.
- Można je usunąć w taki sam sposób, jak zwykłą kolejkę.

Użycie kolejek dynamicznych

Kolejek dynamicznych można używać do:

- Aplikacje, które nie wymagają przechowywania kolejek po zakończeniu działania aplikacji.
- Aplikacje wymagające odpowiedzi na komunikaty, które mają być przetwarzane przez inną aplikację. Takie aplikacje mogą dynamicznie tworzyć kolejkę odpowiedzi, otwierając kolejkę modelową. Na przykład aplikacja kliencka może:
 1. Utwórz kolejkę dynamiczną.
 2. Podaj jego nazwę w polu **ReplyToQ** struktury deskryptora komunikatu żądania.
 3. Umieść żądanie w kolejce przetwarzanej przez serwer.

Serwer może następnie umieścić komunikat odpowiedzi w kolejce odpowiedzi. Na koniec klient może przetworzyć odpowiedź i zamknąć kolejkę odpowiedzi z opcją usuwania.

Uwagi dotyczące używania kolejek dynamicznych

Podczas korzystania z kolejek dynamicznych należy wziąć pod uwagę następujące kwestie:

- W modelu klient-serwer każdy klient musi utworzyć własną dynamiczną kolejkę odpowiedzi i korzystać z niej. Jeśli dynamiczna kolejka odpowiedzi jest współużytkowana przez więcej niż jednego klienta, usunięcie kolejki odpowiedzi może być opóźnione z powodu niezatwierdzonego działania zaległego w stosunku do kolejki lub dlatego, że kolejka jest używana przez innego klienta. Ponadto kolejka może być oznaczona jako usuwana logicznie i niedostępna dla kolejnych żądań API (innych niż MQCLOSE).
- Jeśli środowisko aplikacji wymaga, aby kolejki dynamiczne były współużytkowane przez aplikacje, upewnij się, że kolejka jest zamknięta (z opcją usuwania) tylko wtedy, gdy wszystkie działania

związane z kolejką zostały zatwierdzone. Powinien to być ostatni użytkownik. Dzięki temu usunięcie kolejki nie jest opóźnione i minimalizuje okres, w którym kolejka jest niedostępna, ponieważ została oznaczona jako usunięta logicznie.

Kolejki modelowe

Kolejka modelowa jest szablonem definicji kolejki, który jest używany podczas tworzenia kolejki dynamicznej.

Kolejkę lokalną można utworzyć dynamicznie na podstawie programu IBM MQ, nadając nazwę kolejce modelowej, która ma być używana jako szablon dla atrybutów kolejki. W tym momencie można zmienić niektóre atrybuty nowej kolejki. Nie można jednak zmienić pliku **DefinitionType**. Jeśli na przykład wymagana jest kolejka trwała, należy wybrać kolejkę modelową z typem definicji ustawionym na stałe. Niektóre aplikacje konwersacyjne mogą używać kolejek dynamicznych do przechowywania odpowiedzi na zapytania, ponieważ prawdopodobnie nie muszą one obsługiwać tych kolejek po przetworzeniu odpowiedzi.

Nazwę kolejki modelowej określa się w *deskrytorze obiektu* (MQOD) wywołania MQOPEN. Korzystając z atrybutów kolejki modelowej, menedżer kolejek dynamicznie tworzy kolejkę lokalną.

Można określić nazwę (pełną) dla kolejki dynamicznej lub rdzeń nazwy (na przykład ABC) i pozwolić menedżerowi kolejek na dodanie do niej unikalnej części lub pozwolić menedżerowi kolejek na przypisanie pełnej nazwy unikalnej. Jeśli menedżer kolejek przypisuje nazwę, umieszcza ją w strukturze MQOD.

Nie można wywołać wywołania MQPUT1 bezpośrednio do kolejki modelowej, ale można wprowadzić wywołanie MQPUT1 do kolejki dynamicznej, która została utworzona przez otwarcie kolejki modelowej.

Komendy MQSET i MQINQ nie można wprowadzać dla kolejki modelowej. Otwarcie kolejki modelowej za pomocą opcji MQOO_INQUIRE lub MQOO_SET powoduje wykonanie kolejnych wywołań MQINQ i MQSET dla dynamicznie tworzonej kolejki.

Atrybuty kolejki modelowej są podzbiorem atrybutów kolejki lokalnej. Pełniejszy opis znajduje się w sekcji [Atrybuty kolejek](#).

Kolejki używane do konkretnych celów przez produkt IBM MQ

Produkt IBM MQ używa niektórych kolejek lokalnych do konkretnych celów związanych z jego działaniem.

Należy zdefiniować te kolejki, zanim program IBM MQ będzie mógł ich używać.

Kolejki inicjująca

Kolejki inicjujące są kolejkami używanymi w wyzwaniu. Menedżer kolejek umieszcza komunikat wyzwalacza w kolejce inicjującej, gdy wystąpi zdarzenie wyzwalające. Zdarzenie wyzwalające jest logiczną kombinacją warunków wykrytych przez menedżer kolejek. Na przykład zdarzenie wyzwalające może zostać wygenerowane, kiedy liczba komunikatów w kolejce osiągnie predefiniowaną głębokość. To zdarzenie powoduje, że menedżer kolejek umieści komunikat wyzwalacza w określonej kolejce inicjującej. Ten komunikat wyzwalacza jest pobierany przez *monitor wyzwalacza*, specjalną aplikację monitorującą kolejkę inicjującą. Następnie monitor wyzwalacza uruchamia aplikację, która została określona w komunikacie wyzwalacza.

Jeśli menedżer kolejek ma używać wyzwiania, dla tego menedżera kolejek musi być zdefiniowana co najmniej jedna kolejka inicjująca. Patrz sekcja [Zarządzanie obiektami na potrzeby wyzwiania](#), [runmqtrmi](#) [Uruchamianie aplikacji IBM MQ za pomocą wyzwalaczy](#).

Kolejki transmisji

Kolejki transmisji to kolejki, w których tymczasowo przechowywane są komunikaty przeznaczone dla menedżera kolejek zdalnych. Należy zdefiniować co najmniej jedną kolejkę transmisji dla każdego zdalnego menedżera kolejek, do którego menedżer kolejek lokalnych ma wysyłać komunikaty bezpośrednio. Te kolejki są również używane do zdalnego administrowania. Więcej informacji na ten temat zawiera sekcja [Zdalne administrowanie z lokalnego menedżera kolejek](#). Informacje na temat używania kolejek transmisji w kolejkowaniu rozproszonym zawiera sekcja [Techniki kolejkowania rozproszonego w produkcie IBM MQ](#).

Każdy menedżer kolejek może mieć domyślną kolejkę transmisji. Jeśli menedżer kolejek, który nie jest częścią klastra, umieszcza komunikat w kolejce zdalnej, domyślnym działaniem jest użycie domyślnej kolejki transmisji. Jeśli istnieje kolejka transmisji o takiej samej nazwie jak menedżer kolejek docelowych, komunikat jest umieszczany w tej kolejce transmisji. Jeśli istnieje definicja aliasu menedżera kolejek, w której parametr **RQMNAME** jest zgodny z docelowym menedżerem kolejek i określono parametr **XMITQ**, komunikat jest umieszczany w kolejce transmisji o nazwie **XMITQ**. Jeśli nie ma parametru **XMITQ**, komunikat jest umieszczany w lokalnej kolejce o nazwie określonej w komunikacie.

Kolejki transmisji klastra

Każdy menedżer kolejek w klastrze ma kolejkę transmisji klastra o nazwie `SYSTEM.CLUSTER.TRANSMIT.QUEUE` oraz modelową kolejkę transmisji klastra `SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE`. Definicje tych kolejek są tworzone domyślnie podczas definiowania menedżera kolejek. Jeśli atrybut menedżera kolejek, **DEFCLXQ**, jest ustawiony na wartość `CHANNEL`, dla każdego tworzonego kanału nadawczego klastra automatycznie tworzona jest trwała kolejka transmisji klastra dynamicznego. Kolejki mają nazwę `SYSTEM.CLUSTER.TRANSMIT.ChannelName`. Kolejki transmisji klastra można również zdefiniować ręcznie.

Menedżer kolejek będący częścią klastra wysyła komunikaty z jednej z tych kolejek do innych menedżerów kolejek znajdujących się w tym samym klastrze.

Podczas tłumaczenia nazw kolejka transmisji klastra ma pierwszeństwo przed domyślną kolejką transmisji, a konkretna kolejka transmisji klastra ma pierwszeństwo przed kolejką `SYSTEM.CLUSTER.TRANSMIT.QUEUE`.

Kolejki niedostarczonych komunikatów

Kolejka niedostarczonych komunikatów (niedostarczonych komunikatów) to kolejka, w której przechowywane są komunikaty, których nie można kierować do poprawnych miejsc docelowych. Nie można skierować komunikatu, jeśli na przykład kolejka docelowa jest pełna. Podana kolejka niedostarczonych komunikatów ma nazwę `SYSTEM.DEAD.LETTER.QUEUE`.

W przypadku kolejkowania rozproszonego należy zdefiniować kolejkę niedostarczonych komunikatów w każdym zaangażowanym menedżerze kolejek.

Kolejki komend

Kolejka komend `SYSTEM.ADMIN.COMMAND.QUEUE` jest kolejką lokalną, do której aplikacje z odpowiednią autoryzacją mogą wysyłać komendy MQSC w celu przetwarzania. Komendy te są następnie pobierane przez komponent IBM MQ nazywany serwerem komend. Serwer komend sprawdza poprawność komend, przekazuje poprawne komendy do przetwarzania przez menedżer kolejek i zwraca wszystkie odpowiedzi do odpowiedniej kolejki odpowiedzi.

Kolejka komend jest tworzona automatycznie dla każdego menedżera kolejek podczas tworzenia tego menedżera kolejek.

Kolejki odpowiedzi

Gdy aplikacja wysyła komunikat żądania, aplikacja, która go odbiera, może wysłać komunikat odpowiedzi do aplikacji wysyłającej. Ten komunikat jest umieszczany w kolejce, nazywanej kolejką zwrotną, która jest zwykle kolejką lokalną aplikacji wysyłającej. Nazwa kolejki odpowiedzi jest określana przez aplikację wysyłającą jako część deskryptora komunikatu.

kolejki zdarzeń

Zdarzeń instrumentacji można używać do monitorowania menedżerów kolejek niezależnie od aplikacji MQI.

Po wystąpieniu zdarzenia instrumentacji menedżer kolejek umieszcza komunikat zdarzenia w kolejce zdarzeń. Ten komunikat może zostać odczytany przez aplikację monitorującą, która może poinformować administratora lub zainicjować pewne działanie naprawcze, jeśli zdarzenie wskazuje na problem.

Uwaga: Zdarzenia wyzwalacza różnią się od zdarzeń instrumentacji. Zdarzenia wyzwalające nie są powodowane przez te same warunki i nie generują komunikatów zdarzeń.

Więcej informacji na temat zdarzeń instrumentacji zawiera sekcja [Zdarzenia instrumentacji](#).

Menedżery kolejek

Wprowadzenie do *menedżerów kolejek* i usług kolejkowania, które udostępniają aplikacjom.

Aby program mógł korzystać z usług tego menedżera kolejek, musi mieć połączenie z menedżerem kolejek. Program może nawiązać to połączenie jawnie (za pomocą wywołania MQCONN lub MQCONNX) lub niejawnie (zależy to od platformy i środowiska, w którym program jest uruchomiony).

Menedżer kolejek systemu IBM MQ zapewnia wykonanie następujących działań:

- Atrybuty obiektu są zmieniane zgodnie z odebranymi komendami.
- Zdarzenia specjalne, takie jak zdarzenia wyzwalające lub zdarzenia instrumentacji, są generowane, gdy spełnione są odpowiednie warunki.
- Komunikaty są umieszczane w poprawnej kolejce, zgodnie z żądaniem aplikacji wywołującej wywołanie MQPUT . Jeśli nie można tego zrobić, aplikacja jest informowana o odpowiednim kodzie przyczyny.

Każda kolejka należy do pojedynczego menedżera kolejek i jest *kolejką lokalną* dla tego menedżera kolejek. Menedżer kolejek, z którym połączona jest aplikacja, jest dla tej aplikacji *lokalnym menedżerem kolejek* . W przypadku aplikacji kolejki należące do menedżera kolejek lokalnych są kolejkami lokalnymi.


Kolejka zdalna jest kolejką należącą do innego menedżera kolejek. *Zdalny menedżer kolejek* to dowolny menedżer kolejek inny niż menedżer kolejek lokalnych. Zdalny menedżer kolejek może istnieć na komputerze zdalnym w sieci lub na tym samym komputerze, co lokalny menedżer kolejek. Produkt IBM MQ obsługuje wiele menedżerów kolejek na tym samym komputerze.

Obiekt menedżera kolejek może być używany w niektórych wywołaniach MQI. Na przykład można uzyskać informacje o atrybutach obiektu menedżera kolejek za pomocą wywołania MQI MQINQ.


Atrybuty menedżerów kolejek

Z każdym menedżerem kolejek powiązany jest zestaw atrybutów (lub właściwości), które definiują jego parametry. Niektóre atrybuty menedżera kolejek są stałe podczas tworzenia; inne można zmienić za pomocą komend IBM MQ . Za pomocą wywołania MQINQ można uzyskać informacje o wartościach wszystkich atrybutów, z wyjątkiem tych, które są używane do szyfrowania TLS (Transport Layer Security).

Do atrybutów stałych należą:

- Nazwa menedżera kolejek
- Platforma, na której działa menedżer kolejek (na przykład Windows)
- Poziom komend sterujących systemem obsługiwanych przez menedżer kolejek
- Maksymalny priorytet, który można przypisać do komunikatów przetwarzanych przez menedżer kolejek
- Nazwa kolejki, do której programy mogą wysyłać komendy IBM MQ
- Maksymalna długość komunikatów, które może przetworzyć menedżer kolejek  (wartość stała tylko w programie IBM MQ for z/OS)
- Określa, czy menedżer kolejek obsługuje synchronizację podczas umieszczania i pobierania komunikatów przez programy.

Atrybuty *changeable* obejmują:

- Opis tekstowy menedżera kolejek
- Identyfikator zestawu znaków, którego menedżer kolejek używa dla łańcuchów znaków podczas przetwarzania wywołań MQI
- Przedział czasu używany przez menedżera kolejek do ograniczenia liczby komunikatów wyzwalacza
-  Odstęp czasu używany przez menedżer kolejek do określenia, jak często kolejki mają być skanowane w poszukiwaniu komunikatów, które utraciły ważność (tylko system IBM MQ for z/OS)
- Nazwa kolejki niedostarczonych komunikatów menedżera kolejek
- Nazwa domyślnej kolejki transmisji menedżera kolejek

- Maksymalna liczba otwartych uchwytów dla dowolnego połączenia
- Włączanie i wyłączanie różnych kategorii raportowania zdarzeń
- Maksymalna liczba niezatwierdzonych komunikatów w jednostce pracy

Menedżery kolejek i zarządzanie obciążeniem

Można skonfigurować klaster menedżerów kolejek, który ma więcej niż jedną definicję dla tej samej kolejki (na przykład menedżery kolejek w klastrze mogą być klonami). Komunikaty dla konkretnej kolejki mogą być obsługiwane przez dowolny menedżer kolejek, który udostępnia instancję kolejki. Algorytm zarządzania obciążeniem decyduje o tym, który menedżer kolejek obsługuje komunikat, i rozprzestrzenia obciążenie między menedżerami kolejek. Więcej informacji na ten temat zawiera sekcja [Algorytm zarządzania obciążeniem klastra](#).

Kanały

Kanał jest logicznym łączem komunikacyjnym używanym przez rozproszone menedżery kolejek między serwerem IBM MQ MQI client a serwerem IBM MQ lub między dwoma serwerami IBM MQ.

Kanały są używane do przenoszenia komunikatów z jednego menedżera kolejek do innego i chronią aplikacje przed bazowymi protokołami komunikacyjnymi. Menedżery kolejek mogą istnieć w tym samym systemie, w różnych systemach na tej samej platformie lub na różnych platformach. Wysyłane wiadomości mogą pochodzić z wielu miejsc:

- Aplikacje napisane przez użytkownika, które przesyłają dane z jednego węzła do drugiego.
- Napisane przez użytkownika aplikacje administracyjne, które używają komend PCF lub interfejsu MQAI.
- IBM MQ Explorer.
- Menedżery kolejek, które wysyłają komunikaty zdarzeń instrumentacji do innego menedżera kolejek.
- Menedżery kolejek, które wysyłają zdalne komendy administracyjne do innego menedżera kolejek. Na przykład za pomocą komend MQSC lub administrative REST API.

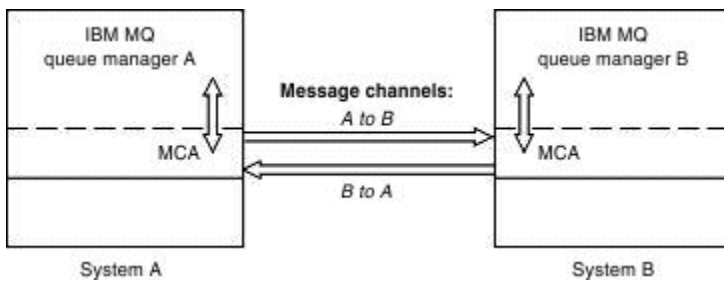
Kanał ma dwie definicje: po jednej na każdym końcu połączenia. Aby menedżery kolejek mogły komunikować się ze sobą, należy zdefiniować jeden obiekt kanału w menedżerze kolejek, który ma wysyłać komunikaty, oraz inny, uzupełniający obiekt, w menedżerze kolejek, który ma je odbierać. Na każdym końcu połączenia musi być używana ta sama *nazwa kanału*, a *typ kanału* musi być zgodny.

W produkcie IBM MQ istnieją trzy kategorie kanałów o różnych typach kanałów w obrębie tych kategorii:

- Kanały komunikatów, które są jednokierunkowe, i przesyłają komunikaty z jednego menedżera kolejek do innego.
- Kanały MQI, które są dwukierunkowe i przesyłają wywołania MQI z programu IBM MQ MQI client do menedżera kolejek oraz odpowiedzi z menedżera kolejek do klienta programu IBM MQ.
- Kanały AMQP, które są dwukierunkowe i łączą klienta AMQP z menedżerem kolejek na serwerze. Produkt IBM MQ używa kanałów AMQP do przesyłania wywołań i odpowiedzi AMQP między aplikacjami AMQP i menedżerami kolejek

Kanały komunikatów

Celem kanału komunikatów jest przesyłanie komunikatów z jednego menedżera kolejek do innego. Kanały komunikatów nie są wymagane przez środowisko serwera klienta.



Rysunek 2. Kanały komunikatów między dwoma menedżerami kolejek

Kanał komunikatów jest łączem jednokierunkowym. Aby zdalny menedżer kolejek odpowiadał na komunikaty wysyłane przez lokalny menedżer kolejek, należy skonfigurować drugi kanał w celu wysłania odpowiedzi z powrotem do lokalnego menedżera kolejek.

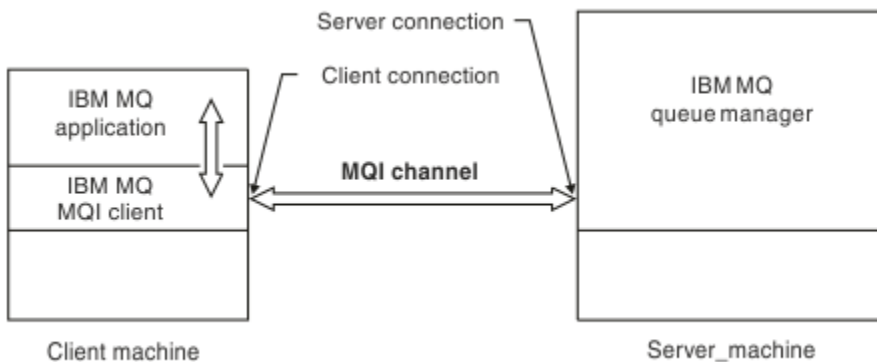
Kanał komunikatów łączy dwa menedżery kolejek za pomocą *agentów kanału komunikatów* (MCAs). Na każdym końcu kanału znajduje się agent kanału komunikatów. Można zezwolić agentowi MCA na przesyłanie komunikatów przy użyciu wielu wątków. Ten proces jest nazywany *potokami*. Potokowanie umożliwia agentowi MCA bardziej wydajne przesyłanie komunikatów, co zwiększa wydajność kanału. Więcej informacji na temat potokowania zawiera sekcja [Atrybuty kanałów](#).

Więcej informacji na temat kanałów zawiera sekcja [Wywołania wyjścia kanału i struktury danych](#) oraz sekcja ["Komponenty kolejkowania rozproszonego"](#) na stronie 47.

Kanały MQI

Kanał MQI (Message Queue Interface) łączy program IBM MQ MQI client z menedżerem kolejek na komputerze serwera i jest ustanawiany po wywołaniu programu MQCONN lub MQCONNX z aplikacji IBM MQ MQI client.

Jest to łącze dwukierunkowe i jest używane tylko do przesyłania wywołań i odpowiedzi MQI, w tym wywołań MQPUT zawierających dane komunikatu i wywołań MQGET zwracających dane komunikatu. Istnieją różne sposoby tworzenia i używania definicji kanałów (patrz sekcja [Definiowanie kanałów MQI](#)).



Rysunek 3. Połączenie klienta i połączenie serwera w kanale MQI

z/OS Kanał MQI może być używany do łączenia klienta z pojedynczym menedżerem kolejek lub z menedżerem kolejek, który jest częścią grupy współużytkownika kolejek (patrz sekcja [łączenie klienta z grupą współużytkownika kolejek](#)).

Istnieją dwa typy kanałów dla definicji kanału MQI. Definiują one dwukierunkowy kanał MQI.

Kanał połączenia klienckiego

Ten typ jest przeznaczony dla IBM MQ MQI client.

Kanał połączenia serwera

Ten typ jest przeznaczony dla serwera, na którym działa menedżer kolejek, z którym aplikacja IBM MQ działająca w środowisku IBM MQ MQI client ma się komunikować.

Kanały protokołu AMQP



Istnieje tylko jeden typ kanału AMQP.

Do nawiązywania połączenia między aplikacją przesyłania komunikatów AMQP i menedżerem kolejek używany jest kanał, umożliwiając aplikacji wymianę komunikatów z aplikacjami produktu IBM MQ. Kanał AMQP umożliwia utworzenie aplikacji przy użyciu produktu MQ Light, a następnie wdrożenie jej jako aplikacji korporacyjnej, dzięki czemu mogą być wykorzystywane funkcje korporacyjne udostępniane przez produkt IBM MQ.

Kanały połączenia klienta

Kanały połączeń klienta to obiekty, które udostępniają ścieżkę komunikacji między programem IBM MQ MQI client a menedżerem kolejek.

Kanały połączeń klienta są używane w kolejkowaniu rozproszonym do przenoszenia komunikatów między menedżerem kolejek a klientem. Chronią aplikacje przed bazowymi protokołami komunikacyjnymi. Klient może istnieć na tej samej lub innej platformie co menedżer kolejek.

Definicje kanałów

Opis każdego typu kanału zawiera sekcja [“Definicje kanałów”](#) na stronie 32 .

Pojęcia pokrewne

[“Rozproszone kolejkovanie i klastry”](#) na stronie 43

Kolejkowanie rozproszone oznacza wysyłanie komunikatów z jednego menedżera kolejek do innego. Odbierający menedżer kolejek może znajdować się na tym samym lub innym komputerze, w pobliżu lub po drugiej stronie świata. Może on działać na tej samej platformie co lokalny menedżer kolejek lub na dowolnej platformie obsługiwanej przez produkt IBM MQ. Można ręcznie zdefiniować wszystkie połączenia w rozproszonym środowisku kolejkowania lub utworzyć klastr i pozwolić produktowi IBM MQ na zdefiniowanie większości szczegółów połączenia.

[Przegląd interfejsu kolejki komunikatów](#)

Zadania pokrewne

[Administrowanie zdalnymi obiektami IBM MQ](#)

[Zatrzymywanie kanałów MQI](#)

[Konfigurowanie połączeń między serwerem i klientem](#)

Odsyłacze pokrewne

[Wywołania wyjścia kanału i struktury danych](#)

[“korporacyjna”](#) na stronie 36

Produkt IBM MQ MQI clients używa kanałów MQI do komunikacji z serwerem.

Definicje kanałów

Tabele opisujące różne typy kanałów komunikatów i kanałów MQI używanych przez produkt IBM MQ .

W kontekście kanałów komunikatów słowo „kanał” często jest używane jako synonim definicji kanału. Najczęściej można zorientować się na podstawie kontekstu, czy omawiany jest kanał kompletny, który ma dwa zakończenia, czy też definicja kanału, która ma tylko jedno zakończenie.

Kanały komunikatów

Definicje kanałów komunikatów mogą być następującego typu:

Typ definicji kanału komunikatów	Opis
Nadawca	Kanał nadawczy jest to kanał komunikatów, za pomocą którego menedżer kolejek wysyła komunikaty do innych menedżerów kolejek. Aby wysyłać komunikaty przy użyciu kanału nadawcy, należy także utworzyć na innym menedżerze kolejek kanał odbiorcy, który ma taką samą nazwę, co kanał nadawcy. Jeśli został zaimplementowany mechanizm wywołania zwrotnego, można używać kanału nadawczego z kanałem requestera.
Serwer	Kanał serwera jest to kanał komunikatów, za pomocą którego menedżer kolejek wysyła komunikaty do innych menedżerów kolejek. Aby wysyłać komunikaty przy użyciu kanału serwera, należy także utworzyć na innym menedżerze kolejek kanał odbiorcy, który ma taką samą nazwę, co kanał serwera. Poza tym możliwe jest używanie kanałów serwera z kanałami requestera. W takim przypadku definicja kanału requestera na pozostałym końcu kanału żąda uruchomienia definicji kanału serwera. Serwer wysyła komunikaty do requestera. Serwer może także inicjować komunikację pod warunkiem, że zna nazwę połączenia kanału partnera.
Odbiornik	Kanał odbiorczy jest to kanał komunikatów, za pomocą którego menedżer kolejek odbiera komunikaty z innych menedżerów kolejek. Aby odbierać komunikaty przy użyciu kanału odbiorcy, należy także utworzyć na innym menedżerze kolejek kanał nadawcy lub kanał serwera, który ma taką samą nazwę, co kanał odbiorcy.
Requester	Kanał requestera to kanał komunikatów używany przez menedżer kolejek w celu odbierania komunikatów od innych menedżerów kolejek. Kanał requestera może utworzyć żądanie uruchomienia kierowane do zdefiniowanego kanału partnera na zdalnym elemencie końcowym połączenia. Jeśli kanałem partnera jest kanał serwera, kanał serwera zatwierdza żądanie uruchomienia i rozpoczyna wysyłanie komunikatów, z kolejki przesyłania określonej w definicji kanału serwera do kanału requestera. Jeśli kanał partnera jest kanałem nadawczym, kanał nadawczy akceptuje żądanie uruchomienia, ale później zamyka połączenie z requesterem. Następnie, kanał nadawczy uruchamia się, negocjuje sesję z kanałem requestera partnera i rozpoczyna wysyłanie komunikatów z kolejki przesyłania określonej w definicji kanału nadawczego. Ten drugi przypadek zasadniczo zapewnia mechanizm wywołania zwrotnego, ponieważ kanał requestera tworzy żądanie wywołania zwrotnego kierowane do kanału nadawczego.

Typ definicji kanału komunikatów	Opis
Nadawca klastra	Definicja kanału wysyłającego klastry (CLUSDR) definiuje wysyłający koniec kanału, z którego menedżer kolejek klastra może wysłać informacje klastra do jednego z pełnych repozytoriów. Kanał wysyłający klastry jest używany w celu powiadomienia repozytorium o wszelkich zmianach w statusie menedżera kolejek, na przykład o dodaniu lub usunięciu kolejki. Jest on również używany w celu przesyłania komunikatów. Menedżery pełnego repozytorium mają kanały wysyłające klastry, które wskazują sobie nawzajem. Używają ich do przekazywania między sobą informacji o zmianach w statusie klastra. Nie ma znaczenia, które pełne repozytorium wskazuje na definicję kanału CLUSDR menedżera kolejek. Po nawiązaniu kontaktu początkowego kolejne obiekty menedżera kolejek klastra są definiowane automatycznie zgodnie z wymaganiami tak, aby menedżer kolejek mógł wysłać informacje o klastrze do każdego pełnego repozytorium oraz komunikaty do każdego menedżera kolejek.
Odbiorca klastra	Definicja kanału odbierającego klastry (CLUSRCVR) definiuje odbierający koniec kanału, na którym menedżer kolejek klastra może odbierać komunikaty od innych menedżerów kolejek w klastrze. Kanałem odbierającym klastry mogą być również przesyłane informacje dotyczące klastra przeznaczone dla repozytorium. Definiując kanał odbierający klastry, menedżer kolejek wskazuje innym menedżerom kolejek klastra, że jest dostępny, aby otrzymywać komunikaty. Przynajmniej jeden kanał odbierający klastry jest potrzebny dla każdego menedżera kolejek klastra.

Należy zdefiniować obydwa końce każdego kanału, aby każdy koniec kanału miał definicję kanału. Obydwa końce kanału muszą być typami zgodnymi.

Możliwe są następujące kombinacje definicji kanałów:

- Kanał nadawcy-Kanał odbiorcy
- Kanał serwera-Kanał odbiorcy
- Kanał requestera-Kanał serwera
- Kanał requestera-Kanał nadawcy (wywołanie zwrotne)
- Kanał wysyłający klastry-Kanał odbierający klastry

Agenci kanału komunikatów

Każda utworzona definicja kanału należy do konkretnego menedżera kolejek. Menedżer kolejek może mieć kilka kanałów tego samego lub różnego typu. Na każdym końcu kanału znajduje się program, którym jest agent kanału komunikatów (MCA). Na jednym końcu kanału wywołujący agent MCA pobiera komunikaty z kolejki transmisji i wysyła je za pośrednictwem kanału. Na drugim końcu kanału odpowiadający agent MCA odbiera komunikaty i dostarcza je do zdalnego menedżera kolejek.

Program wywołujący MCA może zostać przypisany do kanału nadawcy, kanału serwera lub kanału requestera. Program odpowiadający MCA może zostać przypisany do dowolnego typu kanału komunikatów.

Produkt IBM MQ obsługuje następujące kombinacje typów kanałów na dwóch końcach połączenia:

Program wywołujący		Kierunek przepływu informacji	Program odpowiadający	
Typ kanału	Wymagane nastuchiwanie?		Wymagane nastuchiwanie?	Typ kanału
Nadawca	Nie	Program wywołujący do programu odpowiadającego	Tak	Odbiornik
Serwer	Nie	Program wywołujący do programu odpowiadającego	Tak	Odbiornik
Serwer	Nie	Program wywołujący do programu odpowiadającego	Tak	Requester
Requester	Nie	Program odpowiadający do programu wywołującego	Tak	Serwer
Requester	Tak	Program odpowiadający do programu wywołującego	Tak	Nadawca

Kanały MQI

Kanały MQI mogą być następującego typu:

Typ kanału MQI	Opis
Kanał połączenia serwera	Kanał połączenia serwera jest dwukierunkowym kanałem MQI, który jest używany w celu nawiązania połączenia między klientem IBM MQ i serwerem IBM MQ. Kanał połączenia serwera to koniec serwera kanału.
Połączenie klienta	Kanał połączenia klienta jest dwukierunkowym kanałem MQI, który jest używany w celu nawiązania połączenia między klientem IBM MQ i serwerem IBM MQ. Program IBM MQ Explorer także używa połączeń klienta, aby nawiązać połączenie ze zdalnymi menedżerami kolejek. Kanał połączenia klienta to koniec klienta kanału. Podczas tworzenia kanału połączenia klienta tworzony jest plik na komputerze udostępniającym menedżera kolejek. Plik połączenia klienta należy następnie skopiować na komputer klienta produktu IBM MQ.

Opcjonalnie można zezwolić agentowi kanału komunikatów (MCA) na przesyłanie komunikatów przy użyciu wielu wątków. Proces ten, nazywany *potokowaniem*, umożliwia agentowi MCA bardziej wydajne przesyłanie komunikatów przy mniejszej liczbie stanów oczekiwania, co zwiększa wydajność kanału. Każdy agent MCA jest ograniczony do maksymalnie dwóch wątków.

Do sterowania potokami służy parametr *PipeLineLength* w pliku *qm.ini*. Ten parametr jest dodawany do sekcji *Kanały*.

Uwaga: Potoki są skuteczne tylko dla kanałów TCP/IP.

Jeśli używane jest potokowanie, menedżery kolejek na obu końcach kanału muszą być skonfigurowane w taki sposób, aby długość *PipeLine* była większa niż 1.

Uwagi dotyczące wyjścia kanału

Potokowanie może spowodować niepowodzenie niektórych programów obsługi wyjścia, ponieważ:

- Wyjścia mogą nie być wywoływane szeregowo.
- Wyjścia mogą być wywoływane na przemian z różnych wątków.

Przed użyciem potokowania należy sprawdzić projekt programów obsługi wyjścia:

- Wyjścia muszą być wielobieżne na wszystkich etapach ich wykonywania.
- Jeśli używane są wywołania MQI, należy pamiętać, że nie można używać tego samego uchwytu MQI, gdy wyjście jest wywoływane z różnych wątków.




Rozważmy wyjście komunikatu, które otwiera kolejkę i używa jej uchwytu dla wywołań MQPUT we wszystkich kolejnych wywołaniach wyjścia. Działanie to kończy się niepowodzeniem w trybie potokowym, ponieważ wyjście jest wywoływane z różnych wątków. Aby uniknąć tego niepowodzenia, należy zachować uchwyt kolejki dla każdego wątku i sprawdzać identyfikator wątku przy każdym wywołaniu wyjścia.

korporacyjna

Produkt IBM MQ MQI clients używa kanałów MQI do komunikacji z serwerem.

Definicję kanału należy utworzyć zarówno na IBM MQ MQI client, jak i na serwerze po zakończeniu połączenia. Sposób tworzenia definicji kanałów opisano w sekcji [Definiowanie kanałów MQI](#).

Możliwe protokoły transmisji przedstawiono w poniższej tabeli:

Tabela 1. Protokoły transmisji dla kanałów MQI				
Platforma klienta	LU 6.2	TCP/IP	NetBIOS	SPX
 IBM i		Tak		
 Systemy AIX and Linux	Tak ¹	Tak		
 Windows	Tak	Tak	Tak	Tak

Uwaga:

1.  Produkt LU6.2 nie jest obsługiwany na następujących platformach:

- Linux (platforma POWER)
- Linux (platformax86-64)
- Linux (platformazSeries s390x)

Protokoły transmisji-kombinacja platformy IBM MQ MQI client i platformy serwera przedstawia możliwe kombinacje platformy IBM MQ MQI client i platformy serwera, przy użyciu tych protokołów transmisji.

Aplikacja IBM MQ w systemie IBM MQ MQI client może używać wszystkich wywołań MQI w taki sam sposób, jak w przypadku lokalnego menedżera kolejek. **MQCONN** lub **MQCONNX** wiąże aplikację IBM MQ z wybranym menedżerem kolejek, tworząc *uchwyt połączenia*. Inne wywołania używające tego uchwytu połączenia są następnie przetwarzane przez połączony menedżer kolejek. Komunikacja programu IBM MQ MQI client wymaga aktywnego połączenia między klientem i serwerem, w przeciwieństwie do komunikacji między menedżerami kolejek, która jest niezależna od połączenia i od czasu.

Protokół transmisji jest określany przy użyciu definicji kanału i nie ma wpływu na aplikację. Na przykład aplikacja Windows może łączyć się z jednym menedżerem kolejek za pośrednictwem protokołu TCP/IP i z innym menedżerem kolejek za pośrednictwem protokołu NetBIOS.

Zagadnienia związane z wydajnością

Używany protokół transmisji może mieć wpływ na wydajność systemu klienta i serwera IBM MQ. W niektórych sytuacjach, gdy transmisja jest powolna, można użyć kompresji kanału IBM MQ.

Nadawanie nazw obiektom IBM MQ


Konwencja nazewnictwa przyjęta dla obiektów IBM MQ zależy od obiektu. Nazwy komputerów i identyfikatory użytkowników używane z produktem IBM MQ również podlegają pewnym ograniczeniom dotyczącym nazewnictwa.

Każda instancja menedżera kolejek jest znana pod nazwą. Ta nazwa musi być unikalna w obrębie sieci połączonych ze sobą menedżerów kolejek, aby jeden menedżer kolejek mógł jednoznacznie zidentyfikować docelowy menedżer kolejek, do którego jest wysyłany dowolny komunikat.

W przypadku innych typów obiektów każdy obiekt ma powiązaną nazwę i może być przywoływany przez tę nazwę. Te nazwy muszą być unikalne w obrębie jednego menedżera kolejek i jednego typu obiektu. Na przykład może istnieć kolejka i proces o takiej samej nazwie, ale nie mogą istnieć dwie kolejki o takiej samej nazwie.

W systemie IBM MQ nazwy mogą mieć maksymalnie 48 znaków, z wyjątkiem *kanatów*, które mają maksymalnie 20 znaków. Więcej informacji na temat nadawania nazw obiektom IBM MQ zawiera sekcja [“Reguły nazewnictwa obiektów IBM MQ” na stronie 38](#).

Nazwy komputerów i identyfikatory użytkowników używane z produktem IBM MQ również podlegają pewnym ograniczeniom dotyczącym nazewnictwa:

- Upewnij się, że nazwa komputera nie zawiera spacji. IBM MQ nie obsługuje nazw maszyn zawierających spacje. Jeśli produkt IBM MQ jest instalowany na takim komputerze, nie można utworzyć żadnych menedżerów kolejek.
- W przypadku autoryzacji IBM MQ nazwy identyfikatorów użytkowników i grup nie mogą być dłuższe niż 20 znaków (spacje nie są dozwolone).
-  Serwer IBM MQ for Windows nie obsługuje połączenia IBM MQ MQI client, jeśli klient działa z identyfikatorem użytkownika, który zawiera znak @, na przykład abc@d.

Pojęcia pokrewne

[“IBM MQ nazwy plików” na stronie 41](#)

Każdy obiekt informacji o menedżerze kolejek, kolejce, definicji procesu, liście nazw, kanale, kanale połączenia klienckiego, programie nastuchującym, usłudze i uwierzytelnianiu IBM MQ jest reprezentowany przez plik. Ponieważ nazwy obiektów nie muszą być poprawnymi nazwami plików, menedżer kolejek w razie potrzeby przekształca nazwę obiektu w poprawną nazwę pliku.

Odsyłacze pokrewne

[“Reguły nazewnictwa obiektów IBM MQ” na stronie 38](#)

Nazwy obiektów IBM MQ mają maksymalną długość i jest w nich rozróżniana wielkość liter. Nie wszystkie znaki są obsługiwane dla każdego typu obiektu, a wiele obiektów ma reguły dotyczące unikalności nazw.

Reguły nazewnictwa obiektów IBM MQ

Nazwy obiektów IBM MQ mają maksymalną długość i jest w nich rozróżniana wielkość liter. Nie wszystkie znaki są obsługiwane dla każdego typu obiektu, a wiele obiektów ma reguły dotyczące unikalności nazw.

Istnieje wiele różnych typów obiektów IBM MQ, a obiekty każdego typu mogą mieć taką samą nazwę, ponieważ istnieją w oddzielnych przestrzeniach nazw obiektów: na przykład zarówno kolejka lokalna, jak i kanał nadawczy mogą mieć taką samą nazwę. Jednak obiekt nie może mieć tej samej nazwy co inny obiekt w tej samej przestrzeni nazw. Na przykład kolejka lokalna nie może mieć tej samej nazwy co kolejka modelowa, a kanał wysyłający nie może mieć tej samej nazwy co kanał odbierający.

Następujące obiekty IBM MQ istnieją w oddzielnych przestrzeniach nazw obiektów:


- Informacje uwierzytelniające
- Kanał
- Kanał klienta
- Program nasłuchujący
- Lista nazw
- Proces
- Kolejka
- Usługa
- Klasa pamięci masowej
- Subskrypcja
- Temat

Liczba znaków w nazwach obiektów

Zwykle nazwy obiektów IBM MQ mogą mieć długość do 48 znaków. Ta reguła ma zastosowanie do następujących obiektów:








- Informacje uwierzytelniające
- Klaster
- Program nasłuchujący
- Lista nazw
- Definicja procesu
- Kolejka
- Menedżer kolejek
- Usługa
- Subskrypcja
- Temat




Istnieją ograniczenia:

1.  W systemach z/OS menedżery kolejek mogą składać się maksymalnie z 4 znaków i mogą składać się wyłącznie z wielkich liter i cyfr.
2. Maksymalna długość nazw obiektów kanału i nazw kanałów połączenia klienckiego wynosi 20 znaków. Więcej informacji na temat kanałów zawiera sekcja [Definiowanie kanałów](#).
3. Łańcuchy tematów mogą mieć maksymalnie 10240 bajtów. We wszystkich nazwach obiektów IBM MQ rozróżniana jest wielkość liter.
4. Nazwy subskrypcji mogą mieć długość maksymalnie 10240 bajtów i mogą zawierać spacje.
5. Maksymalna długość nazw klas pamięci wynosi 8 znaków.
6. Maksymalna długość nazw struktur CF wynosi 12 znaków.

Znaki w nazwach obiektów

Poprawne znaki dla nazw obiektów IBM MQ to:

Znaków	Ograniczenia
Wielkie litery A-Z	<ul style="list-style-type: none"> • Brak
Małe litery a-z	<ul style="list-style-type: none"> • W skryptach MQSC nazwy zawierające małe litery muszą być ujęte w pojedynczy cudzysłów. Zapobiega to zwiżaniu małych liter na wielkie litery. • Systemy używające EBCDIC Katakana nie mogą używać małych liter a-z w nazwach obiektów. •  W systemach z/OS mogą występować ograniczenia dotyczące używania małych liter, na przykład nazwy menedżerów kolejek nie mogą zawierać małych liter. •  W systemach IBM i podczas korzystania z komend CL nazwy zawierające małe litery muszą być ujęte w pojedynczy cudzysłów. Zapobiega to zwiżaniu małych liter na wielkie litery.
Cyfry 0-9	<ul style="list-style-type: none"> • Brak
Kropka (.)	<ul style="list-style-type: none"> • Brak
Podkreślenie (_)	<ul style="list-style-type: none"> •  Brak •  Należy unikać używania nazw z początkowymi lub końcowymi znakami podkreślenia, ponieważ nie mogą one być obsługiwane przez operacje IBM MQ for z/OS i panele sterowania.
Prawy ukośnik (/)	<ul style="list-style-type: none"> •  W systemach Windows pierwszym znakiem nazwy menedżera kolejek nie może być ukośnik. •  W systemach IBM i, gdy używane są komendy CL, nazwy zawierające ukośnik należy ująć w pojedynczy cudzysłów. •  Brak

Znaków	Ograniczenia
Znak procentu (%)	<ul style="list-style-type: none"> •  Brak •  Jeśli produkt RACF jest używany jako zewnętrzny menedżer zabezpieczeń dla produktu IBM MQ for z/OS, nie należy używać wartości % w nazwach obiektów, ponieważ nazwy te nie są uwzględniane podczas sprawdzania zabezpieczeń, gdy używane są profile ogólne produktu RACF . •  W systemach IBM i , gdy używane są komendy CL, nazwy zawierające znak procentu muszą być ujęte w pojedynczy cudzysłów.

Istnieją również pewne ogólne zasady dotyczące znaków w nazwach obiektów:

1. Czołowe lub wewnętrzne odstępy nie są dozwolone.
2. Znaki języka narodowego są niedozwolone.
3. Każda nazwa, która jest mniejsza niż pełna długość pola, może być dopełniona do prawej strony odstępami. Wszystkie nazwy skrócone zwracane przez menedżer kolejek są zawsze dopełniane po prawej stronie odstępami.


Nazwy kolejek

Nazwa kolejki składa się z dwóch części:

- Nazwa menedżera kolejek
- Nazwa lokalna kolejki znana temu menedżerowi kolejek

Każda część nazwy kolejki ma długość 48 znaków.

Aby odwołać się do kolejki lokalnej, można pominąć nazwę menedżera kolejek (zastępując ją znakami odstępu lub używając znaku o kodzie zero). Jednak wszystkie nazwy kolejek zwracane do programu przez program IBM MQ zawierają nazwę menedżera kolejek.

 Kolejka współużytkowana dostępna dla dowolnego menedżera kolejek w jego grupie współużytkowania kolejek nie może mieć takiej samej nazwy, jak dowolna niewspółużytkowana kolejka lokalna w tej samej grupie współużytkowania kolejek. To ograniczenie pozwala uniknąć możliwości, że aplikacja omyłkowo otworzy kolejkę współużytkowaną, gdy zamierza otworzyć kolejkę lokalną, lub odwrotnie. Kolejki współużytkowane i grupy współużytkowania kolejek są dostępne tylko w systemie IBM MQ for z/OS.

Aby odwołać się do kolejki zdalnej, program musi zawierać nazwę menedżera kolejek w pełnej nazwie kolejki lub musi istnieć lokalna definicja kolejki zdalnej.

Jeśli aplikacja używa nazwy kolejki, nazwa ta może być albo nazwą kolejki lokalnej (lub aliasem), albo nazwą lokalnej definicji kolejki zdalnej, ale aplikacja nie musi wiedzieć, która z nich, chyba że musi pobrać komunikat z kolejki (gdy kolejka musi być lokalna). Gdy aplikacja otwiera obiekt kolejki, wywołanie MQOPEN wykonuje funkcję tłumaczenia nazw, aby określić, w której kolejce wykonać kolejne operacje. Znaczenie tego faktu jest takie, że aplikacja nie ma wbudowanej zależności od konkretnych kolejek definiowanych w określonych miejscach w sieci menedżerów kolejek. Dlatego też, jeśli administrator systemu przemieszcza kolejki w sieci i zmienia ich definicje, aplikacje korzystające z tych kolejek nie muszą być zmieniane.

Zastrzeżone nazwy obiektów

Nazwy obiektów rozpoczynające się od łańcucha SYSTEM. są zarezerwowane dla obiektów zdefiniowanych przez menedżer kolejek. Do zmiany tych definicji obiektów w zależności od instalacji

można użyć komend **Alter**, **Define** i **Replace** . Nazwy zdefiniowane dla parametru IBM MQ są wymienione w sekcji Nazwy kolejek.

 W systemie IBM MQ for z/OS nazwa struktury aplikacji CF CSQSYSAPPL jest zarezerwowana.




Pojęcia pokrewne

Nazwa instalacji w systemie AIX, Linux, and Windows

IBM MQ nazwy plików

Każdy obiekt informacji o menedżerze kolejek, kolejce, definicji procesu, liście nazw, kanale, kanale połączenia klienckiego, programie nastuchującym, usłudze i uwierzytelnianiu IBM MQ jest reprezentowany przez plik. Ponieważ nazwy obiektów nie muszą być poprawnymi nazwami plików, menedżer kolejek w razie potrzeby przekształca nazwę obiektu w poprawną nazwę pliku.

Domyślna ścieżka do katalogu menedżera kolejek jest następująca:

- Przedrostek, który jest zdefiniowany w informacjach konfiguracyjnych systemu IBM MQ :
 -   W systemie AIX and Linux domyślnym przedrostkiem jest `/var/mqm`. Jest on konfigurowany w sekcji `DefaultPrefix` pliku konfiguracyjnego `mqs.ini` .
 -  W 32-bitowych systemach Windows domyślnym przedrostkiem jest `C:\Program Files (x86)\IBM\WebSphere MQ`. W 64-bitowych systemach Windows domyślnym przedrostkiem jest `C:\Program Files\IBM\MQ`. Zarówno w przypadku instalacji 32-bitowych, jak i 64-bitowych, katalogi danych są instalowane w katalogu `C:\ProgramData\IBM\MQ`. Jest on konfigurowany w sekcji `DefaultPrefix` pliku konfiguracyjnego `mqs.ini` .

Jeśli jest to możliwe, przedrostek można zmienić za pomocą strony właściwości IBM MQ w Eksploratorze IBM MQ . W przeciwnym razie należy ręcznie zmodyfikować plik konfiguracyjny `mqs.ini` .

- Nazwa menedżera kolejek jest przekształcana w poprawną nazwę katalogu. Na przykład menedżer kolejek:

```
queue.manager
```

będzie reprezentowana jako:

```
queue!manager
```

Proces ten jest nazywany *transformacją nazwy*.

W programie IBM MQ można nadać menedżerowi kolejek nazwę zawierającą maksymalnie 48 znaków.

Na przykład można nadać nazwę menedżerowi kolejek:

```
QUEUE.MANAGER.ACCOUNTING.SERVICES
```

Jednak każdy menedżer kolejek jest reprezentowany przez plik i istnieją ograniczenia dotyczące maksymalnej długości nazwy pliku oraz znaków, które mogą być używane w nazwie. W rezultacie nazwy plików reprezentujących obiekty są automatycznie transformowane w celu spełnienia wymagań systemu plików.

Reguły rządzące transformacją nazwy menedżera kolejek są następujące:

1. Transformuj pojedyncze znaki:
 - Od. do!
 - Od/do &
2. Jeśli nazwa nadal nie jest poprawna:

- a. Obetnij do ośmiu znaków
- b. Dodaj trzyznakowy przyrostek liczbowy

Na przykład przyjmując domyślny przedrostek i menedżer kolejek o nazwie queue . manager:

- **Windows** W systemie Windows z systemem plików NTFS lub FAT32 nazwa menedżera kolejek jest następująca:

```
C:\Program Files\IBM\MQ\mqgrs\queue!manager
```

- **Windows** W systemie Windows z FAT nazwa menedżera kolejek ma postać:

```
C:\Program Files\IBM\MQ\mqgrs\queue!ma
```

- **Linux** **AIX** W systemie AIX and Linux nazwa menedżera kolejek jest następująca:

```
/var/mqm/mqgrs/queue!manager
```

Algorytm transformacji rozróżnia również nazwy różniące się tylko wielkością liter w systemach plików, w których nie jest rozróżniana wielkość liter.

Transformacja nazwy obiektu

Nazwy obiektów nie muszą być poprawnymi nazwami systemu plików. Może być konieczne przekształcenie nazw obiektów. Używana metoda jest inna niż w przypadku nazw menedżerów kolejek, ponieważ chociaż na każdym komputerze istnieje tylko kilka nazw menedżerów kolejek, dla każdego menedżera kolejek może istnieć duża liczba innych obiektów. W systemie plików reprezentowane są kolejki, definicje procesów, listy nazw, kanały, kanały połączeń klienta, obiekty nasłuchiwanie, usługi i informacje o uwierzytelnianiu.

Gdy nowa nazwa jest generowana przez proces transformacji, nie ma prostej relacji z oryginalną nazwą obiektu. Za pomocą komendy **dspmqls** można dokonać konwersji między rzeczywistymi i przekształconymi nazwami obiektów.

Odsyłacze pokrewne

dspmqls (nazwy plików wyświetlanych)

Informacje pokrewne

Sekcja AllQueueManagers w pliku mqz.ini

IBM i Nazwy obiektów w systemie IBM i

Menedżer kolejek ma powiązaną bibliotekę menedżera kolejek, która ma unikalną nazwę. Nazwy menedżerów kolejek i obiektów mogą wymagać transformacji w celu spełnienia wymagań systemu plików IBM i Integrated File System (IFS).

Po utworzeniu menedżera kolejek program IBM MQ wiąże z nim bibliotekę menedżera kolejek. Tej bibliotece menedżera kolejek nadawana jest unikalna nazwa, nie dłuższa niż 10 znaków, w dużej mierze oparta na nazwie menedżera kolejek zdefiniowanej przez użytkownika. Zarówno menedżer kolejek, jak i biblioteka menedżera kolejek są umieszczane w katalogu, który jest również oparty na nazwie menedżera kolejek z przedrostkiem /QIBM/UserData/mqm. Poniżej przedstawiono przykład menedżera kolejek, biblioteki menedżera kolejek i katalogu:

Nazwa menedżera kolejek	Pomarańczowy
Nazwa biblioteki menedżera kolejek	QMORANGE
Katalog	/QIBM/UserData/mqm/ORANGE

Wszystkie nazwy menedżerów kolejek i biblioteki menedżerów kolejek są zapisywane w sekcjach w pliku `/QIBM/UserData/mqm/mqs.ini`.

IBM MQ katalogi i pliki IFS

System plików IBM i Integrated File System (IFS) jest intensywnie używany przez system IBM MQ do przechowywania danych. Więcej informacji na temat systemu plików IFS zawiera sekcja *Integrated File System -wprowadzenie*.

Każdy obiekt IBM MQ, na przykład kanał lub menedżer kolejek, jest reprezentowany przez plik. Ponieważ nazwy obiektów nie muszą być poprawnymi nazwami plików, menedżer kolejek w razie potrzeby przekształca nazwę obiektu w poprawną nazwę pliku.

Ścieżka do katalogu menedżera kolejek jest tworzona z następujących elementów:

- Przedrostek, który jest zdefiniowany w pliku konfiguracyjnym menedżera kolejek `qm.ini`. Domyślnym przedrostkiem jest `/QIBM/UserData/mqm`.
- Literał `qmgrs`.
- Zakodowana nazwa menedżera kolejek, która jest nazwą menedżera kolejek przekształconą w poprawną nazwę katalogu. Na przykład menedżer kolejek `queue/manager` jest reprezentowany przez wartość `queue&manager`.

Ten proces jest nazywany transformacją nazwy.

Transformacja nazwy menedżera kolejek IFS

W programie IBM MQ można nadać menedżerowi kolejek nazwę zawierającą maksymalnie 48 znaków.

Na przykład można nadać nazwę menedżerowi kolejek `QUEUE/MANAGER/ACCOUNTING/SERVICES`. W ten sam sposób, w jaki biblioteka jest tworzona dla każdego menedżera kolejek, każdy menedżer kolejek jest również reprezentowany przez plik. Ze względu na warianty punktów kodowych w kodzie EBCDIC istnieją ograniczenia dotyczące znaków, które mogą być używane w nazwie. W rezultacie nazwy plików IFS reprezentujących obiekty są automatycznie transformowane w celu spełnienia wymagań systemu plików.

Na podstawie przykładu menedżera kolejek o nazwie `queue/manager`, transformując znak `/` na `&i` przyjmując domyślny przedrostek, nazwa menedżera kolejek w programie IBM MQ for IBM i zmienia się na `/QIBM/UserData/mqm/qmgrs/queue&manager`.

Transformacja nazwy obiektu

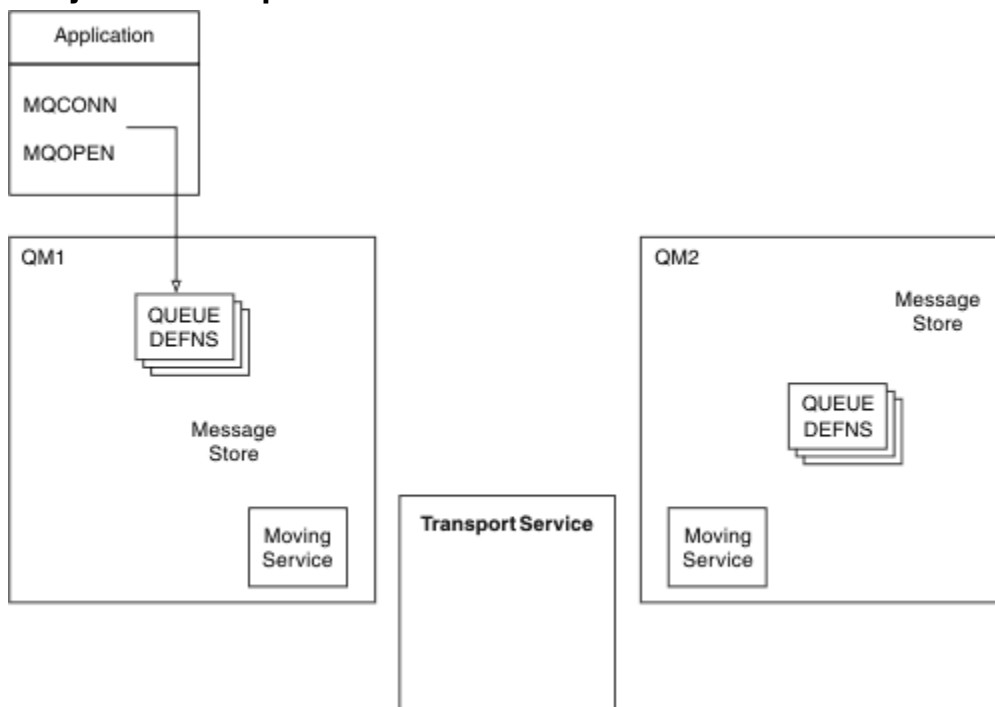
Nazwy obiektów nie muszą być poprawnymi nazwami systemów plików, dlatego nazwy obiektów mogą wymagać transformacji. Używana metoda jest inna niż w przypadku nazw menedżerów kolejek, ponieważ chociaż dla każdego komputera istnieje tylko kilka nazw menedżerów kolejek, dla każdego menedżera kolejek może istnieć duża liczba innych obiektów. W systemie plików reprezentowane są tylko definicje procesów, kolejki i listy nazw. Nie ma to wpływu na kanały.

Gdy nowa nazwa jest generowana przez proces transformacji, nie ma prostej relacji z oryginalną nazwą obiektu. Aby wyświetlić przetransformowane nazwy obiektów IBM MQ, można użyć komendy `DSPMQOBJN`.

Rozproszone kolejkowanie i klastry

Kolejkowanie rozproszone oznacza wysyłanie komunikatów z jednego menedżera kolejek do innego. Odbierający menedżer kolejek może znajdować się na tym samym lub innym komputerze, w pobliżu lub po drugiej stronie świata. Może on działać na tej samej platformie co lokalny menedżer kolejek lub na dowolnej platformie obsługiwanej przez produkt IBM MQ. Można ręcznie zdefiniować wszystkie połączenia w rozproszonym środowisku kolejkowania lub utworzyć klastr i pozwolić produktowi IBM MQ na zdefiniowanie większości szczegółów połączenia.

kolejkowanie rozproszone



Rysunek 4. Przegląd komponentów rozproszonego kolejkowania

Na poprzednim rysunku:

- Aplikacja używa wywołania MQCONN do nawiązania połączenia z menedżerem kolejek. Następnie aplikacja korzysta z wywołania MQOPEN w celu otwarcia kolejki, aby mogła umieścić komunikaty w kolejce.
- Każdy menedżer kolejek ma definicję dla każdej kolejki. Może mieć definicje *kolejek lokalnych* (udostępnianych przez ten menedżer kolejek) i definicje *kolejek zdalnych* (udostępnianych przez inne menedżery kolejek).
- Jeśli komunikaty są przeznaczone dla kolejki zdalnej, menedżer kolejek lokalnych przechowuje je w *kolejce transmisji*, która utrzuwa je w składnicy komunikatów, dopóki nie zostaną przekazane do zdalnego menedżera kolejek.
- Każdy menedżer kolejek zawiera oprogramowanie komunikacyjne nazywane *usługą przenoszenia*, które jest używane przez menedżer kolejek do komunikowania się z innymi menedżerami kolejek.
- *Usługa transportowa* jest niezależna od menedżera kolejek i może mieć jedną z następujących wartości (w zależności od platformy):
 - Zaawansowana komunikacja program-program (SNA APPC) architektury systemów sieciowych
 - protokół TCP/IP (Transmission Control Protocol/Internet Protocol)
 - Podstawowy sieciowy system wejścia/wyjścia (NetBIOS)
 - Sekwencyjna wymiana pakietów (SPX)

Komponenty wymagane do wysłania komunikatu

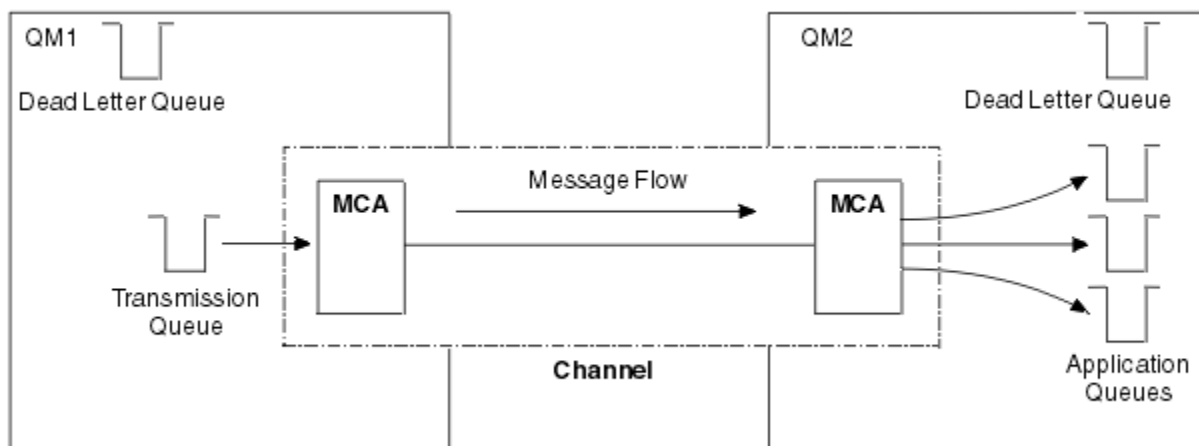
Jeśli komunikat ma zostać wysłany do zdalnego menedżera kolejek, lokalny menedżer kolejek wymaga definicji dla *kolejki transmisji* i *kanatu*. Kanał jest jednokierunkowym łączem komunikacyjnym między dwoma menedżerami kolejek. Może on przenosić komunikaty przeznaczone dla dowolnej liczby kolejek w zdalnym menedżerze kolejek.

Każdy koniec kanału ma oddzielną definicję, definiującą go na przykład jako koniec wysyłający lub odbierający. Prosty kanał składa się z definicji kanału *nadawczego* w menedżerze kolejek lokalnych i definicji kanału *odbiorczego* w menedżerze kolejek zdalnych. Te dwie definicje muszą mieć taką samą nazwę i razem tworzą jeden kanał.

Oprogramowanie obsługujące wysyłanie i odbieranie komunikatów jest nazywane *agentem kanału komunikatów* (Message Channel Agent-MCA). Na każdym końcu kanału znajduje się *agent kanału komunikatów* (MCA).

Każdy menedżer kolejek powinien mieć *kolejkę niedostarczonych komunikatów* (zwaną również *kolejką niedostarczonych komunikatów*). Komunikaty są umieszczane w tej kolejce, jeśli nie można ich dostarczyć do miejsca docelowego.

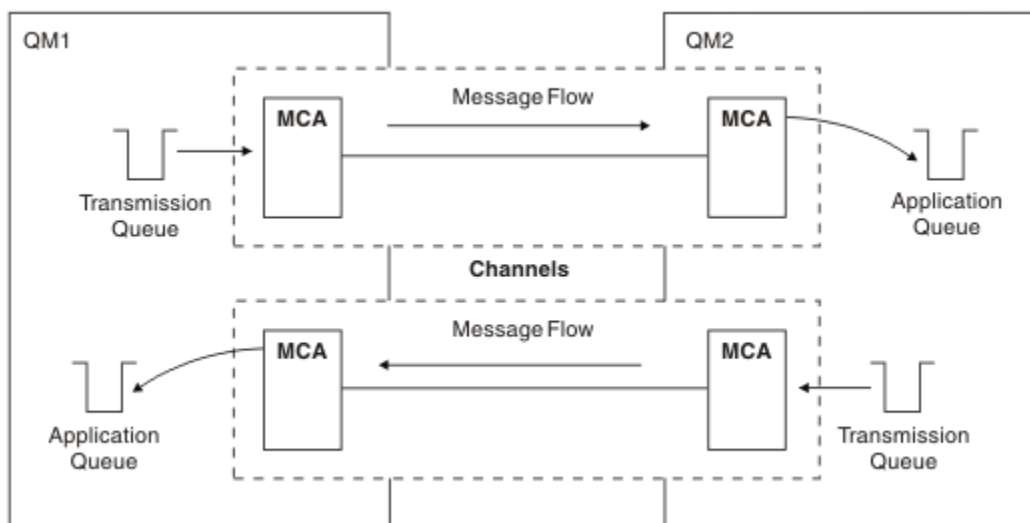
Poniższy rysunek przedstawia relacje między menedżerami kolejek, kolejkami transmisji, kanałami i konsolami HMC:



Rysunek 5. Wysyłanie komunikatów

Komponenty wymagane do zwrócenia komunikatu

Jeśli aplikacja wymaga, aby komunikaty były zwracane ze zdalnego menedżera kolejek, należy zdefiniować inny kanał, który będzie działał w odwrotnym kierunku między menedżerami kolejek, jak pokazano na poniższym rysunku:



Rysunek 6. Wysyłanie komunikatów w obu kierunkach

Klastry

Zamiast ręcznego definiowania wszystkich połączeń w rozproszonym środowisku kolejkowania, można zgrupować zestaw menedżerów kolejek w klastrze. W takim przypadku menedżery kolejek mogą udostępnić kolejki, które udostępniają innym menedżerom kolejek w klastrze, bez konieczności podawania jawnych definicji kanałów, definicji kolejek zdalnych lub kolejek transmisji dla każdego

miejsca docelowego. Każdy menedżer kolejek w klastrze ma pojedynczą kolejkę transmisji, która przesyła komunikaty do dowolnego innego menedżera kolejek w klastrze. Dla każdego menedżera kolejek należy zdefiniować tylko jeden kanał odbiorczy klastra i jeden kanał nadawczy klastra. Dodatkowe kanały są automatycznie zarządzane przez klastr.

Klient IBM MQ może nawiązać połączenie z menedżerem kolejek, który jest częścią klastra, tak samo jak każdy inny menedżer kolejek. Podobnie jak w przypadku ręcznego konfigurowania rozproszonego kolejkowania, wywołanie MQPUT służy do umieszczania komunikatu w kolejce w dowolnym menedżerze kolejek. Wywołanie MQGET służy do pobierania komunikatów z kolejki lokalnej.

Menedżery kolejek na platformach obsługujących klastry nie muszą być częścią klastra. Można kontynuować ręczne konfigurowanie rozproszonego kolejkowania, a także (lub nie) korzystanie z klastrów.

Korzyści z używania klastrów

Technologia klastrowa ma dwie podstawowe zalety:

- Klastry upraszczają administrowanie sieciami IBM MQ, które zwykle wymagają wielu definicji obiektów dla kanałów, kolejek wyjściowych i kolejek zdalnych. Ta sytuacja występuje szczególnie w dużych, potencjalnie zmieniających się sieciach, w których wiele menedżerów kolejek musi być ze sobą połączonych. Ta architektura jest szczególnie trudna do skonfigurowania i aktywnej obsługi.
- Klastry mogą być używane do rozdzielania obciążenia związanego z ruchem komunikatów między kolejki i menedżery kolejek w klastrze. Taka dystrybucja umożliwi rozdzielenie obciążenia komunikatów pojedynczej kolejki między równoważne instancje tej kolejki znajdujące się w wielu menedżerach kolejek. Ten podział obciążenia może być używany w celu uzyskania większej odporności na awarie systemu oraz w celu zwiększenia wydajności skalowania szczególnie aktywnych przepływów komunikatów w systemie. W takim środowisku każda instancja kolejki rozproszonej wykorzystuje aplikacje przetwarzające komunikaty. Więcej informacji na ten temat zawiera sekcja [Używanie klastrów do zarządzania obciążeniem](#).

Sposób kierowania komunikatów w klastrze

Klastrę można traktować jako sieć menedżerów kolejek obsługiwanych przez sumiennego administratora systemów. Za każdym razem, gdy zdefiniowana jest kolejka klastra, administrator systemu automatycznie tworzy odpowiednie definicje kolejek zdalnych w zależności od potrzeb w innych menedżerach kolejek.

Nie trzeba tworzyć definicji kolejek transmisji, ponieważ produkt IBM MQ udostępnia kolejkę transmisji w każdym menedżerze kolejek w klastrze. Ta pojedyncza kolejka transmisji może być używana do przenoszenia komunikatów do dowolnego innego menedżera kolejek w klastrze. Użytkownik nie jest ograniczony do korzystania z pojedynczej kolejki transmisji. Menedżer kolejek może używać wielu kolejek transmisji w celu oddzielenia komunikatów przesyłanych do każdego menedżera kolejek w klastrze. Zwykle menedżer kolejek używa pojedynczej kolejki transmisji klastra. Atrybut menedżera kolejek DEFCLXQ można zmienić w taki sposób, aby menedżer kolejek używał innej kolejki transmisji klastra dla każdego menedżera kolejek w klastrze. Kolejki transmisji klastra można również zdefiniować ręcznie.

Wszystkie menedżery kolejek, które dołączają do klastra, zgadzają się działać w ten sposób. Wysyłają oni informacje o sobie i o kolejkach, które udostępniają, oraz otrzymują informacje o innych elementach klastra.

Aby upewnić się, że żadne informacje nie zostaną utracone, gdy menedżer kolejek stanie się niedostępny, należy określić dwa menedżery kolejek w klastrze, które będą działać jako *pełne repozytoria*. Te menedżery kolejek przechowują pełny zestaw informacji o wszystkich menedżerach kolejek i kolejkach w klastrze. Wszystkie pozostałe menedżery kolejek w klastrze przechowują tylko informacje o tych menedżerach kolejek i kolejkach, z którymi wymieniają się komunikatami. Te menedżery kolejek są nazywane *częściowymi repozytoriami*. Więcej informacji na ten temat zawiera sekcja ["Repozytorium klastra"](#) na stronie 57.

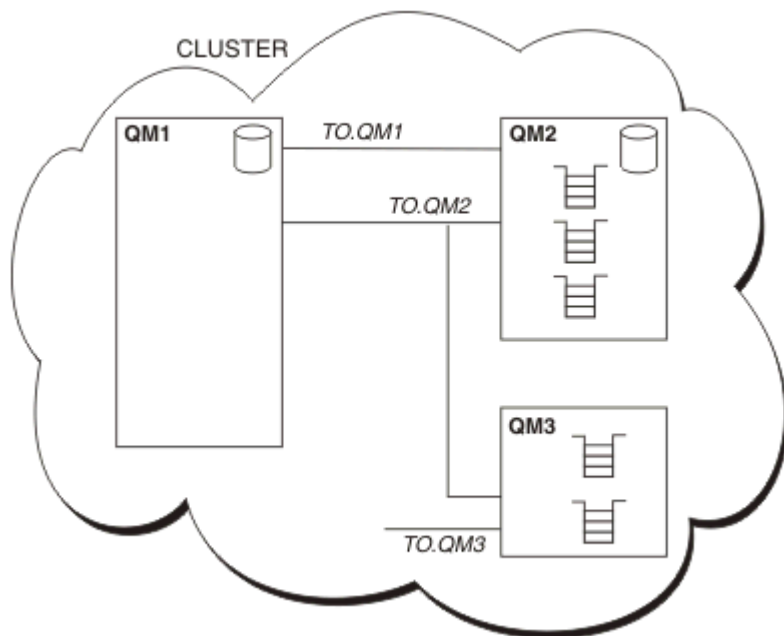
Aby menedżer kolejek stał się częścią klastra, musi mieć dwa kanały: kanał nadawczy klastra i kanał odbiorczy klastra:

- Kanał nadawczy klastra jest kanałem komunikacyjnym podobnym do kanału nadawczego. Należy ręcznie utworzyć jeden kanał nadawczy klastra w menedżerze kolejek, aby połączyć go z pełnym repozytorium, które jest już elementem klastra.

- Kanał odbiorczy klastra jest kanałem komunikacyjnym, takim jak kanał odbiorczy. Należy ręcznie utworzyć jeden kanał odbiorczy klastra. Kanał działa jako mechanizm odbierania komunikacji klastra przez menedżer kolejek.

Wszystkie inne kanały, które są wymagane do komunikacji między tym menedżerem kolejek i innymi elementami klastra, są tworzone automatycznie.

Poniższy rysunek przedstawia komponenty klastra o nazwie CLUSTER:



Rysunek 7. Klaster menedżerów kolejek

- Komenda CLUSTER zawiera trzy menedżery kolejek: QM1, QM2 i QM3.
- QM1 i QM2 udostępniają pełne repozytoria informacji o menedżerach kolejek i kolejkach w klastrze.
- QM2 i QM3 udostępniają niektóre kolejki klastra, to znaczy kolejki, które są dostępne dla dowolnego innego menedżera kolejek w klastrze.
- Każdy menedżer kolejek ma kanał odbiorczy klastra o nazwie TO.qmgr , w którym może odbierać komunikaty.
- Każdy menedżer kolejek ma również kanał nadawczy klastra, w którym może wysyłać informacje do jednego z menedżerów kolejek repozytorium.
- QM1 i QM3 są wysyłane do repozytorium o godz. QM2 i QM2 są wysyłane do repozytorium o godz. QM1.

Komponenty kolejkowania rozproszonego

Komponenty kolejkowania rozproszonego to kanały komunikatów, agenty kanału komunikatów, kolejki transmisji, inicjatory i programy nasłuchujące kanału oraz programy obsługi wyjścia kanału. Definicja każdego końca kanału komunikatów może być jednego z kilku typów.

Kanały komunikatów to kanały, które przenoszą komunikaty z jednego menedżera kolejek do innego. Nie należy mylić kanałów komunikatów z kanałami MQI. Istnieją dwa typy kanału MQI: połączenie serwera (SVRCONN) i połączenie klienta (CLNTCONN). Więcej informacji na ten temat zawiera sekcja [Kanały](#).

Definicja każdego końca kanału komunikatów może być jednego z następujących typów:

- Nadawca (SDR)
- Dziennik (RCVR)
- Serwer (SVR)

- Requester (RQSTR)
- Nadawca klastra (CLUSSDR)
- Odbiornik klastra (CLUSRCVR)

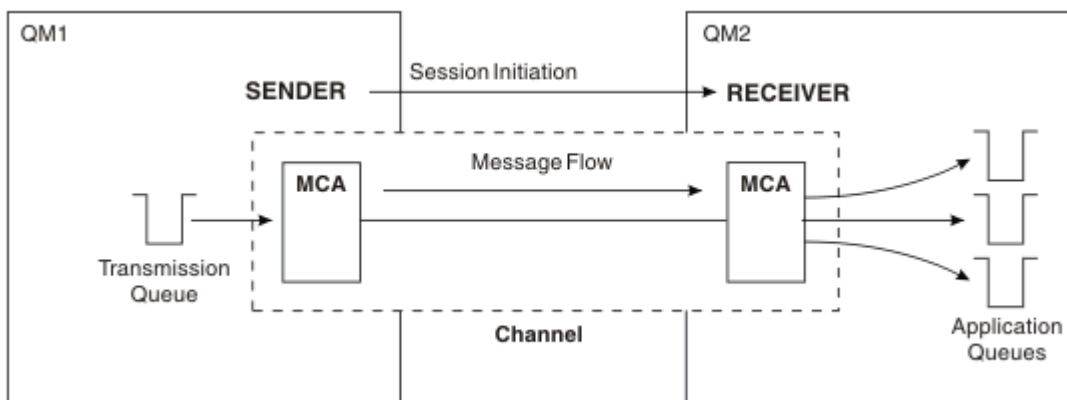
Kanał komunikatów jest definiowany przy użyciu jednego z tych typów zdefiniowanych na jednym końcu i zgodnego typu na drugim końcu. Możliwe kombinacje to:

- Kanał nadawcy-Kanał odbiorcy
- Kanał requestera-Kanał serwera
- Kanał requestera-Kanał nadawcy (wywołanie zwrotne)
- Kanał serwera-Kanał odbiorcy
- Nadawca klastra-odbiorca klastra

Szczegółowe instrukcje dotyczące tworzenia kanału nadawczego i odbiorczego znajdują się w sekcji Definiowanie kanałów. Przykłady parametrów wymaganych do skonfigurowania kanałów nadawczych i odbiorczych zawiera sekcja Przykładowe informacje konfiguracyjne odpowiednia dla danej platformy. Parametry wymagane do zdefiniowania kanału dowolnego typu zawiera sekcja DEFINE CHANNEL.

Kanał nadawcy - kanał odbiorcy

Nadawca w jednym systemie uruchamia kanał, aby mógł wysyłać komunikaty do innego systemu. Nadawca żąda uruchomienia odbiornika na drugim końcu kanału. Nadawca wysyła komunikaty ze swojej kolejki transmisji do odbiorcy. Odbiorca umieszcza komunikaty w kolejce docelowej. Rysunek 8 na stronie 48 ilustruje to.

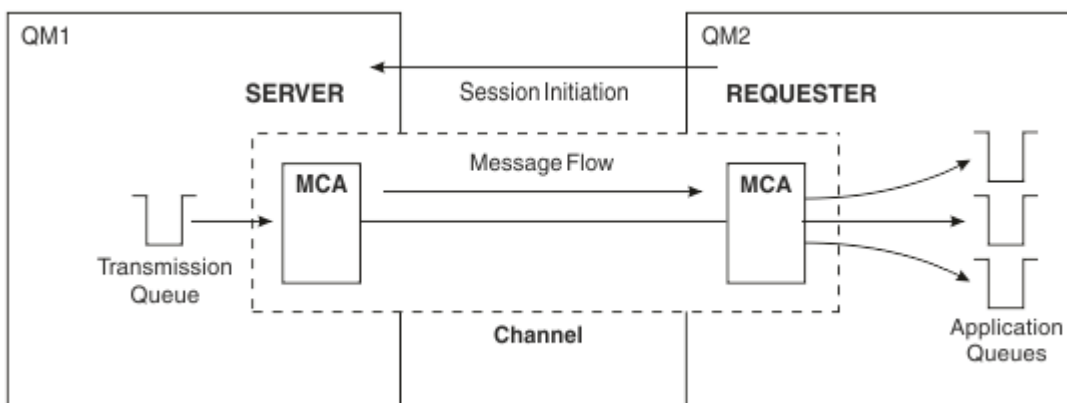


Rysunek 8. Kanał nadawcy-odbiorcy

Kanał requestera - kanał serwera

Requester w jednym systemie uruchamia kanał, aby mógł odbierać komunikaty z innego systemu. Requester żąda uruchomienia serwera na drugim końcu kanału. Serwer wysyła komunikaty do requestera z kolejki transmisji zdefiniowanej w jego definicji kanału.

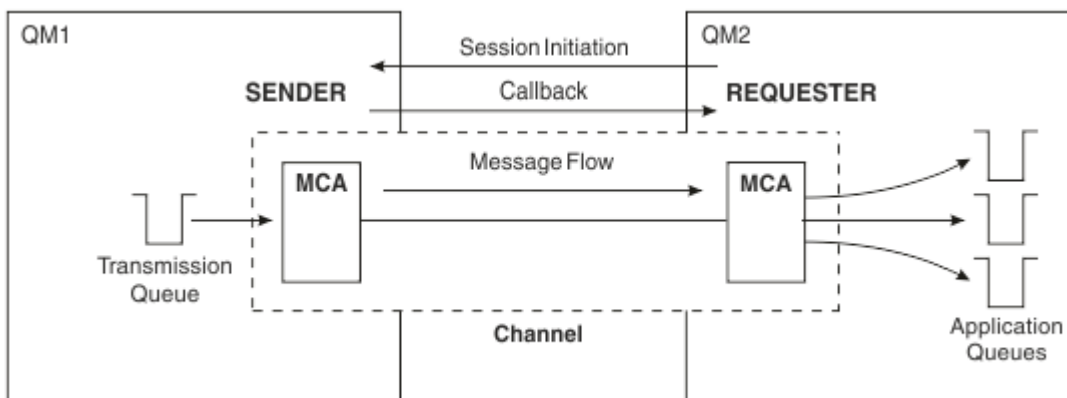
Kanał serwera może również inicjować komunikację i wysyłać komunikaty do requestera. Dotyczy to tylko *pełnych* serwerów, czyli kanałów serwera, które mają nazwę połączenia partnera określoną w definicji kanału. Pełny serwer może zostać uruchomiony przez requester lub może zainicjować komunikację z requestersiem.



Rysunek 9. Requester-kanal serwera

Kanał requestera - kanał nadawcy

Requester uruchamia kanał, a nadawca kończy połączenie. Następnie nadawca restartuje komunikację zgodnie z informacjami w jej definicji kanału (nazywanej *wywołaniem zwrotnym*). Wysyła on komunikaty z kolejki transmisji do requestera.



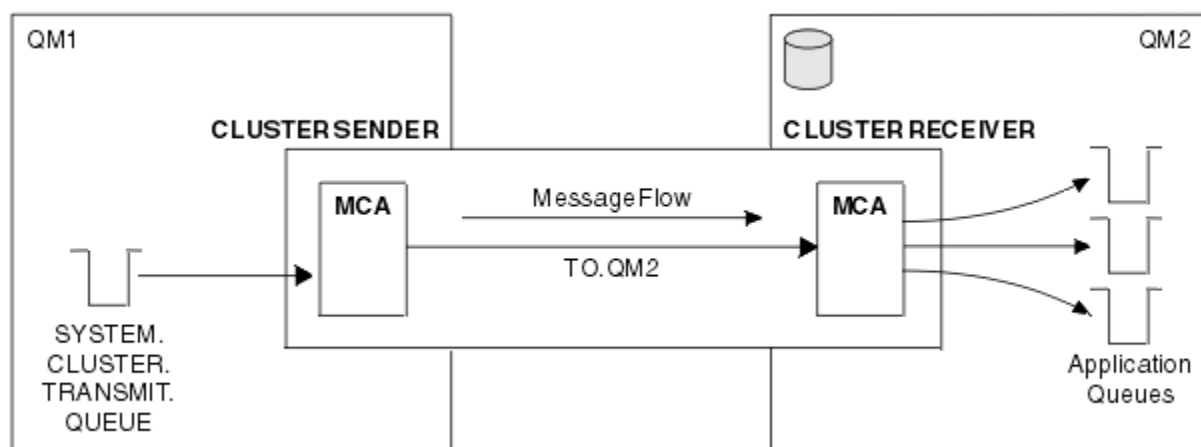
Rysunek 10. Kanał nadawczy requestera

Kanał odbiorczy serwera

Przypomina to nadawcę, ale ma zastosowanie tylko do *pełnych* serwerów, czyli kanałów serwera, które mają nazwę połączenia partnera określoną w definicji kanału. Uruchomienie kanału musi być zainicjowane po stronie serwera łącza. Ilustracja tego przykładu jest podobna do ilustracji w sekcji [Rysunek 8](#) na stronie [48](#).

Kanały wysyłające klastry

W klastrze każdy menedżer kolejek ma kanał nadawczy klastra, w którym może wysyłać informacje o klastrze do jednego z menedżerów kolejek pełnego repozytorium. Menedżery kolejek mogą również wysyłać komunikaty do innych menedżerów kolejek w kanałach nadawczych klastra.



Rysunek 11. Kanał nadawczy klastra

Kanały odbierające klastry

W klastrze każdy menedżer kolejek ma kanał odbiorczy klastra, w którym może odbierać komunikaty i informacje o klastrze. Ilustracja tego przykładu jest podobna do ilustracji w sekcji [Rysunek 11](#) na stronie 50.

Kolejki niedostarczonych komunikatów

Kolejka niedostarczonych komunikatów (lub kolejka niedostarczonych komunikatów) jest kolejką, do której wysyłane są komunikaty, jeśli nie mogą być kierowane do właściwego miejsca docelowego. Każdy menedżer kolejek zwykle ma kolejkę niedostarczonych komunikatów.

Kolejka niedostarczonych komunikatów (DLQ), nazywana czasem *kolejką niedostarczonych komunikatów*, jest kolejką wstrzymującą dla komunikatów, które nie mogą zostać dostarczone do ich kolejek docelowych, na przykład dlatego, że kolejka nie istnieje lub jest pełna. Kolejki niedostarczonych komunikatów są również używane na wysyłającym końcu kanału w przypadku błędów konwersji danych. Każdy menedżer kolejek w sieci zwykle ma kolejkę lokalną, która ma być używana jako kolejka niedostarczonych komunikatów, dzięki czemu komunikaty, których nie można dostarczyć do poprawnego miejsca docelowego, mogą być przechowywane w celu późniejszego odtworzenia.

Komunikaty mogą być umieszczane w kolejce DLQ przez menedżery kolejek, agenty kanału komunikatów (MCA) i aplikacje. Wszystkie komunikaty w kolejce DLQ muszą być poprzedzone *naświetlonym niedostarczonego komunikatu* MQDLH. Pole *Przyczyna* struktury MQDLH zawiera kod przyczyny, który identyfikuje przyczynę, dla której komunikat znajduje się w kolejce DLQ.

Zwykle należy zdefiniować kolejkę niedostarczonych komunikatów dla każdego menedżera kolejek. Jeśli nie, a agent MCA nie może umieścić komunikatu, pozostaje w kolejce transmisji i kanał jest zatrzymany. Również w przypadku szybkich komunikatów nietrwałych (patrz sekcja [Szybkie komunikaty nietrwałe](#)), nie można dostarczyć, a w systemie docelowym nie istnieje kolejka niedostarczonych komunikatów. Komunikaty te są usuwane.

Jednak użycie kolejek niedostarczonych komunikatów może mieć wpływ na kolejność, w jakiej komunikaty są dostarczane, dlatego można zrezygnować z ich używania.

Zadania pokrewne

[Praca z kolejkami niedostarczonych komunikatów](#)

[Rozwiązywanie problemów z niedostarczonych komunikatów](#)

Odsyłacze pokrewne

[runmqdlq \(uruchomienie programu obsługi kolejki niedostarczonych komunikatów-run dead-letter queue handler\)](#)

Definicje kolejek zdalnych

Definicje kolejek zdalnych to definicje kolejek, których właścicielem jest inny menedżer kolejek.

Podczas gdy aplikacje mogą pobierać komunikaty tylko z kolejek lokalnych, mogą umieszczać komunikaty w kolejkach lokalnych lub zdalnych. Z tego powodu, oprócz definicji dla każdej kolejki lokalnej, menedżer kolejek może mieć *definicje kolejek zdalnych*. Zaletą definicji kolejek zdalnych jest to, że umożliwiają one aplikacji umieszczenie komunikatu w kolejce zdalnej bez konieczności określania nazwy kolejki zdalnej, menedżera kolejek zdalnych lub nazwy kolejki transmisji. Definicje kolejek zdalnych zapewniają niezależność miejsca.

Istnieją inne zastosowania definicji kolejek zdalnych, które opisano później.

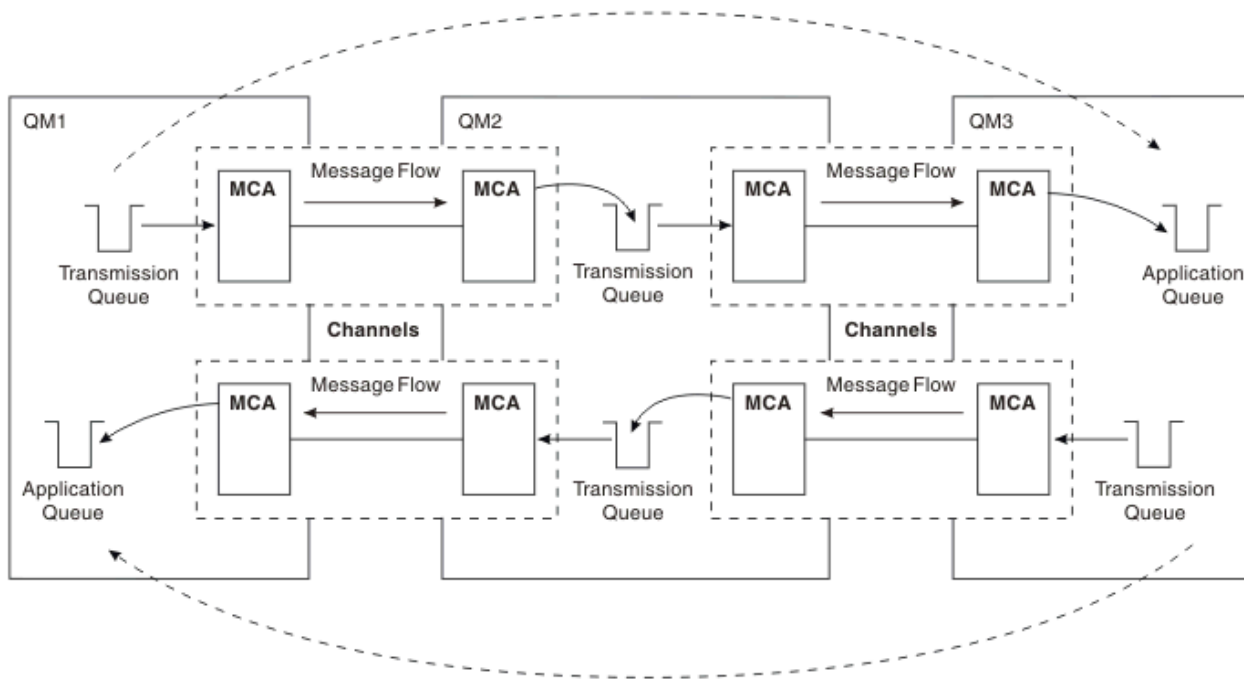
Jak uzyskać dostęp do zdalnego menedżera kolejek

Między każdym źródłowym i docelowym menedżerem kolejek może nie zawsze istnieć jeden kanał. Istnieje wiele innych sposobów łączenia między tymi dwoma kanałami, w tym wiele przeskoków, współużytkowanie kanałów, korzystanie z różnych kanałów i łączenie w klastry.

Wiele przeskoków

Jeśli nie istnieje bezpośrednie łącze komunikacyjne między źródłowym i docelowym menedżerem kolejek, możliwe jest przejście przez co najmniej jeden *pośredni menedżer kolejek* w drodze do docelowego menedżera kolejek. Jest to nazywane *wieloprzeskokowym*.

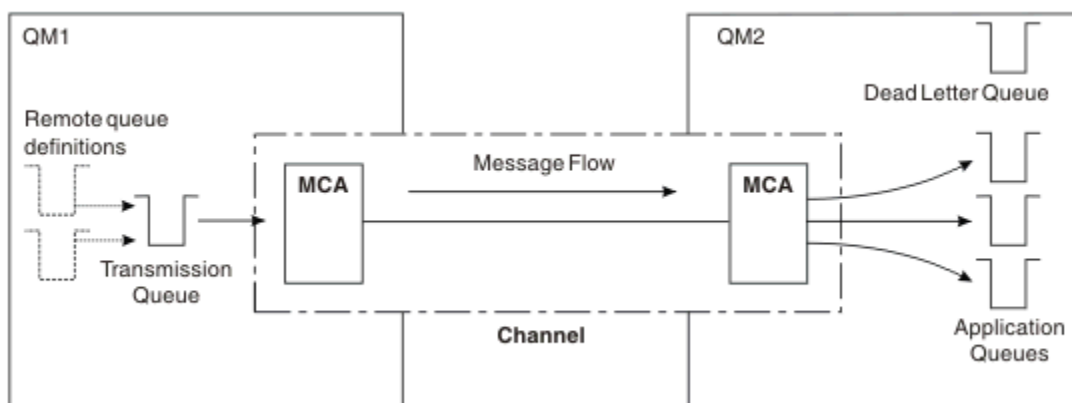
Należy zdefiniować kanały między wszystkimi menedżerami kolejek i kolejkami transmisji w pośrednich menedżerach kolejek. Pokazano to na ilustracji [Rysunek 12 na stronie 51](#).



Rysunek 12. Przekazywanie przez pośrednie menedżery kolejek

Współużytkowanie kanałów

Projektant aplikacji może wymusić na aplikacjach określenie nazwy menedżera kolejek zdalnych wraz z nazwą kolejki lub utworzyć *definicję kolejki zdalnej* dla każdej kolejki zdalnej. Ta definicja zawiera nazwę zdalnego menedżera kolejek, nazwę kolejki i nazwę kolejki transmisji. Tak czy inaczej, wszystkie komunikaty ze wszystkich aplikacji adresujących kolejki w tym samym miejscu zdalnym są wysyłane przez tę samą kolejkę transmisji. Pokazano to na ilustracji [Rysunek 13 na stronie 52](#).



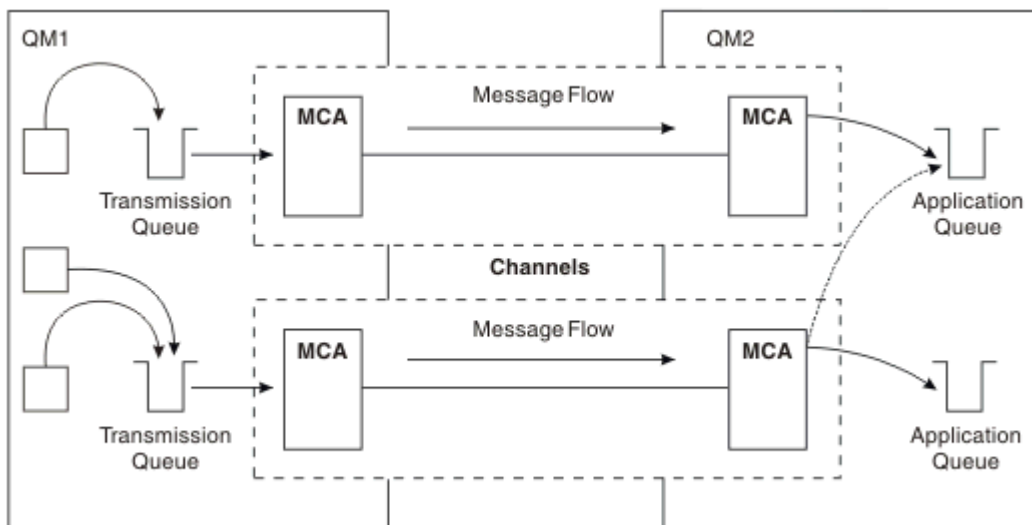
Rysunek 13. Współużytkowanie kolejki transmisji

Rysunek 13 na stronie 52 ilustruje, że komunikaty z wielu aplikacji do wielu kolejek zdalnych mogą korzystać z tego samego kanału.

Korzystanie z różnych kanałów

Jeśli między dwoma menedżerami kolejek mają być wysyłane komunikaty różnych typów, można zdefiniować między nimi więcej niż jeden kanał. Czasami potrzebne są alternatywne kanały, być może ze względów bezpieczeństwa, lub aby wyłączyć szybkość dostarczania w stosunku do dużej części ruchu komunikatów.

Aby skonfigurować drugi kanał, należy zdefiniować inny kanał i inną kolejkę transmisji oraz utworzyć definicję kolejki zdalnej, określając miejsce i nazwę kolejki transmisji. Aplikacje mogą używać obu tych kanałów, ale komunikaty są nadal dostarczane do tych samych kolejek docelowych. Pokazano to na ilustracji Rysunek 14 na stronie 52.



Rysunek 14. Korzystanie z wielu kanałów

Jeśli do określenia kolejki transmisji używane są definicje kolejek zdalnych, aplikacje **nie** mogą samodzielnie określać miejsca (czyli docelowego menedżera kolejek). Jeśli tak, menedżer kolejek nie używa definicji kolejek zdalnych. Definicje kolejek zdalnych zapewniają niezależność miejsca. Aplikacje mogą umieszczać komunikaty w kolejce *logicznej* bez wiedzy o położeniu kolejki, a użytkownik może zmieniać kolejkę *fizyczną* bez konieczności modyfikowania aplikacji.

Korzystanie z technologii klastrowej

Każdy menedżer kolejek w obrębie klastra definiuje kanał odbiorczy klastra. Gdy inny menedżer kolejek chce wysłać komunikat do tego menedżera kolejek, automatycznie definiuje odpowiedni kanał nadawczy klastra. Jeśli na przykład w klastrze istnieje więcej niż jedna instancja kolejki, kanał nadawczy klastra może zostać zdefiniowany dla dowolnego menedżera kolejek udostępniającego kolejkę. Produkt IBM MQ używa algorytmu zarządzania obciążeniem, który używa procedury algorytmu karuzelowego do wybrania dostępnego menedżera kolejek, do którego ma zostać przelożony komunikat. Więcej informacji na ten temat zawiera sekcja [Klastry](#).

Informacje adresowania

Gdy aplikacja umieszcza komunikaty przeznaczone dla zdalnego menedżera kolejek, menedżer kolejek lokalnych dodaje do nich nagłówek transmisji przed umieszczeniem ich w kolejce transmisji. Ten nagłówek zawiera nazwę kolejki docelowej i menedżera kolejek, czyli *informacje o adresowaniu*.

W środowisku pojedynczego menedżera kolejek adres kolejki docelowej jest ustanawiany, gdy aplikacja otwiera kolejkę w celu umieszczania komunikatów. Ponieważ kolejka docelowa znajduje się w tym samym menedżerze kolejek, nie ma potrzeby stosowania żadnych informacji o adresowaniu.

W środowisku kolejkowania rozproszonego menedżer kolejek musi znać nie tylko nazwę kolejki docelowej, ale także położenie tej kolejki (to znaczy nazwę menedżera kolejek) oraz trasę do tego miejsca zdalnego (to jest kolejki transmisji). Te informacje adresowania są zawarte w nagłówku transmisji. Kanał odbiorczy usuwa nagłówek transmisji i używa znajdujących się w nim informacji do znalezienia kolejki docelowej.

W przypadku używania definicji kolejki zdalnej można uniknąć konieczności określania przez aplikację nazwy docelowego menedżera kolejek. Ta definicja określa nazwę kolejki zdalnej, nazwę menedżera kolejek zdalnych, do którego kierowane są komunikaty, oraz nazwę kolejki transmisji używanej do transportu komunikatów.

Czym są aliasy?

Aliaszy są używane w celu zapewnienia jakości usługi dla komunikatów. Alias menedżera kolejek umożliwia administratorowi systemu zmianę nazwy docelowego menedżera kolejek bez konieczności zmiany aplikacji. Umożliwia również administratorowi systemu zmianę trasy do docelowego menedżera kolejek lub skonfigurowanie trasy, która obejmuje przejście przez wiele innych menedżerów kolejek (wiele przeskoków). Alias kolejki odpowiedzi zapewnia jakość usługi dla odpowiedzi.

Aliaszy menedżera kolejek i aliasy kolejki zwrotnej są tworzone przy użyciu definicji kolejki zdalnej, która ma pustą wartość RNAME. Te definicje nie definiują kolejek rzeczywistych. Są one używane przez menedżer kolejek do tłumaczenia nazw kolejek fizycznych, nazw menedżerów kolejek i kolejek transmisji.

Definicje aliasów charakteryzują się pustym RNAME.

Tłumaczenie nazw kolejek

Rozstrzygnięcie nazw kolejek odbywa się w każdym menedżerze kolejek za każdym razem, gdy kolejka jest otwierana. Jego celem jest zidentyfikowanie kolejki docelowej, docelowego menedżera kolejek (który może być lokalny) oraz trasy do tego menedżera kolejek (która może mieć wartość NULL). Rozstrzygnięta nazwa składa się z trzech części: nazwy menedżera kolejek, nazwy kolejki oraz, jeśli menedżer kolejek jest zdalny, kolejki transmisji.


Jeśli istnieje definicja kolejki zdalnej, nie są przywoływane żadne definicje aliasów. Nazwa kolejki podana przez aplikację jest tłumaczona na nazwę kolejki docelowej, menedżera kolejek zdalnych i kolejki transmisji określonej w definicji kolejki zdalnej. Więcej szczegółowych informacji na temat tłumaczenia nazw kolejek zawiera sekcja [Rozstrzygnięcie nazw kolejek](#).

Jeśli nie ma definicji kolejki zdalnej, a nazwa menedżera kolejek została określona lub rozstrzygnięta przez usługę nazw, menedżer kolejek sprawdza, czy istnieje definicja aliasu menedżera kolejek, która jest zgodna z podaną nazwą menedżera kolejek. Jeśli tak, informacje w nim zawarte są używane do przetłumaczenia nazwy menedżera kolejek na nazwę docelowego menedżera kolejek. Definicji aliasu

menedżera kolejek można również użyć do określenia kolejki transmisji do docelowego menedżera kolejek.

Jeśli rozstrzygnięta nazwa kolejki nie jest kolejką lokalną, zarówno nazwa menedżera kolejek, jak i nazwa kolejki są uwzględniane w nagłówku transmisji każdego komunikatu umieszczonego przez aplikację w kolejce transmisji.

Używana kolejka transmisji ma zwykle taką samą nazwę jak rozstrzygnięty menedżer kolejek, chyba że została zmieniona przez definicję kolejki zdalnej lub definicję aliasu menedżera kolejek. Jeśli nie zdefiniowano takiej kolejki transmisji, ale zdefiniowano domyślną kolejkę transmisji, zostanie ona użyta.

 Nazwy menedżerów kolejek działających w systemie z/OS są ograniczone do czterech znaków.

Definicje aliasów menedżera kolejek

Definicje aliasów menedżera kolejek mają zastosowanie, gdy aplikacja otwierająca kolejkę w celu umieszczenia komunikatu, określa nazwę kolejki i nazwę menedżera kolejek.

Definicje aliasów menedżera kolejek mają trzy zastosowania:

- Podczas wysyłania komunikatów zmien odzworowanie nazwy menedżera kolejek
- Podczas wysyłania komunikatów, zmienianie lub określanie kolejki transmisji
- Podczas odbierania komunikatów należy określić, czy lokalny menedżer kolejek jest miejscem docelowym dla tych komunikatów.

Komunikaty wychodzące-zmiana odzworowania nazwy menedżera kolejek

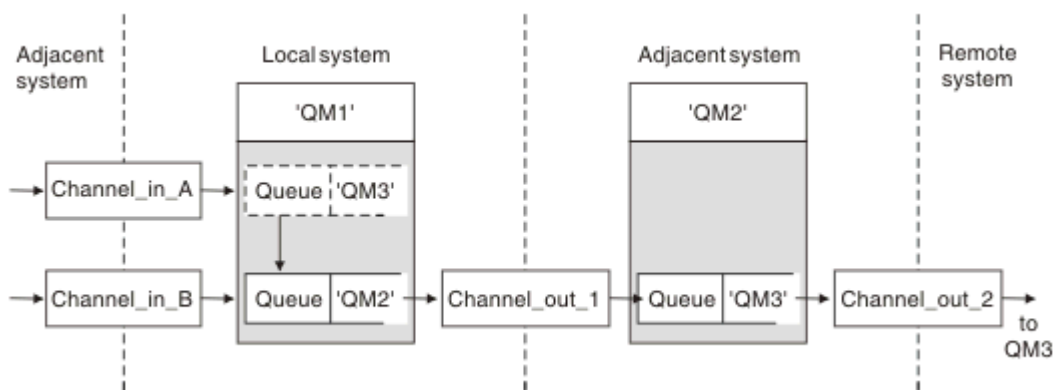
Definicje aliasów menedżera kolejek mogą być używane do zmiany odzworowania nazwy menedżera kolejek określonej w wywołaniu MQOPEN. Na przykład wywołanie MQOPEN określa nazwę kolejki THISQ i nazwę menedżera kolejek YOURQM. W lokalnym menedżerze kolejek istnieje definicja aliasu menedżera kolejek podobna do poniższej:

```
DEFINE QREMOTE (YOURQM) QMNAME(REALQM)
```

Pokazuje to, że rzeczywistym menedżerem kolejek, który ma być używany, gdy aplikacja umieszcza komunikaty w menedżerze kolejek YOURQM, jest REALQM. Jeśli lokalny menedżer kolejek to REALQM, komunikaty są umieszczane w kolejce THISQ, która jest kolejką lokalną. Jeśli lokalny menedżer kolejek nie ma wywołania REALQM, kieruje komunikat do kolejki transmisji o nazwie REALQM. Menedżer kolejek zmienia nagłówek transmisji na REALQM zamiast YOURQM.

Komunikaty wychodzące-zmienianie lub określanie kolejki transmisji

Rysunek 15 na stronie 55 przedstawia scenariusz, w którym komunikaty docierają do menedżera kolejek QM1 z nagłówkami transmisji przedstawiającymi nazwy kolejek w menedżerze kolejek QM3. W tym scenariuszu QM3 jest osiągalny przez wiele przeskoków przez QM2.



Rysunek 15. Alias menedżera kolejek

Wszystkie komunikaty dla produktu QM3 są przechwytywane w katalogu QM1 przy użyciu aliasu menedżera kolejek. Alias menedżera kolejek ma nazwę QM3 i zawiera definicję QM3 za pośrednictwem kolejki transmisji QM2. Definicja wygląda podobnie jak w poniższym przykładzie:

```
DEFINE QREMOTE (QM3) RNAME(' ') RQMNAME(QM3) XMITQ(QM2)
```

Menedżer kolejek umieszcza komunikaty w kolejce transmisji QM2, ale nie zmienia nagłówka kolejki transmisji, ponieważ nazwa docelowego menedżera kolejek QM3 nie ulega zmianie.

Wszystkie komunikaty przychodzące do kolejki QM1 i wyświetlające nagłówek transmisji zawierający nazwę kolejki w QM2 są również umieszczane w kolejce transmisji QM2. W ten sposób komunikaty z różnymi miejscami docelowymi są zbierane do wspólnej kolejki transmisji do odpowiedniego systemu przylegającego, w celu dalszego przesyłania do ich miejsc docelowych.

Komunikaty przychodzące-określanie miejsca docelowego

Odbierający agent MCA otwiera kolejkę przywoływaną w nagłówku transmisji. Jeśli istnieje definicja aliasu menedżera kolejek o takiej samej nazwie jak przywoływany menedżer kolejek, nazwa menedżera kolejek odebrana w nagłówku transmisji jest zastępowana przez nazwę RQMNAME z tej definicji.

Ten proces ma dwa zastosowania:

- Kierowanie komunikatów do innego menedżera kolejek
- Zmiana nazwy menedżera kolejek na taką samą, jak nazwa lokalnego menedżera kolejek

Definicje aliasów kolejki odpowiedzi

Definicja aliasu kolejki odpowiedzi określa alternatywne nazwy informacji odpowiedzi w deskrytorze komunikatu. Zaletą tego rozwiązania jest możliwość zmiany nazwy kolejki lub menedżera kolejek bez konieczności modyfikowania aplikacji.

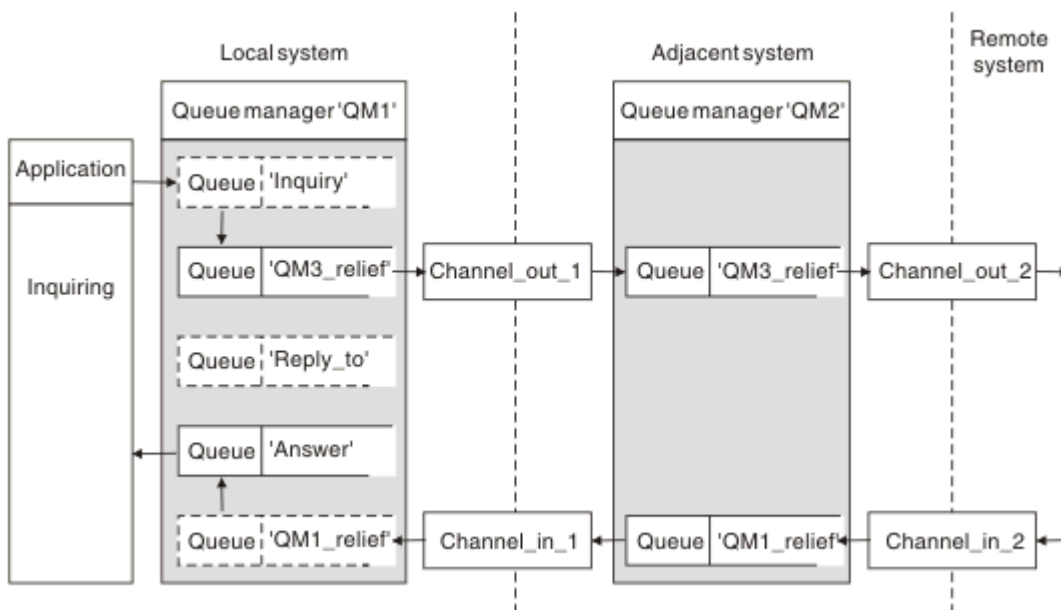
Tłumaczenie nazw kolejek

Gdy aplikacja odpowiada na komunikat, używa danych w *deskrytorze komunikatu* odebranego komunikatu w celu znalezienia nazwy kolejki, na którą ma odpowiedzieć. Aplikacja wysyłająca wskazuje miejsce, do którego są wysyłane odpowiedzi, i dołącza te informacje do swoich komunikatów. Ta koncepcja musi być skoordynowana w ramach projektu aplikacji.

Tłumaczenie nazw kolejek odbywa się na wysyłającym końcu aplikacji, zanim komunikat zostanie umieszczony w kolejce. Dlatego tłumaczenie nazwy kolejki występuje przed interakcją ze zdalną aplikacją, do której wysyłany jest komunikat. Jest to jedyna sytuacja, w której tłumaczenie nazw ma miejsce w czasie, gdy kolejka nie jest otwierana.

Tłumaczenie nazw kolejek przy użyciu aliasu menedżera kolejek

Zwykle aplikacja określa kolejkę odpowiedzi i pozostawia pustą nazwę menedżera kolejek odpowiedzi. Menedżer kolejek kończy własną nazwę w czasie umieszczania. Ta metoda działa poprawnie, z wyjątkiem sytuacji, gdy dla odpowiedzi ma być używany alternatywny kanał, na przykład kanał, który używa kolejki transmisji QM1_relief zamiast domyślnego kanału zwrotnego, który używa kolejki transmisji QM1. W takiej sytuacji nazwy menedżerów kolejek określone w nagłówkach kolejek transmisji nie są zgodne z rzeczywistymi nazwami menedżerów kolejek, ale są ponownie określone przy użyciu definicji aliasów menedżerów kolejek. W celu zwrócenia odpowiedzi na alternatywnych trasach konieczne jest również odwzorowanie danych kolejki odpowiedzi przy użyciu definicji aliasu kolejki odpowiedzi.



Rysunek 16. Alias kolejki odpowiedzi używany do zmiany położenia odpowiedzi

W przykładzie, który przedstawia Rysunek 16 na stronie 56:

1. Aplikacja umieszcza komunikat przy użyciu wywołania MQPUT i określa następujące informacje w deskrytorze komunikatu:

```
ReplyToQ='Reply_to'  
ReplyToQMgr=' '
```

ReplyToMenedżer kolejek musi być pusty, aby można było użyć aliasu kolejki odpowiedzi.

2. Należy utworzyć definicję aliasu kolejki odpowiedzi o nazwie Reply_to, która zawiera nazwę Answer i nazwę menedżera kolejek QM1_relief.

```
DEFINE QREMOTE ('Reply_to') RNAME ('Answer')  
RQMNAME ('QM1_relief')
```

3. Komunikaty są wysyłane z deskrytorem komunikatu o wartości ReplyToQ='Answer' i ReplyToQMgr='QM1_relief'.
4. Specyfikacja aplikacji musi zawierać informacje o tym, że odpowiedzi mają być znajdowane w kolejce Answer, a nie w kolejce Reply_to.

Aby przygotować się do odpowiedzi, należy utworzyć równoległy kanał zwrotu, definiując:

- W QM2: kolejka transmisji o nazwie QM1_relief

```
DEFINE QLOCAL ('QM1_relief') USAGE(XMITQ)
```

- QM1-alias menedżera kolejek QM1_relief

```
DEFINE QREMOTE ('QM1_relief') RNAME() RQMNAME(QM1)
```

Ten alias menedżera kolejek kończy łańcuch równoległych kanałów powrotnych i przechwytuje komunikaty dla QM1.

Jeśli istnieje potrzeba wykonania tej czynności w przyszłości, należy upewnić się, że aplikacje używają aliasu od początku. Obecnie jest to zwykły alias kolejki odpowiedzi, ale później można go zmienić na alias menedżera kolejek.

Nazwa kolejki odpowiedzi

Należy zachować ostrożność przy nadawaniu nazw kolejkom odpowiedzi. Przyczyną umieszczenia przez aplikację w komunikacie nazwy kolejki odpowiedzi jest możliwość określenia kolejki, do której wysyłane są odpowiedzi. Podczas tworzenia definicji aliasu kolejki odpowiedzi o tej nazwie nie może istnieć rzeczywista kolejka odpowiedzi (czyli definicja kolejki lokalnej) o takiej samej nazwie. Dlatego definicja aliasu kolejki odpowiedzi musi zawierać nową nazwę kolejki oraz nazwę menedżera kolejek, a specyfikacja aplikacji musi zawierać informacje o tym, że jej odpowiedzi znajdują się w tej innej kolejce.

Aplikacje muszą teraz wczytywać komunikaty z innej kolejki niż ta, którą nazwały kolejką zwrotną, gdy umieszczały oryginalny komunikat.

Komponenty klastra

Klaster składa się z menedżerów kolejek, repozytoriów klastrów, kanałów klastrów i kolejek klastrów.

Informacje o każdym z komponentów klastra można znaleźć w następujących podtematach:

Pojęcia pokrewne

[Porównanie kolejkowania klastrowego i rozproszonego](#)

Zadania pokrewne

[Konfigurowanie klastra menedżera kolejek](#)

[Konfigurowanie nowego klastra](#)

Repozytorium klastra

Repozytorium jest kolekcją informacji o menedżerach kolejek będących elementami klastra.

Informacje o repozytorium obejmują nazwy menedżerów kolejek, ich miejsca, kanały, kolejki, które udostępniają, oraz inne informacje. Informacje są przechowywane w postaci komunikatów w kolejce o nazwie SYSTEM.CLUSTER.REPOSITORY.QUEUE. Ta kolejka jest jednym z obiektów

domyślnych.  W systemie **Wiele platform** jest on definiowany podczas tworzenia menedżera kolejek systemu IBM MQ.  W systemie **IBM MQ for z/OS** jest on definiowany w ramach dostosowywania menedżera kolejek.

Pełne repozytorium i częściowe repozytorium

Zwykle dwa menedżery kolejek w klastrze przechowują pełne repozytorium. Wszystkie pozostałe menedżery kolejek przechowują częściowe repozytorium.

Menedżer kolejek, który udostępnia pełny zestaw informacji o każdym menedżerze kolejek w klastrze, ma pełne repozytorium. Inne menedżery kolejek w klastrze mają częściowe repozytoria zawierające podzbiór informacji w pełnych repozytoriach.

Częściowe repozytorium zawiera informacje tylko o tych menedżerach kolejek, z którymi menedżer kolejek musi wymieniać komunikaty. Menedżery kolejek żądają aktualizacji potrzebnych informacji, aby w przypadku ich zmiany menedżer kolejek repozytorium pełnego wysyłał do nich nowe informacje. Przez większość czasu częściowe repozytorium zawiera wszystkie informacje, które menedżer kolejek musi wykonać w klastrze. Jeśli menedżer kolejek potrzebuje informacji dodatkowych, wysyła zapytania o pełne repozytorium i aktualizuje swoje częściowe repozytorium. Menedżery kolejek używają kolejki `SYSTEM.CLUSTER.COMMAND.QUEUE` do żądania i odbierania aktualizacji repozytoriów.


Podczas migrowania menedżerów kolejek będących elementami klastra należy migrować pełne repozytoria przed repozytoriami częściowymi. Dzieje się tak, ponieważ starsze repozytorium nie może przechowywać nowszych atrybutów wprowadzonych w nowszej wersji. Toleruje je, ale ich nie przechowuje.

Menedżer kolejek klastra

Menedżer kolejek klastra jest menedżerem kolejek, który jest elementem klastra.

Menedżer kolejek może należeć do kilku klastrów. Każdy menedżer kolejek klastra musi mieć nazwę unikalną we wszystkich klastrach, do których należy.

Menedżer kolejek klastra może udostępniać kolejki, które są ogłaszane innym menedżerom kolejek w klastrze. Nie musi jednak tego robić. Zamiast tego może on wysyłać komunikaty do kolejek udostępnianych w innym miejscu w klastrze i odbierać tylko odpowiedzi, które są do niego jawnie kierowane.

 W programie IBM MQ for z/OS menedżer kolejek klastra może być elementem grupy współużytkowania kolejek. W tym przypadku współużytkuje swoje definicje kolejek z innymi menedżerami kolejek w tej samej grupie współużytkowania kolejek.

Menedżery kolejek klastra są autonomiczne. Mają pełną kontrolę nad kolejkami i kanałami, które definiują. Ich definicje nie mogą być modyfikowane przez inne menedżery kolejek (inne niż menedżery kolejek w tej samej grupie współużytkowania kolejek). Menedżery kolejek repozytorium nie sterują definicjami w innych menedżerach kolejek w klastrze. Zawierają one pełny zestaw wszystkich definicji, do użycia w razie potrzeby. Klaster jest stowarzyszeniem menedżerów kolejek.

Po utworzeniu lub zmodyfikowaniu definicji w menedżerze kolejek klastra informacje są wysyłane do menedżera kolejek pełnego repozytorium. Inne repozytoria w klastrze zostaną później zaktualizowane.

Menedżer kolejek pełnego repozytorium

Menedżer kolejek repozytorium pełnego to menedżer kolejek klastra, który przechowuje pełną reprezentację zasobów klastra. Aby zapewnić dostępność, w każdym klastrze należy skonfigurować co najmniej dwa menedżery kolejek pełnego repozytorium. Menedżery kolejek pełnego repozytorium odbierają informacje wysłane przez inne menedżery kolejek w klastrze i aktualizują swoje repozytoria. Wysyłają do siebie komunikaty w celu upewnienia się, że obie są na bieżąco aktualizowane z nowymi informacjami o klastrze.

Menedżery kolejek i repozytoria

Każdy klaster ma co najmniej jeden (najlepiej dwa) menedżer kolejek przechowujący pełne repozytoria informacji o menedżerach kolejek, kolejkach i kanałach w klastrze. Repozytoria te zawierają również żądania aktualizacji informacji pochodzące od innych menedżerów kolejek w klastrze.

Każdy z pozostałych menedżerów kolejek zawiera częściowe repozytorium zawierające informacje o podzbiorze kolejek i menedżerów kolejek, z którymi muszą się komunikować. Menedżery kolejek tworzą swoje częściowe repozytoria, sprawdzając, kiedy po raz pierwszy potrzebują dostępu do innej kolejki lub menedżera kolejek. Żądają oni, aby zostali powiadomieni o wszelkich nowych informacjach dotyczących tej kolejki lub menedżera kolejek.

Każdy menedżer kolejek przechowuje informacje o swoim repozytorium w komunikatach w kolejce o nazwie SYSTEM.CLUSTER.REPOSITORY.QUEUE. Menedżery kolejek wymieniają informacje o repozytorium w komunikatach w kolejce o nazwie SYSTEM.CLUSTER.COMMAND.QUEUE.

Każdy menedżer kolejek, który dołącza do klastra, definiuje kanał nadawczy klastra (CLUSDR) do jednego z repozytoriów. Natychmiast dowiaduje się, które inne menedżery kolejek w klastrze przechowują pełne repozytoria. Od tego momentu menedżer kolejek może żądać informacji z dowolnego repozytorium. Gdy menedżer kolejek wysyła informacje do wybranego repozytorium, wysyła również informacje do innego repozytorium (jeśli istnieje).


Pełne repozytorium jest aktualizowane, gdy menedżer kolejek, który je udostępnia, otrzymuje nowe informacje z jednego z połączonych z nim menedżerów kolejek. Nowe informacje są również wysyłane do innego repozytorium w celu zmniejszenia ryzyka opóźnienia, jeśli menedżer kolejek repozytorium jest nieużywany. Ponieważ wszystkie informacje są wysyłane dwukrotnie, repozytoria muszą odrzucić duplikaty. Każdy element informacji zawiera numer kolejny, który jest używany przez repozytoria do identyfikowania duplikatów. Wszystkie repozytoria są przechowywane w jednym kroku, wymieniając się komunikatami.

Kolejki klastra

Kolejka klastra to kolejka udostępniana przez menedżer kolejek klastra innym menedżerom kolejek w klastrze.

Definicja kolejki klastra jest ogłaszana w innych menedżerach kolejek w klastrze. Inne menedżery kolejek w klastrze mogą umieszczać komunikaty w kolejce klastra bez konieczności stosowania odpowiadającej jej definicji kolejki zdalnej. Kolejka klastra może zostać ogłoszona w więcej niż jednym klastrze przy użyciu listy nazw klastra.

Po ogłoszeniu kolejki każdy menedżer kolejek w klastrze może umieszczać w niej komunikaty. Aby umieścić komunikat, menedżer kolejek musi ustalić, w którym repozytorium pełnym znajduje się kolejka. Następnie do komunikatu w kolejce transmisji klastra dodawane są niektóre informacje o kierowaniu.

 Kolejka klastra może być kolejką współużytkowaną przez członków grupy współużytkowania kolejki w programie IBM MQ for z/OS.

Zadania pokrewne

[Definiowanie kolejek klastra](#)

Porównanie kolejek współużytkowanych i kolejek klastra

Informacje te mają na celu pomóc w porównaniu kolejek współużytkowanych i kolejek klastra oraz w podjęciu decyzji, które z nich mogą być bardziej odpowiednie dla danego systemu.

Koszty inicjatora kanału

W kolejkach klastra komunikaty są wysyłane przez kanały, dlatego oprócz kosztów aplikacji należy także liczyć na koszty inicjatora kanału. W sieci są koszty, ponieważ kanały dostają i umieszczają komunikaty. Te koszty nie występują w przypadku kolejek współużytkowanych, które z tego powodu podczas przenoszenia komunikatów między menedżerami kolejek w grupie współużytkowania kolejek zużywają mniej mocy przetwarzania niż kolejki klastra.

Dostępność komunikatów

Podczas umieszczania w kolejce kolejki klastra wysyłają komunikat do jednego z menedżerów kolejek z aktywnymi kanałami połączonymi z menedżerem kolejek. Jeśli w zdalnym menedżerze kolejek aplikacje używane do przetwarzania komunikatów nie działają, komunikaty nie są przetwarzane i oczekują na uruchomienie aplikacji. Podobnie, jeśli menedżer kolejek zostanie zamknięty, żadne komunikaty w menedżerze kolejek nie będą dostępne do momentu zrestartowania menedżera kolejek. Te instancje wykazują niższą dostępność komunikatów niż w przypadku korzystania z kolejek współużytkowanych.

Jeśli używane są kolejki współużytkowane, każda aplikacja w grupie współużytkowania kolejek może otrzymać wysyłane komunikaty. W przypadku zamknięcia jednego menedżera kolejek w grupie współużytkowania kolejek komunikaty są dostępne dla innych menedżerów kolejek, co zapewnia wyższą dostępność komunikatów niż w przypadku używania kolejek klastra.

pojemność

Narzędzie CF jest bardziej kosztowne niż dysk, dlatego koszt przechowywania 1 000 000 komunikatów w kolejce lokalnej jest niższy niż koszt posiadania narzędzia CF o pojemności wystarczającej do przechowywania tej samej liczby komunikatów.

Wysyłanie do innych menedżerów kolejek

Komunikaty kolejki współużytkowanej są dostępne tylko w grupie współużytkowania kolejek. Aby używać menedżera kolejek poza grupą współużytkowania kolejek, należy użyć kanałów. Technologiai klastrowej można użyć do równoważenia obciążenia między wieloma zdalnymi rozproszonymi menedżerami kolejek.

Równoważenie obciążenia

Za pomocą technologii klastrowej można nadać wagę tym kanałom i menedżerom kolejek, do których wysyłana jest część komunikatów. Na przykład można wysłać 60% komunikatów do jednego menedżera kolejek i 40% komunikatów do innego menedżera kolejek. Ta instancja nie zależy od możliwości przetwarzania pracy przez zdalny menedżer kolejek. System z pierwszym menedżerem kolejek może być przeciążony, a system z drugim menedżerem kolejek może być beczynny, ale większość komunikatów nadal jest przesyłanych do pierwszego menedżera kolejek.

Dzięki kolejkom współużytkowanym dwa systemy CICS mogą otrzymać komunikaty. Jeśli jeden system jest przeciążony, drugi system przejmuje większość obciążenia.

Kanały klastra

W każdym pełnym repozytorium należy ręcznie zdefiniować kanał odbiorczy klastra oraz zestaw kanałów nadawczych klastra, aby połączyć się z każdym innym pełnym repozytorium w klastrze. Podczas dodawania częściowego repozytorium należy ręcznie zdefiniować kanał odbiorczy klastra oraz pojedynczy kanał nadawczy klastra, który łączy się z jednym z pełnych repozytoriów. W razie potrzeby kolejne kanały nadawcze klastra są definiowane automatycznie przez klastr. Automatycznie definiowane kanały nadawcze klastra pobierają swoje atrybuty z odpowiedniej definicji kanału odbiorczego klastra w odbierającym menedżerze kolejek.

Kanał odbiorczy klastra: CLUSRCVR

Definicja kanału CLUSRCVR definiuje koniec kanału, w którym menedżer kolejek klastra może odbierać komunikaty od innych menedżerów kolejek w klastrze.

Dla każdego menedżera kolejek klastra należy zdefiniować co najmniej jeden kanał CLUSRCVR . Definiując kanał CLUSRCVR , menedżer kolejek wyświetla inne menedżery kolejek klastra, które są dostępne do odbierania komunikatów.

Definicja kanału CLUSRCVR umożliwia również innym menedżerom kolejek automatyczne definiowanie odpowiednich definicji kanału nadawczego klastra. Patrz sekcja [“Automatycznie definiowane kanały nadawcze klastra”](#) na stronie 61 w tym artykule.

Kanał nadawczy klastra: CLUSSDR

Kanał CLUSSDR jest definiowany ręcznie z każdego menedżera kolejek pełnego repozytorium do każdego innego menedżera kolejek pełnego repozytorium w klastrze. Wszystkie aktualizacje wymieniane przez pełne repozytoria przepływają wyłącznie przez te kanały. Ręczne definiowanie tych kanałów pozwala na jawne sterowanie siecią pełnych repozytoriów.

Podczas dodawania częściowego menedżera kolejek repozytorium do klastra należy ręcznie zdefiniować pojedynczy kanał CLUSSDR , aby nawiązać połączenie z jednym z pełnych repozytoriów. Wybór pełnego repozytorium nie ma znaczenia, ponieważ po początkowym kontakcie kolejne obiekty menedżera kolejek klastra dla menedżera kolejek, w tym kanały CLUSSDR , są definiowane automatycznie w razie potrzeby. Umożliwia to menedżerowi kolejek wysyłanie informacji o klastrze do dowolnego pełnego repozytorium i wysyłanie komunikatów do dowolnego menedżera kolejek w klastrze.

Jak wyjaśniono w sekcji tego artykułu, automatycznie definiowane kanały nadawcze są oparte na konfiguracji kanału odbiorczego klastra. Oznacza to, że wszystkie właściwości kanału ustawione w kanałach klastra powinny być ustawione identycznie dla zgodnych kanałów CLUSSDR i kanałów odbiorczych klastra lub ustawione tylko dla kanałów odbiorczych klastra.

Kanały CLUSSDR należy definiować ręcznie tylko z powodów opisanych wcześniej. Oznacza to, że początkowo należy połączyć częściowe repozytorium z pełnym repozytorium lub połączyć ze sobą dwa pełne repozytoria. Ręczne konfigurowanie kanału CLUSSDR łączącego się z częściowym repozytorium lub z menedżerem kolejek spoza klastra powoduje wygenerowanie komunikatów o błędach, takich jak komunikaty [AMQ9427](#) i [AMQ9428](#) . Chociaż czasami może to być nieuniknione jako sytuacja tymczasowa, na przykład podczas modyfikowania położenia pełnego repozytorium, definicja ręczna powinna zostać usunięta tak szybko, jak to możliwe.

Automatycznie definiowane kanały nadawcze klastra

Zwykle podczas dodawania częściowego menedżera kolejek repozytorium do klastra należy ręcznie zdefiniować tylko dwa kanały klastra w menedżerze kolejek:

- Nadawca klastra (CLUSSDR) Kanał do menedżera kolejek pełnego repozytorium dla klastra.
- Odbiornik klastra (CLUSRCVR) Kanał.

Zdefiniowany kanał CLUSSDR umożliwia menedżerowi kolejek nawiązanie początkowego kontaktu z klastrem. Po utworzeniu początkowego kontaktu kolejne kanały CLUSSDR są w razie potrzeby automatycznie definiowane przez klastr.

Automatycznie definiowany kanał CLUSSDR pobiera atrybuty z odpowiedniej definicji kanału CLUSRCVR w odbierającym menedżerze kolejek. Nawet jeśli istnieje ręcznie zdefiniowany kanał CLUSSDR , używane są atrybuty z automatycznie definiowanego kanału CLUSSDR . Na przykład można zdefiniować kanał CLUSRCVR bez określania numeru portu w parametrze **CONNNAME** i ręcznie zdefiniować kanał CLUSSDR , który określa numer portu. Gdy kanał CLUSSDR zdefiniowany automatycznie zastępuje kanał zdefiniowany ręcznie, numer portu (pobierany z kanału CLUSRCVR) staje się pusty. Używany jest domyślny numer portu, a działanie kanału kończy się niepowodzeniem.

Jeśli istnieją różnice w konfiguracji między ręcznie zdefiniowanym kanałem CLUSSDR i odpowiednią definicją kanału CLUSRCVR , niektóre różnice są uwzględniane natychmiast (na przykład parametry równoważenia obciążenia), a niektóre są uwzględniane tylko przy restarcie kanału (na przykład konfiguracja TLS).

Aby uniknąć nieporozumień, w miarę możliwości należy przestrzegać następujących wytycznych:

- Tylko ręcznie zdefiniuj kanały CLUSSDR , aby wskazać pełne repozytoria.
- Jeśli kanały CLUSSDR zostały zdefiniowane ręcznie, skonfiguruj je tak, aby były identyczne z odpowiednią definicją kanału CLUSRCVR w odbierającym menedżerze kolejek.

Patrz także [Praca z kanałami definiowanymi automatycznie](#).

Pojęcia pokrewne

[Praca z kanałami definiowanymi automatycznie](#)

[Praca z kolejkami transmisji klastra i kanałami nadawczymi klastra](#)

Zadania pokrewne

[Konfigurowanie nowego klastra](#)

[Dodawanie menedżera kolejek do klastra](#)

Tematy klastra

Tematy klastra są tematami administracyjnymi ze zdefiniowanym atrybutem **cluster**. Informacje o tematach klastra są przekazywane do wszystkich elementów klastra i łączone z tematami lokalnymi w celu utworzenia fragmentów obszaru tematu, który obejmuje wiele menedżerów kolejek. Umożliwia to dostarczanie komunikatów opublikowanych w temacie w menedżerze kolejek do subskrypcji innych menedżerów kolejek w klastrze.

Po zdefiniowaniu tematu klastra w menedżerze kolejek definicja tematu klastra jest wysyłana do menedżerów kolejek pełnych repozytoriów. Następnie pełne repozytoria propagują definicję tematu klastra do wszystkich menedżerów kolejek w klastrze, udostępniając ten sam temat klastra publikatorom i subskrybentom w każdym menedżerze kolejek w klastrze. Menedżer kolejek, w którym został utworzony temat klastra jest nazywany hostem tematu klastra. Temat klastra może być używany przez dowolny menedżer kolejek w klastrze, ale wszystkie modyfikacje tematu klastra muszą być wprowadzane w menedżerze kolejek, w którym zdefiniowano temat (na hoście). Wówczas modyfikacja jest propagowana do wszystkich elementów klastra przez pełne repozytoria.

Informacje na temat konfigurowania tematów klastra pod kątem używania *kierowania bezpośredniego* lub *kierowania hostami tematów* oraz na temat dziedziczenia tematów w klastrze i subskrypcji ze znakami wieloznacznymi zawiera sekcja [Definiowanie tematów klastra](#).

Więcej informacji na temat komend, których można używać do wyświetlania tematów klastra, można znaleźć w sekcji informacji pokrewnych.

Pojęcia pokrewne

[Praca z tematami administracyjnymi](#)

[Praca z subskrypcjami](#)



Odsyłacze pokrewne

[WYŚWIETL_TEMAT](#)

[STATUS DYSKU](#)

[WYSWIETLANESUB](#)

Domyślne obiekty klastra

 W przypadku wersji wieloplatformowych domyślne obiekty klastra są dołączane do zestawu obiektów domyślnych tworzonych automatycznie podczas definiowania menedżera kolejek.  W systemie z/OS domyślne definicje obiektów klastra można znaleźć w przykładach dostosowywania.

Uwaga: Domyślne definicje kanałów można zmienić w taki sam sposób, jak inne definicje kanałów, uruchamiając komendy MQSC lub PCF. Nie należy zmieniać domyślnych definicji kolejek z wyjątkiem `SYSTEM.CLUSTER.HISTORY.QUEUE`.

SYSTEM.CLUSTER.COMMAND.QUEUE

Każdy menedżer kolejek w klastrze ma lokalną kolejkę o nazwie `SYSTEM.CLUSTER.COMMAND.QUEUE`, która jest używana do przesyłania komunikatów do pełnego repozytorium. Komunikat zawiera nowe lub zmienione informacje o menedżerze kolejek lub żądania informacji o innych menedżerach kolejek. `SYSTEM.CLUSTER.COMMAND.QUEUE` jest zwykle pusta.

SYSTEM.CLUSTER.HISTORY.QUEUE

Każdy menedżer kolejek w klastrze ma lokalną kolejkę o nazwie `SYSTEM.CLUSTER.HISTORY.QUEUE`. `SYSTEM.CLUSTER.HISTORY.QUEUE` służy do przechowywania historii informacji o stanie klastra na potrzeby usługi.

W domyślnych ustawieniach obiektu `SYSTEM.CLUSTER.HISTORY.QUEUE` ma wartość `PUT (ENABLED)`. Aby pominąć gromadzenie danych historycznych, zmień ustawienie na `PUT (DISABLED)`.

SYSTEM.CLUSTER.REPOSITORY.QUEUE

Każdy menedżer kolejek w klastrze ma lokalną kolejkę o nazwie `SYSTEM.CLUSTER.REPOSITORY.QUEUE`. Ta kolejka jest używana do przechowywania wszystkich pełnych informacji o repozytorium. Ta kolejka nie jest zwykle pusta.

SYSTEM.CLUSTER.TRANSMIT.QUEUE

Każdy menedżer kolejek ma definicję kolejki lokalnej o nazwie SYSTEM.CLUSTER.TRANSMIT.QUEUE. SYSTEM.CLUSTER.TRANSMIT.QUEUE jest domyślną kolejką transmisji dla wszystkich komunikatów do wszystkich kolejek i menedżerów kolejek znajdujących się w klastrach. Domyślną kolejkę transmisji dla każdego kanału nadawczego klastra można zmienić na SYSTEM.CLUSTER.TRANSMIT.ChannelName, zmieniając atrybut menedżera kolejek DEFCLXQ. Nie można usunąć pliku SYSTEM.CLUSTER.TRANSMIT.QUEUE. Jest on również używany do definiowania sprawdzania autoryzacji, czy domyślną kolejką transmisji, która jest używana, jest SYSTEM.CLUSTER.TRANSMIT.QUEUE lub SYSTEM.CLUSTER.TRANSMIT.ChannelName.

SYSTEM.DEF.CLUSRCVR

Każdy klaster ma domyślną definicję kanału CLUSRCVR o nazwie SYSTEM.DEF.CLUSRCVR. Parametr SYSTEM.DEF.CLUSRCVR jest używany do podawania wartości domyślnych dla wszystkich atrybutów, które nie są określone podczas tworzenia kanału odbiorczego klastra w menedżerze kolejek w klastrze.

SYSTEM.DEF.CLUSSDR

Każdy klaster ma domyślną definicję kanału CLUSSDR o nazwie SYSTEM.DEF.CLUSSDR. Parametr SYSTEM.DEF.CLUSSDR jest używany do podawania wartości domyślnych dla wszystkich atrybutów, które nie są określone podczas tworzenia kanału nadawczego klastra w menedżerze kolejek w klastrze.

Pojęcia pokrewne

[Praca z domyślnymi obiektami klastra](#)

Przesyłanie komunikatów w trybie publikowania/subskrypcji

Przesyłanie komunikatów w trybie publikowania/subskrypcji umożliwia oddzielenie dostawcy informacji od konsumentów tych informacji. Aplikacja wysyłająca i aplikacja odbierająca nie muszą mieć żadnych informacji o sobie, aby informacje były wysyłane i odbierane.

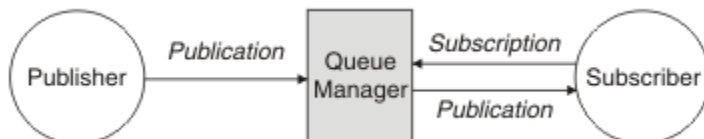
Zanim aplikacja IBM MQ typu punkt z punktem będzie mogła wysłać komunikat do innej aplikacji, musi coś wiedzieć o tej aplikacji. Na przykład musi znać nazwę kolejki, do której mają zostać wysłane informacje, i może również określać nazwę menedżera kolejek.

Publikowanie/subskrypcja produktu IBM MQ eliminuje potrzebę posiadania przez aplikację informacji o aplikacji docelowej. Wszystko, co musi zrobić aplikacja wysyłająca, to:

- Umieść komunikat IBM MQ zawierający informacje wymagane przez aplikację.
- Przypisz wiadomość do tematu, który określa temat informacji.
- Pozwól IBM MQ obsłużyć dystrybucję tych informacji.

Podobnie aplikacja docelowa nie musi mieć żadnych informacji o źródle informacji, które otrzymuje.

Poniższy rysunek przedstawia najprostszy system publikowania/subskrypcji. Istnieje jeden publikator, jeden menedżer kolejek i jeden subskrybent. Subskrybent dokonuje subskrypcji w menedżerze kolejek, publikacja jest wysyłana z publikatora do menedżera kolejek, a następnie jest przekazywana przez menedżera kolejek do subskrybenta.



Rysunek 17. Prosta konfiguracja publikowania/subskrypcji

Typowy system publikowania/subskrypcji ma więcej niż jeden publikator i więcej niż jeden subskrybent w wielu różnych tematach i często ma więcej niż jeden menedżer kolejek. Aplikacja może być zarówno publikatorem, jak i subskrybentem.

Inną istotną różnicą między przesyłaniem komunikatów w trybie publikowania/subskrypcji a przesyłaniem komunikatów w trybie punkt z punktem jest fakt, że komunikat wysyłany do kolejki w trybie punkt z punktem jest przetwarzany tylko przez jedną aplikację konsumującą. Komunikat opublikowany w temacie publikowania/subskrypcji, w którym zainteresowanie zarejestrował więcej niż jeden subskrybent, jest przetwarzany przez każdego zainteresowanego subskrybenta.

Komponenty publikowania/subskrybowania

Publikowanie/subskrybowanie jest mechanizmem, za pomocą którego subskrybenci mogą odbierać informacje od publikatorów w formie komunikatów. Interakcje między publikatorami i subskrybentami są kontrolowane przez menedżery kolejek za pomocą standardowych narzędzi IBM MQ .

Typowy system publikowania/subskrypcji ma więcej niż jeden publikator i więcej niż jeden subskrybent w wielu różnych tematach i często ma więcej niż jeden menedżer kolejek. Aplikacja może być zarówno publikatorem, jak i subskrybentem.

Dostawca informacji jest nazywany *publikatorem*. Wydawcy dostarczają informacji na temat danego tematu bez konieczności znajomości aplikacji, które są nim zainteresowane. Publikatory generują te informacje w formie komunikatów nazywanych *publikacjami* , które mają być publikowane, i definiują temat tych komunikatów.

Konsument informacji jest nazywany *subskrybentem*. Subskrybenci tworzą *subskrypcje* opisujące temat, który jest interesowany przez subskrybent. Oznacza to, że subskrypcja określa, które publikacje są przekazywane do subskrybenta. Subskrybenci mogą tworzyć wiele subskrypcji i otrzymywać informacje od wielu różnych publikatorów.

Publikowane informacje są wysyłane w komunikacie IBM MQ , a temat informacji jest identyfikowany przez *temat*. Publikator określa temat podczas publikowania informacji, a subskrybent określa tematy, o których chce otrzymywać publikacje. Subskrybent otrzymuje informacje tylko o tych tematach, które subskrybuje.

Istnienie tematów pozwala oddzielić dostawców i konsumentów informacji w przesyłaniu komunikatów w trybie publikowania/subskrypcji, eliminując konieczność dołączania konkretnego miejsca docelowego do każdego komunikatu, tak jak jest to wymagane w przesyłaniu komunikatów w trybie punkt z punktem.

Interakcje między publikatorami i subskrybentami są sterowane przez menedżer kolejek. Menedżer kolejek odbiera komunikaty od publikatorów i subskrypcje od subskrybentów (do zakresu tematów). Zadaniem menedżera kolejek jest kierowanie opublikowanych komunikatów do subskrybentów, którzy zarejestrowali zainteresowanie tematem komunikatów.

Do dystrybucji komunikatów używane są standardowe narzędzia IBM MQ , dzięki czemu aplikacje mogą korzystać ze wszystkich funkcji dostępnych dla istniejących aplikacji IBM MQ . Oznacza to, że można użyć trwałych komunikatów, aby uzyskać gwarantowane dostarczenie jednorazowe i że komunikaty mogą być częścią transakcyjnej jednostki pracy, aby zapewnić, że komunikaty będą dostarczane do subskrybenta tylko wtedy, gdy zostaną zatwierdzone przez publikator.

Wydawcy i publikacje

W programie IBM MQ publikowanie/subskrypcja publikator to aplikacja udostępniająca menedżerowi kolejek informacje o określonym temacie w postaci standardowego komunikatu IBM MQ nazywanego publikacją. Publikator może publikować informacje o więcej niż jednym temacie.

Publikatory używają komendy MQPUT do umieszczenia komunikatu w wcześniej otwartym temacie. Ten komunikat jest publikacją. Następnie lokalny menedżer kolejek kieruje publikację do wszystkich subskrybentów, którzy mają subskrypcje tematu publikacji. Opublikowany komunikat może być używany przez więcej niż jednego subskrybenta.

Oprócz dystrybucji publikacji do wszystkich subskrybentów lokalnych, które mają odpowiednie subskrypcje, menedżer kolejek może również dystrybuować publikację do wszystkich połączonych z nią menedżerów kolejek, bezpośrednio lub za pośrednictwem sieci menedżerów kolejek, które mają subskrybentów tematu.

W sieci publikowania/subskrypcji produktu IBM MQ aplikacja publikowania może być również subskrybentem.

Publikacje w punkcie synchronizacji

Publikatory mogą wysyłać wywołania MQPUT lub MQPUT1 w punkcie synchronizacji w celu dołączenia wszystkich komunikatów dostarczonych do subskrybentów w jednostce pracy. Jeśli określono opcję MQPMO_RETAIN lub opcje dostarczania tematu NPMMSGDLV lub PMSGDLV z wartościami ALL lub ALLDUR, menedżer kolejek używa wewnętrznych wywołań MQPUT lub MQPUT1 w punkcie synchronizacji w zasięgu wywołania MQPUT lub MQPUT1 publikatora.

Informacje o stanie i zdarzeniach

Publikacje można sklasyfikować jako publikacje stanowe, takie jak bieżąca cena akcji, lub publikacje dotyczące zdarzeń, takie jak handel tymi akcją.

Publikacje o stanie

Publikacje o stanie zawierają informacje o bieżącym stanie czegoś, takie jak cena akcji lub bieżący wynik w meczu piłki nożnej. W momencie wystąpienia zdarzenia (na przykład zmiany ceny akcji lub zmiany wyniku meczu) poprzednia informacja o stanie nie jest już dłużej wymagana, ponieważ zostaje zastąpiona przez nową informację.

Subskrybent będzie chciał otrzymywać bieżącą wersję informacji o stanie po uruchomieniu i otrzymywać nowe informacje przy każdej zmianie stanu.

Jeśli publikacja zawiera informacje o stanie, jest często publikowana jako zachowana publikacja. Nowy subskrybent zwykle chce natychmiast uzyskać informacje o bieżącym stanie. Subskrybent nie chce czekać na zdarzenie, które powoduje ponowne opublikowanie informacji. Subskrybenty będą automatycznie otrzymywać zachowaną publikację tematu podczas subskrybowania, chyba że subskrybent używa opcji MQSO_PUBLIC_ACTIONS_ON_REQUEST lub MQSO_NEW_PUBLIC.

Publikacje dotyczące zdarzeń

Publikacje o zdarzeniach zawierają informacje o pojedynczych zdarzeniach, które występują, takich jak transakcja giełdowa lub konkretny gol. Każde zdarzenie jest niezależne od innych zdarzeń.

Subskrybent będzie chciał otrzymywać informacje o zdarzeniach w trakcie ich występowania.

Zachowane publikacje

Domyślnie po wystaniu publikacji do wszystkich zainteresowanych subskrybentów jest ona usuwana. Publikator może jednak określić, że kopia publikacji ma zostać zachowana, aby mogła zostać wysłana do przyszłych subskrybentów, którzy zarejestrują zainteresowanie tematem.

Usunięcie publikacji po ich wystaniu do wszystkich zainteresowanych subskrybentów jest odpowiednie dla informacji o zdarzeniu, ale nie zawsze jest odpowiednie dla informacji o stanie. Zachowując komunikat, nowi subskrybenci nie muszą czekać na ponowne opublikowanie informacji, zanim otrzymają początkowe informacje o stanie. Na przykład subskrybent z subskrypcją ceny akcji otrzyma bieżącą cenę od razu, bez czekania na zmianę ceny akcji (i w związku z tym zostanie ponownie opublikowany).

Menedżer kolejek może zachować tylko jedną publikację dla każdego tematu, dlatego istniejąca zachowana publikacja tematu jest usuwana po nadejściu nowej zachowanej publikacji do menedżera kolejek. Jednak usunięcie istniejącej publikacji może nie nastąpić synchronicznie z nadejściem nowej zachowanej publikacji. Dlatego tam, gdzie jest to możliwe, nie więcej niż jeden wydawca wysyła zachowane publikacje na dowolny temat.

Subskrybenci mogą określić, że nie chcą otrzymywać zachowanych publikacji za pomocą opcji subskrypcji MQSO_NEW_PUBLIC_S_ONLY. Istniejący subskrybenci mogą poprosić o wysłanie do nich zduplikowanych kopii zachowanych publikacji.

Czasami nie ma potrzeby zachowywania publikacji, nawet w przypadku informacji o stanie:

- Jeśli wszystkie subskrypcje tematu są wykonywane przed publikacjami w tym temacie, a użytkownik nie oczekuje nowych subskrypcji lub nie zezwala na nie, nie ma potrzeby zachowywania publikacji, ponieważ są one dostarczane do pełnego zestawu subskrybentów przy pierwszej publikacji.
- Jeśli publikacje występują często, na przykład co sekundę, nowy subskrybent (lub subskrybent odzyskujący się po awarii) otrzymuje bieżący stan prawie natychmiast po subskrypcji początkowej, dlatego nie ma potrzeby zachowywania tych publikacji.
- Jeśli publikacje są duże, może być potrzebna duża ilość miejsca w pamięci masowej do przechowywania zachowanych publikacji dla każdego tematu. W środowisku z wieloma menedżerami kolejek zachowane publikacje są przechowywane przez wszystkie menedżery kolejek w sieci, które mają zgodną subskrypcję.

Decydując się na użycie zachowanych publikacji, należy wziąć pod uwagę sposób, w jaki aplikacje subskrybujące odtwarza dane po awarii. Jeśli publikator nie używa zachowanych publikacji, aplikacja subskrybenta może wymagać lokalnego przechowywania bieżącego stanu.

Aby zapewnić zachowanie publikacji, należy użyć opcji put-message komendy MQPMO_RETAIN. Jeśli ta opcja jest używana i nie można zachować publikacji, komunikat nie jest publikowany i wywołanie kończy się niepowodzeniem z wartością MQRC_PUT_NOT_ZACHOWANY.

Jeśli komunikat jest zachowaną publikacją, jest to wskazywane przez właściwość komunikatu MQIsRetained . Trwałość komunikatu jest taka sama, jak w momencie jego pierwotnej publikacji.

Pojęcia pokrewne

Uwagi dotyczące projektowania zachowanych publikacji w klastrach publikowania/subskrypcji

Publikacje w punkcie synchronizacji

W publikacji/subskrypcji produktu IBM MQ punkt synchronizacji może być używany przez publikatory lub wewnętrznie przez menedżer kolejek.

Publikatory używają punktu synchronizacji, gdy wywołują wywołania MQPUT/MQPUT1 z opcją MQPMO_SYNCPOINT. Wszystkie komunikaty dostarczane do subskrybentów są wliczane do maksymalnej liczby niezatwierdzonych komunikatów w jednostce pracy work.The . Po osiągnięciu limitu publikator otrzymuje kod przyczyny 2024 (07E8) (RC2024): MQRC_SYNCPOINT_LIMIT_REACHED .

Gdy publikator wywołuje komendę MQPUT/MQPUT1 przy użyciu wywołania MQPMO_NO_SYNCPOINT z opcją MQPMO_RETAIN lub opcjami dostarczania tematu NPMSGDLV/PMSGDLV z wartościami ALL lub ALLDUR, menedżer kolejek używa wewnętrznych punktów synchronizacji, aby zagwarantować, że komunikaty są dostarczane zgodnie z żądaniem. Publikator może odebrać kod przyczyny 2024 (07E8) (RC2024): MQRC_SYNCPOINT_LIMIT_REACHED , jeśli limit został osiągnięty w zasięgu wywołania publikatora MQPUT/MQPUT1 .

Subskrybenci i subskrypcje

W przypadku publikowania/subskrypcji produktu IBM MQ subskrybent jest aplikacją, która żąda informacji o konkretnym temacie od menedżera kolejek w sieci publikowania/subskrybowania. Subskrybent może odbierać komunikaty dotyczące tych samych lub różnych tematów od więcej niż jednego publikatora.

Subskrypcje mogą być tworzone ręcznie za pomocą komendy MQSC lub przez aplikacje. Te subskrypcje są wysyłane do lokalnego menedżera kolejek i zawierają informacje o publikacjach, które subskrybent chce otrzymywać:

- Temat, który jest interesowany przez subskrybenta, może zostać przetłumaczony na wiele tematów, jeśli używane są znaki wieloznaczne.
- Opcjonalny łańcuch wyboru, który ma zostać zastosowany do publikowanych komunikatów.
- Uchwyt do kolejki (nazywanej *kolejką subskrybenta*), w której powinny zostać umieszczone wybrane publikacje, oraz opcjonalny identyfikator CorrelId.

Menedżer kolejek lokalnych przechowuje informacje o subskrypcji, a po otrzymaniu publikacji skanuje informacje w celu określenia, czy istnieje subskrypcja zgodna z tematem i łańcuchem wyboru publikacji. Dla każdej zgodnej subskrypcji menedżer kolejek kieruje publikację do kolejki subskrybenta. Informacje przechowywane przez menedżera kolejek na temat subskrypcji można wyświetlić za pomocą komend DIS SUB i DIS SBSTATUS.

Subskrypcja jest usuwana tylko wtedy, gdy wystąpi jedno z następujących zdarzeń:

- Subskrybent anuluje subskrypcję przy użyciu wywołania MQCLOSE (jeśli subskrypcja nie została trwale ustawiona).
- Subskrypcja utraci ważność.
- Subskrypcja jest usuwana przez administratora systemu za pomocą komendy DELETE SUB.
- Aplikacja subskrybenta kończy działanie (jeśli subskrypcja została nietrwale ustawiona).
- Menedżer kolejek jest zatrzymywany lub restartowany (jeśli subskrypcja została trwale ustawiona).

Podczas pobierania komunikatów należy użyć odpowiednich opcji w wywołaniu MQGET. Jeśli aplikacja przetwarza tylko komunikaty dla jednej subskrypcji, należy użyć co najmniej wartości `get-by-correlid`, tak jak to pokazano w przykładowym programie w języku C `amqssbxa.c` i niezarządzanym subskrybencie produktu MQ. Wartość **CorrelId**, która ma być używana, jest zwracana z usługi MQSUB w usłudze MQSD. Pole **SubCorrelId**.

Pojęcia pokrewne

Subskrypcje sklonowane i współużytkowane

Odsyłacze pokrewne

Przykłady definiowania właściwości sharedSubscription

Kolejki zarządzane i publikowanie/subskrypcja

Podczas tworzenia subskrypcji można wybrać opcję używania zarządzanego kolejkowania. Jeśli używane jest kolejkowanie zarządzane, kolejka subskrypcji jest tworzona automatycznie podczas tworzenia subskrypcji. Zarządzane kolejki są automatycznie ustawiane zgodnie z trwałością subskrypcji. Korzystanie z kolejek zarządzanych oznacza, że nie trzeba się martwić o tworzenie kolejek w celu odbierania publikacji, a niewykorzystane publikacje są automatycznie usuwane z kolejek subskrybentów, jeśli połączenie subskrypcji nietrwale jest zamknięte.

Jeśli aplikacja nie musi używać określonej kolejki jako kolejki subskrybenta, miejsce docelowe dla odbieranych publikacji może korzystać z *subskrypcji zarządzanych* przy użyciu opcji subskrypcji MQSO_MANAGED. Jeśli zostanie utworzona subskrypcja zarządzana, menedżer kolejek zwraca uchwyt obiektu do subskrybenta dla kolejki subskrybenta tworzonej przez menedżer kolejek, w której będą odbierane publikacje. Jest to spowodowane tym, że *subskrypcje zarządzane* to subskrypcja, w której program IBM MQ obsługuje subskrypcję. Uchwyt obiektu kolejki zostanie zwrócony, umożliwiając przeglądanie, pobieranie lub zadawanie zapytań o kolejkę (nie jest możliwe umieszczenie lub ustawienie atrybutów kolejki zarządzanej, chyba że użytkownik otrzymał jawnie dostęp do tymczasowych kolejek dynamicznych).

Trwałość subskrypcji określa, czy kolejka zarządzana pozostaje po zerwaniu połączenia aplikacji subskrybującej z menedżerem kolejek.

Subskrypcje zarządzane są szczególnie przydatne, gdy są używane z subskrypcjami nietrwałymi, ponieważ po zakończeniu połączenia aplikacji niewykorzystane komunikaty pozostałyby w kolejce subskrybenta, zajmując miejsce w menedżerze kolejek w nieskończoność. Jeśli używana jest subskrypcja zarządzana, kolejka zarządzana będzie tymczasową kolejką dynamiczną i zostanie usunięta wraz ze wszystkimi nieużywanymi komunikatami, gdy połączenie zostanie zerwane z jednego z następujących powodów:

- Opcja MQCLOSE z opcją MQCO_REMOVE_SUB jest używana, a zarządzany obiekt Hobj jest zamknięty.
- utracono połączenie z aplikacją przy użyciu nietrwalej subskrypcji (MQSO_NON_DURABLE).
- Subskrypcja została usunięta, ponieważ utraciła ważność i zarządzany Hobj jest zamknięty.

Subskrypcje zarządzane mogą być również używane z subskrypcjami trwałymi, ale istnieje możliwość pozostawienia niewykorzystanych komunikatów w kolejce subskrybentów, aby można je było pobrać po ponownym otwarciu połączenia. Z tego powodu kolejki zarządzane dla trwałych subskrypcji mają formę trwałej kolejki dynamicznej i pozostaną po zerwaniu połączenia aplikacji subskrybującej z menedżerem kolejek.

Istnieje możliwość ustawienia utraty ważności subskrypcji, jeśli ma być używana stała dynamiczna kolejka zarządzana, tak aby mimo tego, że kolejka nadal będzie istnieć po zerwaniu połączenia, nie będzie ona istnieć w nieskończoność.

Usunięcie kolejki zarządzanej spowoduje wyświetlenie komunikatu o błędzie.

Tworzone kolejki zarządzane są nazywane numerami na końcu (datowniki), dzięki czemu każda z nich jest unikalna.

Trwałość subskrypcji

Subskrypcje mogą być skonfigurowane jako trwałe lub nietrwałe. Trwałość subskrypcji określa, co dzieje się z subskrypcjami, gdy aplikacje subskrybujące rozłączają się z menedżerem kolejek.

Subskrypcje stałe

Trwałe subskrypcje nadal istnieją, gdy połączenie aplikacji subskrybującej z menedżerem kolejek jest zamknięte. Jeśli subskrypcja jest trwała, po odłączeniu aplikacji subskrybującej subskrypcja pozostaje na miejscu i może być używana przez aplikację subskrybującą podczas ponownego nawiązywania połączenia z żądaniem subskrypcji przy użyciu programu **SubName**, który został zwrócony podczas tworzenia subskrypcji.

W przypadku subskrypcji w sposób dualisowy nazwa subskrypcji (**SubName**) jest wymagana. Nazwy subskrypcji muszą być unikalne w obrębie menedżera kolejek, aby można było ich używać do identyfikowania subskrypcji. Ten sposób identyfikacji jest niezbędny podczas określania subskrypcji, która ma zostać wznowiona, jeśli połączenie z subskrypcją zostało celowo zamknięte (przy użyciu opcji MQCO_KEEP_SUB) lub zostało odłączone od menedżera kolejek. Istniejącą subskrypcję można wznowić za pomocą wywołania MQSUB z opcją MQSO_RESUME. Nazwy subskrypcji są również wyświetlane w przypadku użycia komendy DISPLAY SBSTATUS z parametrem SUBTYPE ALL lub ADMIN.

Jeśli aplikacja nie wymaga już trwałej subskrypcji, można ją usunąć za pomocą wywołania funkcji MQCLOSE z opcją MQCO_REMOVE_SUB lub można ją usunąć ręcznie za pomocą komendy MQSC DELETE SUB.

Atrybut tematu **DURSUB** umożliwia określenie, czy można tworzyć trwałe subskrypcje tematu.

Po powrocie z wywołania MQSUB z opcją MQSO_RESUME utrata ważności subskrypcji jest ustawiana na pierwotny termin ważności subskrypcji, a nie na pozostały czas ważności.

Menedżer kolejek kontynuuje wysyłanie publikacji w celu spełnienia subskrypcji trwałej, nawet jeśli aplikacja subskrybenta nie jest połączona. Prowadzi to do budowania komunikatów w kolejce subskrybenta. Najprostszym sposobem uniknięcia tego problemu jest użycie nietrwałej subskrypcji tam, gdzie jest to konieczne. Jeśli jednak konieczne jest użycie trwałych subskrypcji, można uniknąć budowania komunikatów, jeśli subskrybent subskrybuje przy użyciu opcji Zachowane publikacje. Subskrybent może następnie sterować odbieraniem publikacji za pomocą wywołania MQSUBRQ.

Subskrypcje nietrwałe

Subskrypcje nietrwałe istnieją tylko wtedy, gdy połączenie aplikacji subskrybującej z menedżerem kolejek pozostaje otwarte. Subskrypcja zostaje usunięta, gdy aplikacja subskrybująca odłącza się od menedżera kolejek celowo lub ze względu na utratę połączenia. Po zamknięciu połączenia informacje o subskrypcji są usuwane z menedżera kolejek i nie są już wyświetlane, jeśli subskrypcje są wyświetlane za pomocą komendy DISPLAY SBSTATUS. W kolejce subskrybenta nie są umieszczane żadne komunikaty.

Informacje o tym, co dzieje się z niewykorzystywaną publikacjami w kolejce subskrybentów dla subskrypcji nietrwałych, są określane w następujący sposób.

- Jeśli aplikacja subskrybująca używa zarządzanego miejsca docelowego, wszystkie publikacje, które nie zostały wykorzystane, są automatycznie usuwane.
- Jeśli aplikacja subskrybująca udostępnia uchwyt do własnej kolejki subskrybenta podczas subskrybowania, niewykorzystane komunikaty nie są automatycznie usuwane. Aplikacja jest odpowiedzialna za usunięcie zawartości kolejki, jeśli jest to właściwe. Jeśli kolejka jest współużytkowana przez więcej niż jeden subskrybent lub inne aplikacje typu punkt z punktem, może nie być właściwe całkowite skasowanie kolejki.

Chociaż nie jest to wymagane w przypadku subskrypcji nietrwających, menedżer kolejek używa nazwy subskrypcji, jeśli została podana. Nazwy subskrypcji muszą być unikalne w obrębie menedżera kolejek, aby można było ich używać do identyfikowania subskrypcji.

Pojęcia pokrewne

Subskrypcje sklonowane i współużytkowane

Zadania pokrewne

Korzystanie ze współużytkowanych subskrypcji programu JMS 2.0

Odsyłacze pokrewne

Przykłady definiowania właściwości sharedSubscription

Łańcuchy wyboru

Łańcuch wyboru to wyrażenie stosowane do publikacji w celu określenia, czy jest ona zgodna z subskrypcją. Łańcuchy wyboru mogą zawierać znaki wieloznaczne.

Po zasubskrybowaniu, oprócz określenia tematu, można określić łańcuch wyboru w celu wybrania publikacji zgodnie z ich właściwościami komunikatu.

Łańcuch wyboru jest wartościowany względem komunikatu umieszczonego przez publikator przed zmodyfikowaniem go w celu dostarczenia do każdego subskrybenta. Należy zachować ostrożność podczas używania pól w łańcuchu wyboru, które mogą być modyfikowane w ramach operacji publikowania. Na przykład pola MQMD UserIdentifier, MsgIdi CorrelId.

Łańcuchy wyboru nie powinny odwoływać się do żadnych pól właściwości komunikatu dodanych przez menedżer kolejek w ramach operacji publikowania (patrz sekcja Właściwości komunikatu publikowania/subskrypcji), z wyjątkiem właściwości komunikatu MQTopicString, która zawiera łańcuch tematu dla publikacji.

Pojęcia pokrewne

Reguły i ograniczenia dotyczące łańcuchów wyboru

Tematy

Temat to temat informacji publikowanych w komunikacie publikowania/subskrypcji.

Komunikaty w systemach typu punkt z punktem są wysyłane do określonego adresu docelowego. Wiadomości w systemach publikowania/subskrypcji opartych na temacie są wysyłane do subskrybentów na podstawie tematu, który opisuje treść wiadomości. W systemach opartych na treści komunikaty są wysyłane do subskrybentów na podstawie treści samego komunikatu.

System publikowania/subskrypcji IBM MQ jest systemem publikowania/subskrypcji opartym na temacie. Publikator tworzy komunikat i publikuje go z łańcuchem tematu, który najlepiej pasuje do tematu publikacji. Aby otrzymywać publikacje, subskrybent tworzy subskrypcję ze wzorcem zgodnym z łańcuchem tematu w celu wybrania tematów publikacji. Menedżer kolejek dostarcza publikacje do subskrybentów, którzy mają subskrypcje zgodne z tematem publikacji i mają autoryzację do odbierania publikacji. Artykuł „Łańcuchy tematów” na stronie 70 opisuje składnię łańcuchów tematów, które identyfikują temat publikacji. Subskrybenci tworzą również łańcuchy tematów, aby wybrać, które tematy mają być odbierane. Łańcuchy tematów tworzone przez subskrybentów mogą zawierać jeden z dwóch alternatywnych schematów znaków wieloznacznych w celu dopasowania wzorca do łańcuchów tematów w publikacjach. Dopasowywanie wzorca jest opisane w sekcji “Schematy znaków wieloznacznych” na stronie 71.

W publikacji/subskrypcji opartej na podmiotach publikatorzy lub administratorzy są odpowiedzialni za klasyfikowanie tematów do tematów. Zwykle tematy są zorganizowane hierarchicznie, w postaci drzew tematów, przy użyciu znaku ' / ' do tworzenia podtematów w łańcuchu tematu. Przykłady drzew tematów można znaleźć w sekcji [“Drzewa tematów”](#) na stronie 77 . Tematy są węzłami w drzewie tematów. Tematy mogą być węzłami-liśćmi bez dalszych podtematów lub węzłami pośrednimi z podtematami.

Równoległe ze zorganizowaniem tematów w hierarchiczne drzewo tematów można powiązać tematy z obiektami tematów administracyjnych. Atrybuty są przypisywane do tematu, na przykład w przypadku, gdy temat jest rozproszony w klastrze, przez powiązanie go z obiektem tematu administracyjnego. Powiązanie jest tworzone przez nadanie tematowi nazwy przy użyciu atrybutu TOPICSTR obiektu tematu administracyjnego. Jeśli obiekt tematu administracyjnego nie zostanie jawnie powiązany z tematem, temat dziedziczy atrybuty najbliższego przodka w drzewie tematów, które *zostały* powiązane z obiektem tematu administracyjnego. Jeśli nie zdefiniowano żadnego tematu nadrzędnego, dziedziczy on z systemu SYSTEM.BASE.TOPIC. Obiekty tematów administracyjnych są opisane w sekcji [“Obiekty tematów administracyjnych”](#) na stronie 78.

Uwaga: Nawet jeśli wszystkie atrybuty tematu są dziedziczone z systemu SYSTEM.BASE.TOPIC, zdefiniuj temat główny dla tematów, które bezpośrednio dziedziczą z systemu SYSTEM.BASE.TOPIC. Na przykład w obszarze tematu Stany Zjednoczone, produkt USA/Alabama USA/Alaskaitd. USA jest tematem głównym. Głównym przeznaczeniem tematu głównego jest tworzenie dyskretnych, nienakładających się obszarów tematów, aby uniknąć publikowania zgodnych z niepoprawnymi subskrypcjami. Oznacza to również, że można zmienić atrybuty tematu głównego, aby wpłynąć na cały obszar tematu. Na przykład można ustawić nazwę atrybutu **CLUSTER** .

W przypadku odwoływania się do tematu jako publikatora lub subskrybenta można podać łańcuch tematu lub odwołać się do obiektu tematu. Można również wykonać obie te czynności. W takim przypadku podany łańcuch tematu definiuje podtemat obiektu tematu. Menedżer kolejek identyfikuje temat, dodając łańcuch tematu do przedrostka łańcucha tematu określonego w obiekcie tematu, wstawiając dodatkowy łańcuch ' / ' między dwoma łańcuchami tematów, na przykład *łańcuch tematu/łańcuch obiektu*. [“Łączenie łańcuchów tematów”](#) na stronie 75 opisuje to dalej. Wynikowy łańcuch tematu jest używany do identyfikowania tematu i powiązania go z obiektem tematu administracyjnego. Obiekt tematu administracyjnego nie musi być tym samym obiektem tematu co obiekt tematu odpowiadający tematykowi głównemu.

W przypadku publikowania/subskrybowania opartego na treści można zdefiniować komunikaty, które mają być odbierane, podając łańcuchy wyboru, które będą przeszukiwać treść każdego komunikatu. Produkt IBM MQ udostępnia pośrednią formę opartego na treści publikowania/subskrybowania przy użyciu selektorów komunikatów, które skanują właściwości komunikatu, a nie pełną treść komunikatu. Więcej informacji na ten temat zawiera sekcja [Selektory](#). Archetypem użycia selektorów komunikatów jest subskrybowanie tematu, a następnie kwalifikowanie wyboru za pomocą właściwości liczbowej. Selektor umożliwia określenie, że użytkownik jest zainteresowany tylko wartościami z określonego zakresu. Nie można użyć znaków wieloznacznych ani tematu. Jeśli konieczne jest filtrowanie na podstawie pełnej treści komunikatu, należy użyć opcji IBM Integration Bus.

Łańcuchy tematów

Etykietowanie informacji publikowanych jako temat przy użyciu łańcucha tematu. Subskrybuj grupy tematów, używając łańcuchów tematu ze znakami wieloznacznymi lub tematami.

Tematy

Łańcuch tematu to łańcuch znaków identyfikujący temat komunikatu publikowania/subskrypcji. Podczas tworzenia łańcucha tematu można używać dowolnych znaków.



Trzy znaki mają specjalne znaczenie w programie IBM WebSphere MQ 7 publish/subscribe. Są one dozwolone w dowolnym miejscu łańcucha tematu, ale należy ich używać ostrożnie. Sposób użycia znaków specjalnych opisano w sekcji [“Schemat znaków wieloznacznych na podstawie tematu”](#) na stronie 72.

Ukośnik (/)

Separator poziomu tematu. Znak ' / ' służy do ustrukturyzowania tematu w drzewo tematów.

Należy unikać pustych poziomów tematu, ' / / ', jeśli jest to możliwe. Odpowiadają one węzłom w hierarchii tematów bez łańcucha tematu. Wiodący lub kończący węzeł ' / ' w łańcuchu tematu odpowiada początkowemu lub końcowemu pustemu węzłowi i należy go unikać.

Znak krzyżyka (#)

Używany w połączeniu z ' / ' do tworzenia wielopoziomowego znaku wieloznacznego w subskrypcjach. Należy zachować ostrożność, używając łańcucha '# ' obok łańcucha ' / ' w łańcuchach tematów używanych do nazywania publikowanych tematów. [“Przykłady łańcuchów tematów”](#) na stronie 71 przedstawia sensowne użycie języka '# '.

Łańcuchy '.../#/...', '#/...' i '.../#' mają specjalne znaczenie w łańcuchach tematów subskrypcji. Łańcuchy są zgodne ze wszystkimi tematami na co najmniej jednym poziomie w hierarchii tematów. Dlatego jeśli został utworzony temat z jedną z tych sekwencji, nie można go zasubskrybować bez subskrybowania wszystkich tematów na wielu poziomach w hierarchii tematów.

Znak plus (+)

Używany w połączeniu z ' / ' do tworzenia jednopoziomowego znaku wieloznacznego w subskrypcjach. Należy zachować ostrożność, używając łańcucha '+ ' obok łańcucha ' / ' w łańcuchach tematów używanych do nazywania publikowanych tematów.

Łańcuchy '.../+/...', '+/...' i '.../+' mają specjalne znaczenie w łańcuchach tematów subskrypcji. Łańcuchy są zgodne ze wszystkimi tematami na jednym poziomie w hierarchii tematów. Dlatego jeśli utworzono temat z jedną z tych sekwencji, nie można go zasubskrybować bez subskrybowania wszystkich tematów na jednym poziomie w hierarchii tematów.

Przykłady łańcuchów tematów

```
IBM/Business Area#/Results
IBM/Diversity/%African American
```

Odsyłacze pokrewne

[TOPIC](#)

Schematy znaków wieloznacznych

Istnieją dwa schematy znaków wieloznacznych używane do subskrybowania wielu tematów. Wybór schematu jest opcją subskrypcji.

MQSO_WILDCARD_TOPIC

Wybierz tematy, które mają być subskrybowane przy użyciu schematu ze znakami wieloznacznymi opartymi na tematach.

Jest to ustawienie domyślne, jeśli nie wybrano jawnie schematu ze znakami wieloznacznymi.

MQSO_WILDCARD_CHAR

Wybierz tematy, które mają być subskrybowane przy użyciu schematu znaków wieloznacznych opartego na znakach.

Ustaw jeden z tych schematów, podając parametr **wschema** w komendzie DEFINE SUB. Więcej informacji na ten temat zawiera sekcja [DEFINE SUB](#).

Uwaga: Subskrypcje utworzone przed programem IBM WebSphere MQ 7.0 zawsze używają schematu znaków wieloznacznych opartego na znakach.

Przykłady

```
IBM+/Results
#/Results
IBM/Software/Results
IBM/*ware/Results
```

Schemat znaków wieloznacznych na podstawie tematu

Znaki wieloznaczne oparte na tematach umożliwiają subskrybentom subskrybowanie więcej niż jednego tematu jednocześnie.

Znaki wieloznaczne oparte na tematach są potężną funkcją systemu tematów w produkcie IBM MQ publish/subscribe. Wielopoziomowe i jednopoziomowe znaki wieloznaczne mogą być używane w przypadku subskrypcji, ale nie mogą być używane w temacie przez publikator komunikatu.

Schemat znaków wieloznacznych opartych na tematach umożliwia wybranie publikacji pogrupowanych według poziomu tematu. Dla każdego poziomu w hierarchii tematów można wybrać, czy łańcuch w subskrypcji dla tego poziomu tematu musi być dokładnie zgodny z łańcuchem w publikacji, czy nie. Na przykład subskrypcja IBM+/Results wybiera wszystkie tematy,

```
IBM/Software/Results
IBM/Services/Results
IBM/Hardware/Results
```

Istnieją dwa typy znaków wieloznacznych.

Wielopoziomowy znak wieloznaczny

- Wielopoziomowy znak wieloznaczny jest używany w subskrypcjach. W przypadku użycia w publikacji jest ona traktowana jako literał.
- Wielopoziomowy znak wieloznaczny '#' jest używany do dopasowania dowolnej liczby poziomów w temacie. Na przykład przy użyciu przykładowego drzewa tematów, jeśli subskrybujesz 'USA/Alaska/#', otrzymasz komunikaty dotyczące tematów 'USA/Alaska' i 'USA/Alaska/Juneau'.
- Wielopoziomowy znak wieloznaczny może reprezentować zero lub więcej poziomów. Dlatego wartość 'USA/#' może być również zgodna z osobliwą wartością 'USA', gdzie '#' oznacza zero poziomów. Separator poziomu tematu nie ma znaczenia w tym kontekście, ponieważ nie ma poziomów do oddzielenia.
- Wielopoziomowy znak wieloznaczny ma zastosowanie tylko wtedy, gdy jest określony samodzielnie lub obok znaku separatora poziomu tematu. Dlatego '#' i 'USA/#' są poprawnymi tematami, w których znak '#' jest traktowany jako znak wieloznaczny. Jednak mimo że 'USA#' jest również poprawnym łańcuchem tematu, znak '#' nie jest traktowany jako znak wieloznaczny i nie ma żadnego specjalnego znaczenia. Więcej informacji można znaleźć w sekcji [“Gdy znaki wieloznaczne oparte na temacie nie są znakami wieloznacznymi”](#) na stronie 74.

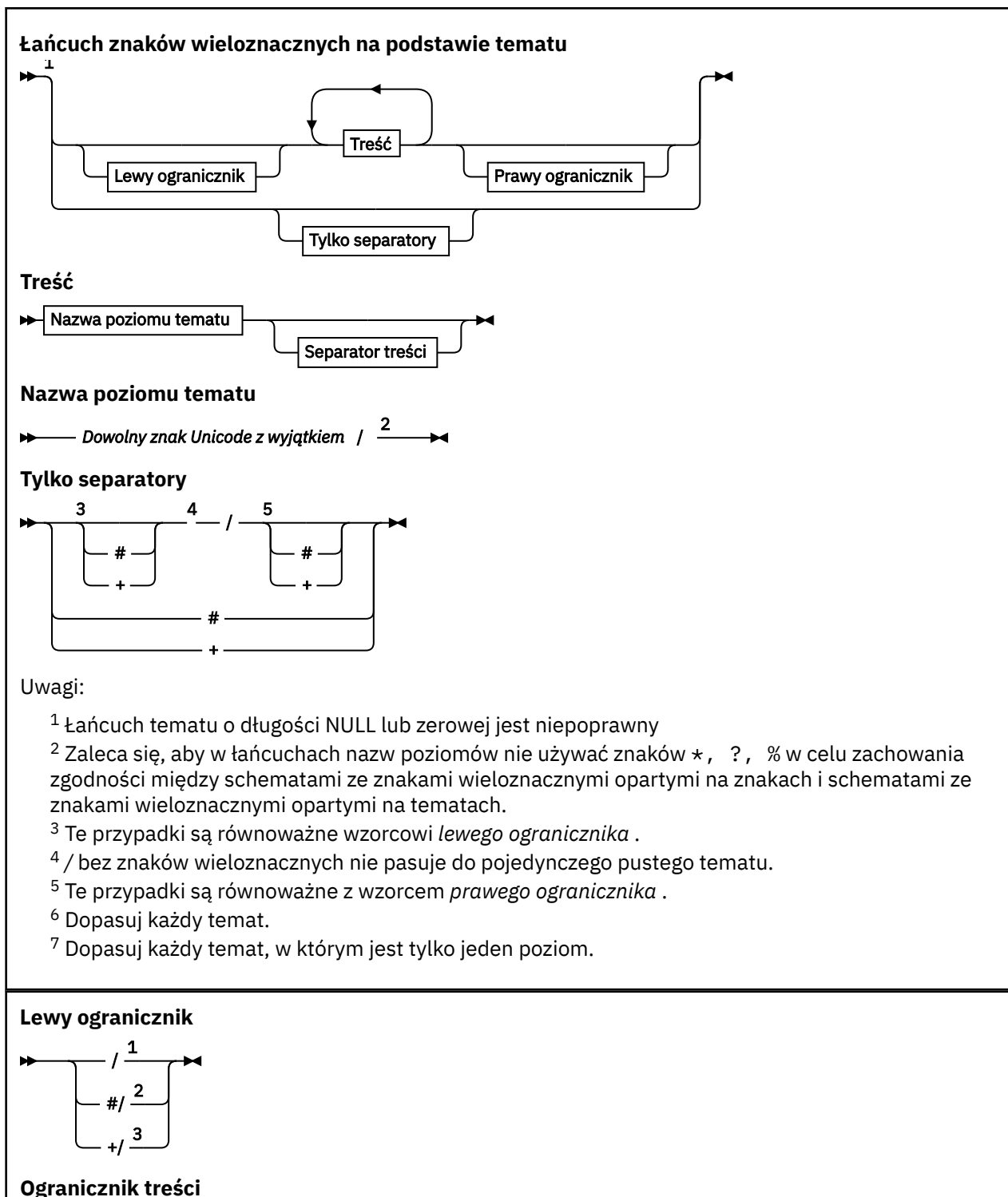
Jednopoziomowy znak wieloznaczny

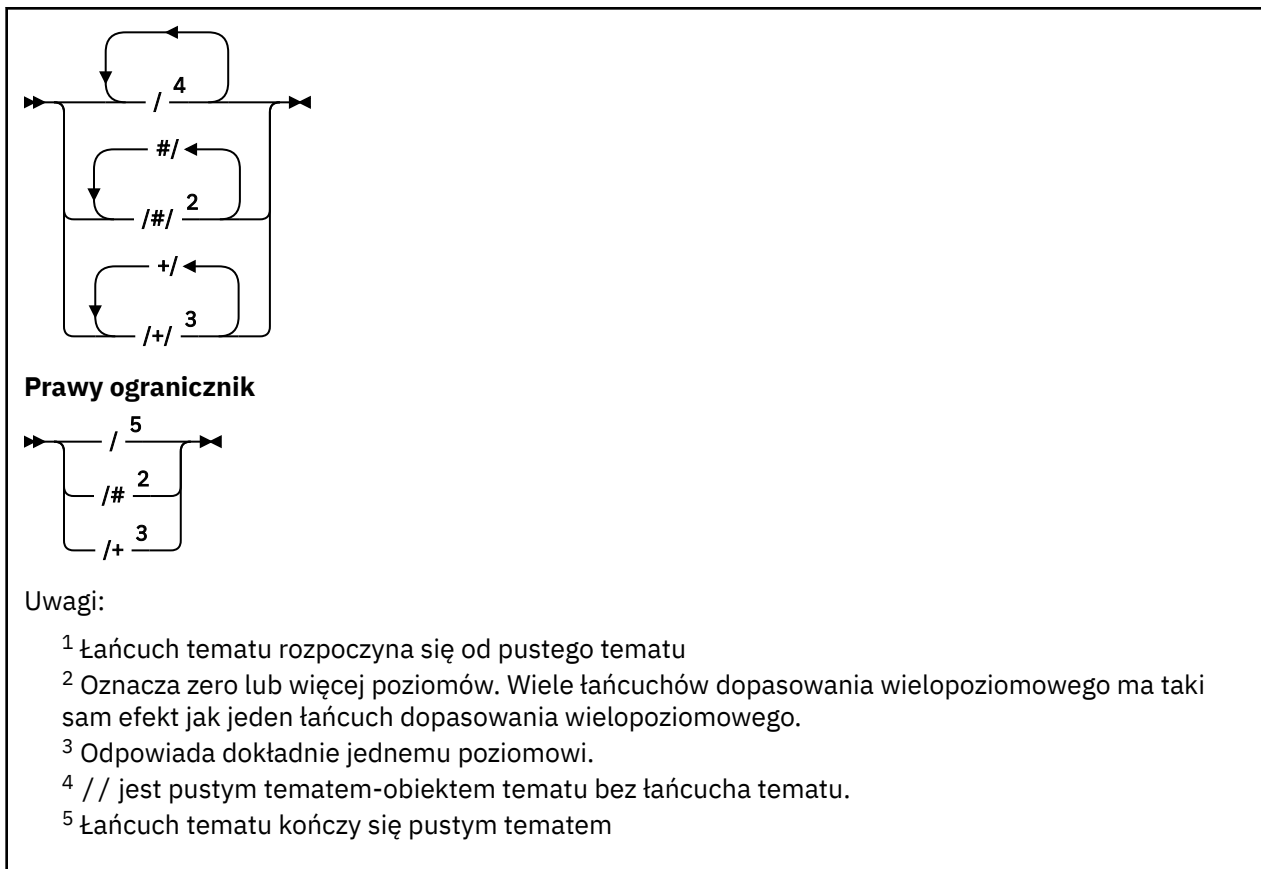
- Pojedynczy znak wieloznaczny jest używany w subskrypcjach. W przypadku użycia w publikacji jest ona traktowana jako literał.
- Jednopoziomowy znak wieloznaczny '+' jest zgodny z jednym i tylko jednym poziomem tematu. Na przykład wartość 'USA/+' jest zgodna z wartością 'USA/Alabama', ale nie z wartością 'USA/Alabama/Auburn'. Ponieważ jednopoziomowy znak wieloznaczny jest zgodny tylko z jednym poziomem, 'USA/+' nie jest zgodny z 'USA'.
- Jednopoziomowego znaku wieloznacznego można używać na dowolnym poziomie w drzewie tematów oraz w połączeniu z wielopoziomowym znakiem wieloznacznym. Jednopoziomowy znak wieloznaczny musi być określony obok separatora poziomu tematu, z wyjątkiem sytuacji, gdy jest określony samodzielnie. Dlatego '+' i 'USA/+' są poprawnymi tematami, w których znak '+' jest traktowany jako znak wieloznaczny. Jednak mimo że 'USA+' jest również poprawnym łańcuchem tematu, znak '+' nie jest traktowany jako znak wieloznaczny i nie ma żadnego specjalnego

znaczenia. Więcej informacji można znaleźć w sekcji “Gdy znaki wieloznaczne oparte na temacie nie są znakami wieloznacznymi” na stronie 74.

Składnia schematu znaków wieloznacznych opartego na tematach nie zawiera znaków zmiany znaczenia. To, czy znaki '#' i '+' są traktowane jako znaki wieloznaczne, zależy od ich kontekstu. Więcej informacji można znaleźć w sekcji “Gdy znaki wieloznaczne oparte na temacie nie są znakami wieloznacznymi” na stronie 74.

Uwaga: Początek i koniec łańcucha tematu są traktowane w specjalny sposób. Znak '\$' oznacza koniec łańcucha, znak '\$#/...' jest wielopoziomowym znakiem wieloznacznym, a znak '\$/#/...' jest pustym węzłem w katalogu głównym, po którym występuje wielopoziomowy znak wieloznaczny.





Gdy znaki wieloznaczne oparte na temacie nie są znakami wieloznacznymi

Znaki wieloznaczne '+' i '#' nie mają specjalnego znaczenia, jeśli są mieszane z innymi znakami (w tym z samymi sobą) na poziomie tematu.

Oznacza to, że można publikować tematy zawierające znaki '+' lub '#' wraz z innymi znakami na poziomie tematu.

Rozważmy na przykład następujące dwa tematy:

1. level0/level1+/level4/#
2. level0/level1/#+/level4/level#

W pierwszym przykładzie znaki '+' i '#' są traktowane jako znaki wieloznaczne i dlatego nie są poprawne w łańcuchu tematu, który ma zostać opublikowany, ale jest poprawny w subskrypcji.

W drugim przykładzie znaki '+' i '#' nie są traktowane jako znaki wieloznaczne, dlatego łańcuch tematu może być zarówno publikowany, jak i subskrybowany.

Przykłady

```
IBM/+/Results
#/Results
IBM/Software/Results
```

Schemat znaków wieloznacznych

Schemat znaków wieloznacznych oparty na znakach umożliwia wybieranie tematów na podstawie tradycyjnego dopasowywania znaków.

Za pomocą łańcucha '*' można wybrać wszystkie tematy na wielu poziomach w hierarchii tematów. Użycie '*' w schemacie znaków zastępczych opartym na znakach jest równoważne z użyciem łańcucha znaków wieloznacznych opartego na tematach '#'

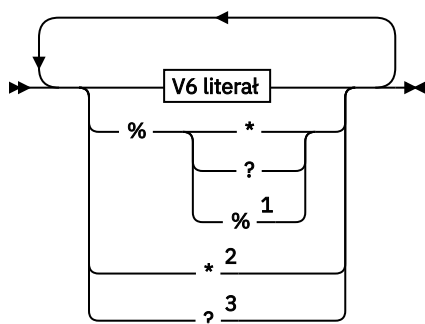
' $x/*/y$ ' jest odpowiednikiem ' $x/#/y$ ' w schemacie opartym na temacie i wybiera wszystkie tematy w hierarchii tematów między poziomami ' x i y ', gdzie ' x ' i ' y ' są nazwami tematów, które nie znajdują się w zestawie poziomów zwróconych przez znak wieloznaczny.

' $/+/'$ w schemacie opartym na temacie nie ma dokładnego odpowiednika w schemacie znakowym. ' $IBM/*/Results$ ' również wybierze ' $IBM/Patents/Software/Results$ '. Tylko wtedy, gdy zestaw nazw tematów na każdym poziomie hierarchii jest unikalny, można zawsze tworzyć zapytania z dwoma schematami, które dają identyczne dopasowania.

Ogólnie rzecz biorąc, ' $*$ ' i ' $?'$ ' w schemacie opartym na znakach nie mają odpowiedników w schemacie opartym na tematach. Schemat oparty na temacie nie wykonuje częściowego dopasowywania przy użyciu znaków wieloznacznych. Subskrypcja znaku wieloznacznego opartego na znakach ' $IBM/*ware/Results$ ' nie ma odpowiednika opartego na tematach.

Uwaga: Dopasowania używające subskrypcji ze znakami wieloznacznymi są wolniejsze niż dopasowania używające subskrypcji opartych na tematach.

Łańcuch znaków wieloznacznych



V6 literat

► Dowolny znak Unicode z wyjątkiem $*;?i\%$ ◄

Uwagi:

- ¹ Oznacza, że należy zmienić znaczenie następującego znaku, tak aby był on traktowany jako literał. Po wartości ' $\%$ ' musi nastąpić łańcuch ' $*$ ', ' $?'$ lub ' $\%$ '. Patrz ["Przykłady łańcuchów tematów"](#) na stronie 71.
- ² Oznacza zgodność z zerem lub większą liczbą znaków w subskrypcji.
- ³ Oznacza dopasowanie dokładnie jednego znaku w subskrypcji.

Przykłady

```
IBM/*/Results
IBM/*ware/Results
```

Łączenie łańcuchów tematów

Podczas tworzenia subskrypcji lub otwierania tematów w celu publikowania w nich komunikatów można utworzyć łańcuch tematu, łącząc dwa oddzielne łańcuchy tematu podrzędnego lub "podtematy". Jeden podtemat jest udostępniany przez aplikację lub komendę administracyjną jako łańcuch tematu, a drugi jest łańcuchem tematu powiązany z obiektem tematu. Tematu podrzędnego można użyć jako samego łańcucha tematu lub połączyć je w nową nazwę tematu.

Na przykład podczas definiowania subskrypcji za pomocą komendy MQSC **DEFINE SUB** komenda może przyjąć atrybut **TOPICSTR** (łańcuch tematu) lub **TOPICOBJ** (obiekt tematu) albo oba te atrybuty razem. Jeśli zostanie podana tylko wartość **TOPICOBJ**, jako łańcuch tematu zostanie użyty łańcuch tematu powiązany z tym obiektem tematu. Jeśli podano tylko parametr **TOPICSTR**, jest on używany jako łańcuch tematu. Jeśli oba są podane, są konkatelowane w celu utworzenia pojedynczego łańcucha tematu

w postaci **TOPICOBJ** / **TOPICSTR**, gdzie **TOPICOBJ** skonfigurowany łańcuch tematu jest zawsze pierwszy, a dwie części łańcucha są zawsze rozdzielone znakiem "/" .

Podobnie w programie MQI pełna nazwa tematu jest tworzona przez MQOPEN. Składa się on z dwóch pól używanych w wywołaniach MQI publikowania/subskrypcji, w następującej kolejności:

1. Atrybut **TOPICSTR** obiektu tematu o nazwie podanej w polu **ObjectName** .
2. Parametr **ObjectString** definiujący podtemat udostępniany przez aplikację.

Wynikowy łańcuch tematu jest zwracany w parametrze **ResObjectString** .

Te pola są traktowane jako obecne, jeśli pierwszy znak każdego pola nie jest znakiem odstępu lub znakiem o kodzie zero, a długość pola jest większa od zera. Jeśli istnieje tylko jedno pole, jest ono używane bez zmian jako nazwa tematu. Jeśli żadne z pól nie ma wartości, wywołanie kończy się niepowodzeniem z kodem przyczyny MQRC_UNKNOWN_OBJECT_NAME lub MQRC_TOPIC_STRING_ERROR , jeśli pełna nazwa tematu jest niepoprawna.

Jeśli oba pola są obecne, między dwoma elementami wynikowej połączonej nazwy tematu wstawiany jest znak "/" .

W poniższej tabeli przedstawiono przykłady konkatencji łańcuchów tematów:

<i>Tabela 2. Przykłady konkatencji łańcuchów tematów</i>			
TOPICSTR obiektu tematu	Łańcuch tematu udostępniany przez aplikację lub komendę DEFINE SUB	Pełna nazwa tematu	Komentarz
Piłka nożna/Wyniki	' '	Piłka nożna/Wyniki	Parametr TOPICSTR obiektu tematu jest używany samodzielnie.
' '	Piłka nożna/Wyniki	Piłka nożna/Wyniki	Łańcuch ObjectString/ TOPICSTR jest używany samodzielnie.
Piłka nożna	Oceny	Piłka nożna/Wyniki	W punkcie konkatencji dodawany jest znak "/" .
Piłka nożna	/Wyniki	Piłka nożna//Wyniki	Między dwoma łańcuchami tworzony jest pusty węzeł. Różni się to od "Piłka nożna/ Wyniki".
/Piłka nożna	Oceny	/Piłka nożna/Wyniki	Temat rozpoczyna się od pustego węzła. Różni się to od "Piłka nożna/ Wyniki".

Znak "/" jest traktowany jako znak specjalny, który zapewnia strukturę pełnej nazwy tematu w pliku "Drzewa tematów" na stronie 77. Znaku "/" nie można używać z żadnej innej przyczyny, ponieważ ma to wpływ na strukturę drzewa tematów. Temat "/Football" różni się od tematu "Football".

Uwaga: Jeśli podczas tworzenia subskrypcji używany jest obiekt tematu, wartość łańcucha tematu obiektu tematu jest poprawiana w subskrypcji w czasie definiowania. Każda kolejna zmiana obiektu tematu nie ma wpływu na łańcuch tematu, w którym jest zdefiniowana subskrypcja.

Znaki wieloznaczne w łańcuchach tematów

Następujące znaki wieloznaczne są znakami specjalnymi:

- znak plus (+)
- znak numeru (#)
- gwiazdka (*)
- znak zapytania (?)

Znaki wieloznaczne mają specjalne znaczenie tylko wtedy, gdy są używane przez subskrypcję. Znaki te nie są uważane za niepoprawne, jeśli są używane w innym miejscu, jednak należy zrozumieć sposób ich używania i może być wskazane, aby nie używać tych znaków w łańcuchach tematów podczas publikowania lub definiowania obiektów tematów.

W przypadku publikowania w łańcuchu tematu z użyciem # lub + mieszanego z innymi znakami (w tym ze sobą) na poziomie tematu, łańcuch tematu może zostać zasubskrybowany przy użyciu dowolnego ze schematów ze znakami wieloznacznymi.

W przypadku publikowania w łańcuchu tematu, w którym # lub + jest jedynym znakiem między dwoma znakami /, łańcuch tematu nie może zostać jawnie zasubskrybowany przez aplikację przy użyciu schematu wieloznacznego MQSO_WILDCARD_TOPIC. Ta sytuacja powoduje, że aplikacja otrzymuje więcej publikacji niż oczekiwano.

Nie należy używać znaku wieloznacznego w łańcuchu tematu zdefiniowanego obiektu tematu. W takim przypadku znak jest traktowany jako znak literału, gdy obiekt jest używany przez publikator, i jako znak wieloznaczny, gdy jest używany przez subskrypcję. Może to prowadzić do zamieszania.

Przykładowy fragment kodu

Ten fragment kodu wyodrębniony z przykładowego programu [Przykład 2: Publikator do tematu](#) zmieniający obiekt tematu ze zmiennym łańcuchem tematu:

```
MQOD   td = {MQOD_DEFAULT}; /* Object Descriptor      */
td.ObjectType = MQOT_TOPIC; /* Object is a topic      */
td.Version = MQOD_VERSION_4; /* Descriptor needs to be V4 */
stncpy(td.ObjectName, topicName, MQ_TOPIC_NAME_LENGTH);
td.ObjectString.VSPtr = topicString;
td.ObjectString.VSLength = (MQLONG)strlen(topicString);
td.ResObjectString.VSPtr = resTopicStr;
td.ResObjectString.VSBufSize = sizeof(resTopicStr)-1;
MQOPEN(Hconn, &td, MQOO_OUTPUT | MQOO_FAIL_IF QUIESCING, &Hobj, &CompCode, &Reason);
```

Drzewa tematów

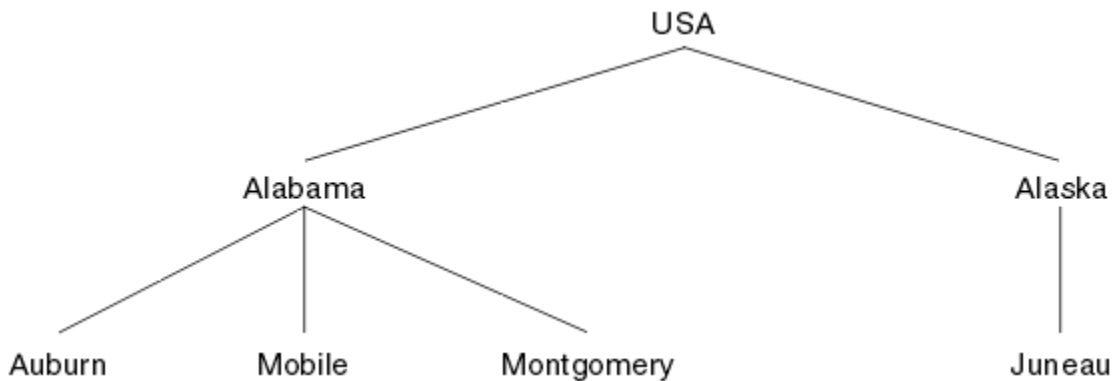
Każdy definiowany temat stanowi element lub węzeł drzewa tematów. Drzewo tematów może być puste, aby rozpocząć się od tematów lub zawierać tematy, które zostały wcześniej zdefiniowane za pomocą komend MQSC lub PCF. Nowy temat można zdefiniować przy użyciu komend tworzenia tematu lub przez określenie tematu po raz pierwszy w publikacji lub subskrypcji.

Chociaż do zdefiniowania łańcucha tematu można użyć dowolnego łańcucha znaków, zaleca się wybranie łańcucha tematu, który pasuje do hierarchicznej struktury drzewa. Przemyślany projekt stingów tematów i drzew tematów może pomóc w następujących operacjach:

- Subskrybowanie wielu tematów.
- Ustanawianie strategii bezpieczeństwa.

Mimo że drzewo tematów można utworzyć jako płaską strukturę liniową, lepiej jest zbudować drzewo tematów w strukturze hierarchicznej z jednym lub większą liczbą tematów głównych. Więcej informacji na temat planowania zabezpieczeń i tematów zawiera sekcja [Zabezpieczenia publikowania/subskrypcji](#).

Rysunek 18 na stronie 78 przedstawia przykład drzewa tematów z jednym tematem głównym.



Rysunek 18. Przykład drzewa tematów

Każdy łańcuch znaków na rysunku reprezentuje węzeł w drzewie tematów. Pełny łańcuch tematu jest tworzony przez zagregowanie węzłów z jednego lub większej liczby poziomów w drzewie tematów. Poziomy są rozdzielane znakiem "/". Format w pełni określonego łańcucha tematu jest następujący: "root/level2/level3".

Poprawne tematy w drzewie tematów, które przedstawia [Rysunek 18 na stronie 78](#) , to:

"USA"
 "USA/Alabama"
 "USA/Alaska"
 "USA/Alabama/Auburn"
 "USA/Alabama/Mobile"
 "USA/Alabama/Montgomery"
 "USA/Alaska/Juneau"

Podczas projektowania łańcuchów tematów i drzew tematów należy pamiętać, że menedżer kolejek nie interpretuje samego łańcucha tematu ani nie próbuje na jego podstawie wyprowadzić jego znaczenia. Po prostu używa łańcucha tematu do wysyłania wybranych komunikatów do subskrybentów tego tematu.

Do konstrukcji i treści drzewa tematów mają zastosowanie następujące zasady:

- Liczba poziomów w drzewie tematów nie jest ograniczona.
- Długość nazwy poziomu w drzewie tematów nie jest ograniczona.
- Może istnieć dowolna liczba węzłów "głównych", czyli może istnieć dowolna liczba drzew tematów.

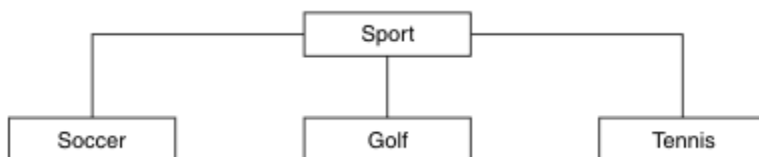
Zadania pokrewne

[Zmniejszanie liczby niechcianych tematów w drzewie tematów](#)

Obiekty tematów administracyjnych

Za pomocą obiektu tematu administracyjnego można przypisać konkretne, inne niż domyślne atrybuty do tematów.

[Rysunek 19 na stronie 78](#) pokazuje, w jaki sposób można zwizualizować w postaci drzewa tematów temat wysokiego poziomu Sport podzielony na osobne tematy dotyczące różnych dyscyplin sportowych:



Rysunek 19. Wizualizacja drzewa tematów

[Rysunek 20 na stronie 79](#) pokazuje, w jaki sposób można podzielić drzewo tematów, aby oddzielić różne typy informacji o każdym sporcie:



Rysunek 20. Rozszerzone drzewo tematów

Aby utworzyć ilustrowane drzewo tematów, nie trzeba definiować żadnych obiektów tematów administracyjnych. Każdy węzeł w tym drzewie jest zdefiniowany przez łańcuch tematu utworzony w operacji publikowania lub subskrybowania. Każdy temat w drzewie dziedziczy atrybuty ze swojego tematu nadrzędnego. Atrybuty są dziedziczone z nadrzędnego obiektu tematu, ponieważ domyślnie wszystkie atrybuty są ustawione na wartość ASPARENT. W tym przykładzie każdy temat ma takie same atrybuty jak temat Sport. Temat Sport nie ma obiektu tematu administracyjnego i dziedziczy atrybuty z systemu `SYSTEM.BASE.TOPIC`.

Należy zauważyć, że nie zaleca się nadawania uprawnień użytkownikom innym niż mqm w węźle głównym drzewa tematów, którym jest `SYSTEM.BASE.TOPIC`, ponieważ uprawnienia są dziedziczone, ale nie mogą być ograniczone. W związku z tym, dając uprawnienia na tym poziomie, dajesz uprawnienia do całego drzewa. Należy nadać uprawnienia na niższym poziomie tematu w hierarchii.

Obiekty tematów administracyjnych mogą być używane do definiowania konkretnych atrybutów dla poszczególnych węzłów w drzewie tematów. W poniższym przykładzie zdefiniowano obiekt tematu administracyjnego w celu ustawienia właściwości trwałych subskrypcji `DURSUB` tematu piłki nożnej na wartość `NO`:

```

DEFINE TOPIC(FOOTBALL.EUROPEAN)
TOPICSTR('Sport/Soccer')
DURSUB(NO)
DESCR('Administrative topic object to disallow durable subscriptions')
  
```

Drzewo tematów można teraz zwizualizować w następujący sposób:



Rysunek 21. Wizualizacja obiektu tematu administracyjnego powiązanego z tematem Sport/Piłka nożna

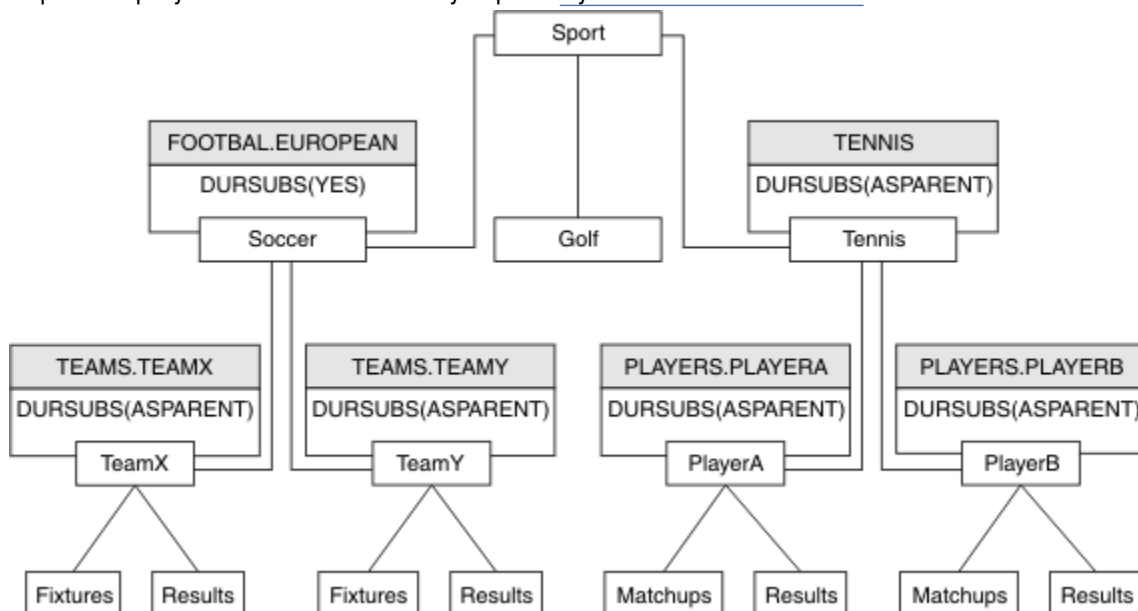
Wszystkie aplikacje subskrybujące tematy poniżej poziomu Piłka nożna w drzewie nadal mogą używać łańcuchów tematów, które były używane przed dodaniem obiektu tematu administracyjnego. Jednak aplikacja może być teraz napisana do subskrypcji przy użyciu nazwy obiektu `FOOTBALL.EUROPEAN` zamiast łańcucha `/Sport/Soccer`. Na przykład, aby zasubskrybować `/Sport/Soccer/Results`, aplikacja może określić `MQSD.ObjectName` jako `FOOTBALL.EUROPEAN`, a `MQSD.ObjectString` jako `Results`.

Dzięki tej funkcji można ukryć część drzewa tematów przed twórcami aplikacji. Zdefiniuj obiekt tematu administracyjnego w konkretnym węźle drzewa tematów, a twórcy aplikacji będą mogli zdefiniować

własne tematy jako elementy potomne węzła. Programiści muszą wiedzieć o temacie nadrzędnym, ale nie o innych węzłach w drzewie nadrzędnym.

Dziedziczenie atrybutów

Jeśli drzewo tematów zawiera wiele obiektów tematów administracyjnych, każdy obiekt tematu administracyjnego domyślnie dziedziczy atrybuty z najbliższego nadrzędnego tematu administracyjnego. Poprzedni przykład został rozszerzony w pliku Rysunek 22 na stronie 80:



Rysunek 22. Drzewo tematów z kilkoma obiektami tematów administracyjnych

Na przykład użyj dziedziczenia, aby nadać wszystkim tematom potomnym właściwości /Sport/Soccer, że subskrypcje są nietrwałe. Zmień wartość atrybutu DURSUB w pliku FOOTBALL.EUROPEAN na NO.

Atrybut ten można ustawić za pomocą następującej komendy:

```
ALTER TOPIC(FOOTBALL.EUROPEAN) DURSUB(NO)
```

Wszystkie obiekty tematów administracyjnych tematów potomnych Sport/Soccer mają właściwość DURSUB ustawioną na wartość domyślną ASPARENT. Po zmianie wartości właściwości DURSUB z FOOTBALL.EUROPEAN na NO tematy podrzędne elementu Sport/Soccer dziedziczą wartość właściwości DURSUB NO. Wszystkie tematy podrzędne Sport/Tennis dziedziczą wartość DURSUB z obiektu SYSTEM.BASE.TOPIC.SYSTEM.BASE.TOPIC ma wartość YES.

Próba wykonania trwałej subskrypcji tematu Sport/Soccer/TeamX/Results zakończy się teraz niepowodzeniem, jednak próba wykonania trwałej subskrypcji tematu Sport/Tennis/PlayerB/Results zakończy się powodzeniem.

Sterowanie użyciem znaków wieloznacznych za pomocą właściwości WILDCARD

Użyj właściwości MQSC **Topic** WILDCARD lub równoważnej właściwości PCF Topic WildcardOperation, aby sterować dostarczaniem publikacji do aplikacji subskrybenta, które używają łańcuchów tematów ze znakami wieloznacznymi. Właściwość WILDCARD może mieć jedną z dwóch możliwych wartości:

WILDCARD

Zachowanie subskrypcji ze znakami wieloznacznymi w odniesieniu do tego tematu.

PASSTHRU

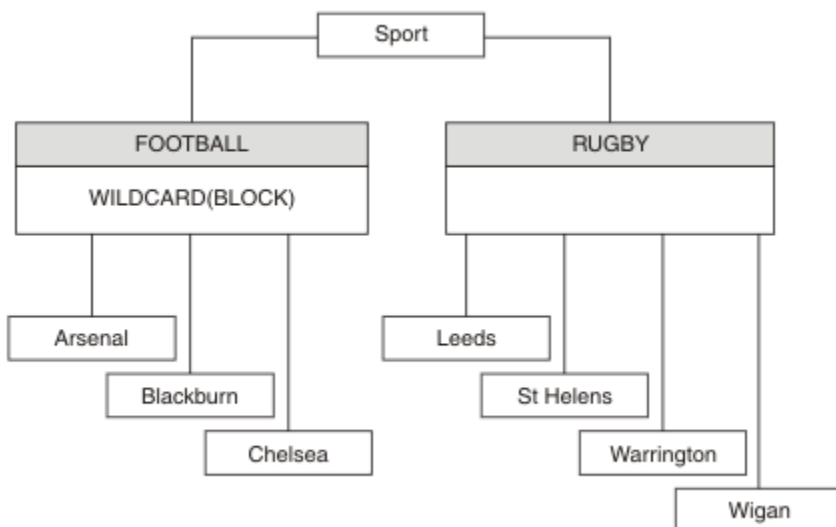
Subskrypcje tematu ze znakami wieloznacznymi, który jest mniej konkretny niż łańcuch tematu w tym obiekcie tematu, otrzymują publikacje zamieszczone w tym temacie i w łańcuchach tematów bardziej konkretnych niż ten temat.

BLOCK

Subskrypcje tematu ze znakami wieloznacznymi, który jest mniej konkretny niż łańcuch tematu w tym obiekcie tematu, nie otrzymują publikacji zamieszczonych w tym temacie i w łańcuchach tematów bardziej konkretnych niż ten temat.

Wartość tego atrybutu jest używana podczas definiowania subskrypcji. Jeśli ten atrybut zostanie zmieniony, modyfikacja nie będzie mieć wpływu na zestaw tematów objętych istniejącymi subskrypcjami. Ten scenariusz dotyczy również sytuacji, w której topologia jest zmieniana podczas tworzenia lub usuwania obiektów tematu. Zestaw tematów zgodnych z subskrypcjami utworzonymi po zmodyfikowaniu atrybutu WILDCARD jest tworzony przy użyciu zmodyfikowanej topologii. Aby wymusić ponowną ocenę zgodnego zestawu tematów pod kątem istniejących subskrypcji, należy zrestartować menedżer kolejek.

W przykładzie “Przykład: tworzenie klastra publikowania/subskrypcji produktu Sport” na stronie 84 można wykonać kroki, aby utworzyć strukturę drzewa tematów, które przedstawia [Rysunek 23](#) na stronie 81.



Rysunek 23. Drzewo tematów, w którym używana jest właściwość WILDCARD , BLOCK

Subskrybent używający łańcucha tematu ze znakami wieloznacznymi # odbiera wszystkie publikacje do tematu Sport i poddrzewa Sport/Rugby . Subskrybent nie otrzymuje żadnych publikacji w poddrzewie Sport/Football , ponieważ właściwość WILDCARD tematu Sport/Football ma wartość BLOCK.

PASSTHRU jest ustawieniem domyślnym. Wartość właściwości WILDCARD można ustawić PASSTHRU na węzły w drzewie Sport . Jeśli węzły nie mają wartości właściwości WILDCARD BLOCK, ustawienie PASSTHRU nie zmienia zachowania obserwowanego przez subskrybentów węzłów w drzewie Sports .

W tym przykładzie należy utworzyć subskrypcje, aby sprawdzić, w jaki sposób ustawienie znaku wieloznacznego wpływa na dostarczane publikacje (patrz sekcja [Rysunek 27](#) na stronie 86). Uruchom komendę publikowania w pliku [Rysunek 30](#) na stronie 87 , aby utworzyć publikacje.

pub QMA

Rysunek 24. Publikuj w QMA

Wyniki są wyświetlane w sekcji Tabela 3 na stronie 82. Należy zauważyć, że ustawienie wartości właściwości WILDCARD BLOCK zapobiega odbieraniu publikacji do tematów w zasięgu znaku wieloznacznego przez subskrypcje ze znakami wieloznacznymi.

Tabela 3. Publikacje otrzymane w serwisie QMA			
Subskrypcja	Łańcuch tematu	Odebrane publikacje	Uwagi
SPORTS	Sports/#	Sports Sports/Rugby Sports/Rugby/Leeds	Wszystkie publikacje w podrzewie piłkarskim zablokowane przez WILDCARD (BLOCK) na Sports/Football
SARSENAL	Sports/#/Arsenal	-	WILDCARD (BLOCK) on Sports/Football zapobiega subskrypcji znaków wieloznacznych w systemie Arsenal
SLEEDS	Sports/#/Leeds	Sports/Rugby/Leeds	Użycie wartości domyślnej WILDCARD w sporcie/rugby nie zapobiega subskrypcji znaku wieloznacznego w produkcie Leeds.

Uwaga:

Załóżmy, że subskrypcja zawiera znak wieloznaczny, który jest zgodny z obiektem tematu o wartości właściwości WILDCARD BLOCK. Jeśli subskrypcja zawiera również łańcuch tematu po prawej stronie zgodnego znaku wieloznacznego, subskrypcja nigdy nie otrzyma publikacji. Zestaw publikacji, które nie są zablokowane, to publikacje do tematów, które są elementami nadrzędnymi zablokowanego znaku wieloznacznego. Publikacje do tematów, które są elementami potomnymi tematu z wartością właściwości BLOCK, są blokowane przez znak wieloznaczny. Dlatego łańcuchy tematów subskrypcji, które zawierają temat po prawej stronie znaku wieloznacznego, nigdy nie otrzymują żadnych zgodnych publikacji.

Ustawienie właściwości WILDCARD na wartość BLOCK nie oznacza, że nie można subskrybować przy użyciu łańcucha tematu zawierającego znaki wieloznaczne. Taka subskrypcja jest normalna. Subskrypcja ma jawny temat, który jest zgodny z tematem z obiektem tematu o wartości właściwości WILDCARD BLOCK. W przypadku tematów będących elementami nadrzędnymi lub podrzędnymi tematu z wartością właściwości WILDCARD BLOCK używane są znaki wieloznaczne. W przykładzie przedstawionym w sekcji Rysunek 23 na stronie 81 subskrypcja, taka jak Sports/Football/#, może odbierać publikacje.

Znaki wieloznaczne i tematy klastra

Definicje tematów klastra są propagowane do każdego menedżera kolejek w klastrze. Subskrypcja tematu klastra w jednym menedżerze kolejek w klastrze powoduje, że menedżer kolejek tworzy subskrypcje proxy. Subskrypcja proxy jest tworzona w każdym innym menedżerze kolejek w klastrze. Subskrypcje używające łańcuchów tematów zawierających znaki wieloznaczne w połączeniu z tematami klastra mogą utrudniać przewidywanie zachowania. Zachowanie zostało wyjaśnione w poniższym przykładzie.

W klastrze, w którym skonfigurowano przykład “Przykład: tworzenie klastra publikowania/subskrypcji produktu Sport” na stronie 84, QMB ma taki sam zestaw subskrypcji jak QMA, ale QMB nie otrzymał żadnych publikacji po opublikowaniu publikatora w QMA, patrz sekcja Rysunek 24 na stronie 81. Mimo że tematy Sports/Football i Sports/Rugby są tematami klastra, subskrypcje zdefiniowane w pliku fullsubs.tst nie odwołują się do tematu klastra. Żadne subskrypcje proxy nie są propagowane z produktu QMB do produktu QMA. Bez subskrypcji proxy żadne publikacje do produktu QMA nie są przekazywane do produktu QMB.

Niektóre subskrypcje, takie jak Sports/#/Leeds, mogą odwoływać się do tematu klastra, w tym przypadku do tematu Sports/Rugby. W rzeczywistości subskrypcja Sports/#/Leeds jest tłumaczona na obiekt tematu SYSTEM.BASE.TOPIC.

Reguła rozstrzygnięcia obiektu tematu, do którego odwołuje się subskrypcja, na przykład Sports/#/Leeds, jest następująca. Obiekt łańcuch tematu do pierwszego znaku wieloznacznego. Skanowanie w lewo przez łańcuch tematu w poszukiwaniu pierwszego tematu, z którym jest powiązany obiekt tematu administracyjnego. Obiekt tematu może określać nazwę klastra lub definiować lokalny obiekt tematu. W przykładzie Sports/#/Leeds łańcuch tematu po obcięciu to łańcuch Sports, który nie ma obiektu tematu, a więc Sports/#/Leeds dziedziczy z SYSTEM.BASE.TOPIC, który jest lokalnym obiektem tematu.

Aby zobaczyć, w jaki sposób subskrybowanie tematów w klastrze może zmienić sposób działania propagacji z użyciem znaków wieloznacznych, należy uruchomić skrypt wsadowy `upsubs.bat`. Skrypt kasuje kolejki subskrypcji i dodaje subskrypcje tematów klastra w pliku `fullsubs.tst`. Ponownie uruchom plik `puba.bat`, aby utworzyć partię publikacji; patrz sekcja [Rysunek 24](#) na stronie 81.

Tabela 4 na stronie 83 przedstawia wynik dodania dwóch nowych subskrypcji do tego samego menedżera kolejek, w którym zostały opublikowane publikacje. Wynik jest zgodny z oczekiwaniami, nowe subskrypcje otrzymują po jednej publikacji, a liczba publikacji odebranych przez inne subskrypcje pozostaje niezmienną. Nieoczekiwane wyniki wystąpiły w innym menedżerze kolejek klastra (patrz sekcja Tabela 5 na stronie 84).

<i>Tabela 4. Publikacje otrzymane w serwisie QMA</i>			
Subskrypcja	Łańcuch tematu	Odebrane publikacje	Uwagi
SPORTS	Sports/#	Sports Sports/Rugby Sports/Rugby/Leeds	Wszystkie publikacje w poddrzewie piłkarskim zablokowane przez WILDCARD (BLOCK) na Sports/Football
SARSENAL	Sports/#/Arsenal	-	WILDCARD (BLOCK) on Sports/Football zapobiega subskrypcji znaków wieloznacznych w systemie Arsenal
SLEEDS	Sports/#/Leeds	Sports/Rugby/Leeds	Użycie wartości domyślnej WILDCARD w sporcie/rugby nie zapobiega subskrypcji znaku wieloznacznego w produkcie Leeds.
FARSENAL	Sports/Football/Arsenal	Sports/Football/Arsenal	Arsenal otrzymuje publikację, ponieważ subskrypcja nie zawiera znaku wieloznacznego.
FLEEDS	Sports/Rugby/Leeds	Sports/Rugby/Leeds	Leeds otrzyma publikację w dowolnym przypadku.

Tabela 5 na stronie 84 przedstawia wyniki dodania dwóch nowych subskrypcji w systemie QMB i publikowania w systemie QMA. Należy pamiętać, że program QMB nie otrzymał żadnych publikacji bez tych dwóch nowych subskrypcji. Zgodnie z oczekiwaniami dwie nowe subskrypcje otrzymują publikacje, ponieważ Sports/Football i Sports/Rugby są tematami klastra. QMB przekazała subskrypcje proxy dla Sports/Football/Arsenal i Sports/Rugby/Leeds do QMA, a następnie wysłała publikacje do QMB.

Nieoczekiwany wynik jest to, że te dwie subskrypcje Sports/# i Sports/#/Leeds, które wcześniej nie otrzymywały żadnych publikacji, teraz otrzymują publikacje. Wynika to z faktu, że publikacje Sports/Football/Arsenal i Sports/Rugby/Leeds przekazane do QMB dla innych subskrypcji są teraz dostępne dla każdego subskrybenta przyłączonego do produktu QMB. W związku z tym subskrypcje tematów lokalnych Sports/# i Sports/#/Leeds otrzymują publikację Sports/Rugby/Leeds. Produkt Sports/#/Arsenal nadal nie otrzymuje publikacji, ponieważ właściwość WILDCARD sportu/piłki nożnej ma wartość BLOCK.

Tabela 5. Publikacje otrzymane w serwisie QMB

Subskrypcja	Łańcuch tematu	Odebrane publikacje	Uwagi
SPORTS	Sports/#	Sports/Rugby/Leeds	Wszystkie publikacje w poddrzewie piłki nożnej zablokowane przez WILDCARD (BLOCK) w systemie Sports/Football
SARSENAL	Sports/#/Arsenal	-	WILDCARD (BLOCK) on Sports/Football zapobiega subskrypcji znaków wieloznacznych w systemie Arsenal
SLEEDS	Sports/#/Leeds	Sports/Rugby/Leeds	Wartość domyślna WILDCARD w przypadku produktu Sports/Rugby nie zapobiega subskrypcji znaku wieloznacznego w produkcie Leeds.
FARSENAL	Sports/Football/Arsenal	Sports/Football/Arsenal	Arsenal otrzymuje publikację, ponieważ subskrypcja nie zawiera znaku wieloznacznego.
FLEEDS	Sports/Rugby/Leeds	Sports/Rugby/Leeds	Leeds otrzyma publikację w dowolnym przypadku.

W większości aplikacji nie jest wskazane, aby jedna subskrypcja miała wpływ na zachowanie innej subskrypcji. Ważnym zastosowaniem właściwości WILDCARD z wartością BLOCK jest zapewnienie jednolitego zachowania subskrypcji tego samego łańcucha tematu zawierającego znaki wieloznaczne. Niezależnie od tego, czy subskrypcja znajduje się w tym samym menedżerze kolejek co publikator, czy w innym menedżerze kolejek, wyniki subskrypcji są takie same.

Znaki wieloznaczne i strumienie

W przypadku nowej aplikacji zapisanej w interfejsie API publikowania/subskrypcji subskrypcja produktu * nie odbiera żadnych publikacji. Aby otrzymywać wszystkie publikacje dotyczące sportu, należy zasubskrybować publikacje Sports/*lub Sports/#oraz publikacje dotyczące produktu Business w podobny sposób.

Zachowanie istniejącej umieszczonej w kolejce aplikacji publikowania/subskrypcji nie zmienia się, gdy broker publikowania/subskrypcji jest migrowany do nowszej wersji produktu IBM MQ. Właściwość **StreamName** w komendach **Publish**, **Register Publisher** lub **Subscriber** jest odwzorowywana na nazwę tematu, do którego strumień został zmigrowany.

Znaki wieloznaczne i punkty subskrypcji

W przypadku nowej aplikacji napisanej w interfejsie API publikowania/subskrypcji efektem migracji jest brak publikacji dla subskrypcji produktu *. Aby otrzymywać wszystkie publikacje dotyczące sportu, należy zasubskrybować publikacje Sports/*lub Sports/#oraz publikacje dotyczące produktu Business w podobny sposób.

Zachowanie istniejącej umieszczonej w kolejce aplikacji publikowania/subskrypcji nie zmienia się, gdy broker publikowania/subskrypcji jest migrowany do nowszej wersji produktu IBM MQ. Właściwość **SubPoint** w komendach **Publish**, **Register Publisher** lub **Subscriber** jest odwzorowywana na nazwę tematu, do którego została zmigrowana subskrypcja.

Przykład: tworzenie klastra publikowania/subskrypcji produktu Sport

Poniższe kroki umożliwiają utworzenie klastra CL1z czterema menedżerami kolejek: dwoma pełnymi repozytoriami, CL1A i CL1B oraz dwoma częściowymi repozytoriami QMA i QMB. Pełne repozytoria są

używane do przechowywania tylko definicji klastrów. QMA jest nazwą hosta tematu klastra. Trwałe subskrypcje są definiowane zarówno w systemie QMA, jak i w systemie QMB.

Uwaga: Przykład jest zakodowany dla Windows. Należy ponownie zakodować komendę `Create qmgrs.bat` i utworzyć plik `pub.bat`, aby skonfigurować i przetestować przykład na innych platformach.

1. Utwórz pliki skryptowe.
 - a. Utwórz plik topics.tst
 - b. Utwórz wilddsubs.tst
 - c. Utwórz fullsubs.tst
 - d. Utwórz plik qmgrs.bat
 - e. tworzenie pub.bat
2. Uruchom komendę `Create qmgrs.bat`, aby utworzyć konfigurację.

```
qmgrs
```

Utwórz tematy w sekcji [Rysunek 23 na stronie 81](#). Skrypt na rysunku 5 tworzy tematy klastra Sports/Football i Sports/Rugby.

Uwaga: Opcja REPLACE nie zastępuje właściwości TOPICSTR tematu. Właściwość TOPICSTR jest właściwością, która jest używana w przykładzie do testowania różnych drzew tematów. Aby zmienić tematy, najpierw usuń temat.

```
DELETE TOPIC ('Sports')
DELETE TOPIC ('Football')
DELETE TOPIC ('Arsenal')
DELETE TOPIC ('Blackburn')
DELETE TOPIC ('Chelsea')
DELETE TOPIC ('Rugby')
DELETE TOPIC ('Leeds')
DELETE TOPIC ('Wigan')
DELETE TOPIC ('Warrington')
DELETE TOPIC ('St. Helens')

DEFINE TOPIC ('Sports') TOPICSTR('Sports')
DEFINE TOPIC ('Football') TOPICSTR('Sports/Football') CLUSTER(CL1) WILDCARD(BLOCK)
DEFINE TOPIC ('Arsenal') TOPICSTR('Sports/Football/Arsenal')
DEFINE TOPIC ('Blackburn') TOPICSTR('Sports/Football/Blackburn')
DEFINE TOPIC ('Chelsea') TOPICSTR('Sports/Football/Chelsea')
DEFINE TOPIC ('Rugby') TOPICSTR('Sports/Rugby') CLUSTER(CL1)
DEFINE TOPIC ('Leeds') TOPICSTR('Sports/Rugby/Leeds')
DEFINE TOPIC ('Wigan') TOPICSTR('Sports/Rugby/Wigan')
DEFINE TOPIC ('Warrington') TOPICSTR('Sports/Rugby/Warrington')
DEFINE TOPIC ('St. Helens') TOPICSTR('Sports/Rugby/St. Helens')
```

Rysunek 25. Usuwanie i tworzenie tematów: topics.tst

Uwaga: Usuń tematy, ponieważ REPLACE nie zastępuje łańcuchów tematów.

Utwórz subskrypcje ze znakami wieloznacznymi. Znaki wieloznaczne odpowiadające tematom z obiektami tematów w produkcie [Rysunek 23 na stronie 81](#). Utwórz kolejkę dla każdej subskrypcji. Po uruchomieniu lub ponownym uruchomieniu skryptu kolejki są czyszczone, a subskrypcje usuwane.

Uwaga: Opcja REPLACE nie zastępuje właściwości TOPICOBJ ani TOPICSTR subskrypcji. TOPICOBJ lub TOPICSTR to właściwości, które są używane w tym przykładzie do testowania różnych subskrypcji. Aby je zmienić, najpierw usuń subskrypcję.

```

DEFINE QLOCAL(QSPORTS) REPLACE
DEFINE QLOCAL(QSARSENAL) REPLACE
DEFINE QLOCAL(QSLEEDS) REPLACE
CLEAR QLOCAL(QSPORTS)
CLEAR QLOCAL(QSARSENAL)
CLEAR QLOCAL(QSLEEDS)

DELETE SUB (SPORTS)
DELETE SUB (SARSENAL)
DELETE SUB (SLEEDS)
DEFINE SUB (SPORTS) TOPICSTR('Sports/#') DEST(QSPORTS)
DEFINE SUB (SARSENAL) TOPICSTR('Sports+/Arsenal') DEST(QSARSENAL)
DEFINE SUB (SLEEDS) TOPICSTR('Sports+/Leeds') DEST(QSLEEDS)

```

Rysunek 26. Utwórz subskrypcje ze znakami wieloznacznymi: wildsubs.tst

Utwórz subskrypcje, które odwołują się do obiektów tematu klastra.

Uwaga:

Separator / jest automatycznie wstawiany między łańcuchem tematu, do którego odwołuje się obiekt TOPICOBJ, a łańcuchem tematu zdefiniowanym w parametrze TOPICSTR.

Definicja DEFINE SUB(FARSENAL) TOPICSTR('Sports/Football/Arsenal') DEST(QFARSENAL) tworzy tę samą subskrypcję. TOPICOBJ jest używana jako szybki sposób odwoływania się do już zdefiniowanego łańcucha tematu. Po utworzeniu subskrypcja nie odwołuje się już do obiektu tematu.

```

DEFINE QLOCAL(QFARSENAL) REPLACE
DEFINE QLOCAL(QRLEEDS) REPLACE
CLEAR QLOCAL(QFARSENAL)
CLEAR QLOCAL(QRLEEDS)

DELETE SUB (FARSENAL)
DELETE SUB (RLEEDS)
DEFINE SUB (FARSENAL) TOPICOBJ('Football') TOPICSTR('Arsenal') DEST(QFARSENAL)
DEFINE SUB (RLEEDS) TOPICOBJ('Rugby') TOPICSTR('Leeds') DEST(QRLEEDS)

```

Rysunek 27. Usuwanie i tworzenie subskrypcji: fullsubs.tst

Utwórz klaster z dwoma repozytoriami. Utwórz dwa częściowe repozytoria na potrzeby publikowania i subskrybowania. Uruchom ponownie skrypt, aby usunąć wszystko i rozpocząć od nowa. Skrypt tworzy również hierarchię tematów i początkowe subskrypcje ze znakami wieloznacznymi.

Uwaga:

Na innych platformach napisz podobny skrypt lub wpisz wszystkie komendy. Użycie skryptu umożliwia szybkie usunięcie wszystkich elementów i rozpoczęcie od nowa z identyczną konfiguracją.

```

@echo off
set port.CL1B=1421
set port.CL1A=1420
for %%A in (CL1A CL1B QMA QMB) do call :createQM %%A
call :configureQM CL1A CL1B %port.CL1B% full
call :configureQM CL1B CL1A %port.CL1A% full
for %%A in (QMA QMB) do call :configureQM %%A CL1A %port.CL1A% partial
for %%A in (topics.tst wildsubs.tst) do runmqsc QMA < %%A
for %%A in (wildsubs.tst) do runmqsc QMB < %%A
goto:eof

:createQM
echo Configure Queue manager %1
endmqm -p %1
for %%B in (dlt crt str) do %%Bmqm %1
goto:eof

:configureQM
if %1==CL1A set p=1420
if %1==CL1B set p=1421
if %1==QMA set p=1422
if %1==QMB set p=1423
echo configure %1 on port %p% connected to repository %2 on port %3 as %4 repository
echo DEFINE LISTENER(LST%1) TRPTYPE(TCP) PORT(%p%) CONTROL(QMGR) REPLACE | runmqsc %1
echo START LISTENER(LST%1) | runmqsc %1
if full==%4 echo ALTER QMGR REPOS(CL1) DEADQ(SYSTEM.DEAD.LETTER.QUEUE) | runmqsc %1
echo DEFINE CHANNEL(TO.%2) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME('LOCALHOST(%3)') CLUSTER(CL1)
REPLACE | runmqsc %1
echo DEFINE CHANNEL(TO.%1) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME('LOCALHOST(%p%)')
CLUSTER(CL1) REPLACE | runmqsc %1
goto:eof

```

Rysunek 28. Utwórz menedżery kolejek: qmgrs.bat

Zaktualizuj konfigurację, dodając subskrypcje do tematów klastra.

```

@echo off
for %%A in (QMA QMB) do runmqsc %%A < wildsubs.tst
for %%A in (QMA QMB) do runmqsc %%A < upsubs.tst

```

Rysunek 29. Aktualizuj subskrypcje: upsubs.bat

Uruchom program pub.bat z menedżerem kolejek jako parametrem, aby opublikować komunikaty zawierające łańcuch tematu publikacji. Pub.bat korzysta z programu przykładowego **amqspub**.

```

@echo off
@rem Provide queue manager name as a parameter
set S=Sports
set S=6 Sports/Football Sports/Football/Arsenal
set S=6 Sports/Rugby Sports/Rugby/Leeds
for %%B in (6) do echo %%B | amqspub %%B %1

```

Rysunek 30. Publikowanie: pub.bat

Strumienie i tematy

W kolejce publikowania/subskrypcji występuje pojęcie strumienia publikacji, który nie istnieje w zintegrowanym modelu publikowania/subskrybowania. W kolejkowanych publikacji/subskrypcji strumienie umożliwiają oddzielenie przepływu informacji dla różnych tematów. Strumień jest implementowany jako temat najwyższego poziomu, który można odwzorować administracyjnie na inny identyfikator tematu.

Strumień domyślny SYSTEM.BROKER.DEFAULT.STREAM jest konfigurowany automatycznie dla wszystkich brokerów i menedżerów kolejek w sieci i nie jest wymagana żadna dodatkowa konfiguracja w celu użycia strumienia domyślnego. Strumień domyślny należy traktować jako nienazwany domyślny obszar tematu. Tematy opublikowane w strumieniu domyślnym są natychmiast dostępne dla wszystkich połączonych menedżerów kolejek z włączoną kolejką publikowania/subskrybowania. Nazwane strumienie są podobne do oddzielnych, nazwanych obszarów tematów. Nazwany strumień musi być zdefiniowany w każdym brokerze, w którym jest używany.

Jeśli publikatory i subskrybenty znajdują się w różnych menedżerach kolejek, po połączeniu brokerów w tej samej hierarchii brokerów nie jest wymagana żadna dodatkowa konfiguracja dla publikacji i subskrypcji, które mają przepływać między nimi. Ta sama interoperacyjność działa również w odwrotnym kierunku.

Nazwane strumienie

Projektant rozwiązania pracujący z umieszczonym w kolejce modelem programowania publikowania/subskrypcji może zdecydować o umieszczeniu wszystkich publikacji sportowych w nazwanym strumieniu o nazwie Sport. Aby strumień był dostępny dla menedżera kolejek działającego w systemie IBM MQ z włączoną kolejką publikowania/subskrypcji, należy dodać strumień ręcznie.

Umieszczone w kolejce aplikacje publikowania/subskrypcji, które subskrybują Soccer/Results w strumieniu Sport, działają bez zmian. Zintegrowane aplikacje publikowania/subskrybowania, które subskrybują temat Sport przy użyciu MQSUBi dostarczają łańcuch tematu Soccer/Results, również otrzymują te same publikacje.

Zadanie dodawania strumienia zostało opisane w temacie [Dodawanie strumienia](#). Konieczne może być ręczne dodanie strumieni z dwóch powodów.

1. Zamiast migrowania aplikacji do zintegrowanego interfejsu publikowania/subskrybowania MQI, należy kontynuować tworzenie umieszczonych w kolejce aplikacji publikowania/subskrybowania, które działają w menedżerach kolejek w nowszej wersji.
2. Domyślne odwzorowanie strumieni na tematy prowadzi do "kolizji" w obszarze tematu, a publikacje w strumieniu mają ten sam łańcuch tematu co publikacje z innego miejsca.

Uprawnienia

Domyślnie w katalogu głównym drzewa tematów znajduje się wiele obiektów tematu: SYSTEM.BASE.TOPIC, SYSTEM.BROKER.DEFAULT.STREAM, SYSTEM.BROKER.DEFAULT.SUBPOINT. Uprawnienia (na przykład do publikowania lub subskrybowania) są określane przez uprawnienia na SYSTEM.BASE.TOPIC; wszelkie uprawnienia w systemie SYSTEM.BROKER.DEFAULT.STREAM lub SYSTEM.BROKER.DEFAULT.SUBPOINT są ignorowane. Jeśli jeden z obiektów SYSTEM.BROKER.DEFAULT.STREAM lub SYSTEM.BROKER.DEFAULT.SUBPOINT zostanie usunięty i ponownie utworzony z niepustym łańcuchem tematu, uprawnienia zdefiniowane dla tych obiektów będą używane w taki sam sposób, jak normalny obiekt tematu.

Odwzorowanie między strumieniami i tematami

Umieszczony w kolejce strumień publikowania/subskrypcji jest naśladowany w programie IBM MQ przez utworzenie kolejki i nadanie mu takiej samej nazwy jak strumieniowi. Czasami kolejka jest nazywana kolejką strumienia, ponieważ w ten sposób jest wyświetlana w kolejkach aplikacji publikowania/subskrybowania. Kolejka jest identyfikowana przez mechanizm publikowania/subskrypcji przez dodanie jej do specjalnej listy nazw o nazwie SYSTEM.QPUBSUB.QUEUE.NAMELIST. Można dodać dowolną liczbę strumieni, dodając dodatkowe kolejki specjalne do listy nazw. Na koniec należy dodać tematy o takich samych nazwach jak strumienie i tych samych łańcuchach tematów co nazwa strumienia, aby można było publikować i subskrybować tematy.

Jednak w wyjątkowych okolicznościach można nadać tematami odpowiadającym strumieniom dowolny łańcuch tematu wybrany podczas definiowania tematów. Celem łańcucha tematu jest nadanie tematowi unikalnej nazwy w obszarze tematu. Zazwyczaj nazwa strumienia służy temu celowi doskonale. Czasami nazwa strumienia i istniejąca nazwa tematu kolidują ze sobą. Aby rozwiązać ten problem, wybierz inny łańcuch tematu dla tematu powiązanego ze strumieniem. Wybierz dowolny łańcuch tematu, upewniając się, że jest on unikalny.

Łańcuch tematu zdefiniowany w definicji tematu jest dodawany w normalny sposób do łańcucha tematu udostępnianego przez publikatory i subskrybentów przy użyciu wywołań MQI produktu MQOPEN lub MQSUB. Wybór przedrostka łańcucha tematu nie ma wpływu na aplikacje odwotujące się do tematów korzystających z obiektów tematu. Dlatego można wybrać dowolny łańcuch tematu, który zachowuje unikalność publikacji w obszarze tematu.

Zmiana odwzorowania różnych strumieni na różne tematy jest zależna od przedrostków używanych dla łańcuchów tematów, które są unikalne, w celu całkowitego oddzielenia jednego zestawu tematów od innego. Należy zdefiniować uniwersalną konwencję nazewnictwa tematów, która będzie ściśle przestrzegana, aby odwzorowanie działało.

W produkcie IBM MQ do zmiany odwzorowania łańcucha tematu na inne miejsce w obszarze tematu używany jest mechanizm przedrostkowy.

Uwaga: Podczas usuwania strumienia najpierw należy usunąć wszystkie subskrypcje w strumieniu. To działanie jest najważniejsze, jeśli dowolna z subskrypcji pochodzi z innych brokerów w hierarchii brokerów.

Punkty subskrypcji i tematy

Nazwane punkty subskrypcji są emulowane przez tematy i obiekty tematów.

Informacje na temat ręcznego dodawania punktów subskrypcji zawiera sekcja [Dodawanie punktu subskrypcji](#).

Punkty subskrypcji w programie IBM MQ

Produkt IBM MQ odwzorowuje punkty subskrypcji na różne obszary tematów w drzewie tematów produktu IBM MQ. Tematy w komunikatach komend bez punktu subskrypcji są odwzorowywane bez zmian na element główny drzewa tematów IBM MQ i dziedziczą właściwości z pliku SYSTEM.BASE.TOPIC.

Komunikaty komend z punktem subskrypcji są przetwarzane przy użyciu listy obiektów tematu w produkcie SYSTEM.QPUBSUB.SUBPOINT.NAMELIST. Nazwa punktu subskrypcji w komunikacie komendy jest dopasowywana do łańcucha tematu dla każdego obiektu tematu na liście. Jeśli zostanie znalezione dopasowanie, nazwa punktu subskrypcji jest dołączana do łańcucha tematu jako węzeł tematu. Temat dziedziczy właściwości z powiązanego obiektu tematu znajdującego w pliku SYSTEM.QPUBSUB.SUBPOINT.NAMELIST.

Efektom użycia punktów subskrypcji jest utworzenie osobnego obszaru tematu dla każdego punktu subskrypcji. Obszar tematu jest głównym tematem, który ma taką samą nazwę jak punkt subskrypcji. Tematy w każdym obszarze tematu dziedziczą swoje właściwości z obiektu tematu o takiej samej nazwie jak punkt subskrypcji.

Wszystkie właściwości, które nie zostały ustawione w zgodnym obiekcie tematu, są dziedziczone w normalny sposób z pliku SYSTEM.BASE.TOPIC.

Istniejące kolejgowane aplikacje publikowania/subskrybowania, korzystające z nagłówek komunikatów MQRFH2, kontynuują pracę, ustawiając właściwość **SubPoint** w komunikatach komend `Publish` lub `Register subscriber`. Punkt subskrypcji jest łączony z łańcuchem tematu w komunikacie komendy, a temat wynikowy jest przetwarzany jak każdy inny.

Punkty subskrypcji nie mają wpływu na aplikacje IBM MQ. Jeśli aplikacja korzysta z tematu, który dziedziczy informacje z jednego ze zgodnych obiektów tematu, aplikacja ta współdziała z aplikacją umieszczoną w kolejce przy użyciu zgodnego punktu subskrypcji.

Przykład

Istniejący WebSphere Message Broker (obecnie znany jako IBM Integration Bus) Aplikacja publikowania/subskrypcji, która została zmigrowana do IBM MQ utworzyła dwa obiekty tematu, GBP i USD, z odpowiednimi łańcuchami tematów 'GBP' i 'USD'.

Istniejące publikatory do tematu NYSE/IBM/SPOT, poddane migracji w celu uruchomienia w systemie IBM MQ, które korzystają z punktu subskrypcji USD tworzą publikacje w temacie USD/NYSE/IBM/SPOT. Podobnie jak istniejący subskrybenci programu NYSE/IBM/SPOT, korzystając z punktu subskrypcji USD utwórz subskrypcje programu USD/NYSE/IBM/SPOT.

Zasubskrybuj cenę punktową w programie publikowania/subskrybowania IBM MQ , wywołując metodę MQSUB. Utwórz subskrypcję przy użyciu obiektu tematu USD i łańcucha tematu 'NYSE/IBM/SPOT', jak to pokazano we fragmencie kodu "C".

```
stncpy(sd.ObjectName, "USD", MQ_TOPIC_NAME_LENGTH);  
sd.ObjectString.VSPtr = "NYSE/IBM/SPOT";  
sd.ObjectString.VSLength = MQVS_NULL_TERMINATED;  
MQSUB(Hconn, &sd, &Hobj, &Hsub, &CompCode, &Reason);
```

1. Ustaw atrybut CLUSTER obiektów tematu USD i GBP na hoście tematu klastra.
2. Usuń wszystkie kopie obiektów tematów USD i GBP w innych menedżerach kolejek w klastrze.
3. Upewnij się, że wartości USD i GBP są zdefiniowane w pliku SYSTEM.QPUBSUB.SUBPOINT.NAMELIST w każdym menedżerze kolejek w klastrze.

Przykład konfiguracji publikowania/subskrybowania pojedynczego menedżera kolejek

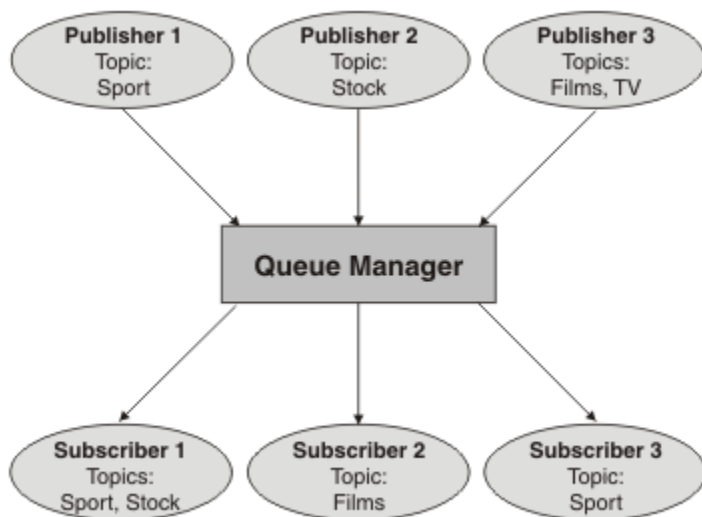
Na ilustracji [Rysunek 31](#) na stronie 90 przedstawiono podstawową konfigurację publikowania/subskrybowania pojedynczego menedżera kolejek. Przykład przedstawia konfigurację usługi wiadomości, w której informacje o kilku tematach są dostępne od wydawców:

- Wydawca 1 publikuje informacje o wynikach sportowych przy użyciu tematu Sport
- Wydawca 2 publikuje informacje o cenach akcji używając tematu Akcje
- Wydawca 3 publikuje informacje o recenzjach filmów za pomocą tematu Filmy, oraz o listingach telewizyjnych za pomocą tematu TV

Trzech subskrybentów zgłosiło zainteresowanie różnymi tematami, dlatego menedżer kolejek wysyła im informacje, którymi są zainteresowani:

- Abonent 1 otrzymuje wyniki sportowe i ceny akcji
- Abonent 2 otrzymuje recenzje filmów
- Subskrybent 3 otrzymuje wyniki sportowe

Żaden z abonentów nie zarejestrował udziału w programach telewizyjnych, więc nie są one dystrybuowane.



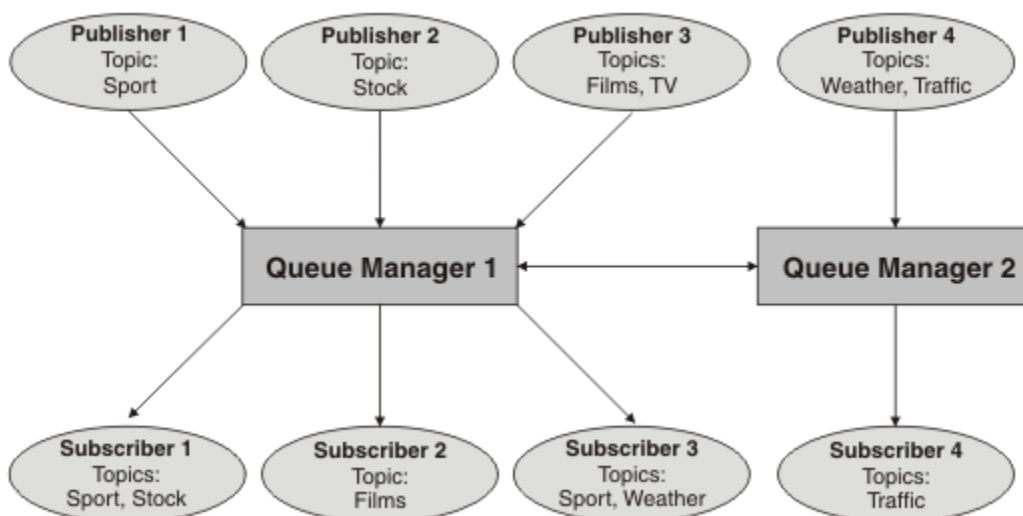
Rysunek 31. Przykład publikowania/subskrybowania pojedynczego menedżera kolejek

Rozproszone sieci publikowania/subskrypcji

Każdy menedżer kolejek dopasowuje komunikaty opublikowane w temacie do lokalnie utworzonych subskrypcji, które zasubskrybowały ten temat. Sieć menedżerów kolejek można skonfigurować w taki sposób, aby komunikaty publikowane przez aplikację połączoną z jednym menedżerem kolejek były dostarczane do zgodnych subskrypcji utworzonych w innych menedżerach kolejek w sieci. Wymaga to dodatkowej konfiguracji w prostych kanałach między menedżerami kolejek.

Konfiguracja rozproszonego publikowania/subskrypcji to zestaw połączonych ze sobą menedżerów kolejek. Wszystkie menedżery kolejek mogą znajdować się w tym samym systemie fizycznym lub mogą być rozproszone w kilku systemach fizycznych. Po połączeniu menedżerów kolejek subskrybenci mogą subskrybować jeden menedżer kolejek i odbierać komunikaty, które zostały początkowo opublikowane w innym menedżerze kolejek. Aby to zilustrować, poniższy rysunek dodaje drugi menedżer kolejek do konfiguracji opisanej w sekcji [“Przykład konfiguracji publikowania/subskrybowania pojedynczego menedżera kolejek”](#) na stronie 90.

- Menedżer kolejek 2 jest używany przez publikator 4 do publikowania informacji o prognozie pogody, przy użyciu tematu Pogoda, oraz informacji o warunkach ruchu na głównych drogach, przy użyciu tematu Ruch.
- Subskrybent 4 używa również tego menedżera kolejek i subskrybuje informacje o warunkach ruchu danych przy użyciu tematu Ruch danych.
- Subskrybent 3 subskrybuje również informacje o warunkach pogodowych, nawet jeśli używa innego menedżera kolejek niż publikator. Jest to możliwe, ponieważ menedżery kolejek są ze sobą powiązane.

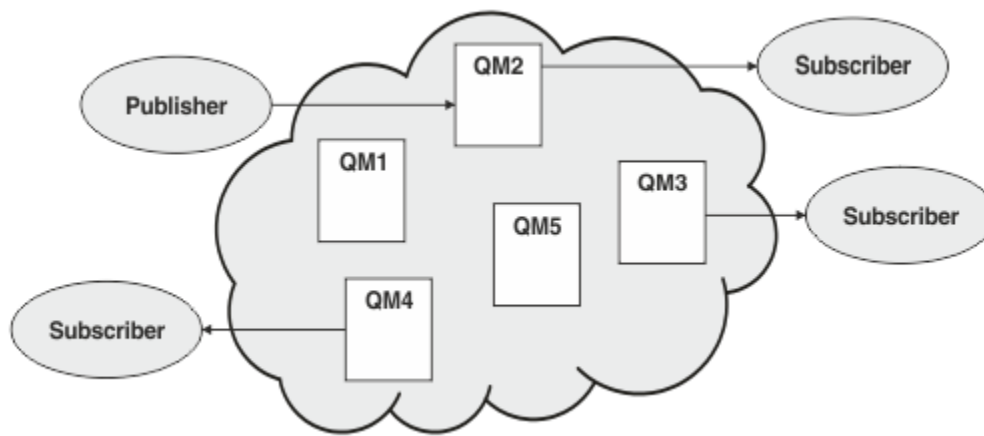


Rysunek 32. Przykład publikowania/subskrybowania z dwoma menedżerami kolejek

Można ręcznie połączyć menedżery kolejek w hierarchii nadrzędnej i podrzędnej lub utworzyć klaster publikowania/subskrypcji i pozwolić produktowi IBM MQ na zdefiniowanie większości szczegółów połączenia. Można również użyć obu topologii w połączeniu, na przykład łącząc kilka klastrów w hierarchii.

Przegląd klastrów publikowania/subskrypcji

Klaster publikowania/subskrypcji jest standardowym klastrem z co najmniej jednym obiektem tematu dodanym do klastra. Po zdefiniowaniu obiektu tematu administracyjnego w dowolnym menedżerze kolejek w klastrze i utworzeniu klastra dla tego obiektu tematu przez podanie nazwy klastra, publikatory i subskrybenci tematu mogą łączyć się z dowolnym menedżerem kolejek w klastrze, a publikowane komunikaty są kierowane do subskrybentów za pośrednictwem kanałów klastra między menedżerami kolejek.



Rysunek 33. klastro publikowania/subskrypcji

Istnieją dwa sposoby konfiguracji sposobu kierowania komunikatami publikowania/subskrypcji w klastrze:

- kierowanie bezpośrednie
- kierowanie hostami tematów

Podczas konfiguracji bezpośredniego kierowanego tematu klastrowego komunikaty publikowane w jednym menedźerze kolejek są wysyłane bezpośrednio z tego menedźera kolejek do każdej subskrypcji w dowolnym innym menedźerze kolejek w klastrze. Może to być najbardziej bezpośrednia ścieżka dla publikacji, ale powoduje, że wszystkie menedżery kolejek w klastrze będą informować o wszystkich innych menedżerach kolejek, z których każdy może mieć ustanowione kanały klastra między nimi.

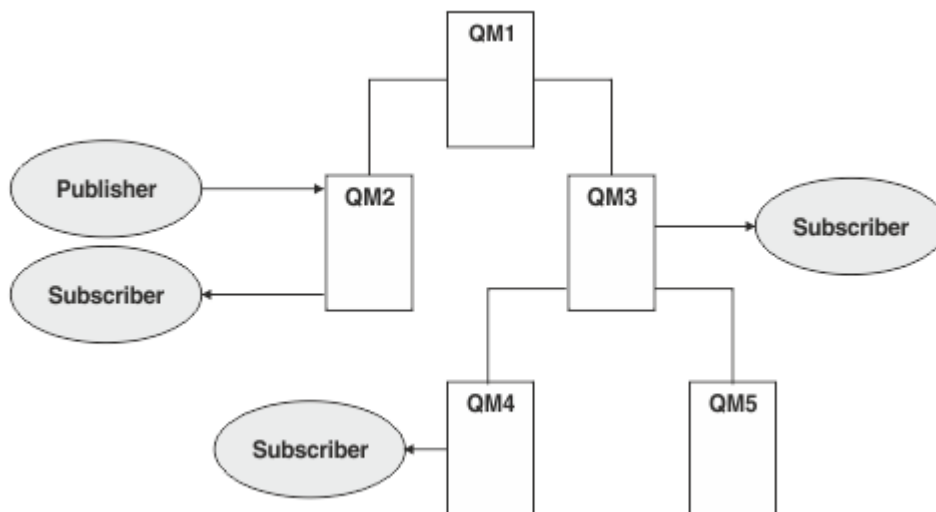
Jeśli używane jest kierowanie hostami tematów, komunikaty publikowane w jednym menedźerze kolejek są wysyłane z tego menedźera do menedźera kolejek, który udostępnia definicję administrowanego obiektu tematu. Ten *menedźer kolejek hosta tematu* kieruje komunikatami do każdej subskrypcji we wszystkich pozostałych menedżerach kolejek w klastrze. Jeśli publikatory lub subskrybenci nie znajdują się w menedżerach kolejek hostów tematów, spowoduje to dłuższą trasę dla publikacji. Korzyścią jest jednak to, że tylko menedżery kolejek hostów tematów mają informacje o wszystkich innych menedżerach kolejek w klastrze i mogą mieć ustanowione kanały klastra.

Więcej informacji na ten temat zawiera sekcja [“Klastry publikowania/subskrypcji”](#) na stronie 94.

Przegląd hierarchii publikowania/subskrypcji

Hierarchia publikowania/subskrypcji to zestaw menedżerów kolejek połączonych kanałami w strukturę hierarchiczną. Każdy menedźer kolejek identyfikuje swój *nadrzędny* menedźer kolejek zgodnie z opisem w sekcji [Nawiązywanie połączenia menedźera kolejek z hierarchią publikowania/subskrypcji](#).

Publikatory i subskrybenci tematu mogą łączyć się z dowolnym menedżerem kolejek w hierarchii, a komunikaty przepływają między nimi za pomocą hierarchicznego połączenia menedźera kolejek.



Rysunek 34. Hierarchia publikowania/subskrypcji

Na poprzednim rysunku publikacje dostarczone do subskrybentów QM3 i QM4 zostały przekierowane z QM2 do QM1, a następnie do QM3 i na końcu QM4.

Hierarchie dają bezpośrednią kontrolę nad relacjami między każdym menedżerem kolejek w hierarchii. Umożliwia to precyzyjną kontrolę nad kierowaniem komunikatów z publikatorów do subskrybentów i jest szczególnie przydatne w przypadku kierowania komunikatów między sieciami menedżera kolejek z ograniczonymi połączeniami. Należy dokładnie rozważyć dostępność i możliwości każdego menedżera kolejek, przez który komunikat jest kierowany z publikatora do subskrybentów.

Więcej informacji na ten temat zawiera sekcja [“Hierarchie publikowania/subskrypcji”](#) na stronie 96.

Dystrybucja publikacji między menedżerami kolejek

Oprócz opcji routingu istnieją dwa podejścia do dystrybucji publikacji w sieci menedżerów kolejek:

- Publikacje są wysyłane tylko z jednego menedżera kolejek do menedżerów kolejek, które aktualnie udostępniają subskrypcję tej publikacji.
- Wyślij każdą publikację do wszystkich menedżerów kolejek i pozwól im dopasować ją do swoich subskrypcji.

Pierwsze z nich powoduje wysyłanie komunikatów publikowania tylko wtedy, gdy jest to konieczne, ale wymaga, aby poziom wiedzy o subskrypcji był współużytkowany przez menedżery kolejek. Ta ostatnia nie wymaga współużytkowania informacji o subskrypcji, ale może spowodować wysyłanie zbędnych komunikatów publikowania między menedżerami kolejek.

Domyślnie program IBM MQ używa poprzedniej metody, w której publikacje są wysyłane tylko do menedżerów kolejek, które mają dla nich subskrypcje. Wiedza o subskrypcji jest propagowana między menedżerami kolejek w postaci *subskrypcji proxy*. Zależy to od dystrybucji i czasu życia subskrypcji oraz częstotliwości publikacji, która jest najbardziej wydajna w topologii rozproszonego publikowania/subskrypcji. Patrz sekcja [Wydajność subskrypcji w sieciach publikowania/subskrypcji](#).

Pojęcia pokrewne

“Drzewa tematów” na stronie 77

Każdy definiowany temat stanowi element lub węzeł drzewa tematów. Drzewo tematów może być puste, aby rozpocząć się od tematów lub zawierać tematy, które zostały wcześniej zdefiniowane za pomocą komend MQSC lub PCF. Nowy temat można zdefiniować przy użyciu komend tworzenia tematu lub przez określenie tematu po raz pierwszy w publikacji lub subskrypcji.

[Scenariusze hierarchii publikowania/subskrypcji](#)

Zadania pokrewne

[Projektowanie klastrów publikowania/subskrypcji](#)

Klustry publikowania/subskrypcji

Klaster publikowania/subskrypcji to standardowy klaster połączonych ze sobą menedżerów kolejek, w którym publikacje są automatycznie przenoszone z aplikacji publikowania do subskrypcji istniejących w dowolnym z menedżerów kolejek w klastrze. Istnieją dwie opcje kierowania publikacji w ramach klastra publikowania/subskrybowania: *kierowanie bezpośrednio* i *kierowanie hostami tematów*. Wybrany routing zależy od wielkości i oczekiwanych wzorców aktywności dla klastra.

Klaster używany na potrzeby przesyłania komunikatów w trybie publikowania/subskrypcji nie różni się od standardowego klastra IBM MQ. W związku z tym menedżery kolejek w klastrze publikowania/subskrypcji mogą istnieć na fizycznie oddzielnych komputerach, a każda para menedżerów kolejek jest w razie potrzeby automatycznie połączona ze sobą za pomocą kanałów klastra. Więcej informacji na ten temat zawiera sekcja [Klustry](#).

Aby skonfigurować standardowy klaster menedżerów kolejek na potrzeby przesyłania komunikatów publikacji/subskrypcji, należy zdefiniować co najmniej jeden administrowany obiekt tematu w menedżerze kolejek klastra. Aby temat był tematem klastra, należy skonfigurować właściwość **CLUSTER** z nazwą klastra. W takim przypadku każdy temat używany przez publikator lub subskrybent w danym punkcie lub poniżej [drzewa tematów](#) jest współużytkowany przez wszystkie menedżery kolejek w klastrze, a komunikaty publikowane w klastrowej gałęzi drzewa tematów są automatycznie kierowane do subskrypcji w innych menedżerach kolejek w klastrze.

Między menedżerem kolejek publikatora i każdym innym menedżerem kolejek wysyłana jest tylko jedna kopia każdego komunikatu, niezależnie od liczby subskrybentów komunikatu w docelowym menedżerze kolejek. Po nadejściu do menedżera kolejek z co najmniej jedną subskrypcją komunikat jest duplikowany we wszystkich subskrypcjach.

Każdy menedżer kolejek dołączający do klastra automatycznie otrzymuje informacje o tematach w klastrze, a publikatory i subskrybenty w tym menedżerze kolejek automatycznie uczestniczą w klastrze.

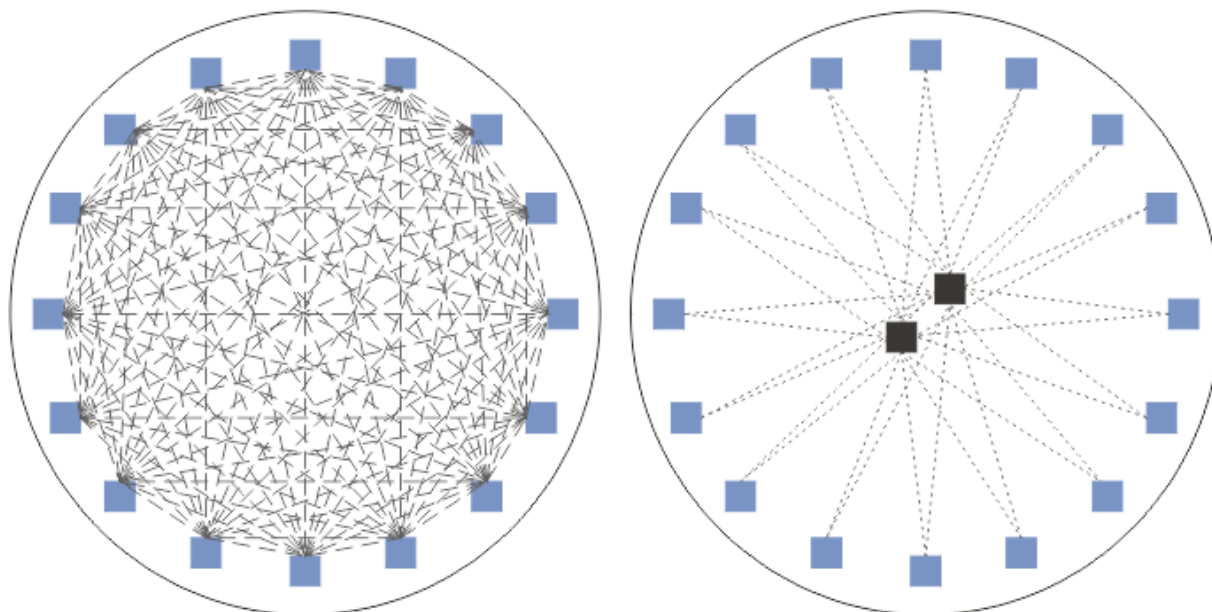
Działanie publikowania/subskrybowania nieklastrowego może również odbywać się w klastrze publikowania/subskrybowania, pracując z łańcuchami tematów, które nie należą do klastrowego obiektu tematu.

Istnieją dwie opcje kierowania publikacji w ramach klastra publikowania/subskrybowania: *kierowanie bezpośrednio* i *kierowanie hostami tematów*. Aby wybrać metodę kierowania komunikatów do użycia w ramach klastra, należy ustawić właściwość **CLROUTE** w administrowanym obiekcie tematu na jedną z następujących wartości:

- **DIRECT**
- **TOPICHOST**

Domyślnie kierowanie tematami odbywa się w systemie **DIRECT**. Była to jedyna opcja w produktach starszych niż IBM MQ 8.0. Po skonfigurowaniu bezpośredniego kierowanego tematu klastra w menedżerze kolejek wszystkie menedżery kolejek w klastrze będą powiadomione o obecności wszystkich innych menedżerów kolejek w klastrze. Podczas wykonywania operacji publikowania i subskrypcji każdy menedżer kolejek może nawiązać bezpośrednie połączenie z dowolnym innym menedżerem kolejek w klastrze.

W produkcie IBM MQ 8.0 można zamiast tego skonfigurować kierowanie tematami za pomocą opcji **TOPICHOST**. Jeśli używane jest kierowanie hostami tematów, wszystkie menedżery kolejek w klastrze będą powiadomione o menedżerach kolejek klastra, które udostępniają definicje kierowanych tematów (czyli o menedżerach kolejek, w których zdefiniowano obiekt tematu). Podczas wykonywania operacji publikowania i subskrypcji menedżery kolejek w klastrze nawiązują połączenie tylko z tymi menedżerami kolejek hostów tematów, a nie bezpośrednio ze sobą. Menedżery kolejek hostów tematów są odpowiedzialne za kierowanie publikacji z menedżerów kolejek, na których publikacje są publikowane, do menedżerów kolejek ze zgodnymi subskrypcjami.



Rysunek 35. Kierowanie bezpośrednio i kierowanie hostami tematów

Przegląd kierowania bezpośrednio

Jeśli administrowany obiekt tematu jest skonfigurowany na potrzeby kierowania bezpośrednio, obiekt tematu musi być zdefiniowany tylko w jednym z menedżerów kolejek w klastrze, aby wszystkie menedżery kolejek mogły się o nim dowiedzieć. Wybór menedżera kolejek, w którym zdefiniowano temat, nie ma wpływu na zachowanie przesyłania komunikatów w trybie publikowania/subskrypcji dla tematu.

Każdy komunikat przepływa bezpośrednio z menedżera kolejek publikatora do każdej subskrypcji w innych menedżerach kolejek w klastrze, bez przekazywania przez pośrednie menedżery kolejek.

Domyślnie komunikaty są wysyłane tylko do innych menedżerów kolejek w klastrze, które udostępniają co najmniej jedną subskrypcję.

- Zależy to od tego, czy każdy menedżer kolejek bezpośrednio informuje wszystkie inne menedżery kolejek w klastrze o wszystkich tematach, które aktualnie mają co najmniej jedną subskrypcję. Powoduje to, że wszystkie menedżery kolejek w klastrze mają informacje o wszystkich subskrybowanych tematach oraz o każdym menedżerze kolejek, który udostępnia subskrypcję ustanawiającą kanał dla każdego innego menedżera kolejek. Jest to niezależne od tego, czy każdy menedżer kolejek ma publikator.
- Znajomość poszczególnych subskrybowanych tematów we wszystkich menedżerach kolejek można usunąć, zmieniając model wysyłania wszystkich publikacji do wszystkich menedżerów kolejek w klastrze, bez względu na to, czy mają one subskrypcje. Zmniejsza to ruch wiedzy o subskrypcji, ale prawdopodobnie zwiększa ruch związany z publikacją i liczbę kanałów ustanawianych przez każdy menedżer kolejek. Patrz sekcja [Wydajność subskrypcji w sieciach publikowania/subskrypcji](#).

Przepływy komunikatów publikowania/subskrypcji używające bezpośrednio kierowanych tematów klastrowych mogą obejmować wiele klastrów publikowania/subskrypcji, dodając po jednym menedżerze kolejek z każdego klastra do hierarchii publikowania/subskrypcji. Patrz sekcja [Łączenie obszarów tematów wielu klastrów](#).

Bardziej szczegółową eksplorację kierowania bezpośrednio można znaleźć w sekcji [Kierowanie bezpośrednio w klastrach publikowania/subskrypcji](#).

Przegląd kierowania hostami tematów

Jeśli administrowany obiekt tematu jest skonfigurowany na potrzeby kierowania hostami tematów, publikacje z menedżera kolejek w klastrze są kierowane przez menedżer kolejek, w którym jest

skonfigurowany obiekt tematu ("host tematu"), a stamtąd do menedżerów kolejek, w których istnieją subskrypcje.

- Zależy to od tego, czy każdy menedżer kolejek informuje wszystkie hosty tematów o każdym temacie, który aktualnie ma co najmniej jedną subskrypcję. Każdy menedżer kolejek udostępniający subskrypcję ustanawia kanał dla każdego hosta tematu, do którego odnosi się subskrypcja.
- Menedżery kolejek, które nie są udostępniane w ramach tematu, nie są informowane o innych menedżerach kolejek, które nie są udostępniane w ramach tematu w klastrze na potrzeby publikowania/subskrybowania, a kanały nie są ustanawiane między nimi w tym celu.
- Jeśli aplikacja publikująca jest połączona z menedżerem kolejek udostępniającym temat, opublikowane komunikaty są kierowane bezpośrednio do menedżerów kolejek, w których utworzono zgodne subskrypcje, bez konieczności dodatkowego przeskoku. Podobnie, jeśli zgodne subskrypcje są tworzone w jedynym menedżerze kolejek udostępniającym temat, komunikaty publikowane w tym temacie są kierowane bezpośrednio do tego menedżera kolejek bez konieczności dodatkowego przeskoku.
- Subskrypcje w tym samym menedżerze kolejek co publikator są spełnione bez uprzedniego kierowania publikacji do hostów obiektu tematu.

Podobnie jak w przypadku kolejek klastrowych, wiele menedżerów kolejek może skonfigurować ten sam obiekt tematu administracyjnego. Zapewnia to większą dostępność routingu komunikatów i skalowanie poziome poprzez równoważenie obciążenia. W przypadku obiektów tematu kierowanych przez hosta tematów, gdy wiele menedżerów kolejek konfiguruje ten sam nazwany temat dla tej samej gałęzi drzewa tematów, każdy host tematu jest informowany o subskrybowanych tematach przez każdy menedżer kolejek udostępniający subskrypcję.

- Po opublikowaniu komunikatu jest on wysyłany do jednego z menedżerów kolejek hostów tematów w celu przekazania go do menedżerów kolejek udostępniających subskrypcję. Wybór menedżera kolejek hosta tematu jest zgodny z tymi samymi domyślnymi regułami równoważenia obciążenia, co w przypadku kolejek klastrowych typu punkt z punktem.
- Jeśli menedżer kolejek publikowania nie może skontaktować się z co najmniej jednym menedżerem kolejek hostów tematów, komunikaty są kierowane do pozostałych dostępnych menedżerów kolejek hostów tematów.

Każda publikacja tematu w kierowanej gałęzi drzewa tematów jest przekazywana do jednego z hostów tematów, nawet jeśli w dowolnym miejscu w klastrze nie ma subskrypcji tego tematu. Domyślnie komunikaty są wysyłane z tego miejsca tylko do innych menedżerów kolejek w klastrze, które udostępniają co najmniej jedną subskrypcję.

- Zależy to od tego, czy każdy menedżer kolejek hosta tematu jest informowany o wszystkich subskrybowanych łańcuchach tematów w każdym menedżerze kolejek w klastrze.
- Wiedzę na temat poszczególnych subskrybowanych tematów można usunąć, zmieniając na model wysyłania wszystkich publikacji kierowanych do hosta tematu do wszystkich menedżerów kolejek w klastrze, niezależnie od tego, czy mają one subskrypcje. Zmniejsza to ruch wiedzy o subskrypcji, ale prawdopodobnie zwiększy ruch związany z publikacją i potencjalnie liczbę kanałów ustanowionych dla każdego tematu udostępniającego menedżer kolejek. Patrz sekcja [Wydajność subskrypcji w sieciach publikowania/subskrypcji](#).

Przepływy komunikatów publikowania/subskrypcji używające tematów klastrowych kierowanych przez hosta tematów **nie mogą** obejmować wielu klastrów publikowania/subskrypcji za pomocą hierarchii publikowania/subskrybowania.

Bardziej szczegółowe informacje na temat kierowania hostami tematów zawiera sekcja [Kierowanie hostami tematów w klastrach publikowania/subskrypcji](#).

Hierarchie publikowania/subskrypcji

Hierarchię publikowania/subskrypcji można zbudować, łącząc menedżery kolejek za pomocą kanałów, a następnie definiując relację podrzędny-nadrzędny między parami menedżerów kolejek. Komunikat przepływa od publikatora do subskrypcji przez bezpośrednie relacje w hierarchii. Należy zauważyć, że może to oznaczać wiele "przeskoków" .

Tylko jedna kopia komunikatu jest wysyłana między dowolną parą menedżerów kolejek, niezależnie od liczby subskrybentów komunikatu w docelowym menedżerze kolejek. Po nadejściu do menedżera kolejek z co najmniej jedną subskrypcją komunikat jest duplikowany we wszystkich subskrypcjach.

Domyślnie komunikaty są wysyłane tylko do innych menedżerów kolejek w hierarchii, które znajdują się w trasie do subskrypcji w innym menedżerze kolejek:

- Zależy to od każdego menedżera kolejek informującego o każdej bezpośredniej relacji wszystkich tematów, które aktualnie mają jedną lub więcej subskrypcji w tym menedżerze kolejek lub w jednej z jego innych relacji. Spowoduje to, że wszystkie menedżery kolejek w hierarchii będą informowane o wszystkich subskrybowanych tematach.
- To zachowanie można zmienić, aby zawsze wysyłać publikacje do wszystkich menedżerów kolejek w hierarchii, bez względu na istniejące subskrypcje. Eliminuje to potrzebę propagowania informacji o subskrypcji w hierarchii, ale może zwiększyć ruch publikowania.

Podczas tworzenia klastra należy uważać, aby nie utworzyć pętli powodującej powtarzanie się komunikatów w sieci w nieskończoność. W hierarchii nie można tworzyć takich pętli.

Każdy menedżer kolejek musi mieć unikalną nazwę menedżera kolejek.

Przeptywy komunikatów publikowania/subskrybowania mogą obejmować wiele klastrów publikowania/subskrybowania. W tym celu należy dodać po jednym menedżerze kolejek z każdego klastra do hierarchii publikowania/subskrypcji.

Bardziej szczegółową eksplorację zawiera sekcja [Kierowanie w hierarchiach publikowania/subskrypcji](#).

Subskrypcje proxy w sieci publikowania/subskrypcji

Subskrypcja proxy to subskrypcja utworzona przez jeden menedżer kolejek i dotycząca tematów publikowanych w innym menedżerze kolejek. Subskrypcja proxy przepływa między menedżerami kolejek dla każdego pojedynczego łańcucha tematów, który subskrybuje subskrypcja. Nie można utworzyć subskrypcji proxy jawnie. Menedżer kolejek robi to w imieniu bieżącego użytkownika.

Menedżery kolejek można połączyć ze sobą w klastrze publikowania/subskrybowania lub w hierarchii publikowania/subskrybowania. Przepływ subskrypcji proxy między połączonymi menedżerami kolejek. Subskrypcje proxy powodują, że publikacje w temacie utworzonym przez publikator połączony z jednym menedżerem kolejek są odbierane przez subskrybentów tego tematu połączonych z innymi menedżerami kolejek. Patrz ["Rozproszone sieci publikowania/subskrypcji"](#) na stronie 91.

W topologiach publikowania/subskrypcji z wieloma tysiącami subskrypcji pojedynczych łańcuchów tematów lub w miejscach, w których istnienie tych subskrypcji może się szybko zmieniać, należy wziąć pod uwagę narzut propagacji subskrypcji proxy. Oprócz automatycznej agregacji opisanej w dalszej części tego tematu można wprowadzić ręczne zmiany w konfiguracji, które dodatkowo ograniczają przepływ subskrypcji proxy i publikacji między połączonymi menedżerami kolejek oraz zmniejszają opóźnienie oczekiwania na propagację subskrypcji proxy do wszystkich połączonych menedżerów kolejek. Patrz sekcja [Wydajność subskrypcji w sieciach publikowania/subskrypcji](#).

Subskrypcje proxy nie zawierają żadnych selektorów używanych przez subskrypcje lokalne, a łańcuchy tematów subskrypcji zawierające znaki wieloznaczne mogą zostać uproszczone. Może to spowodować, że publikacje będą zgodne z subskrypcjami proxy, a rzeczywiste subskrypcje nie, co spowoduje dodatkowy przepływ publikacji między menedżerami kolejek. Menedżer kolejek udostępniający subskrypcje odfiltrowuje takie rozbieżności, aby dodatkowe publikacje nie były zwracane do subskrypcji.

Agregacja subskrypcji proxy

Subskrypcje proxy są agregowane przy użyciu systemu eliminowania duplikatów. W przypadku konkretnego przetłumaczonego łańcucha tematu subskrypcja proxy jest wysyłana w pierwszej subskrypcji lokalnej lub otrzymanej subskrypcji proxy. Kolejne subskrypcje tego samego łańcucha tematu korzystają z tej istniejącej subskrypcji proxy.

Subskrypcja proxy jest anulowana po anulowaniu ostatniej subskrypcji lokalnej lub odebranej subskrypcji proxy.

Agregacja publikacji

Jeśli w menedżerze kolejek istnieje więcej niż jedna subskrypcja tego samego łańcucha tematu, tylko jedna kopia każdej publikacji zgodnej z tym łańcuchem tematu jest wysyłana z innych menedżerów kolejek w topologii publikowania/subskrybowania. Po nadejściu komunikatu menedżer kolejek lokalnych dostarcza kopię komunikatu do każdej zgodnej subskrypcji.

Możliwe jest, że więcej niż jedna subskrypcja proxy jest zgodna z łańcuchem tematu pojedynczej publikacji, gdy subskrypcje proxy zawierają znaki wieloznaczne. Jeśli komunikat jest publikowany w menedżerze kolejek, który jest zgodny z co najmniej dwiema subskrypcjami proxy utworzonymi przez pojedynczy połączony menedżer kolejek, tylko jedna kopia publikacji jest przekazywana do zdalnego menedżera kolejek w celu spełnienia wymagań wielu subskrypcji proxy.

Pojęcia pokrewne

[Wykrywanie pętli w rozproszonej sieci publikowania/subskrypcji](#)

Znaki wieloznaczne w subskrypcjach proxy

Subskrypcje mogą używać znaków wieloznacznych w łańcuchach tematów w celu dopasowania ich do wielu łańcuchów tematów w publikacjach.

Subskrypcja może korzystać z dwóch schematów ze znakami wieloznacznymi: *opartego na tematach* i *znakowego*. Patrz "Schematy znaków wieloznacznych" na stronie 71.

W produkcie IBM MQ wszystkie subskrypcje proxy dla subskrypcji ze znakami wieloznacznymi są przekształcane w znaki wieloznaczne oparte na tematach. Jeśli zostanie znaleziony znak wieloznaczny oparty na znakach, zostanie on zastąpiony znakiem # , z powrotem do najbliższego znaku / . Na przykład wartość /aaa/bbb/c*d jest przekształcana w wartość /aaa/bbb/#. W wyniku konwersji menedżery kolejek zdalnych wysyłają nieco więcej publikacji, niż zostały jawnie zasubskrybowane. Dodatkowe publikacje są odfiltrowywane przez menedżer kolejek lokalnych, gdy dostarcza je do subskrybentów lokalnych.

Sterowanie użyciem znaków wieloznacznych za pomocą właściwości WILDCARD

Użyj właściwości MQSC **Topic WILDCARD** lub równoważnej właściwości PCF **Topic WildcardOperation** , aby sterować dostarczaniem publikacji do aplikacji subskrybenta, które używają łańcuchów tematów ze znakami wieloznacznymi. Właściwość WILDCARD może mieć jedną z dwóch możliwych wartości:

WILDCARD

Zachowanie subskrypcji ze znakami wieloznacznymi w odniesieniu do tego tematu.

PASSTHRU

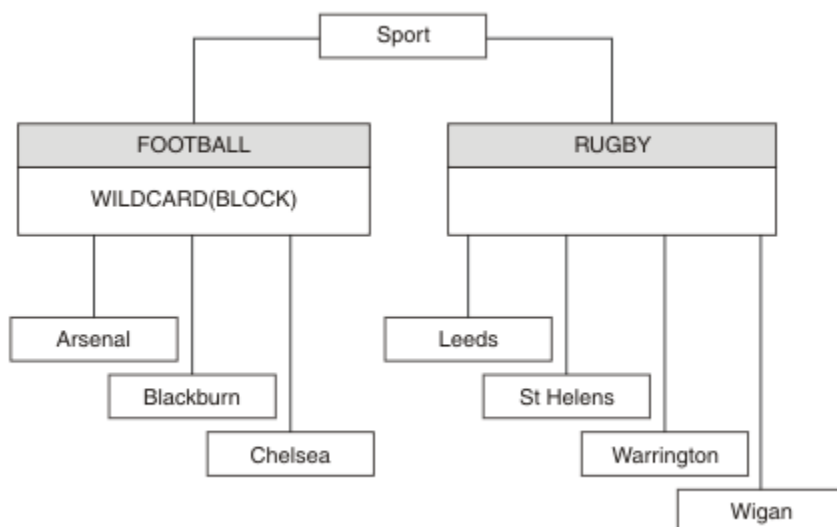
Subskrypcje tematu ze znakami wieloznacznymi, który jest mniej konkretny niż łańcuch tematu w tym obiekcie tematu, otrzymują publikacje zamieszczone w tym temacie i w łańcuchach tematów bardziej konkretnych niż ten temat.

BLOCK

Subskrypcje tematu ze znakami wieloznacznymi, który jest mniej konkretny niż łańcuch tematu w tym obiekcie tematu, nie otrzymują publikacji zamieszczonych w tym temacie i w łańcuchach tematów bardziej konkretnych niż ten temat.

Wartość tego atrybutu jest używana podczas definiowania subskrypcji. Jeśli ten atrybut zostanie zmieniony, modyfikacja nie będzie mieć wpływu na zestaw tematów objętych istniejącymi subskrypcjami. Ten scenariusz dotyczy również sytuacji, w której topologia jest zmieniana podczas tworzenia lub usuwania obiektów tematu. Zestaw tematów zgodnych z subskrypcjami utworzonymi po zmodyfikowaniu atrybutu WILDCARD jest tworzony przy użyciu zmodyfikowanej topologii. Aby wymusić ponowną ocenę zgodnego zestawu tematów pod kątem istniejących subskrypcji, należy zrestartować menedżer kolejek.

W przykładzie "Przykład: tworzenie klastra publikowania/subskrypcji produktu Sport" na stronie 84 można wykonać kroki, aby utworzyć strukturę drzewa tematów, które przedstawia [Rysunek 23](#) na stronie 81.



Rysunek 36. Drzewo tematów, w którym używana jest właściwość WILDCARD , BLOCK

Subskrybent używający łańcucha tematu ze znakami wieloznacznymi # odbiera wszystkie publikacje do tematu Sport i poddrzewa Sport/Rugby . Subskrybent nie otrzymuje żadnych publikacji w poddrzewie Sport/Football , ponieważ właściwość WILDCARD tematu Sport/Football ma wartość BLOCK.

PASSTHRU jest ustawieniem domyślnym. Wartość właściwości WILDCARD można ustawić PASSTHRU na węzły w drzewie Sport . Jeśli węzły nie mają wartości właściwości WILDCARD BLOCK, ustawienie PASSTHRU nie zmienia zachowania obserwowanego przez subskrybentów węzłów w drzewie Sports .

W tym przykładzie należy utworzyć subskrypcje, aby sprawdzić, w jaki sposób ustawienie znaku wieloznacznego wpływa na dostarczane publikacje (patrz sekcja Rysunek 27 na stronie 86). Uruchom komendę publikowania w pliku Rysunek 30 na stronie 87 , aby utworzyć publikacje.

pub QMA

Rysunek 37. Publikuj w QMA

Wyniki są wyświetlane w sekcji Tabela 3 na stronie 82. Należy zauważyć, że ustawienie wartości właściwości WILDCARD BLOCK zapobiega odbieraniu publikacji do tematów w zasięgu znaku wieloznacznego przez subskrypcje ze znakami wieloznacznymi.

Tabela 6. Publikacje otrzymane w serwisie QMA			
Subskrypcja	Łańcuch tematu	Odebrane publikacje	Uwagi
SPORTS	Sports/#	Sports Sports/Rugby Sports/Rugby/Leeds	Wszystkie publikacje w poddrzewie piłkarskim zablokowane przez WILDCARD (BLOCK) na Sports/Football
SARSENAL	Sports/#/Arsenal	-	WILDCARD (BLOCK) on Sports/Football zapobiega subskrypcji znaków wieloznacznym w systemie Arsenal
SLEEDS	Sports/#/Leeds	Sports/Rugby/Leeds	Użycie wartości domyślnej WILDCARD w sporcie/rugby nie zapobiega subskrypcji znaku wieloznacznego w produkcie Leeds.

Uwaga:

Założmy, że subskrypcja zawiera znak wieloznaczny, który jest zgodny z obiektem tematu o wartości właściwości WILDCARD BLOCK. Jeśli subskrypcja zawiera również łańcuch tematu po prawej stronie zgodnego znaku wieloznacznego, subskrypcja nigdy nie otrzyma publikacji. Zestaw publikacji, które nie są zablokowane, to publikacje do tematów, które są elementami nadrzędnymi zablokowanego znaku wieloznacznego. Publikacje do tematów, które są elementami potomnymi tematu z wartością właściwości BLOCK, są blokowane przez znak wieloznaczny. Dlatego łańcuchy tematów subskrypcji, które zawierają temat po prawej stronie znaku wieloznacznego, nigdy nie otrzymują żadnych zgodnych publikacji.

Ustawienie właściwości WILDCARD na wartość BLOCK nie oznacza, że nie można subskrybować przy użyciu łańcucha tematu zawierającego znaki wieloznaczne. Taka subskrypcja jest normalna. Subskrypcja ma jawny temat, który jest zgodny z tematem z obiektem tematu o wartości właściwości WILDCARD BLOCK. W przypadku tematów będących elementami nadrzędnymi lub podrzędnymi tematu z wartością właściwości WILDCARD BLOCK używane są znaki wieloznaczne. W przykładzie przedstawionym w sekcji Rysunek 23 na stronie 81 subskrypcja, taka jak `Sports/Football/#`, może odbierać publikacje.

Znaki wieloznaczne i tematy klastra

Definicje tematów klastra są propagowane do każdego menedżera kolejek w klastrze. Subskrypcja tematu klastra w jednym menedżerze kolejek w klastrze powoduje, że menedżer kolejek tworzy subskrypcje proxy. Subskrypcja proxy jest tworzona w każdym innym menedżerze kolejek w klastrze. Subskrypcje używające łańcuchów tematów zawierających znaki wieloznaczne w połączeniu z tematami klastra mogą utrudniać przewidywanie zachowania. Zachowanie zostało wyjaśnione w poniższym przykładzie.

W klastrze, w którym skonfigurowano przykład “Przykład: tworzenie klastra publikowania/subskrypcji produktu Sport” na stronie 84, QMB ma taki sam zestaw subskrypcji jak QMA, ale QMB nie otrzymał żadnych publikacji po opublikowaniu publikatora w QMA, patrz sekcja Rysunek 24 na stronie 81. Mimo że tematy `Sports/Football` i `Sports/Rugby` są tematami klastra, subskrypcje zdefiniowane w pliku `fullsubs.tst` nie odwołują się do tematu klastra. Żadne subskrypcje proxy nie są propagowane z produktu QMB do produktu QMA. Bez subskrypcji proxy żadne publikacje do produktu QMA nie są przekazywane do produktu QMB.

Niektóre subskrypcje, takie jak `Sports/#/Leeds`, mogą odwoływać się do tematu klastra, w tym przypadku do tematu `Sports/Rugby`. W rzeczywistości subskrypcja `Sports/#/Leeds` jest tłumaczona na obiekt tematu `SYSTEM.BASE.TOPIC`.

Reguła rozstrzygania obiektu tematu, do którego odwołuje się subskrypcja, na przykład `Sports/#/Leeds`, jest następująca. Obecnij łańcuch tematu do pierwszego znaku wieloznacznego. Skanowanie w lewo przez łańcuch tematu w poszukiwaniu pierwszego tematu, z którym jest powiązany obiekt tematu administracyjnego. Obiekt tematu może określać nazwę klastra lub definiować lokalny obiekt tematu. W przykładzie `Sports/#/Leeds` łańcuch tematu po obcięciu to łańcuch `Sports`, który nie ma obiektu tematu, a więc `Sports/#/Leeds` dziedziczy z `SYSTEM.BASE.TOPIC`, który jest lokalnym obiektem tematu.

Aby zobaczyć, w jaki sposób subskrybowanie tematów w klastrze może zmienić sposób działania propagacji z użyciem znaków wieloznacznych, należy uruchomić skrypt wsadowy `upsubs.bat`. Skrypt kasuje kolejki subskrypcji i dodaje subskrypcje tematów klastra w pliku `fullsubs.tst`. Ponownie uruchom plik `puba.bat`, aby utworzyć partię publikacji; patrz sekcja Rysunek 24 na stronie 81.

Tabela 4 na stronie 83 przedstawia wynik dodania dwóch nowych subskrypcji do tego samego menedżera kolejek, w którym zostały opublikowane publikacje. Wynik jest zgodny z oczekiwaniami, nowe subskrypcje otrzymują po jednej publikacji, a liczba publikacji odebranych przez inne subskrypcje pozostaje niezmienną. Nieoczekiwane wyniki wystąpiły w innym menedżerze kolejek klastra (patrz sekcja Tabela 5 na stronie 84).

<i>Tabela 7. Publikacje otrzymane w serwisie QMA</i>			
Subskrypcja	Łańcuch tematu	Odebrane publikacje	Uwagi
SPORTS	Sports/#	Sports Sports/Rugby Sports/Rugby/Leeds	Wszystkie publikacje w poddrzewie piłkarskim zablokowane przez WILDCARD (BLOCK) na Sports/ Football
SARSENAL	Sports/#/Arsenal	-	WILDCARD (BLOCK) on Sports/ Football zapobiega subskrypcji znaków wieloznacznych w systemie Arsenal
SLEEDS	Sports/#/Leeds	Sports/Rugby/Leeds	Użycie wartości domyślnej WILDCARD w sporcie/rugby nie zapobiega subskrypcji znaku wieloznacznego w produkcie Leeds.
FARSENAL	Sports/Football/ Arsenal	Sports/Football/ Arsenal	Arsenal otrzymuje publikację, ponieważ subskrypcja nie zawiera znaku wieloznacznego.
FLEEDS	Sports/Rugby/Leeds	Sports/Rugby/Leeds	Leeds otrzyma publikację w dowolnym przypadku.

Tabela 5 na stronie 84 przedstawia wyniki dodania dwóch nowych subskrypcji w systemie QMB i publikowania w systemie QMA. Należy pamiętać, że program QMB nie otrzymał żadnych publikacji bez tych dwóch nowych subskrypcji. Zgodnie z oczekiwaniami dwie nowe subskrypcje otrzymują publikacje, ponieważ Sports/FootBall i Sports/Rugby są tematami klastra. QMB przekazała subskrypcje proxy dla Sports/Football/Arsenal i Sports/Rugby/Leeds do QMA, a następnie wysłała publikacje do QMB.

Nieoczekiwany wynik jest to, że te dwie subskrypcje Sports/# i Sports/#/Leeds, które wcześniej nie otrzymywały żadnych publikacji, teraz otrzymują publikacje. Wynika to z faktu, że publikacje Sports/Football/Arsenal i Sports/Rugby/Leeds przekazane do QMB dla innych subskrypcji są teraz dostępne dla każdego subskrybenta przyłączonego do produktu QMB. W związku z tym subskrypcje tematów lokalnych Sports/# i Sports/#/Leeds otrzymują publikację Sports/Rugby/Leeds. Produkt Sports/#/Arsenal nadal nie otrzymuje publikacji, ponieważ właściwość WILDCARD sportu/piłki nożnej ma wartość BLOCK.

<i>Tabela 8. Publikacje otrzymane w serwisie QMB</i>			
Subskrypcja	Łańcuch tematu	Odebrane publikacje	Uwagi
SPORTS	Sports/#	Sports/Rugby/ Leeds	Wszystkie publikacje w poddrzewie piłki nożnej zablokowane przez WILDCARD (BLOCK) w systemie Sports/Football
SARSENAL	Sports/#/Arsenal	-	WILDCARD (BLOCK) on Sports/ Football zapobiega subskrypcji znaków wieloznacznych w systemie Arsenal
SLEEDS	Sports/#/Leeds	Sports/Rugby/ Leeds	Wartość domyślna WILDCARD w przypadku produktu Sports/Rugby nie zapobiega subskrypcji znaku wieloznacznego w produkcie Leeds.

Tabela 8. Publikacje otrzymane w serwisie QMB (kontynuacja)			
Subskrypcja	Łańcuch tematu	Odebrane publikacje	Uwagi
FARSENAL	Sports/Football/Arsenal	Sports/Football/Arsenal	Arsenal otrzymuje publikację, ponieważ subskrypcja nie zawiera znaku wieloznacznego.
FLEEDS	Sports/Rugby/Leeds	Sports/Rugby/Leeds	Leeds otrzyma publikację w dowolnym przypadku.

W większości aplikacji nie jest wskazane, aby jedna subskrypcja miała wpływ na zachowanie innej subskrypcji. Ważnym zastosowaniem właściwości WILDCARD z wartością BLOCK jest zapewnienie jednolitego zachowania subskrypcji tego samego łańcucha tematu zawierającego znaki wieloznaczne. Niezależnie od tego, czy subskrypcja znajduje się w tym samym menedżerze kolejek co publikator, czy w innym menedżerze kolejek, wyniki subskrypcji są takie same.

Znaki wieloznaczne i strumienie

W przypadku nowej aplikacji zapisanej w interfejsie API publikowania/subskrypcji subskrypcja produktu * nie odbiera żadnych publikacji. Aby otrzymywać wszystkie publikacje dotyczące sportu, należy zasubskrybować publikacje Sports/*lub Sports/#oraz publikacje dotyczące produktu Business w podobny sposób.

Zachowanie istniejącej umieszczonej w kolejce aplikacji publikowania/subskrypcji nie zmienia się, gdy broker publikowania/subskrypcji jest migrowany do nowszej wersji produktu IBM MQ. Właściwość **StreamName** w komendach **Publish**, **Register Publisher** lub **Subscriber** jest odwzorowywana na nazwę tematu, do którego strumień został zmigrowany.

Znaki wieloznaczne i punkty subskrypcji

W przypadku nowej aplikacji napisanej w interfejsie API publikowania/subskrypcji efektem migracji jest brak publikacji dla subskrypcji produktu *. Aby otrzymywać wszystkie publikacje dotyczące sportu, należy zasubskrybować publikacje Sports/*lub Sports/#oraz publikacje dotyczące produktu Business w podobny sposób.

Zachowanie istniejącej umieszczonej w kolejce aplikacji publikowania/subskrypcji nie zmienia się, gdy broker publikowania/subskrypcji jest migrowany do nowszej wersji produktu IBM MQ. Właściwość **SubPoint** w komendach **Publish**, **Register Publisher** lub **Subscriber** jest odwzorowywana na nazwę tematu, do którego została zmigrowana subskrypcja.

Przykład: tworzenie klastra publikowania/subskrypcji produktu Sport

Poniższe kroki umożliwiają utworzenie klastra CL1z czterema menedżerami kolejek: dwoma pełnymi repozytoriami, CL1A i CL1B oraz dwoma częściowymi repozytoriami QMA i QMB. Pełne repozytoria są używane do przechowywania tylko definicji klastrów. QMA jest nazwą hosta tematu klastra. Trwałe subskrypcje są definiowane zarówno w systemie QMA, jak i w systemie QMB.

Uwaga: Przykład jest zakodowany dla Windows. Należy ponownie zakodować komendę [Create qmgrs.bat](#) i utworzyć plik [pub.bat](#), aby skonfigurować i przetestować przykład na innych platformach.

1. Utwórz pliki skryptowe.
 - a. [Utwórz plik topics.tst](#)
 - b. [Utwórz wilddsubs.tst](#)
 - c. [Utwórz fullsubs.tst](#)
 - d. [Utwórz plik qmgrs.bat](#)
 - e. [tworzenie pub.bat](#)

2. Uruchom komendę `Create qmgrs.bat` , aby utworzyć konfigurację.

```
qmgrs
```

Utwórz tematy w sekcji [Rysunek 23 na stronie 81](#). Skrypt na rysunku 5 tworzy tematy klastra `Sports/Football` i `Sports/Rugby`.

Uwaga: Opcja `REPLACE` nie zastępuje właściwości `TOPICSTR` tematu. Właściwość `TOPICSTR` jest właściwością, która jest używana w przykładzie do testowania różnych drzew tematów. Aby zmienić tematy, najpierw usuń temat.

```
DELETE TOPIC ('Sports')
DELETE TOPIC ('Football')
DELETE TOPIC ('Arsenal')
DELETE TOPIC ('Blackburn')
DELETE TOPIC ('Chelsea')
DELETE TOPIC ('Rugby')
DELETE TOPIC ('Leeds')
DELETE TOPIC ('Wigan')
DELETE TOPIC ('Warrington')
DELETE TOPIC ('St. Helens')

DEFINE TOPIC ('Sports') TOPICSTR('Sports')
DEFINE TOPIC ('Football') TOPICSTR('Sports/Football') CLUSTER(CL1) WILDCARD(BLOCK)
DEFINE TOPIC ('Arsenal') TOPICSTR('Sports/Football/Arsenal')
DEFINE TOPIC ('Blackburn') TOPICSTR('Sports/Football/Blackburn')
DEFINE TOPIC ('Chelsea') TOPICSTR('Sports/Football/Chelsea')
DEFINE TOPIC ('Rugby') TOPICSTR('Sports/Rugby') CLUSTER(CL1)
DEFINE TOPIC ('Leeds') TOPICSTR('Sports/Rugby/Leeds')
DEFINE TOPIC ('Wigan') TOPICSTR('Sports/Rugby/Wigan')
DEFINE TOPIC ('Warrington') TOPICSTR('Sports/Rugby/Warrington')
DEFINE TOPIC ('St. Helens') TOPICSTR('Sports/Rugby/St. Helens')
```

Rysunek 38. Usuwanie i tworzenie tematów: topics.tst

Uwaga: Usuń tematy, ponieważ `REPLACE` nie zastępuje łańcuchów tematów.

Utwórz subskrypcje ze znakami wieloznacznymi. Znaki wieloznaczne odpowiadające tematom z obiektami tematów w produkcie [Rysunek 23 na stronie 81](#). Utwórz kolejki dla każdej subskrypcji. Po uruchomieniu lub ponownym uruchomieniu skryptu kolejki są czyszczone, a subskrypcje usuwane.

Uwaga: Opcja `REPLACE` nie zastępuje właściwości `TOPICOBJ` ani `TOPICSTR` subskrypcji. `TOPICOBJ` lub `TOPICSTR` to właściwości, które są używane w tym przykładzie do testowania różnych subskrypcji. Aby je zmienić, najpierw usuń subskrypcję.

```
DEFINE QLOCAL(QSPORTS) REPLACE
DEFINE QLOCAL(QSARSENAL) REPLACE
DEFINE QLOCAL(QSLEEDS) REPLACE
CLEAR QLOCAL(QSPORTS)
CLEAR QLOCAL(QSARSENAL)
CLEAR QLOCAL(QSLEEDS)

DELETE SUB (SPORTS)
DELETE SUB (SARSENAL)
DELETE SUB (SLEEDS)
DEFINE SUB (SPORTS) TOPICSTR('Sports/#') DEST(QSPORTS)
DEFINE SUB (SARSENAL) TOPICSTR('Sports+/Arsenal') DEST(QSARSENAL)
DEFINE SUB (SLEEDS) TOPICSTR('Sports+/Leeds') DEST(QSLEEDS)
```

Rysunek 39. Utwórz subskrypcje ze znakami wieloznacznymi: wildsubs.tst

Utwórz subskrypcje, które odwołują się do obiektów tematu klastra.

Uwaga:

Separator `/` jest automatycznie wstawiany między łańcuchem tematu, do którego odwołuje się obiekt `TOPICOBJ`, a łańcuchem tematu zdefiniowanym w parametrze `TOPICSTR`.

Definicja DEFINE SUB(FARSENAL) TOPICSTR('Sports/Football/Arsenal') DEST(QFARSENAL) tworzy tę samą subskrypcję. TOPICOBJ jest używana jako szybki sposób odwoływania się do już zdefiniowanego łańcucha tematu. Po utworzeniu subskrypcja nie odwołuje się już do obiektu tematu.

```
DEFINE QLOCAL(QFARSENAL) REPLACE
DEFINE QLOCAL(QRLEEDS) REPLACE
CLEAR QLOCAL(QFARSENAL)
CLEAR QLOCAL(QRLEEDS)

DELETE SUB (FARSENAL)
DELETE SUB (RLEEDS)
DEFINE SUB (FARSENAL) TOPICOBJ('Football') TOPICSTR('Arsenal') DEST(QFARSENAL)
DEFINE SUB (RLEEDS) TOPICOBJ('Rugby') TOPICSTR('Leeds') DEST(QRLEEDS)
```

Rysunek 40. Usuwanie i tworzenie subskrypcji: fullsubs.tst

Utwórz klaster z dwoma repozytoriami. Utwórz dwa częściowe repozytoria na potrzeby publikowania i subskrybowania. Uruchom ponownie skrypt, aby usunąć wszystko i rozpocząć od nowa. Skrypt tworzy również hierarchię tematów i początkowe subskrypcje ze znakami wieloznacznymi.

Uwaga:

Na innych platformach napisz podobny skrypt lub wpisz wszystkie komendy. Użycie skryptu umożliwia szybkie usunięcie wszystkich elementów i rozpoczęcie od nowa z identyczną konfiguracją.

```
@echo off
set port.CL1B=1421
set port.CL1A=1420
for %%A in (CL1A CL1B QMA QMB) do call :createQM %%A
call :configureQM CL1A CL1B %port.CL1B% full
call :configureQM CL1B CL1A %port.CL1A% full
for %%A in (QMA QMB) do call :configureQM %%A CL1A %port.CL1A% partial
for %%A in (topics.tst wildsubs.tst) do runmqsc QMA < %%A
for %%A in (wildsubs.tst) do runmqsc QMB < %%A
goto:eof

:createQM
echo Configure Queue manager %1
endmqm -p %1
for %%B in (dlt crt str) do %%Bmqm %1
goto:eof

:configureQM
if %1==CL1A set p=1420
if %1==CL1B set p=1421
if %1==QMA set p=1422
if %1==QMB set p=1423
echo configure %1 on port %p% connected to repository %2 on port %3 as %4 repository
echo DEFINE LISTENER(LST%1) TRPTYPE(TCP) PORT(%p%) CONTROL(QMGR) REPLACE | runmqsc %1
echo START LISTENER(LST%1) | runmqsc %1
if full==%4 echo ALTER QMGR REPOS(CL1) DEADQ(SYSTEM.DEAD.LETTER.QUEUE) | runmqsc %1
echo DEFINE CHANNEL(TO.%2) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME('LOCALHOST(%3)') CLUSTER(CL1)
REPLACE | runmqsc %1
echo DEFINE CHANNEL(TO.%1) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME('LOCALHOST(%p%)')
CLUSTER(CL1) REPLACE | runmqsc %1
goto:eof
```

Rysunek 41. Utwórz menedżery kolejek: qmgrs.bat

Zaktualizuj konfigurację, dodając subskrypcje do tematów klastra.

```
@echo off
for %%A in (QMA QMB) do runmqsc %%A < wildsubs.tst
for %%A in (QMA QMB) do runmqsc %%A < upsubs.tst
```

Rysunek 42. Aktualizuj subskrypcje: upsubs.bat

Uruchom program pub.bat z menedżerem kolejek jako parametrem, aby opublikować komunikaty zawierające łańcuch tematu publikacji. Pub.bat korzysta z programu przykładowego **amqspub**.


```
@echo off
@rem Provide queue manager name as a parameter
set S=Sports
set S=6 Sports/Football Sports/Football/Arsenal
set S=6 Sports/Rugby Sports/Rugby/Leeds
for %%B in (6) do echo %%B | amqspub %%B %1
```

Rysunek 43. Publikowanie: pub.bat

Pojęcia pokrewne

Subskrypcje ze znakami wieloznacznymi i zachowane publikacje

Zasięg publikacji

Podczas konfigurowania klastra lub hierarchii publikowania/subskrypcji zasięg publikacji dodatkowo określa, czy menedżery kolejek przekazują publikację do zdalnych menedżerów kolejek. Atrybut tematu **PUBSCOPE** służy do administrowania zasięgiem publikacji.

Jeśli publikacja nie jest przekazywana do menedżerów kolejek zdalnych, publikacja jest odbierana tylko przez subskrybentów lokalnych.

Jeśli używany jest klaster publikowania/subskrypcji, zasięg publikacji jest przede wszystkim kontrolowany przez definicję klastrowych obiektów tematu w określonych punktach drzewa tematów. Zasięg publikowania musi być ustawiony, aby umożliwić przepływ publikacji do innych menedżerów kolejek w klastrze. Zasięg publikowania dla tematu klastrowego należy ograniczyć tylko wtedy, gdy konieczne jest precyzyjne sterowanie konkretnymi tematami w określonych menedżerach kolejek.

Jeśli używana jest hierarchia publikowania/subskrypcji, zasięg publikacji jest przede wszystkim kontrolowany przez ten atrybut w połączeniu z atrybutem zasięgu subskrypcji.

Atrybut **PUBSCOPE** służy do określania zasięgu publikacji dotyczących konkretnego tematu. Atrybut można ustawić na jedną z następujących wartości:

QMGR

Publikacja jest dostarczana tylko do subskrybentów lokalnych. Te publikacje są nazywane *publikacjami lokalnymi*. Publikacje lokalne nie są przekazywane do zdalnych menedżerów kolejek i dlatego nie są odbierane przez subskrybentów połączonych ze zdalnymi menedżerami kolejek.

ALL

Publikacja jest dostarczana do lokalnych subskrybentów i subskrybentów połączonych ze zdalnymi menedżerami kolejek w klastrze lub hierarchii publikowania/subskrypcji. Publikacje te są nazywane *publikacjami globalnymi*.

NIEISKRZĄCY

Użyj ustawienia **PUBSCOPE** tematu nadrzędnego w drzewie tematów.

Publikatory mogą również określać, czy publikacja jest lokalna, czy globalna, za pomocą opcji umieszczania komunikatu MQPMO_SCOPE_QMGR. Jeśli ta opcja jest używana, nadpisuje ona każde zachowanie, które zostało ustawione za pomocą atrybutu tematu **PUBSCOPE**.

Pojęcia pokrewne

“Obiekty tematów administracyjnych” na stronie 78

Za pomocą obiektu tematu administracyjnego można przypisać konkretne, inne niż domyślne atrybuty do tematów.

Zadania pokrewne

Konfigurowanie rozproszonych sieci publikowania/subskrypcji

Zasięg subskrypcji

Zasięg subskrypcji określa, czy subskrypcja w jednym menedżerze kolejek odbiera publikacje publikowane w innym menedżerze kolejek w klastrze lub hierarchii publikowania/subskrypcji, czy tylko publikacje z publikatorów lokalnych.

Ograniczenie zasięgu subskrypcji do menedżera kolejek powoduje, że subskrypcje proxy nie są przekazywane do innych menedżerów kolejek w topologii publikowania/subskrypcji. Zmniejsza to ruch przesyłania komunikatów w trybie publikowania/subskrypcji między menedżerami kolejek.

Jeśli używany jest klaster publikowania/subskrypcji, zasięg subskrypcji jest przede wszystkim kontrolowany przez definicję klastrowych obiektów tematu w określonych punktach drzewa tematów. Zasięg subskrypcji musi być ustawiony, aby umożliwić przepływ subskrypcji proxy do innych menedżerów kolejek w klastrze. Zasięg subskrypcji dla tematu klastrowego należy ograniczyć tylko wtedy, gdy konieczne jest precyzyjne sterowanie konkretnymi tematami w określonych menedżerach kolejek.

Jeśli używana jest hierarchia publikowania/subskrypcji, zasięg subskrypcji jest przede wszystkim kontrolowany przez ten atrybut w połączeniu z atrybutem zasięgu publikacji.

Atrybut tematu **SUBSCOPE** służy do określania zasięgu subskrypcji konkretnego tematu. Atrybut można ustawić na jedną z następujących wartości:

QMGR

Subskrypcja odbiera tylko publikacje lokalne, a subskrypcje proxy nie są propagowane do zdalnych menedżerów kolejek.

ALL

Subskrypcja proxy jest propagowana do zdalnych menedżerów kolejek w klastrze lub hierarchii publikowania/subskrypcji, a subskrybent odbiera publikacje lokalne i zdalne.

NIEISKRZĄCY

Użyj ustawienia **SUBSCOPE** tematu nadrzędnego w drzewie tematów.

Jeśli zasięg subskrypcji dla tematu jest ustawiony na ALL (bezpośrednio lub za pomocą opcji ASPARENT), poszczególne subskrypcje tego tematu mogą ograniczyć swój zasięg do QMGR, określając parametr MQSO_SCOPE_QMGR podczas tworzenia subskrypcji. Subskrypcja tematu, który ma zasięg QMGR, nie może rozszerzać zasięgu do wartości ALL.

Pojęcia pokrewne

“Obiekty tematów administracyjnych” na stronie 78

Za pomocą obiektu tematu administracyjnego można przypisać konkretne, inne niż domyślne atrybuty do tematów.

Zadania pokrewne

Konfigurowanie rozproszonych sieci publikowania/subskrypcji

Obszary tematów

Obszar tematu to zestaw tematów, które można subskrybować i publikować. Menedżer kolejek w topologii rozproszonego publikowania/subskrypcji ma obszar tematu, który potencjalnie obejmuje tematy, które zostały zasubskrybowane i opublikowane w połączonych menedżerach kolejek w tej topologii.

Uwaga: Przegląd tematów w menedżerze kolejek, takich jak obiekty tematów administracyjnych, łańcuchy tematów i drzewa tematów, zawiera sekcja “Tematy” na stronie 69. Dodatkowe odwołania do *tematów* w bieżącej artykule dotyczą *łańcuchów tematów*, chyba że określono inaczej.

Tematy są początkowo tworzone w jeden z następujących sposobów:

- Administracyjnie, podczas definiowania obiektu tematu lub trwałej subskrypcji.
- dynamicznie, gdy aplikacja tworzy dynamicznie publikację lub subskrypcję nowego tematu.

Tematy są propagowane do innych menedżerów kolejek zarówno przez subskrypcje proxy, jak i przez tworzenie obiektów tematów klastra administracyjnego. Subskrypcje proxy powodują przekazywanie publikacji z menedżera kolejek, z którym połączony jest publikator, do menedżerów kolejek subskrybentów.

Subskrypcje proxy są propagowane między wszystkimi menedżerami kolejek, które są połączone relacjami nadrzędny-podrzędny w hierarchii menedżerów kolejek. W wyniku tego można zasubskrybować w jednym menedżerze kolejek temat zdefiniowany w dowolnym innym menedżerze kolejek w hierarchii.

Dopóki istnieje połączona ścieżka między menedżerami kolejek, nie ma znaczenia, w jaki sposób menedżery kolejek są połączone.

Subskrypcje proxy są również propagowane w przypadku subskrypcji tematów klastra w klastrze publikowania/subskrypcji. Temat klastra jest tematem przyłączonym do obiektu tematu, który ma atrybut **CLUSTER** lub dziedziczny atrybut z obiektu nadrzędnego. Tematy, które nie są tematami klastra, są nazywane tematami lokalnymi i nie są replikowane do klastra. Żadne subskrypcje proxy nie są propagowane do klastra z subskrypcji do tematów lokalnych.

Podsumowując, subskrypcje proxy są tworzone dla subskrybentów w dwóch okolicznościach.

1. Menedżer kolejek jest elementem hierarchii, a subskrypcja proxy jest przekazywana do elementu nadrzędnego i elementów potomnych menedżera kolejek.
2. Menedżer kolejek jest elementem klastra, a łańcuch tematu subskrypcji jest tłumaczony na temat powiązany z obiektem tematu klastra. Jeśli temat jest tematem klastra *kierowanym bezpośrednio*, subskrypcje proxy są przekazywane do wszystkich elementów klastra. Jeśli temat jest tematem klastra *kierowanym przez hosta tematu*, subskrypcje proxy są przekazywane tylko do menedżerów kolejek w klastrze, które zdefiniowały obiekt tematu klastra. Więcej informacji na ten temat zawiera sekcja [“Klustry publikowania/subskrypcji”](#) na stronie 94.

Jeśli menedżer kolejek jest elementem klastra i hierarchii, subskrypcje proxy są propagowane przez oba mechanizmy bez dostarczania zduplikowanych publikacji do subskrybenta.

Obszary tematów w trzech topologiach publikowania/subskrypcji są opisane na poniższej liście:

- [“Przypadek 1. Klustry publikowania/subskrypcji”](#) na stronie 107.
- [“Przypadek 2. Hierarchie publikowania/subskrypcji”](#) na stronie 108.

W osobnych tematach poniższe czynności konfiguracyjne opisują sposób łączenia obszarów tematów.

- [Tworzenie pojedynczego obszaru tematu w klastrze publikowania/subskrybowania.](#)
- [Łączenie obszarów tematów wielu klastrów.](#)
- [Łączenie i izolowanie obszarów tematów w wielu klastrach.](#)
- [Publikowanie i subskrybowanie obszarów tematów w wielu klastrach.](#)

Przypadek 1. Klustry publikowania/subskrypcji

W przykładzie założono, że menedżer kolejek nie jest połączony z hierarchią publikowania/subskrypcji.

Jeśli menedżer kolejek jest elementem klastra publikowania/subskrybowania, jego obszar tematu składa się z tematów lokalnych i tematów klastra. Tematy lokalne są powiązane z obiektami tematów bez atrybutu **CLUSTER**. Jeśli menedżer kolejek ma definicje lokalnych obiektów tematu, jego obszar tematu różni się od innego menedżera kolejek w klastrze, który ma również własne lokalnie zdefiniowane obiekty tematu.

W klastrze publikowania/subskrybowania nie można subskrybować tematu zdefiniowanego w innym menedżerze kolejek, chyba że subskrybowany temat jest rozstrzygany jako obiekt tematu klastra.

Jeśli te same nazwane definicje obiektu tematu klastra są wymagane w wielu menedżerach kolejek, na przykład podczas używania *kierowania hostami tematów*, ważne jest, aby wszystkie definicje były zgodne w razie potrzeby. Więcej informacji na ten temat zawiera sekcja [Tworzenie pojedynczego obszaru tematu w klastrze publikowania/subskrybowania](#).

Lokalna definicja obiektu tematu, niezależnie od tego, czy dotyczy tematu klastra, czy tematu lokalnego, ma pierwszeństwo przed tym samym obiektem tematu zdefiniowanym w innym miejscu w klastrze. Lokalnie zdefiniowany temat jest używany, nawet jeśli obiekt zdefiniowany w innym miejscu jest nowszy.

Ważne jest, aby obiekt tematu klastra był powiązany z tym samym łańcuchem tematu w każdym miejscu klastra. Nie można modyfikować łańcucha tematu, z którym jest powiązany obiekt tematu. Aby powiązać ten sam obiekt tematu z innym łańcuchem tematu, należy usunąć obiekt tematu i utworzyć go ponownie z nowym łańcuchem tematu. Jeśli temat jest sklastrowany, efektem jest usunięcie kopii obiektu tematu zapisanych w innych elementach klastra, a następnie utworzenie kopii nowego obiektu tematu w każdym miejscu w klastrze. Wszystkie kopie obiektu tematu odnoszą się do tego samego łańcucha tematu.

Możliwe jest przypadkowe utworzenie dwóch definicji tego samego nazwanego obiektu tematu w różnych menedżerach kolejek w klastrze z różnymi łańcuchami tematów. Może to powodować niejasne zachowanie, ponieważ wiele definicji tego samego obiektu tematu z różnymi łańcuchami tematów może generować różne wyniki w zależności od tego, w jaki sposób i gdzie temat jest przywoływany. Więcej informacji na ten temat zawiera sekcja [Wiele definicji tematów klastra o tej samej nazwie](#).

Przypadek 2. Hierarchie publikowania/subskrypcji

W przykładzie założono, że menedżer kolejek nie jest elementem klastra publikowania/subskrybowania.

Jeśli w produkcie IBM MQ menedżer kolejek jest elementem hierarchii publikowania/subskrypcji, jego obszar tematu składa się ze wszystkich tematów zdefiniowanych lokalnie oraz w połączonych menedżerach kolejek. Obszar tematu wszystkich menedżerów kolejek w hierarchii jest taki sam. Nie ma podziału tematów na tematy lokalne i globalne.

Należy ustawić jedną z opcji **PUBSCOPE** i **SUBSCOPE** na wartość QMGR, aby zapobiec publikowaniu w temacie przepływającym od publikatora do subskrybenta połączonych z różnymi menedżerami kolejek w hierarchii.

Założmy, że zdefiniowano obiekt tematu Alabama za pomocą łańcucha tematu USA/Alabama w menedżerze kolejek QMA. Wynik jest następujący:

1. Obszar tematu w QMA zawiera teraz obiekt tematu Alabama i łańcuch tematu USA/Alabama.
2. Aplikacja lub administrator może utworzyć subskrypcję w QMA, korzystając z nazwy obiektu tematu Alabama.
3. Aplikacja może utworzyć subskrypcję dowolnego tematu, w tym tematu USA/Alabama, w dowolnym menedżerze kolejek w hierarchii. Jeśli QMA nie została zdefiniowana lokalnie, temat USA/Alabama jest tłumaczony na obiekt tematu SYSTEM.BASE.TOPIC.

Pojęcia pokrewne

[“Zasięg publikacji” na stronie 105](#)

Podczas konfigurowania klastra lub hierarchii publikowania/subskrypcji zasięg publikacji dodatkowo określa, czy menedżery kolejek przekazują publikację do zdalnych menedżerów kolejek. Atrybut tematu **PUBSCOPE** służy do administrowania zasięgiem publikacji.

[“Zasięg subskrypcji” na stronie 105](#)

Zasięg subskrypcji określa, czy subskrypcja w jednym menedżerze kolejek odbiera publikacje publikowane w innym menedżerze kolejek w klastrze lub hierarchii publikowania/subskrypcji, czy tylko publikacje z publikatorów lokalnych.

Zadania pokrewne

[Konfigurowanie rozproszonych sieci publikowania/subskrypcji](#)

IBM MQ Rozsyłanie grupowe

Funkcja rozsyłania grupowego w produkcie IBM MQ umożliwia niezawodne rozsyłanie grupowe komunikatów z zachowaniem małych opóźnień i wysokiego zwielokrotnienia.

Rozsyłanie grupowe jest wydajną formą przesyłania komunikatów w trybie publikowania/subskrypcji, ponieważ może być skalowane do dużej liczby subskrybentów bez negatywnego wpływu na wydajność. Produkt IBM MQ umożliwia niezawodne przesyłanie komunikatów oparte na rozsyłaniu grupowym przez korzystanie z potwierdzeń, potwierdzeń negatywnych i numerów kolejnych przy zachowaniu małych opóźnień i wysokiego zwielokrotnienia.

Funkcja rozsyłania grupowego z bezstronnym dostarczaniem produktu IBM MQ umożliwia prawie równoczesne dostarczenie przy zapewnieniu, że żaden z odbiorców nie ma przewagi. Ponieważ funkcja rozsyłania grupowego produktu IBM MQ używa sieci do dostarczania komunikatów, mechanizm publikowania/subskrypcji nie jest potrzebny do zwielokrotnienia danych. Po odwzorowaniu tematu na adres grupy nie ma potrzeby stosowania menedżera kolejek, ponieważ publikatory i subskrybenty mogą działać w trybie równorzędnym. Pozwala to na zmniejszenie obciążenia serwerów menedżera kolejek, przez co serwer menedżera kolejek przestaje być potencjalnym punktem awarii.

Początkowe pojęcia dotyczące rozsyłania grupowego

IBM MQ Rozsyłanie grupowe można łatwo zintegrować z istniejącymi systemami i aplikacjami za pomocą obiektu Informacje o komunikacji (COMMINFO). Dwa pola obiektu TOPIC umożliwiają szybką konfigurację istniejących obiektów TOPIC w celu obsługi lub ignorowania rozsyłania grupowego.

Obiekty potrzebne do rozsyłania grupowego

Poniżej przedstawiono krótki przegląd dwóch obiektów potrzebnych do rozsyłania grupowego produktu IBM MQ :

COMMINFO, obiekt

Obiekt COMMINFO zawiera atrybuty powiązane z transmisją rozsyłania grupowego. Więcej informacji na temat parametrów obiektu COMMINFO zawiera sekcja [DEFINE COMMINFO](#).

Jedynym polem COMMINFO, które MUSI być ustawione, jest nazwa obiektu COMMINFO. Ta nazwa jest następnie używana do identyfikowania obiektu COMMINFO w temacie. Pole **GRPADDR** obiektu COMMINFO należy sprawdzić, aby upewnić się, że wartość jest poprawnym adresem grupy rozsyłania.

Obiekt tematu

Temat jest tematem informacji publikowanych w komunikacie publikowania/subskrybowania, a temat jest definiowany przez utworzenie obiektu TOPIC. Więcej informacji na temat parametrów obiektu TOPIC zawiera sekcja [DEFINE TOPIC](#).

Istniejących tematów można używać z rozsyłaniem grupowym, zmieniając wartości następujących parametrów obiektu TOPIC: **COMMINFO** i **MCAST**.

- **COMMINFO** Ten parametr określa nazwę obiektu informacji o komunikacji rozsyłania grupowego.
- **MCAST** Ten parametr określa, czy rozsyłanie grupowe jest dozwolone w danej pozycji drzewa tematów. Domyślnie parametr **MCAST** ma wartość **ASPARENT**, co oznacza, że atrybut rozsyłania tematu jest dziedziczony z elementu nadrzędnego. Ustawienie parametru **MCAST** na wartość **ENABLED** umożliwia rozsyłanie grupowe w tym węźle.

Sieci i tematy rozsyłania grupowego

Poniższe informacje zawierają przegląd informacji o tym, co dzieje się z subskrypcjami o różnych typach subskrypcji i definicjach tematów. W tych przykładach założono, że parametr **COMMINFO** obiektu TOPIC jest ustawiony na nazwę poprawnego obiektu COMMINFO:

Dla tematu włączono rozsyłanie grupowe

Jeśli parametr łańcucha tematu **MCAST** jest ustawiony na wartość **ENABLED**, subskrypcje klientów obsługujących rozsyłanie grupowe są dozwolone i jest wykonywane subskrypcja grupowa, chyba że:

- Jest to trwała subskrypcja od klienta obsługującego rozsyłanie grupowe.
- Jest to niezarządzana subskrypcja od klienta obsługującego rozsyłanie grupowe.
- Jest to subskrypcja od klienta, który nie obsługuje rozsyłania grupowego.

W takich przypadkach subskrypcja nie jest rozsyłana grupowo, a subskrypcje są przywracane do normalnej wersji publikowania/subskrypcji.

Temat ustawiony na rozsyłanie grupowe jest wyłączony

Jeśli parametr łańcucha tematu **MCAST** jest ustawiony na wartość **DISABLED**, subskrypcja bez rozsyłania grupowego jest zawsze dokonywana i subskrypcje są przywracane do normalnej wersji publikowania/subskrypcji.

Temat ustawiony tylko na rozsyłanie grupowe

Jeśli parametr łańcucha tematu **MCAST** jest ustawiony na wartość **ONLY**, subskrypcje klientów obsługujących rozsyłanie grupowe są dozwolone i są wykonywane subskrypcje rozsyłania grupowego, chyba że:

- Jest to subskrypcja trwała: trwałe subskrypcje zostały odrzucone z kodem przyczyny [2436 \(0984\) \(RC2436\)](#): `MQRC_DURABILITY_NOT_ALLOWED`

- Jest to subskrypcja niezarządzana: subskrypcje niezarządzane są odrzucane z kodem przyczyny 2046 (07FE) (RC2046): MQRC_OPTIONS_ERROR
- Jest to subskrypcja klienta, który nie obsługuje rozsyłania grupowego: te subskrypcje są odrzucane z kodem przyczyny 2560 (0A00) (RC2560): MQRC_MULTICAST_ONLY
- Jest to subskrypcja z aplikacji powiązanej lokalnie: Te subskrypcje są odrzucane z kodem przyczyny 2560 (0A00) (RC2560): MQRC_MULTICAST_ONLY

Windows

Linux

AIX

MQ TelemetryPrzegląd

Produkt MQ Telemetry zawiera usługę telemetryczną (MQXR), która jest częścią menedżera kolejek, klienty telemetryczne, które można napisać samodzielnie lub pobrać bezpłatnie, oraz interfejsy administracyjne wiersza komend i eksploratora. Telemetria odnosi się do gromadzenia danych z wielu urządzeń zdalnych i administrowania nimi. Za pomocą produktu MQ Telemetry można zintegrować gromadzenie danych i sterowanie urządzeniami z aplikacjami WWW.

Program MQ Telemetry jest komponentem aplikacji IBM MQ. Aktualizacja dla tych wersji polega zasadniczo na zainstalowaniu nowszej wersji produktu IBM MQ.

Przykładowe aplikacje są nadal dostępne bezpłatnie w serwisie Eclipse Paho i MQTT.org. Patrz [programy przykładoweIBM MQ Telemetry Transport](#).

Ponieważ produkt MQ Telemetry jest komponentem produktu IBM MQ, produkt MQ Telemetry można zainstalować razem z produktem głównym lub po zainstalowaniu produktu głównego. Informacje na temat migracji zawiera sekcja [Migrowanie MQ Telemetry w systemie Windows](#) oraz sekcja [Migrowanie MQ Telemetry w systemie Linux](#).

Produkt MQ Telemetry zawiera następujące komponenty:

Kanały pomiarowe

Kanały pomiarowe umożliwiają zarządzanie połączeniami klientów MQTT z produktem IBM MQ. Kanały pomiarowe używają nowych obiektów produktu IBM MQ, takich jak SYSTEM.MQTT.TRANSMIT.QUEUE, do interakcji z produktem IBM MQ.

Usługa telemetryczna (MQXR)

Klienty produktu MQTT używają usługi telemetrycznej produktu SYSTEM.MQXR.SERVICE do nawiązywania połączeń z kanałami telemetrycznymi.

IBM MQ Explorer obsługa MQ Telemetry

MQ Telemetry można administrować za pomocą programu IBM MQ Explorer.

Dokumentacja

Dokumentacja produktu MQ Telemetry jest zawarta w standardowej dokumentacji produktu IBM MQ. Dokumentacja pakietu SDK dla klientów w systemach Java i C jest dostępna w dokumentacji produktu oraz w formacie Javadoc i HTML.

Pojęcia związane z telemetrią

W celu podjęcia decyzji o tym, co należy zrobić, należy zebrać informacje ze środowiska. Jako konsument sprawdzasz, co masz w sklepie, przed podjęciem decyzji o tym, co kupić. Chcesz wiedzieć, jak długo podróż będzie trwać, jeśli wyjedziesz teraz, przed rezerwacją połączenia. Przed podjęciem decyzji o wizycie u lekarza należy sprawdzić objawy. Przed podjęciem decyzji o tym, czy czekać, należy sprawdzić, kiedy zostanie nadjeżdżona magistrala. Informacje dotyczące tych decyzji pochodzą bezpośrednio z liczników i urządzeń, z pisemnego słowa na papierze lub z ekranu, a także od użytkownika. Gdziekolwiek jesteś i kiedy będziesz potrzebował, zbieraj informacje, zbieraj je razem, analizuj i działaj na nich.

Jeśli źródła informacji są szeroko rozproszone lub niedostępne, zbieranie najdokładniejszych informacji staje się trudne i kosztowne. Jeśli istnieje wiele zmian, które mają zostać wprowadzone, lub jeśli wprowadzanie zmian jest trudne, zmiany nie są wprowadzane lub są wprowadzane, gdy są mniej skuteczne.

Co się stanie, jeśli koszty gromadzenia i kontrolowania informacji z urządzeń rozproszonych na szeroką skalę zostaną znacznie obniżone dzięki podłączeniu urządzeń z technologią cyfrową do Internetu?

Informacje mogą być analizowane przy użyciu zasobów Internetu i przedsiębiorstwa. Masz więcej możliwości podejmowania świadomych decyzji i podejmowania na nich działań.

Tendencje technologiczne oraz presja na środowisko i gospodarkę są siłą napędową tych zmian:

1. Koszt podłączenia i sterowania czujnikami i napędami zmniejsza się ze względu na standaryzację i podłączenie do tanich procesorów cyfrowych.
2. Internet i technologie internetowe są coraz częściej używane do łączenia urządzeń. W niektórych krajach liczba połączeń z aplikacjami internetowymi telefonów komórkowych przekracza liczbę komputerów osobistych. Inne urządzenia z pewnością podążają za.
3. Internet i technologie internetowe ułatwiają aplikacje pobieranie danych. Łatwy dostęp do danych pozwala wykorzystać analizy danych do przetwarzania danych z czujników w informacje przydatne w wielu innych rozwiązaniach.
4. Inteligentne wykorzystanie zasobów jest często szybszym i tańszym sposobem na zmniejszenie emisji dwutlenku węgla i kosztów. Alternatywy: znalezienie nowych zasobów lub opracowanie nowych technologii w celu wykorzystania istniejących zasobów może być rozwiązaniem długoterminowym. W perspektywie krótkoterminowej rozwój nowych technologii lub znalezienie nowych zasobów jest często bardziej ryzykowne, wolniejsze i bardziej kosztowne niż poprawa istniejących rozwiązań.

Przykład

Przykład pokazuje, w jaki sposób te trendy tworzą nowe możliwości inteligentnej interakcji ze środowiskiem.

Międzynarodowa konwencja o bezpieczeństwie życia na morzu (SOLAS) wymaga, aby system automatycznej identyfikacji (AIS) był wdrożony na wielu statkach. Jest on wymagany na statkach handlowych o toni ponad 300 ton i na statkach pasażerskich. AIS jest przede wszystkim systemem unikania kolizji dla żegluga przybrzeżnej. Jest on wykorzystywany przez władze morskie do monitorowania i kontroli wód przybrzeżnych.

Entuzjaści na całym świecie wdrażają tanie stacje śledzenia AIS i umieszczają informacje o żegludze przybrzeżnej w Internecie. Inni entuzjaści piszą aplikacje, które łączą informacje z AIS z innymi informacjami z Internetu. Wyniki są umieszczane w serwisach WWW i publikowane za pomocą serwisów Twitter i SMS.

W jednej aplikacji informacje ze stacji AIS w pobliżu Southampton są połączone z własnością statku i informacjami geograficznymi. Aplikacja przekazuje na żywo informacje o przylocach i wylotach promów do serwisu Twitter. Zwykli użytkownicy korzystający z promów między Southampton a Isle of Wight subskrybują kanał informacyjny za pomocą Twittera lub SMS-ów. Jeśli kanał pokazuje, że ich prom jest spóźniony, dojeżdżający mogą opóźnić odlot i złapać prom, gdy dokuje później niż planowany czas przylotu.

Więcej przykładów zawiera sekcja [“Przypadki użycia danych telemetrycznych”](#) na stronie 113.

Zadania pokrewne

[Instalowanie produktu MQ Telemetry](#)

[administrowanieMQ Telemetry](#)

[Migrowanie produktu MQ Telemetry w systemie Windows](#)

[Migrowanie produktu MQ Telemetry w systemie Linux](#)

[Tworzenie aplikacji dla składnika MQ Telemetry](#)

[Rozwiązywanie problemów z systemem MQ Telemetry](#)

Odsyłacze pokrewne

[Informacje uzupełniające dotyczące produktu MQ Telemetry](#)

Windows

Linux

AIX

MQ Telemetry - wprowadzenie

Ludzie, firmy i rządy coraz częściej chcą używać produktu MQ Telemetry do bardziej inteligentnego współdziałania ze środowiskiem, w którym żyjemy i w którym pracujemy. MQ Telemetry łączy wszystkie

rodzaje urządzeń z Internetem i przedsiębiorstwem, a także obniża koszty tworzenia aplikacji dla inteligentnych urządzeń.

Co to jest MQ Telemetry?

- Jest to funkcja systemu IBM MQ, która rozszerza uniwersalny rdzeń sieci przesyłania komunikatów udostępniany przez IBM MQ na szeroką gamę zdalnych czujników, siłowników i urządzeń telemetrycznych. MQ Telemetry rozszerza IBM MQ, dzięki czemu może łączyć inteligentne aplikacje korporacyjne, usługi i osoby podejmujące decyzje z sieciami oprzyrządowanych urządzeń.
- Podstawowe części systemu MQ Telemetry to:

Usługa MQ Telemetry (MQXR).

Ta usługa działa na serwerze IBM MQ i używa protokołu IBM MQ Telemetry Transport (MQTT) do komunikacji z urządzeniami pomiarowymi.

Aplikacje MQTT, które są zapisywane.

Te aplikacje sterują informacjami przekazywanymi między urządzeniami telemetrycznymi i menedżerem kolejek produktu IBM MQ oraz wszelkimi działaniami podejmowanymi w odpowiedzi na te informacje. Aby ułatwić tworzenie tych aplikacji, należy użyć bibliotek klienta MQTT.

Co to może dla mnie zrobić?

- MQTT to otwarty transport przesyłania komunikatów, który umożliwia tworzenie implementacji języka MQTT dla wielu różnych urządzeń.
- Klienci MQTT mogą działać na urządzeniach o małej ilości miejsca, które mają ograniczone zasoby.
- Produkt MQTT działa wydajnie w sieciach, w których przepustowość jest niska, gdzie koszt wysyłania danych jest kosztowny lub może być niestabilny.
- Dostarczanie komunikatów jest zapewnione i oddzielone od aplikacji.
- Programiści aplikacji nie muszą posiadać wiedzy na temat programowania komunikacji.
- Komunikaty mogą być wymieniane z innymi aplikacjami do przesyłania komunikatów. Mogą to być inne aplikacje telemetryczne, MQI, JMS lub aplikacje do przesyłania komunikatów w przedsiębiorstwie.

Jak go używać?

- Udostępniono przykładowe skrypty, które działają z przykładową aplikacją kliencką IBM MQ Telemetry Transport v3 (mqttv3app.jar). Patrz [programy przykładowe IBM MQ Telemetry Transport](#).
- Produkt IBM MQ Explorer i powiązane z nim narzędzia umożliwiają administrowanie funkcją telemetryczną produktu IBM MQ.
- Biblioteki klienckie ułatwiają tworzenie aplikacji produktu MQTT, które łączą się z menedżerem kolejek i korzystają z przesyłania komunikatów w trybie publikowania/subskrypcji.
- Dystrybuuj aplikację i bibliotekę kliencką do urządzenia, na którym aplikacja ma być uruchomiona.

Jak to działa?

- MQTT jest protokołem publikowania i subskrybowania. Aplikacja kliencka MQTT może publikować komunikaty na serwerze MQTT lub subskrybować komunikaty wysyłane przez aplikacje łączące się z serwerem MQTT.
- Aplikacje klienckie MQTT używają bibliotek klienckich, które implementują transport komunikatów MQTT.
- Podstawowa aplikacja kliencka systemu MQTT działa jak standardowy klient MQ, ale może działać na znacznie szerszym zakresie platform i sieci.
- Usługa MQ Telemetry (MQXR) przekształca menedżer kolejek IBM MQ w serwer MQTT.
- Jeśli menedżer kolejek produktu IBM MQ działa jako serwer produktu MQTT, inne aplikacje nawiązujące połączenie z menedżerem kolejek mogą subskrybować i odbierać komunikaty z klienta produktu MQTT.

- Menedżer kolejek działa jako router dystrybuujący komunikaty z aplikacji publikujących do aplikacji subskrybujących.
- Komunikaty mogą być rozdzielane między różne typy aplikacji klienckich. Na przykład między klientami produktu Telemetry a klientami produktu JMS .

Uwaga: MQ Telemetry zastępuje węzły SCADA, które zostały wycofane w wersji 7 produktu WebSphere Message Broker (obecnie znane jako IBM Integration Bus) i działa w systemach Windows, Linux i AIX.

Windows

Linux

AIX

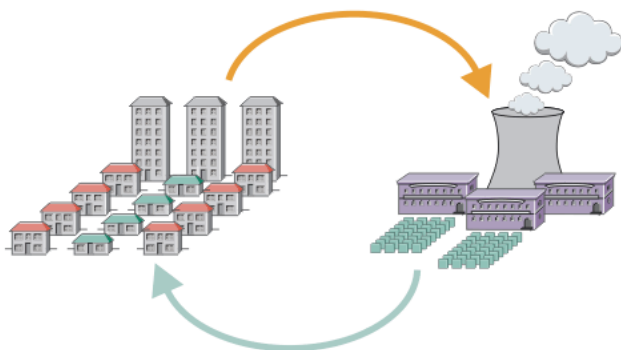
Przypadki użycia danych telemetrycznych

Telemetria jest zautomatyzowanym wykryciem, pomiarem danych i sterowaniem urządzeniami zdalnymi. Nacisk kładzie się na transmisję danych z urządzeń do centralnego punktu kontrolnego. Telemetria obejmuje również wysyłanie informacji konfiguracyjnych i kontrolnych do urządzeń.

Program MQ Telemetry łączy małe urządzenia za pomocą programu MQTT protocol, a urządzenia z innymi aplikacjami za pomocą programu IBM MQ. MQ Telemetry stanowi most między urządzeniami i Internetem, co ułatwia tworzenie "inteligentnych rozwiązań". Inteligentne rozwiązania odblokowują bogactwo informacji dostępnych w Internecie i w aplikacjach korporacyjnych dla aplikacji, które monitorują i kontrolują urządzenia.

Na poniższych diagramach przedstawiono niektóre typowe zastosowania produktu MQ Telemetry:

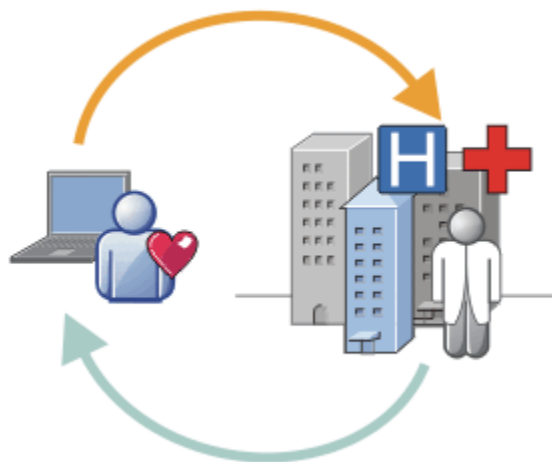
Telemetria: Inteligentna Energia Elektryczna



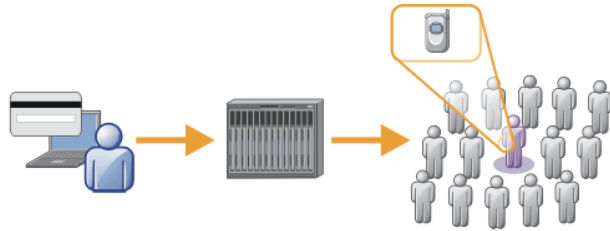
- Komunikat MQTT zawierający dane dotyczące zużycia energii wysłane do dostawcy usług.
- Program MQ Telemetry wysyła komendę CONTROL COMMANDS na podstawie analizy danych dotyczących zużycia energii.
- Więcej informacji na ten temat zawiera następujący przypadek użycia: [“Telemetria przypadek użycia: Home Energy Monitoring and Control”](#) na stronie 116

Telemetria: Smart Health Services

- Program MQ Telemetry wysyła dane dotyczące zdrowia do szpitala i lekarza.
- Alerty lub informacje zwrotne komunikatów MQTT mogą być wysyłane na podstawie analizy Danych dotyczących Zdrowia.
- Więcej informacji na ten temat zawiera następujący przypadek użycia: [“Telemetria przypadek użycia: Home patient monitoring”](#) na stronie 114



Telemetria: Jeden w tłumie



- Prosta transakcja kartana jest wysyłana do serwera banku.
- MQ Telemetry identyfikuje jedną osobę z tysięcy, ostrzegając klienta, że jej karta została użyta.
- Program MQ Telemetry może użyć najprostszych danych wejściowych i zlokalizować daną osobę.

Przypadki użycia opisane w podtematach pochodzą z rzeczywistych przykładów. Ilustrują one niektóre sposoby korzystania z telemetrii i niektóre z powszechnych problemów, które technologia telemetryczna musi rozwiązać.

Windows

Linux

AIX

Telemetria przypadek użycia: Home patient monitoring

We współpracy między firmą IBM i placówką służby zdrowia w systemie opieki nad pacjentami chorymi na serce, wszczepiony defibrylator cardioverter komunikuje się ze szpitalem. Dane o pacjencie i wszczepionym urządzeniu są przekazywane za pomocą telemetrii RF do urządzenia MQTT w domu pacjenta.

Zazwyczaj transfer odbywa się co noc do nadajnika znajdującego się przy łóżku. Transmitter przesyła dane bezpiecznie przez system telefoniczny do szpitala, gdzie dane są analizowane.

System zmniejsza liczbę wizyt, które pacjent musi wykonać u lekarza. Wykrywa, kiedy pacjent lub urządzenie wymaga uwagi, a w razie nagłego wypadku powiadamia lekarza dyżurnego.

Współpraca między produktem IBM i dostawcą opieki zdrowotnej ma cechy wspólne dla wielu przypadków użycia telemetrycznego:

Invisibility

Urządzenie nie wymaga interwencji użytkownika poza zasilaniem, linią telefoniczną i pozostaniem w pobliżu urządzenia przez część dnia. Jego działanie jest niezawodne i proste w obsłudze.

Aby usunąć konieczność skonfigurowania urządzenia przez pacjenta, dostawca urządzenia wstępnie konfiguruje urządzenie. Pacjent musi go tylko podłączyć. Eliminacja konfiguracji przez pacjenta upraszcza działanie urządzenia i zmniejsza prawdopodobieństwo błędnej konfiguracji urządzenia.

Klient MQTT jest osadzony jako część urządzenia. Programista urządzenia osadza implementację klienta MQTT na urządzeniu, a programista lub dostawca konfiguruje klienta MQTT jako część konfiguracji wstępnej.

Klient MQTT jest dostarczany jako plik JAR Java SE, który programista dołącza do swojej aplikacji Java. W środowiskach innych niż Java, takich jak to, programista urządzenia może zaimplementować klienta w innym języku przy użyciu opublikowanych formatów i protokołu MQTT. Alternatywnie programista może użyć jednego z klientów w języku C dostarczonych jako biblioteki współużytkowane dla platform Windows, Linux i ARM.

Połączenie nierównomierne

Komunikacja między defibrylatorem a szpitalem ma nierównomierną charakterystykę sieci. Dwie różne sieci są używane do rozwiązywania różnych problemów z gromadzeniem danych od pacjenta i wysyłaniem danych do szpitala. Między pacjentem a urządzeniem MQTT używana jest sieć RF o niskim poborze mocy. Nadajnik łączy się ze szpitalem za pomocą połączenia VPN TCP/IP przez linię telefoniczną o niskiej przepustowości.

Często niepraktyczne jest znalezienie sposobu na bezpośrednie połączenie każdego urządzenia z siecią Internet Protocol. Wspólne rozwiązanie polega na wykorzystaniu dwóch sieci połączonych za

pomocą koncentratora. Urządzenie MQTT jest prostym koncentratorom, przechowującym informacje od pacjenta i przekazującym je do szpitala.

Zabezpieczenia

Lekarz musi być w stanie zaufać autentyczności danych pacjenta, a pacjent chce, aby jego dane były respektowane.

W niektórych sytuacjach wystarczy zaszyfrować połączenie przy użyciu sieci VPN lub TLS. W innych sytuacjach wskazane jest zabezpieczenie danych nawet po ich zapisaniu.

Czasami urządzenie telemetryczne nie jest bezpieczne. Może to być na przykład wspólne mieszkanie. Użytkownik urządzenia musi być uwierzytelniony, aby upewnić się, że dane pochodzą od właściwego pacjenta. Samo urządzenie może zostać uwierzytelnione na serwerze za pomocą protokołu TLS, a serwer może zostać uwierzytelniony na urządzeniu.

Kanał pomiarowy między urządzeniem a menedżerem kolejek obsługuje usługę JAAS na potrzeby uwierzytelniania użytkowników i protokołu TLS na potrzeby szyfrowania komunikacji oraz uwierzytelniania urządzeń. Dostęp do publikacji jest kontrolowany przez menedżera uprawnień do obiektów w systemie IBM MQ.

Identyfikator używany do uwierzytelniania użytkownika może zostać odwzorowany na inny identyfikator, taki jak wspólna tożsamość pacjenta. Wspólny identyfikator upraszcza konfigurowanie autoryzacji do tematów publikacji w programie IBM MQ.

Połączenia

Połączenie między urządzeniem MQTT a szpitalem wykorzystuje połączenie modemowe i działa z przepustowością nawet 300 bodów.

Aby działać wydajnie z prędkością 300 bodów, MQTT protocol dodaje tylko kilka dodatkowych bajtów do komunikatu oprócz nagłówek TCP/IP.

Usługa MQTT protocol udostępnia pojedyncze przesyłanie komunikatów *fire and forget*, które utrzymuje niskie opóźnienia. Może również użyć wielu transmisji, aby zagwarantować *co najmniej raz i dokładnie raz* dostarczenie, jeśli gwarantowane dostarczenie jest ważniejsze niż czas odpowiedzi. Aby zagwarantować dostarczenie, komunikaty są przechowywane w urządzeniu do momentu ich pomyślnego dostarczenia. Jeśli urządzenie jest podłączone bezprzewodowo, gwarantowana dostawa jest szczególnie przydatna.

skalowalnością;

Urządzenia telemetryczne są zwykle wdrażane w dużych ilościach, od dziesiątek tysięcy do milionów.

Podłączenie wielu urządzeń do systemu powoduje duże zapotrzebowanie na rozwiązanie.

Istnieją wymagania biznesowe, takie jak koszt urządzeń i ich oprogramowania oraz wymagania administracyjne dotyczące zarządzania licencjami, urządzeniami i użytkownikami. Wymagania techniczne obejmują obciążenie sieci i serwerów.

Otwieranie połączeń wykorzystuje więcej zasobów serwera niż utrzymywanie otwartych połączeń. Jednak w przypadku użycia takiego jak ten, w którym używane są linie telefoniczne, koszt połączeń oznacza, że połączenia pozostają otwarte nie dłużej niż jest to wymagane. Transfery danych mają w dużej mierze charakter wsadowy. Połączenia mogą być zaplanowane na całą noc, aby uniknąć nagłego szczytowego połączenia przed snem.

Na kliencie skalowalność klientów jest wspomagana przez minimalną wymaganą konfigurację klienta. Klient MQTT jest wbudowany w urządzenie. Nie jest wymagane, aby krok konfiguracji lub akceptacji licencji klienta MQTT był wbudowany we wdrażanie urządzeń u pacjentów.

Na serwerze MQ Telemetry ma początkowe miejsce docelowe 50 000 otwartych połączeń dla każdego menedżera kolejek.

Połączenia są zarządzane za pomocą programu IBM MQ Explorer. Program IBM MQ Explorer filtruje połączenia, które mają być wyświetlane, do liczby możliwej do zarządzania. Przy odpowiednio wybranym schemacie przydzielania identyfikatorów do klientów można filtrować połączenia na podstawie położenia geograficznego lub alfabetycznie według nazwy pacjenta.

Monitoring and Control

Inteligentne liczniki zbierają więcej szczegółów na temat zużycia energii niż tradycyjne liczniki.

Inteligentne mierniki są często połączone z lokalną siecią telemetryczną do monitorowania i kontroli poszczególnych urządzeń w domu. Niektóre są również zdalnie podłączone do monitorowania i kontroli na odległość.

Połączenie zdalne może być skonfigurowane przez osobę fizyczną, przez narzędzie zasilające lub przez centralny punkt kontrolny. Zdalny punkt kontrolny może odczytywać zużycie energii i udostępniać dane o wykorzystaniu. Może udostępniać dane wpływające na wykorzystanie, takie jak ciągłe ustalanie cen i informacje o pogodzie. Może ograniczyć obciążenie, aby poprawić ogólną efektywność wytwarzania energii.

Inteligentne liczniki zaczynają być szeroko wdrażane. Na przykład rząd Wielkiej Brytanii konsuluje się w sprawie wdrożenia inteligentnych liczników do każdego domu w Wielkiej Brytanii do 2020 roku.

Domowy pomiar przypadków użycia mają wiele wspólnych cech:

Invisibility

Jeśli użytkownik nie chce być zaangażowany w oszczędzanie energii za pomocą licznika, licznik nie może wymagać interwencji użytkownika. Nie może on zmniejszać niezawodności dostaw energii do poszczególnych urządzeń.

Klient MQTT może być wbudowany w oprogramowanie wdrożone z licznikiem i nie wymaga oddzielnej instalacji ani konfiguracji.

Połączenie nierównomierne

Komunikacja między urządzeniami i inteligentnym miernikiem wymaga innych standardów łączności niż między miernikiem i zdalnym punktem połączenia.

Połączenie między inteligentnym miernikiem a urządzeniami musi być wysoce dostępne i zgodne ze standardami sieciowymi dla sieci macierzystych.

Sieć zdalna prawdopodobnie używa różnych połączeń fizycznych. Niektóre z nich, np. komórkowe, mają wysokie koszty transmisji i mogą być sporadyczne. Specyfikacja MQTT v3 dotyczy połączeń zdalnych oraz połączeń między adapterami lokalnymi i licznikiem inteligentnym.

Połączenie między gniazdami zasilającymi i zastosowanymi, a miernikiem, wykorzystuje sieć domową, taką jak Zigbee. MQTT dla sieci z czujnikami (MQTT-S), jest przeznaczony do pracy z protokołami Zigbee i innymi protokołami sieciowymi o niskiej przepustowości. System MQ Telemetry nie obsługuje bezpośrednio opcji MQTT-S. Wymaga bramy do połączenia systemu MQTT-S z systemem MQTT v3.

Podobnie jak w przypadku monitorowania pacjentów w domu, rozwiązania do monitorowania i kontroli energii w domu wymagają wielu sieci połączonych za pomocą inteligentnego miernika jako koncentratora.

Zabezpieczenia

Istnieje wiele problemów z bezpieczeństwem powiązanych z licznikami inteligentnymi. Problemy te obejmują niezaprzeczalność transakcji, autoryzację wszelkich zainicjowanych działań kontrolnych oraz prywatność danych dotyczących zużycia energii.

Aby zapewnić prywatność, dane przesyłane między licznikiem a zdalnym punktem kontrolnym przez produkt MQTT mogą być szyfrowane za pomocą protokołu TLS. Aby zapewnić autoryzację działań kontrolnych, połączenie MQTT między licznikiem i zdalnym punktem kontrolnym może być wzajemnie uwierzytelnianie przy użyciu protokołu TLS.

Połączenia

Fizyczna natura sieci zdalnej może się znacznie różnić. Może korzystać z istniejącego połączenia szerokopasmowego lub z sieci komórkowej o wysokich kosztach połączeń i sporadycznych dostępności. W przypadku połączeń nieregularnych o wysokim koszcie MQTT jest wydajnym

i niezawodnym protokołem; patrz [“Telemetria przypadek użycia: Home patient monitoring”](#) na stronie 114.

skalowalnością;

W końcu firmy energetyczne, czyli centralne punkty kontrolne, planują wdrożyć dziesiątki milionów inteligentnych liczników. Początkowo liczba metrów na wdrożenie wynosi od dziesiątek do setek tysięcy. Ta liczba jest porównywalna z początkową wartością docelową produktu MQTT wynoszącą 50 000 otwartych połączeń klienckich na menedżer kolejek.

Kluczowym aspektem architektury dla domowego monitorowania i sterowania energią jest użycie inteligentnego miernika jako koncentratora sieciowego. Każdy adapter urządzenia jest osobnym czujnikiem. Łącząc je z lokalnym serwerem koncentrującym za pomocą MQTT, serwer koncentrujący może skoncentrować przepływ danych na pojedynczej sesji TCP/IP z centralnym punktem kontrolnym, a także przechowywać komunikaty przez krótki czas w celu wyeliminowania wyłączeń sesji.

Połączenia zdalne muszą być otwarte w domowych przypadkach użycia energii z dwóch powodów. Najpierw, ponieważ otwieranie połączeń zajmuje dużo czasu w stosunku do wysyłania żądań. Czas otwierania wielu połączeń w celu wystąpienia żądań "load-limit" w krótkim odstępie czasu jest zbyt długi. Po drugie, aby odbierać żądania ograniczenia obciążenia od przedsiębiorstwa energetycznego, połączenie musi być najpierw otwarte przez klienta. W przypadku systemu MQTT połączenia są zawsze inicjowane przez klienta, a aby odbierać żądania ograniczenia obciążenia od przedsiębiorstwa energetycznego, połączenie musi pozostać otwarte.

Jeśli szybkość otwierania połączeń jest krytyczna lub serwer inicjuje żądania o newralgicznym czasie, zwykle rozwiązanie utrzymuje wiele otwartych połączeń.

Przypadki użycia telemetrii: Identyfikacja radiowa (RFID)

RFID to użycie osadzonego znacznika RFID do identyfikacji i śledzenia obiektu bezprzewodowo. Znaczniki RFID mogą być odczytywane do zakresu kilku metrów i poza zasięgiem wzroku czytnika RFID. Znaczniki pasywne są aktywowane przez czytnik RFID. Aktywne znaczniki są przesyłane bez zewnętrznej aktywacji. Aktywne znaczniki muszą mieć źródło zasilania. Znaczniki pasywne mogą zawierać źródło zasilania w celu zwiększenia ich zakresu.

RFID jest używany w wielu aplikacjach, a typy przypadków użycia są bardzo zróżnicowane. Przypadki użycia RFID oraz monitorowanie pacjentów w domu i monitorowanie energii w domu, a także kontrola przypadków użycia, mają pewne podobieństwa i różnice.

Invisibility

W wielu przypadkach użycia czytnik RFID jest wdrażany w dużej liczbie i musi działać bez interwencji użytkownika. Czytnik zawiera wbudowany klient MQTT do komunikacji z centralnym punktem kontrolnym.

Na przykład w magazynie dystrybucyjnym czytnik używa czujnika ruchu do wykrywania palet. Aktywuje znaczniki RFID elementów na palecie i wysyła dane i żądania do centralnych aplikacji. Dane są używane do aktualizowania lokalizacji magazynu. Żądania sterują tym, co dzieje się z paletą w następnej kolejności, na przykład przenosząc ją do konkretnej zatoki. Linie lotnicze i systemy bagażowe na lotniskach używają w ten sposób RFID.

W niektórych przypadkach użycia RFID czytnik ma standardowe środowisko obliczeniowe, takie jak Java Platform, Micro Edition (Java ME). W takich przypadkach klient MQTT może zostać wdrożony w odrębnym kroku konfiguracji po zakończeniu produkcji.

Połączenie nierównomierne

Czytniki RFID mogą być oddzielone od lokalnego urządzenia sterującego, które zawiera klienta MQTT, lub każdy z nich może zawierać klienta MQTT. Zwykle czynniki geograficzne lub komunikacyjne wskazują na wybór topologii.

Zabezpieczenia

Prywatność i autentyczność to kwestie bezpieczeństwa związane z dołączaniem znaczników RFID. Znaczniki RFID są niezauważalne i mogą być potajemnie monitorowane, podszywane lub manipulowane.

Rozwiązanie problemów z bezpieczeństwem RFID zwiększa możliwość wdrożenia nowych rozwiązań RFID. Chociaż ryzyko naruszenia bezpieczeństwa jest w znaczniku RFID, a lokalny czytnik, korzystanie z centralnego przetwarzania informacji sugeruje podejście do przeciwdziałania różnym zagrożeniom. Na przykład manipulowanie oznakowaniem może być wykrywane przez dynamiczne korelowanie poziomów zapasów z dostawami i wysyłkami.

Połączenia

Aplikacje RFID zwykle obejmują zarówno przechowywanie wsadowe, jak i przekazywanie informacji zebranych z czytników RFID i zapytań bezpośrednich. W przypadku użycia magazynu dystrybucyjnego czytnik RFID jest cały czas podłączony. Gdy znacznik jest odczytywany, jest on publikowany razem z informacjami o czytniku. Aplikacja opracowywania danych publikuje odpowiedź z powrotem do czytnika.

W aplikacji opracowywania danych sieć jest zwykle niezawodna, a bezpośrednie żądania mogą korzystać z komunikatów *fire and forget* w celu zapewnienia niskiej wydajności z opóźnieniem. Dane przechowywane i przekazywane w trybie wsadowym mogą korzystać z mechanizmu przesyłania komunikatów *dokładnie jeden raz* w celu zminimalizowania kosztów administracyjnych związanych z traceniem danych.

skalowalnością;

Jeśli aplikacja RFID wymaga natychmiastowych reakcji, w kolejności sekund lub dwóch, czytniki RFID muszą pozostać połączone.

Przypadki użycia danych telemetrycznych: czujniki środowiska

Czujniki środowiska wykorzystują telemetrię do gromadzenia informacji na temat poziomu i jakości wód rzecznych, zanieczyszczeń atmosferycznych i innych danych środowiskowych.

Czujniki są często umieszczane w odległych miejscach, bez dostępu do komunikacji przewodowej. Przepustowość sieci bezprzewodowej jest kosztowna, a niezawodność może być niska. Zwykle pewna liczba czujników otoczenia w małym obszarze geograficznym jest podłączona do lokalnego urządzenia monitorującego w bezpiecznym miejscu. Połączenia lokalne mogą być przewodowe lub bezprzewodowe.

Invisibility

Urządzenia z czujnikami są prawdopodobnie mniej dostępne, zasilane i wdrażane w większej liczbie niż centralne urządzenie monitorujące. Czujniki są czasami "głupie", a lokalne urządzenie monitorujące zawiera adaptery do transformacji i przechowywania danych z czujników. Urządzenie monitorujące może zawierać komputer ogólnego przeznaczenia, który obsługuje Java Platform, Standard Edition (Java SE) lub Java Platform, Micro Edition (Java ME). Podczas konfigurowania klienta MQTT jest mało prawdopodobne, aby widoczność była głównym wymaganiem.

Połączenie nierównomierne

Możliwości czujników oraz koszt i przepustowość połączenia zdalnego zwykle skutkuje lokalnym serwerem koncentrującym monitorowanie połączonym z serwerem centralnym.

Zabezpieczenia

Jeśli rozwiązanie nie jest używane w przypadku użycia wojskowego lub obronnego, bezpieczeństwo nie jest głównym wymogiem.

Połączenia

Wiele zastosowań nie wymaga ciągłego monitorowania ani natychmiastowej dostępności danych. Dane wyjątku, takie jak alert na poziomie powodziowym, muszą zostać natychmiast przekazane. Dane z czujników są agregowane w monitorze lokalnym w celu zmniejszenia kosztów połączenia

i komunikacji, a następnie przesyłane za pomocą zaplanowanych połączeń. Dane wyjątku są przekazywane natychmiast po wykryciu w monitorze.

skalowalnością;

Czujniki są skupione wokół lokalnych koncentratorów, a dane z czujników są agregowane w pakiety, które są przesyłane zgodnie z harmonogramem. Oba te czynniki zmniejszają obciążenie serwera centralnego, które może być związane z używaniem czujników podłączonych bezpośrednio.

Przypadki użycia danych telemetrycznych: aplikacje dla urządzeń przenośnych

Aplikacje dla urządzeń przenośnych to aplikacje działające na urządzeniach bezprzewodowych. Są to albo ogólne platformy aplikacji, albo urządzenia niestandardowe.

Platformy ogólne obejmują urządzenia kieszonkowe, takie jak telefony i asystenty danych osobowych, oraz urządzenia przenośne, takie jak notebooki. Urządzenia niestandardowe używają sprzętu specjalnego przeznaczenia dostosowanego do konkretnych aplikacji. Przykładem niestandardowego urządzenia mobilnego jest urządzenie rejestrujące przesyłkę "signed-for". Aplikacje na niestandardowych urządzeniach mobilnych są często budowane na ogólnej platformie oprogramowania.

Invisibility

Wdrożenie niestandardowych aplikacji mobilnych jest zarządzane i może obejmować konfigurację aplikacji klienckiej MQTT. Podczas konfigurowania klienta MQTT jest mało prawdopodobne, aby widoczność była głównym wymaganiem.

Połączenie nierównomierne

W przeciwieństwie do lokalnej topologii koncentratora w poprzednich przypadkach użycia, klienci mobilne łączą się zdalnie. Warstwa aplikacji klienckiej łączy się bezpośrednio z aplikacją w centralnym koncentratorze.

Zabezpieczenia

Przy niewielkim poziomie bezpieczeństwa fizycznego należy uwierzytelnić urządzenie mobilne i użytkownika mobilnego. Do potwierdzenia tożsamości urządzenia używany jest protokół TLS, a do uwierzytelnienia użytkownika używany jest protokół JAAS.

Połączenia

Jeśli aplikacja mobilna zależy od zasięgu sieci bezprzewodowej, musi być w stanie działać w trybie bez połączenia i efektywnie obsługiwać przerwane połączenie. W tym środowisku celem jest pozostawienie połączenia, ale aplikacja musi mieć możliwość zapisywania i przekazywania komunikatów. Często komunikaty są zamówieniami lub potwierdzeniami dostawy i mają ważną wartość biznesową. Muszą być przechowywane i przekazywane niezawodnie.

skalowalnością;

Skalowalność nie jest głównym problemem. W niestandardowych przypadkach użycia aplikacji mobilnych liczba klientów aplikacji prawdopodobnie nie przekroczy liczby tysięcy lub dziesiątek tysięcy.

Łączenie urządzeń pomiarowych z menedżerem kolejek

Urządzenia pomiarowe łączą się z menedżerem kolejek przy użyciu klienta MQTT v3. Klient MQTT v3 używa protokołu TCP/IP do nawiązania połączenia z programem nasłuchującym TCP/IP o nazwie usługa telemetryczna (MQXR).

Po połączeniu urządzenia pomiarowego z menedżerem kolejek klient MQTT inicjuje połączenie TCP/IP przy użyciu metody `MqttClient.connect`. Podobnie jak klienci IBM MQ, klient MQTT musi być połączony z menedżerem kolejek w celu wysyłania i odbierania komunikatów. Połączenie jest nawiązywane na serwerze przy użyciu programu nasłuchującego TCP/IP zainstalowanego z produktem

MQ Telemetrii nazywanego usługą telemetryczną (MQXR). Każdy menedżer kolejek uruchamia maksymalnie jedną usługę telemetryczną (MQXR).

Usługa telemetryczna (MQXR) używa adresu gniazda zdalnego ustawionego przez każdy klient w metodzie `MqttClient.connect` w celu przydzielenia połączenia do kanału pomiarowego. Adres gniazda jest kombinacją nazwy hosta TCP/IP i numeru portu. Wiele klientów używających tego samego adresu gniazda zdalnego jest połączonych z tym samym kanałem pomiarowym przez usługę telemetryczną (MQXR).

Jeśli na serwerze znajduje się wiele menedżerów kolejek, należy rozdzielić kanały pomiarowe między menedżery kolejek. Przydziel adresy gniazd zdalnych między menedżerami kolejek. Zdefiniuj każdy kanał pomiarowy z unikalnym adresem gniazda zdalnego. Dwa kanały pomiarowe nie mogą używać tego samego adresu gniazda.

Jeśli ten sam adres gniazda zdalnego jest skonfigurowany dla kanałów pomiarowych w wielu menedżerach kolejek, wygrywa pierwszy kanał pomiarowy do nawiązania połączenia. Kolejne kanały łączące się pod tym samym adresem nie powiodą się.

Jeśli na serwerze istnieje wiele adapterów sieciowych, podziel adresy zdalnych gniazd między kanały pomiarowe. Przydzielanie adresów gniazd jest całkowicie dowolne, o ile dowolny konkretny adres gniazda jest skonfigurowany tylko w jednym kanale pomiarowym.

Skonfiguruj program IBM MQ do łączenia się z klientami MQTT za pomocą kreatorów dostępnych w suplemencie MQ Telemetry dla IBM MQ Explorer. Alternatywnie można postępować zgodnie z instrukcjami zawartymi w sekcji [Konfigurowanie menedżera kolejek na potrzeby pomiarów telemetrycznych w systemach Linux i AIX](#) oraz [Konfigurowanie menedżera kolejek na potrzeby pomiarów telemetrycznych w systemie Windows](#), aby skonfigurować telemetrię ręcznie.

Odsyłacze pokrewne

[Właściwości MQXR](#)

Windows

Linux

AIX

Protokoły połączeń telemetrycznych

Produkt MQ Telemetry obsługuje protokoły TCP/IP IPv4 i IPv6 oraz TLS.

Windows

Linux

AIX

Usługa telemetryczna (MQXR)

Usługa telemetryczna (MQXR) jest programem nasłuchującym TCP/IP, który jest zarządzany jako usługa systemu IBM MQ. Utwórz usługę za pomocą kreatora IBM MQ Explorer lub za pomocą komendy `runmqsc`.




Usługa MQ Telemetry (MQXR) nosi nazwę `SYSTEM.MQXR.SERVICE`.


Przykładowy kreator konfiguracji produktu Telemetry udostępniony w funkcji MQ Telemetry dla produktu IBM MQ Explorer tworzy usługę telemetryczną i przykładowy kanał telemetryczny. Więcej informacji na ten temat zawiera sekcja [Weryfikowanie instalacji produktu MQ Telemetry przy użyciu produktu IBM MQ Explorer](#).

Utwórz przykładową konfigurację z poziomu wiersza komend. Informacje na ten temat zawiera sekcja [Weryfikowanie instalacji produktu MQ Telemetry za pomocą wiersza komend](#).

Usługa telemetryczna (MQXR) jest uruchamiana i zatrzymywana automatycznie z menedżerem kolejek. Sterowanie usługą za pomocą folderu usług w katalogu IBM MQ Explorer. Aby wyświetlić usługę, należy kliknąć ikonę w celu zatrzymania IBM MQ Explorer filtrowania obiektów `SYSTEM` z ekranu.

Przykład ręcznego tworzenia usługi zawiera sekcja

-   [Tworzenie `SYSTEM.MQXR.SERVICE` w systemie Linux.](#)
-  [Tworzenie `SYSTEM.MQXR.SERVICE` w systemie Windows.](#)

 Poczynając od wersji IBM MQ 9.3.0, [Tworzenie pliku `SYSTEM.MQXR.SERVICE` w systemie Linux](#) i [Tworzenie pliku `SYSTEM.MQXR.SERVICE` w systemie Windows](#) są aktualizowane w celu określenia

domyślnego klucza, który wymaga szyfrowania fraz haseł dla kanałów TLS MQTT. Więcej informacji na ten temat zawiera sekcja [Szyfrowanie fraz haseł dla kanałów TLS produktu MQTT](#).

Windows

Linux

AIX

Kanały pomiarowe

Kanały pomiarowe umożliwiają tworzenie połączeń o różnych właściwościach, takich jak usługa Java Authentication and Authorization Service (JAAS) lub uwierzytelnianie TLS, a także zarządzanie grupami klientów.

Tworzenie kanałów pomiarowych przy użyciu kreatora **New Telemetry Channel** udostępnionego w funkcji MQ Telemetry dla produktu IBM MQ Explorer. Skonfiguruj kanał, korzystając z kreatora, w celu akceptowania połączeń od klientów MQTT na określonym porcie TCP/IP. Od wersji IBM WebSphere MQ 7.1 można skonfigurować program MQ Telemetry za pomocą programu wiersza komend **runmqsc**.

Aby ułatwić zarządzanie dużą liczbą połączeń klienckich, należy utworzyć wiele kanałów pomiarowych na różnych portach, dzieląc klienty na grupy. Każdy kanał pomiarowy ma inną nazwę.

Istnieje możliwość skonfigurowania kanałów pomiarowych z różnymi atrybutami zabezpieczeń w celu utworzenia różnych typów połączeń. Utwórz wiele kanałów, aby akceptować połączenia klientów na różnych adresach TCP/IP. Protokół TLS służy do szyfrowania komunikatów i uwierzytelniania kanału i klienta telemetrycznego. Więcej informacji na ten temat zawiera sekcja [Konfiguracja protokołu TLS klientów i kanałów telemetrycznych produktu MQTT](#). Podaj ID użytkownika, aby uprościć autoryzowanie dostępu do obiektów IBM MQ. Określ konfigurację JAAS w celu uwierzytelnienia użytkownika serwera MQTT za pomocą usługi JAAS. Więcej informacji na ten temat zawiera sekcja [Identyfikacja klienta MQTT, autoryzacja i uwierzytelnianie](#).

Windows

Linux

AIX

IBM MQ Telemetry Transport :NONE.

Protokół w systemie IBM MQ Telemetry Transport (MQTT) v3 jest przeznaczony do wymiany komunikatów między małymi urządzeniami przy małej przepustowości lub dużych połączeniach, a także do niezawodnego wysyłania komunikatów. Używa protokołu TCP/IP.

Plik MQTT protocol jest publikowany; patrz [IBM MQ Telemetry Transport format i protokół](#). Wersja 3 protokołu używa publikowania/subskrypcji i obsługuje trzy jakości usługi: *fire and forget, co najmniej raz i dokładnie raz*.

Mała wielkość nagłówek protokołu i ładunku komunikatu tablicy bajtów powoduje, że komunikaty są małe. Nagłówki składają się z 2-bajtowego stałego nagłówka i do 12 bajtów dodatkowych nagłówek zmiennych. Protokół używa 12-bajtowych nagłówek zmiennych do subskrybowania i łączenia się oraz tylko 2-bajtowych nagłówek zmiennych dla większości publikacji.

W przypadku trzech elementów jakości usługi można dokonać wymiany między niskimi opóźnieniami i niezawodnością. Więcej informacji na ten temat zawiera sekcja [Qualności usług udostępnianych przez MQTT klienta](#). Opcja *Uruchom i zapomnij* nie używa trwałej pamięci masowej urządzenia i tylko jednej transmisji w celu wysłania lub odebrania publikacji. *Co najmniej raz i dokładnie jeden raz* wymagają trwałej pamięci masowej na urządzeniu, aby zachować stan protokołu i zapisać komunikat do czasu jego potwierdzenia.

Windows

Linux

AIX

MQTT klienty

Aplikacja kliencka MQTT jest odpowiedzialna za gromadzenie informacji z urządzenia pomiarowego, nawiązywanie połączenia z serwerem i publikowanie informacji na serwerze. Może również subskrybować tematy, odbierać publikacje i sterować urządzeniem telemetrycznym.

W przeciwieństwie do aplikacji klienckich IBM MQ, aplikacje klienckie MQTT nie są aplikacjami IBM MQ. Nie określają one menedżera kolejek, z którym ma zostać nawiązane połączenie. Nie są one ograniczone do korzystania z konkretnych interfejsów programistycznych języka IBM MQ. Zamiast tego klienty MQTT implementują protokół MQTT 3. Użytkownik może napisać własną bibliotekę kliencką w celu połączenia się z MQTT protocol w języku programowania i na wybranej platformie. Patrz [IBM MQ Telemetry Transport format i protokół \(format and protocol\)](#).

Aby uprościć pisanie aplikacji klienckich MQTT, należy użyć bibliotek klienta C, Java i JavaScript, które obudowują MQTT protocol dla wielu platform. Jeśli te biblioteki zostaną włączone do aplikacji MQTT, w pełni funkcjonalny klient MQTT może być skrócony do 15 wierszy kodu. Biblioteki klienta MQTT są dostępne bezpłatnie w serwisie Eclipse Paho i MQTT.org. Patrz [programy przykładowe IBM MQ Telemetry Transport](#).

Aplikacja kliencka MQTT jest zawsze odpowiedzialna za inicjowanie połączenia z kanałem pomiarowym. Po nawiązaniu połączenia aplikacja kliencka MQTT lub aplikacja IBM MQ mogą rozpocząć wymianę komunikatów.

Aplikacje klienckie MQTT i aplikacje IBM MQ publikują i subskrybują ten sam zestaw tematów. Aplikacja IBM MQ może również wysłać komunikat bezpośrednio do aplikacji klienckiej MQTT bez uprzedniego tworzenia subskrypcji przez aplikację kliencką. Patrz sekcja [Konfigurowanie rozproszonego kolejkowania w celu wysyłania komunikatów do klientów MQTT](#).

Aplikacje klienckie MQTT są połączone z produktem IBM MQ przy użyciu kanału pomiarowego. Kanał pomiarowy działa jako most między różnymi typami komunikatów używanymi przez produkt MQTT i produkt IBM MQ. Tworzy on publikacje i subskrypcje w menedżerze kolejek w imieniu aplikacji klienckiej MQTT. Kanał pomiarowy wysyła publikacje zgodne z subskrypcjami aplikacji klienckiej MQTT z menedżera kolejek do aplikacji klienckiej MQTT.

Windows

Linux

AIX

Wysyłanie komunikatu do klienta MQTT

Aplikacje IBM MQ mogą wysyłać komunikaty klientów w wersji MQTT v3, publikując je w subskrypcjach utworzonych przez klienty lub wysyłając komunikaty bezpośrednio. Klienci MQTT mogą wysyłać komunikaty do siebie nawzajem, publikując je w tematach zasubskrybowanych przez inne klienty.

Klient MQTT subskrybuje publikację, którą otrzymuje od IBM MQ

Wykonaj zadanie "[Publikowanie komunikatu w programie narzędziowym klienta MQTT z serwisu IBM MQ Explorer](#)" na stronie 124, aby wysłać publikację z programu IBM MQ do klienta MQTT.

Standardowym sposobem odbierania komunikatów przez klienta MQTT v3 jest utworzenie przez niego subskrypcji tematu lub zbioru tematów. W przykładowym fragmencie kodu [Rysunek 44 na stronie 123](#) klient MQTT subskrybuje przy użyciu łańcucha tematu "MQTT Examples". Aplikacja w języku C (IBM MQ C application, [Rysunek 45 na stronie 123](#)) publikuje w temacie przy użyciu łańcucha tematu "MQTT Examples". We fragmencie kodu [Rysunek 46 na stronie 124](#) klient MQTT odbiera publikację w metodzie wywołania zwrotnego `messageArrived`.

Więcej informacji na temat konfigurowania produktu IBM MQ do wysyłania publikacji w odpowiedzi na subskrypcje od klientów MQTT zawiera sekcja [Publikowanie komunikatu w odpowiedzi na subskrypcję klienta MQTT](#).

Aplikacja IBM MQ wysyła komunikat bezpośrednio do klienta MQTT

Wykonaj czynność "[Wysyłanie komunikatu do klienta MQTT za pomocą programu IBM MQ Explorer](#)" na stronie 129, aby wysłać komunikat bezpośrednio z produktu IBM MQ do klienta MQTT.

Komunikat wysyłany w ten sposób do klienta MQTT jest nazywany niechcianym komunikatem. Klienci MQTT v3 odbierają niezamówione komunikaty jako publikacje z ustawioną nazwą tematu. Usługa telemetryczna (MQXR) ustawia nazwę tematu na nazwę kolejki zdalnej.

Więcej informacji na temat konfigurowania produktu IBM MQ do wysyłania komunikatów bezpośrednio do klientów MQTT zawiera sekcja [Wysyłanie komunikatu bezpośrednio do klienta](#).

Klient MQTT publikuje komunikat

Klient MQTT v3 może publikować komunikat, który jest odbierany przez innego klienta MQTT v3, ale nie może wysyłać niezamówionego komunikatu. Fragment kodu [Rysunek 47 na stronie 124](#) pokazuje, w jaki sposób klient MQTT v3 napisany w języku Java publikuje komunikat.

Typowym wzorcem wysyłania komunikatu do jednego konkretnego klienta MQTT v3 jest utworzenie przez każdego klienta subskrypcji na własny identyfikator `ClientIdentifier`. Wykonaj czynność [“Publikowanie komunikatu w konkretnym kliencie MQTT v3”](#) na stronie 130 , aby opublikować komunikat z jednego klienta MQTT na innym kliencie MQTT przy użyciu łańcucha tematu `ClientIdentifier` .

Przykładowe fragmenty kodu

Fragment kodu w pliku [Rysunek 44](#) na stronie 123 pokazuje, w jaki sposób klient MQTT napisany w języku Java tworzy subskrypcję. Do odbierania publikacji dla subskrypcji wymagana jest także metoda wywołania zwrótnego `messageArrived` .

```
String    clientId = String.format("%-23.23s",
                                System.getProperty("user.name") + "_" +
                                (UUID.randomUUID().toString()).trim()).replace('-', '_');
MqttClient client = new MqttClient("localhost", clientId);
String topicString = "MQTT Examples";
int       QoS = 1;
client.subscribe(topicString, QoS);
```

Rysunek 44. Subskrybent klienta MQTT v3

Fragment kodu w pliku [Rysunek 45](#) na stronie 123 pokazuje, w jaki sposób aplikacja IBM MQ napisana w języku C wysyła publikację. Fragment kodu jest wyodrębniany z czynności [Tworzenie publikatora do tematu zmiennej](#) .

```
/* Define and set variables to defaults */
/* Omitted lines declaring variables */
char * topicName   = ""
char * topicString = "MQTT Examples"
char * publication = "Hello world!";
do {
    MQCONN(qMgrName, &Hconn, &CompCode, &Reason);
    if (CompCode != MQCC_OK) break;
    td.ObjectType = MQOT_TOPIC; /* Object is a topic */
    td.Version = MQOD_VERSION_4; /* Descriptor needs to be V4 */
    strncpy(td.ObjectName, topicName, MQ_TOPIC_NAME_LENGTH);
    td.ObjectString.VSPtr = topicString;
    td.ObjectString.VSLength = (MQLONG)strlen(topicString);
    MQOPEN(Hconn, &td, MQOO_OUTPUT | MQOO_FAIL_IF QUIESCING, &Hobj, &CompCode, &Reason);
    if (CompCode != MQCC_OK) break;
    pmo.Options = MQPMO_FAIL_IF QUIESCING | MQPMO_RETAIN;
    MQPUT(Hconn, Hobj, &md, &pmo, (MQLONG)strlen(publication)+1, publication, &CompCode, &Reason);
    if (CompCode != MQCC_OK) break;
    MQCLOSE(Hconn, &Hobj, MQCO_NONE, &CompCode, &Reason);
    if (CompCode != MQCC_OK) break;
    MQDISC(&Hconn, &CompCode, &Reason);
} while (0);
```

Rysunek 45. IBM MQ publikator

Po nadejściu publikacji klient MQTT wywołuje metodę `messageArrived` klasy MQTT application client `MqttCallback` .

```

public class Callback implements MqttCallback {
    public void messageArrived(MqttTopic topic, MqttMessage message) {
        try {
            System.out.println("Message arrived: \"" + message.toString()
                + "\" on topic \"" + topic.toString() + "\"");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    // ... Other callback methods
}

```

Rysunek 46. Metoda `messageArrived`

Rysunek 47 na stronie 124 przedstawia MQTT v3 publikujący komunikat w subskrypcji utworzonej w wersji Rysunek 44 na stronie 123.

```

String address = "localhost";
String clientId = String.format("%-23.23s",
    System.getProperty("user.name") + "_" +
    (UUID.randomUUID().toString()).trim()).replace('-', '_');
MqttClient client = new MqttClient(address, clientId);
String topicString = "MQTT Examples";
MqttTopic topic = client.getTopic(Example.topicString);
String publication = "Hello world";
MqttMessage message = new MqttMessage(publication.getBytes());
MqttDeliveryToken token = topic.publish(message);

```

Rysunek 47. Publikator klienta MQTT v3

Windows Linux AIX Publikowanie komunikatu w programie narzędziowym klienta MQTT z serwisu IBM MQ Explorer

Wykonaj kroki opisane w tej czynności, aby opublikować komunikat przy użyciu produktu IBM MQ Explorer, a następnie zasubskrybuj go za pomocą programu narzędziowego klienta MQTT. Dodatkowe zadanie przedstawia sposób konfigurowania aliasu menedżera kolejek zamiast ustawiania domyślnej kolejki transmisji na wartość `SYSTEM.MQTT.TRANSMIT.QUEUE`.

Zanim rozpoczniesz

W tej czynności przyjęto założenie, że użytkownik jest zaznajomiony z produktem IBM MQ i produktem IBM MQ Explorer oraz że zainstalowano opcje IBM MQ i MQ Telemetry.

Użytkownik tworzący zasoby menedżera kolejek dla tego zadania musi mieć wystarczające uprawnienia, aby to zrobić. W celach demonstracyjnych przyjmuje się, że ID użytkownika IBM MQ Explorer należy do grupy `mqm`.

O tym zadaniu

W ramach tej czynności użytkownik tworzy temat w produkcie IBM MQ i subskrybuje ten temat przy użyciu programu narzędziowego klienta MQTT. Po opublikowaniu w temacie za pomocą programu IBM MQ Explorer klient MQTT otrzymuje publikację.

Procedura

Wykonaj jedną z następujących czynności:

- Zainstalowano produkt MQ Telemetry, ale nie został on jeszcze uruchomiony. Wykonaj czynność: ["Uruchom zadanie bez zdefiniowanej usługi telemetrycznej \(MQXR\)" na stronie 125.](#)

- Telemetrię produktu IBM MQ uruchomiono wcześniej, ale w celu wykonania demonstracji należy użyć nowego menedżera kolejek. Wykonaj czynność: [“Uruchom zadanie bez zdefiniowanej usługi telemetrycznej \(MQXR\)”](#) na stronie 125.
- Zadanie ma zostać wykonane przy użyciu istniejącego menedżera kolejek, który nie ma zdefiniowanych zasobów telemetrycznych. Kreator **Definiowanie przykładowej konfiguracji** nie ma być uruchamiany.
 - a. Aby skonfigurować telemetrię, wykonaj jedną z następujących czynności:
 - [Konfigurowanie menedżera kolejek na potrzeby telemetrii w systemach Linux i AIX](#)
 - [Konfigurowanie menedżera kolejek na potrzeby telemetrii w systemie Windows](#)
 - b. Wykonaj czynność: [“Uruchom zadanie z uruchomioną usługą telemetryczną \(MQXR\)”](#) na stronie 126
- Aby wykonać to zadanie przy użyciu istniejącego menedżera kolejek, który ma już zdefiniowane zasoby pomiarowe, należy wykonać następującą czynność: [“Uruchom zadanie z uruchomioną usługą telemetryczną \(MQXR\)”](#) na stronie 126.

Co dalej

Wykonaj [“Wysyłanie komunikatu do klienta MQTT za pomocą programu IBM MQ Explorer”](#) na stronie 129 , aby wysłać komunikat bezpośrednio do programu narzędziowego klienta.

Uruchom zadanie bez zdefiniowanej usługi telemetrycznej (MQXR)

Utwórz menedżer kolejek i uruchom komendę **Definiowanie przykładowej konfiguracji** , aby zdefiniować przykładowe zasoby telemetryczne dla menedżera kolejek. Opublikuj komunikat przy użyciu programu IBM MQ Explorera zasubskrybuj go za pomocą programu narzędziowego klienta MQTT .

O tym zadaniu

Podczas konfigurowania przykładowych zasobów pomiarowych przy użyciu opcji **Zdefiniuj przykładową konfigurację** kreator ustawia uprawnienia identyfikatora użytkownika gościa. Należy dokładnie rozważyć, czy identyfikator użytkownika gościa ma być autoryzowany w ten sposób. `guest` w systemie Windowsi `nobody` w systemie Linux mają uprawnienie do publikowania i subskrybowania elementu głównego drzewa tematów oraz do umieszczania komunikatów w katalogu `SYSTEM.MQTT.TRANSMIT.QUEUE`.

Kreator ustawia również domyślną kolejkę transmisji na wartość `SYSTEM.MQTT.TRANSMIT.QUEUE`, co może kolidować z aplikacjami działającymi w istniejącym menedżerze kolejek. Możliwe jest, ale pracochłonne, skonfigurowanie telemetrii i nieużywanie domyślnej kolejki transmisji. W tym celu należy wykonać następującą czynność: [“Korzystanie z aliasu menedżera kolejek”](#) na stronie 127. W tym zadaniu tworzony jest menedżer kolejek, aby uniknąć możliwości ingerencji w istniejącą domyślną kolejkę transmisji.

Procedura

1. Za pomocą programu IBM MQ Explorer utwórz i uruchom nowy menedżer kolejek.
 - a) Kliknij prawym przyciskiem myszy folder `Queue Managers` > **Nowy** > **Menedżer kolejek** Wpisz nazwę menedżera kolejek > **Zakończ**.
Utwórz nazwę menedżera kolejek, na przykład `MQTTQMGR`.
2. Utwórz i uruchom usługę telemetryczną (MQXR) oraz utwórz przykładowy kanał telemetryczny.
 - a) Otwórz folder `Queue Managers\QmgrName\Telemetry` .
 - b) Kliknij przycisk **Zdefiniuj przykładową konfigurację ...** > **Zakończ** .
Pozostaw zaznaczone pole wyboru **Uruchom program narzędziowy klienta MQTT** .
3. Utwórz subskrypcję programu MQTT Example za pomocą programu narzędziowego klienta MQTT .
 - a) Kliknij przycisk **Połącz**.

Historia klienta rejestruje zdarzenie `Connected` .

b) Wpisz MQTT Example w polu **Subskrypcja \ Temat > Subskrybuj**.

Historia klienta rejestruje zdarzenie Subscribed .

4. Utwórz plik MQTTExampleTopic w katalogu IBM MQ.

a) Kliknij prawym przyciskiem myszy folder Queue Managers\QmgrName\Topics w programie **MQ Explorer> Nowy > Temat**.

b) Wpisz MQTTExampleTopic w polu **Nazwa > Dalej**.

c) Wpisz MQTT Example w polu **Łańcuch tematu > Zakończ**.

d) Kliknij przycisk **OK** , aby zamknąć okno potwierdzenia.

5. Opublikuj plik Hello World! w temacie MQTT Example przy użyciu programu IBM MQ Explorer.

a) Kliknij folder Queue Managers\QmgrName\Topics w katalogu IBM MQ Explorer.

b) Kliknij prawym przyciskiem myszy opcję MQTTExampleTopic > **Testuj publikację ...**

c) Wpisz Hello World! w polu **Dane komunikatu > Publikuj komunikat > Przejdź do okna programu narzędziowego klienta MQTT** .

Historia klienta rejestruje zdarzenie Received .

Uruchom zadanie z uruchomioną usługą telemetryczną (MQXR)

Utwórz kanał pomiarowy i temat. Autoryzuj użytkownika do korzystania z tematu i kolejki wyjściowej telemetrycznej. Opublikuj komunikat przy użyciu programu IBM MQ Explorer i zasubskrybuj go za pomocą programu narzędziowego klienta MQTT .

Zanim rozpocznie

W tej wersji zadania jest zdefiniowany i uruchomiony menedżer kolejek QmgrName. Usługa telemetryczna (MQXR) jest zdefiniowana i uruchomiona. Usługa telemetryczna (MQXR) mogła zostać utworzona ręcznie lub przez uruchomienie kreatora **Definiowanie przykładowej konfiguracji** .

O tym zadaniu

W ramach tego zadania należy skonfigurować istniejący menedżer kolejek w taki sposób, aby wysłał publikację do programu narzędziowego klienta MQTT .

Krok "1" na stronie 126 zadania ustawia domyślną kolejkę transmisji na wartość SYSTEM.MQTT.TRANSMIT.QUEUE, co może kolidować z aplikacjami działającymi w istniejącym menedżerze kolejek. Możliwe jest, ale pracochłonne, skonfigurowanie telemetry i nieużywanie domyślnej kolejki transmisji. W tym celu należy wykonać następującą czynność: "Korzystanie z aliasu menedżera kolejek" na stronie 127.

Procedura

1. Ustaw SYSTEM.MQTT.TRANSMIT.QUEUE jako domyślną kolejkę wyjściową.

a) Kliknij prawym przyciskiem myszy opcję Queue Managers\QmgrName folder > **Właściwości ...**

b) W nawigatorze kliknij opcję **Komunikacja** .

c) Kliknij przycisk **Wybierz > Wybierz SYSTEM.MQTT.TRANSMIT.QUEUE > OK > OK**.

2. Utwórz kanał pomiarowy MQTTExampleChannel , aby połączyć program narzędziowy klienta MQTT z produktem IBM MQ, i uruchom program narzędziowy klienta MQTT .

a) Kliknij prawym przyciskiem myszy folder Queue Managers\QmgrName \Telemetry\Channels w programie **MQ Explorer> Nowy > Kanał pomiarowy ...**

b) W polu **Nazwa kanału** wpisz MQTTExampleChannel > **Dalej > Dalej**.

c) Zmień **Stały identyfikator użytkownika** na panelu autoryzacji klienta na identyfikator użytkownika, który będzie publikować i subskrybować MQTTExample > **Dalej**.

- d) Pozostaw zaznaczone pole wyboru **Launch Client Utility** (Uruchom program narzędziowy klienta) > **Finish**(Zakończ).
3. Utwórz subskrypcję programu MQTT Example za pomocą programu narzędziowego klienta MQTT .
- Kliknij przycisk **Połącz**.
Historia klienta rejestruje zdarzenie Connected .
 - Wpisz MQTT Example w polu **Subskrypcja \ Temat** > **Subskrybuj**.
Historia klienta rejestruje zdarzenie Subscribed .
4. Utwórz plik MQTTExampleTopic w katalogu IBM MQ.
- Kliknij prawym przyciskiem myszy folder Queue Managers\QmgrName\Topics w programie **MQ Explorer**> **Nowy** > **Temat**.
 - Wpisz MQTTExampleTopic w polu **Nazwa** > **Dalej**.
 - Wpisz MQTT Example w polu **Łańcuch tematu** > **Zakończ**.
 - Kliknij przycisk **OK** , aby zamknąć okno potwierdzenia.
5. Aby użytkownik, który nie należy do grupy mqm , mógł publikować i subskrybować temat MQTTExample , wykonaj następujące czynności:
- Autoryzuj użytkownika do publikowania i subskrybowania tematu MQTTExampleTopic:

```
setmqaut -m qMgrName -t topic -n MQTTExampleTopic -p User ID -all +pub +sub
```

- Autoryzuj użytkownika, aby umieścić komunikat w pliku SYSTEM.MQTT.TRANSMIT.QUEUE:

```
setmqaut -m qMgrName -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p User ID -all +put
```

6. Opublikuj plik Hello World! w temacie MQTT Example przy użyciu programu IBM MQ Explorer.
- Kliknij folder Queue Managers\QmgrName\Topics w katalogu IBM MQ Explorer.
 - Kliknij prawym przyciskiem myszy opcję MQTTExampleTopic > **Testuj publikację ...**
 - Wpisz Hello World! w polu **Dane komunikatu** > **Publikuj komunikat** > Przejdź do okna programu narzędziowego klienta MQTT .

Historia klienta rejestruje zdarzenie Received .

Korzystanie z aliasu menedżera kolejek

Publikowanie komunikatu w programie narzędziowym klienta MQTT za pomocą komendy IBM MQ Explorer bez ustawiania domyślnej kolejki transmisji na wartość SYSTEM.MQTT.TRANSMIT.QUEUE.

Zadanie jest kontynuacją poprzedniego zadania i używa aliasu menedżera kolejek, aby uniknąć ustawienia domyślnej kolejki transmisji na wartość SYSTEM.MQTT.TRANSMIT.QUEUE.

Zanim rozpoczniesz

Wykonaj zadanie [“Uruchom zadanie bez zdefiniowanej usługi telemetrycznej \(MQXR\)”](#) na stronie 125 lub zadanie [“Uruchom zadanie z uruchomioną usługą telemetryczną \(MQXR\)”](#) na stronie 126.

O tym zadaniu

Gdy klient MQTT tworzy subskrypcję, program IBM MQ wysyła odpowiedź, używając identyfikatora ClientIdentifier jako nazwy zdalnego menedżera kolejek. W tym zadaniu używany jest identyfikator klienta (ClientIdentifier, MyClient).

Jeśli nie ma kolejki transmisji lub aliasu menedżera kolejek o nazwie MyClient, odpowiedź jest umieszczana w domyślnej kolejce transmisji. Ustawienie domyślnej kolejki transmisji na SYSTEM.MQTT.TRANSMIT.QUEUE powoduje, że klient MQTT otrzymuje odpowiedź.

Nie należy ustawiać domyślnej kolejki transmisji na wartość `SYSTEM.MQTT.TRANSMIT.QUEUE`, jeśli używane są aliasy menedżera kolejek. Należy skonfigurować alias menedżera kolejek dla każdego `ClientIdentifier`. Zwykle istnieje zbyt wiele klientów, aby można było używać aliasów menedżera kolejek. Często wartość `ClientIdentifier` jest nieprzewidywalna, co uniemożliwia skonfigurowanie telemetrii w ten sposób.

Jednak w pewnych okolicznościach konieczne może być skonfigurowanie domyślnej kolejki transmisji na wartość inną niż `SYSTEM.MQTT.TRANSMIT.QUEUE`. Kroki opisane w sekcji *Procedura* umożliwiają skonfigurowanie aliasu menedżera kolejek zamiast ustawiania domyślnej kolejki transmisji na wartość `SYSTEM.MQTT.TRANSMIT.QUEUE`.

Procedura

- Usuń `SYSTEM.MQTT.TRANSMIT.QUEUE` jako domyślną kolejkę wyjściową.
 - Kliknij prawym przyciskiem myszy opcję `Queue Managers\QmgrName folder > Właściwości ...`
 - W nawigatorze kliknij opcję **Komunikacja**.
 - Usuń łańcuch `SYSTEM.MQTT.TRANSMIT.QUEUE` z pola **Domyślna kolejka transmisji > OK**.
- Sprawdź, czy nie można już utworzyć subskrypcji za pomocą programu narzędziowego klienta MQTT:
 - Kliknij przycisk **Połącz**.
Historia klienta rejestruje zdarzenie `Connected`.
 - Wpisz `MQTT Example` w polu **Subskrypcja \ Temat > Subskrybuj**.
Historia klienta rejestruje zdarzenie `Subscribe failed iConnection lost`.
- Utwórz alias menedżera kolejek dla `ClientIdentifier`, `MyClient`.
 - Kliknij prawym przyciskiem myszy folder `Queue Managers\QmgrName\Queues > Nowy > Definicja kolejki zdalnej`.
 - Nadaj definicji nazwę `MyClient > Dalej`.
 - W polu **Zdalny menedżer kolejek** wpisz `MyClient`.
 - W polu **Kolejka transmisji** wpisz `SYSTEM.MQTT.TRANSMIT.QUEUE > Zakończ`.
- Ponownie nawiąż połączenie z programem narzędziowym klienta MQTT.
 - Sprawdź, czy właściwość **Identyfikator klienta** jest ustawiona na wartość `MyClient`.
 - Połączenie**
Historia klienta rejestruje zdarzenie `Connected`.
- Utwórz subskrypcję programu `MQTT Example` za pomocą programu narzędziowego klienta MQTT.
 - Kliknij przycisk **Połącz**.
Historia klienta rejestruje zdarzenie `Connected`.
 - Wpisz `MQTT Example` w polu **Subskrypcja \ Temat > Subskrybuj**.
Historia klienta rejestruje zdarzenie `Subscribed`.
- Opublikuj plik `Hello World!` w temacie `MQTT Example` przy użyciu programu `IBM MQ Explorer`.
 - Kliknij folder `Queue Managers\QmgrName\Topics` w katalogu `IBM MQ Explorer`.
 - Kliknij prawym przyciskiem myszy opcję `MQTTExampleTopic > Testuj publikację ...`
 - Wpisz `Hello World!` w polu **Dane komunikatu > Publikuj komunikat > Przejdź do okna programu narzędziowego klienta MQTT**.

Historia klienta rejestruje zdarzenie `Received`.

pomocą programu IBM MQ Explorer

Wyślij komunikat do programu narzędziowego klienta MQTT, umieszczając komunikat w kolejce IBM MQ za pomocą komendy IBM MQ Explorer. Zadanie przedstawia sposób konfigurowania definicji kolejki zdalnej w celu wysłania komunikatu bezpośrednio do klienta MQTT.

Zanim rozpocznieš

Wykonaj czynnořć [“Publikowanie komunikatu w programie narzędziowym klienta MQTT z serwisu IBM MQ Explorer”](#) na stronie 124. Pozostaw połączony program narzędziowy klienta MQTT.

O tym zadaniu

Zadanie demonstruje wysyłanie komunikatu do klienta MQTT przy użyciu kolejki, a nie publikowania w temacie. W kliencie nie jest tworzona subskrypcja. Krok [“2”](#) na stronie 129 zadania pokazuje, że poprzednia subskrypcja została usunięta.

Procedura

1. Usuń wszystkie istniejące subskrypcje, rozłączając i ponownie nawiązując połączenie z programem narzędziowym klienta MQTT.

Subskrypcja jest usuwana, ponieważ, o ile nie zostaną zmienione wartości domyślne, program narzędziowy klienta MQTT łączy się z czystą sesją; patrz sekcja [Czyste sesje](#).

Aby ułatwić wykonanie zadania, należy wpisać własny identyfikator `ClientIdentifier`, a nie używać wygenerowanego `ClientIdentifier` utworzonego przez program narzędziowy klienta MQTT.

- a) Kliknij opcję **Rozłącz**, aby odłączyć program narzędziowy klienta MQTT od kanału pomiarowego.

Historia klienta rejestruje zdarzenie `Disconnected`

- b) Zmień wartość w polu **Client Identifier** (Identyfikator klienta) na `MyClient`.
- c) Kliknij przycisk **Połącz**.

Historia klienta rejestruje zdarzenie `Connected`

2. Sprawdź, czy program narzędziowy klienta MQTT nie odbiera już publikacji dla `MQTTExampleTopic`.

- a) Kliknij folder `Queue Managers\QmgrName\Topics` w katalogu IBM MQ Explorer.
- b) Kliknij prawym przyciskiem myszy opcję `MQTTExampleTopic` > **Testuj publikację ...**
- c) Wpisz `Hello World!` w polu **Dane komunikatu** > **Publikuj komunikat** > Przejdź do okna programu narzędziowego klienta MQTT.

Źadne zdarzenie nie jest rejestrowane w **historii klienta**.

3. Utwórz definicję kolejki zdalnej dla klienta.

W definicji kolejki zdalnej jako nazwę menedżera kolejek zdalnych ustaw `ClientIdentifier`, `MyClient`. Jako nazwy kolejki zdalnej należy użyć dowolnej nazwy. Nazwa kolejki zdalnej jest przekazywana do klienta MQTT jako nazwa tematu.

- a) Kliknij prawym przyciskiem myszy folder `Queue Managers\QmgrName\Queues` > **Nowy** > **Definicja kolejki zdalnej**.
- b) Nadaj definicji nazwę `MyClientRemoteQueue` > **Dalej**.
- c) W polu **Kolejka zdalna** wpisz `MQTTExampleQueue`.
- d) W polu **Zdalny menedżer kolejek** wpisz `MyClient`.
- e) W polu **Kolejka transmisji** wpisz `SYSTEM.MQTT.TRANSMIT.QUEUE` > **Zakończ**.

4. Umieść komunikat testowy w pliku `MyClientRemoteQueue`.

- a) Kliknij prawym przyciskiem myszy opcję `MyClientRemoteQueue` > **Umieść komunikat testowy ...**

b) Wpisz `Hello queue!` w polu Dane komunikatu > **Umieść komunikat** > **Zamknij**

Historia klienta rejestruje zdarzenie `Received`.

5. Usuń `SYSTEM.MQTT.TRANSMIT.QUEUE` jako domyślną kolejkę wyjściową.

a) Kliknij prawym przyciskiem myszy opcję `Queue Managers\QmgrName folder` > **Właściwości ...**

b) W nawigаторze kliknij opcję **Komunikacja**.

c) Usuń łańcuch `SYSTEM.MQTT.TRANSMIT.QUEUE` z pola **Domyślna kolejka transmisji** > **OK**.

6. Wykonaj ponownie krok "4" na stronie 129.

`MyClientRemoteQueue` to definicja kolejki zdalnej, która jawnie określa kolejkę transmisji. Nie ma potrzeby definiowania domyślnej kolejki transmisji w celu wystania komunikatu do `MyClient`.

Co dalej

Jeśli domyślna kolejka transmisji nie jest już ustawiona na wartość `SYSTEM.MQTT.TRANSMIT.QUEUE`, program narzędziowy klienta MQTT nie może utworzyć nowej subskrypcji, dopóki nie zostanie zdefiniowany alias menedżera kolejek dla `ClientIdentifier`, `MyClient`. Odtwórz domyślną kolejkę transmisji do `SYSTEM.MQTT.TRANSMIT.QUEUE`.

Windows Linux AIX Publikowanie komunikatu w konkretnym kliencie MQTT v3

Opublikuj komunikat z jednego klienta MQTT v3 do innego, używając `ClientIdentifier` jako nazwy tematu i IBM MQ jako brokera publikowania/subskrybowania.

Zanim rozpocznie

Wykonaj czynność "Publikowanie komunikatu w programie narzędziowym klienta MQTT z serwisu IBM MQ Explorer" na stronie 124. Pozostaw połączony program narzędziowy klienta MQTT.

O tym zadaniu

Zadanie demonstruje dwie rzeczy:

1. Subskrybowanie tematu w jednym kliencie MQTT i otrzymywanie publikacji z innego klienta MQTT.
2. Konfigurowanie subskrypcji "punkt z punktem" za pomocą łańcucha tematu `ClientIdentifier`.

Procedura

1. Usuń wszystkie istniejące subskrypcje, rozłączając i ponownie nawiązując połączenie z programem narzędziowym klienta MQTT.

Subskrypcja jest usuwana, ponieważ, o ile nie zostaną zmienione wartości domyślne, program narzędziowy klienta MQTT łączy się z czystą sesją; patrz sekcja [Czyste sesje](#).

Aby ułatwić wykonanie zadania, należy wpisać własny identyfikator `ClientIdentifier`, a nie używać wygenerowanego `ClientIdentifier` utworzonego przez program narzędziowy klienta MQTT.

a) Kliknij opcję **Rozłącz**, aby odłączyć program narzędziowy klienta MQTT od kanału pomiarowego.

Historia klienta rejestruje zdarzenie `Disconnected`

b) Zmień wartość w polu **Client Identifier** (Identyfikator klienta) na `MyClient`.

c) Kliknij przycisk **Połącz**.

Historia klienta rejestruje zdarzenie `Connected`

2. Utwórz subskrypcję tematu, `MyClient`

`MyClient` to `ClientIdentifier` tego klienta.

a) Wpisz MyClient w polu **Subskrypcja \ Temat > Subskrybuj**.

Historia klienta rejestruje zdarzenie Subscribed .

3. Uruchom inny program narzędziowy klienta MQTT .

a) Otwórz folder Queue Managers\QmgrName\Telemetry\channels .

b) Kliknij prawym przyciskiem myszy kanał **PlainText > Uruchom program narzędziowy klienta MQTT ...**

c) Kliknij przycisk **Połącz**.

Historia klienta rejestruje zdarzenie Connected

4. Opublikuj Hello MyClient! w temacie MyClient.

a) Skopiuj temat subskrypcji MyClientz programu narzędziowego klienta MQTT działającego z identyfikatorem ClientIdentifier(MyClient).

b) Wklej plik MyClient do pola **Publication \ Topic** w każdej instancji programu narzędziowego klienta MQTT .

c) Wpisz Hello MyClient! w polu **Publikacja \ komunikat** .

d) Kliknij opcję **Publikuj** w obu instancjach.

Wyniki

Historia klienta w programie narzędziowym klienta MQTT z identyfikatorem ClientIdentifier, MyClient, rejestruje dwa zdarzenia **Odebrane** i jedno zdarzenie **Opublikowane** . Druga instancja programu narzędziowego klienta MQTT rejestruje jedno zdarzenie **Opublikowane** .

Jeśli wyświetlane jest tylko jedno zdarzenie **Odebrano** , sprawdź następujące możliwe przyczyny:

1. Czy domyślna kolejka transmisji dla menedżera kolejek jest ustawiona na wartość SYSTEM.MQTT.TRANSMIT.QUEUE ?
2. Czy podczas wykonywania innych ćwiczeń zostały utworzone aliasy menedżera kolejek lub definicje kolejek zdalnych odwołujące się do MyClient ? W przypadku problemów z konfiguracją należy usunąć wszystkie zasoby, które odwołują się do produktu MyClient, takie jak aliasy menedżera kolejek lub kolejki transmisji. Rozłącz programy narzędziowe klienta, zatrzymaj i zrestartuj usługę telemetryczną (MQXR).

Windows

Linux

AIX

Wysyłanie komunikatu do aplikacji IBM MQ z klienta MQTT

Aplikacja IBM MQ może odbierać komunikat od klienta MQTT v3 , subskrybując temat. Klient MQTT nawiązuje połączenie z produktem IBM MQ przy użyciu kanału pomiarowego i wysyła komunikat do aplikacji IBM MQ , publikując go w tym samym temacie.

Wykonaj czynność [“Publikowanie komunikatu w produkcie IBM MQ z poziomu klienta MQTT”](#) na stronie 131, aby dowiedzieć się, jak wysłać publikację z klienta MQTT do subskrypcji zdefiniowanej w produkcie IBM MQ.

Jeśli temat znajduje się w klastrze lub jest dystrybuowany przy użyciu hierarchii publikowania/subskrypcji, subskrypcja może znajdować się w innym menedżerze kolejek niż menedżer kolejek, z którym jest połączony klient MQTT .

Windows

Linux

AIX

Publikowanie komunikatu w produkcie IBM MQ z poziomu klienta MQTT

Tworzenie subskrypcji tematu przy użyciu programu IBM MQ Explorer i publikowanie w temacie przy użyciu programu narzędziowego klienta MQTT .

Zanim rozpoczniesz

Wykonaj czynność “Publikowanie komunikatu w programie narzędziowym klienta MQTT z serwisu IBM MQ Explorer” na stronie 124. Pozostaw połączony program narzędziowy klienta MQTT .

O tym zadaniu

Zadanie demonstruje publikowanie komunikatu przy użyciu klienta MQTT i odbieranie publikacji przy użyciu niezarządzanej subskrypcji trwałej utworzonej przy użyciu produktu IBM MQ Explorer.

Procedura

1. Utwórz trwałą subskrypcję łańcucha tematu MQTT Example.
Wykonaj następujące kroki, aby utworzyć kolejkę i subskrypcję za pomocą programu IBM MQ Explorer.
 - a) Kliknij prawym przyciskiem myszy folder Queue Managers*QmgrName*\Queues w obszarze IBM MQ Explorer > **Nowa** > **Kolejka lokalna ...**.
 - b) Wpisz MQTTExampleQueue jako nazwę kolejki > **Zakończ**.
 - c) Kliknij prawym przyciskiem myszy folder Queue Managers*QmgrName*\Subscriptions w obszarze IBM MQ Explorer > **Nowa** > **Subskrypcja ...**.
 - d) Wpisz MQTTExampleSubscription jako nazwę kolejki > **Dalej**.
 - e) Kliknij przycisk **Wybierz** > MQTTExampleTopic > **OK**.

Temat MQTTExampleTopic został już utworzony w kroku “4” na stronie 126 procedury “Publikowanie komunikatu w programie narzędziowym klienta MQTT z serwisu IBM MQ Explorer” na stronie 124.

- f) Wpisz MQTTExampleQueue jako nazwę miejsca docelowego > **Zakończ**.
2. Jako krok opcjonalny, ustaw kolejkę do użycia przez innego użytkownika, bez uprawnień mqm .
Jeśli konfigurujesz użytkowników z uprawnieniami mniejszymi niż mqm, musisz nadać uprawnienia put i get do MQTTExampleQueue. Dostęp do tematu i kolejki transmisji został skonfigurowany w sekcji “Publikowanie komunikatu w programie narzędziowym klienta MQTT z serwisu IBM MQ Explorer” na stronie 124.
 - a) Autoryzuj użytkownika do umieszczania i pobierania z kolejki MQTTExampleQueue:

```
setmqaut -m qMgrName -t queue -n MQTTExampleQueue -p User ID -all +put +get
```

3. Opublikuj plik Hello IBM MQ! w temacie MQTT Example przy użyciu programu narzędziowego klienta MQTT .
Jeśli program narzędziowy klienta MQTT nie jest połączony, kliknij prawym przyciskiem myszy kanał **PlainText** > **Uruchom program narzędziowy klienta MQTT ...** > **Połącz**.
 - a) W polu **Publikacja \ Temat** wpisz MQTT Example .
 - b) Wpisz wartość Hello IBM MQ! w polu **Publikacja \ Komunikat** > **Publikuj**.
4. Otwórz folder Queue Managers*QmgrName*\Queues i znajdź plik MQTTExampleQueue.
Pole **Bieżące zapełnienie kolejki** ma wartość 1
5. Kliknij prawym przyciskiem myszy opcję MQTTExampleQueue > **Przeglądaj komunikaty ...** i zapoznaj się z publikacją.

Windows

Linux

AIX

Aplikacje publikowania/subskrybowania MQTT

Publikowanie/subskrybowanie oparte na tematach umożliwia pisanie aplikacji MQTT .

Gdy klient MQTT jest połączony, publikacje przepływają między klientem a serwerem w dowolnym kierunku. Publikacje są wysyłane od klienta, gdy informacje są publikowane na kliencie. Publikacje są odbierane przez klienta, gdy komunikat jest publikowany w temacie zgodnym z subskrypcją utworzoną przez klienta.

Broker publikowania/subskrypcji produktu IBM MQ zarządza tematami i subskrypcjami utworzonymi przez klienty MQTT. Tematy utworzone przez klienty MQTT współużytkują ten sam obszar tematu co tematy utworzone przez aplikacje IBM MQ.

Publikacje zgodne z łańcuchem tematu w subskrypcji klienta MQTT są umieszczane w katalogu SYSTEM.MQTT.TRANSMIT.QUEUE, a nazwa zdalnego menedżera kolejek jest ustawiona na wartość ClientIdentifier klienta. Usługa telemetryczna (MQXR) przekazuje publikacje do klienta, który utworzył subskrypcję. Używa on ClientIdentifier, który został ustawiony jako nazwa zdalnego menedżera kolejek w celu zidentyfikowania klienta.

Zwykle SYSTEM.MQTT.TRANSMIT.QUEUE musi być zdefiniowany jako domyślna kolejka transmisji. Możliwe, ale uciążliwe, jest skonfigurowanie produktu MQTT tak, aby nie używał domyślnej kolejki transmisji. Więcej informacji na ten temat zawiera sekcja [Konfigurowanie rozproszonego kolejkowania w celu wysyłania komunikatów do klientów MQTT](#).

Klient MQTT może utworzyć sesję trwałą; patrz sekcja [“Sesje bezstanowe i stanowe MQTT”](#) na stronie 136. Subskrypcje utworzone w sesji trwałej są trwałe. Publikacje dostarczane dla klienta z trwałą sesją są przechowywane w katalogu SYSTEM.MQTT.TRANSMIT.QUEUEi przekazywane do klienta po ponownym nawiązaniu połączenia.

Klient MQTT może również publikować i subskrybować zachowane publikacje; patrz sekcja [Zachowane publikacje i klienty MQTT](#). Subskrybent zachowanego tematu publikacji otrzymuje najnowszą publikację tematu. Subskrybent otrzymuje zachowaną publikację podczas tworzenia subskrypcji lub po ponownym nawiązaniu połączenia z wcześniejszą sesją.

Windows

Linux

AIX

Aplikacje telemetryczne

Tworzenie aplikacji telemetrycznych przy użyciu przepływów komunikatów produktu IBM MQ lub IBM Integration Bus.

Do programowania aplikacji telemetrycznych w produkcie IBM MQ należy używać interfejsów programistycznych JMS, MQI lub innych interfejsów programistycznych IBM MQ.

Usługa telemetryczna (MQXR) przekształca komunikaty produktu MQTT v3 i komunikaty produktu IBM MQ. Tworzy on subskrypcje i publikacje w imieniu klientów MQTT oraz przekazuje publikacje do klientów MQTT. Publikacja jest ładunkiem komunikatu MQTT v3. Ładunek składa się z nagłówek komunikatów i tablicy bajtów w formacie jms-bytes. Serwer pomiarowy odwzorowuje nagłówki między komunikatem MQTT v3 i komunikatem IBM MQ (patrz sekcja [“Integracja programu MQ Telemetry z menedżerami kolejek”](#) na stronie 133).

Węzły Publication, MQInput i JMSInput służą do wysyłania i odbierania publikacji między klientami IBM Integration Bus i MQTT.

Za pomocą przepływów komunikatów można zintegrować dane telemetryczne z serwisami WWW przy użyciu protokołu HTTP oraz z innymi aplikacjami przy użyciu produktu IBM MQ i adapterów WebSphere.

Windows

Linux

AIX

Integracja programu MQ Telemetry

z menedżerami kolejek

Klient MQTT jest zintegrowany z produktem IBM MQ jako aplikacja publikowania/subskrybowania. Może publikować lub subskrybować tematy w produkcie IBM MQ, tworzyć nowe tematy lub korzystać z istniejących tematów. Odbiera on publikacje z serwisu IBM MQ w wyniku działania klientów MQTT, w tym własnych, lub innych aplikacji IBM MQ, które publikują w tematach swoich subskrypcji. Reguły są stosowane do decydowania o atrybutach publikacji.

Wiele atrybutów powiązanych z tematami, publikacjami, subskrypcjami i komunikatami udostępnianymi przez produkt IBM MQ nie jest obsługiwanych. [“Klient MQTT do IBM MQ brokera publikowania/subskrypcji”](#) na stronie 134 i [“IBM MQ do klienta MQTT”](#) na stronie 135 opisują sposób ustawiania atrybutów publikacji. Ustawienia zależą od tego, czy publikacja ma zostać opublikowana w brokerze publikowania/subskrypcji produktu IBM MQ.

Tematy publikowania/subskrypcji produktu IBM MQ są powiązane z obiektami tematów administracyjnych. Tematy utworzone przez klienty MQTT nie różnią się. Gdy klient MQTT tworzy łańcuch tematu dla publikacji, broker publikowania/subskrypcji produktu IBM MQ wiąże go z obiektem tematu administracyjnego. Broker odwzorowuje łańcuch tematu w publikacji na najbliższy nadrzędny obiekt tematu administracyjnego. Odwzorowanie jest takie samo jak w przypadku aplikacji IBM MQ . Jeśli nie ma tematu utworzonego przez użytkownika, temat publikacji jest odwzorowywany na temat SYSTEM.BASE.TOPIC. Atrybuty zastosowane do publikacji pochodzą z obiektu tematu.

Gdy aplikacja IBM MQ lub administrator tworzy subskrypcję, jej nazwa jest nadawana. Wyświetlanie listy subskrypcji przy użyciu programu IBM MQ Explorer lub komend `runmqsc` lub PCF. Zostaną nazwane wszystkie subskrypcje klienta MQTT. Otrzymują one nazwę w postaci: *ClientIdentifier:Topic name*

Klient MQTT do IBM MQ brokera publikowania/subskrypcji

Klient MQTT wysłał publikację do IBM MQ. Usługa telemetryczna (MQXR) przekształca publikację w komunikat IBM MQ . Komunikat IBM MQ składa się z trzech części:

1. MQMD
2. RFH2
3. Komunikat

Właściwości MQMD są ustawiane na wartości domyślne, z wyjątkiem sytuacji opisanych w sekcji [Tabela 9 na stronie 134](#).

Tabela 9. Ustawienia właściwości MQMD		
Pole MQMD	Typ	Wartość
Format	MQCHAR8	MQFMT_RF_HEADER_2
UserIdentifier	MQCHAR12	Ustaw jedną z następujących wartości: MqttClient.ClientIdentifier MqttConnectOptions.UserName Identyfikator użytkownika ustawiony przez administratora produktu IBM MQ dla kanału pomiarowego.
Priority	MQLONG	MQPRI_PRIORITY_AS_Q_DEF (Wartość domyślna dla systemu IBM MQ, która jest inna niż dla systemu JMS , który ma wartość domyślną 4).
Persistence	MQLONG	QoS=0→MQPER_NOT_PERSISTENT QoS=1→MQPER_PERSISTENT QoS=2→MQPER_PERSISTENT

Nagłówek RFH2 nie zawiera folderu <msd> definiującego typ komunikatu JMS . Usługa telemetryczna (MQXR) tworzy komunikat IBM MQ jako domyślny komunikat JMS . Domyślnym typem komunikatu JMS jest komunikat `jms-bytes` . Aplikacja może uzyskać dostęp do dodatkowych informacji nagłówka jako właściwości komunikatu; patrz sekcja [Właściwości komunikatu](#).

RFH2 wartości są ustawione w sposób przedstawiony na rysunku ([Tabela 10 na stronie 134](#)). Właściwość Format jest ustawiana w stałym nagłówku RFH2 , a inne wartości są ustawiane w folderach RFH2 .

Tabela 10. Ustawienia dla właściwości RFH2		
Właściwość RFH2	Typ/folder	Nagłówek
Formatowanie	MQCHAR8	MQFMT_NONE

Tabela 10. Ustawienia dla właściwości RFH2 (kontynuacja)

Właściwość RFH2	Typ/folder	Nagłówek
ClientIdentifier	mqttd/clientId	Skopiuj plik MqttClient.ClientIdentifier o długości 1...23 bajtów.
QoS	mqttd/qos	Skopiuj komunikat QoS z przychodzącego komunikatu MQTT .
ID komunikatu	mqttd/msgid	Skopiuj ID komunikatu z przychodzącego komunikatu MQTT , jeśli QoS ma wartość 1 lub 2.
MQIsRetained	mmps/Ret	Określa, czy oryginalna publikacja MQTT została wysłana z ustawionym zestawem właściwości RETAIN i czy komunikat został odebrany jako zachowana publikacja.
MQTopicString	mmps/Top	Temat, w którym został opublikowany komunikat MQTT .

Ładunek w publikacji MQTT jest odwzorowywany na treść komunikatu IBM MQ :

Tabela 11. Sposób odwzorowania ładunku w publikacji MQTT na treść komunikatu IBM MQ

Treść komunikatu	Typ	Treść komunikatu IBM MQ
Buforuj	MQBYTE <i>n</i>	Kopia bajtów z przychodzącego komunikatu MQTT . Długość może wynosić zero.

IBM MQ do klienta MQTT

Klient zasubskrybował temat publikacji. Aplikacja IBM MQ opublikowała temat, co spowodowało wysłanie publikacji do subskrybenta MQTT przez broker publikowania/subskrypcji produktu IBM MQ . Alternatywnie aplikacja IBM MQ wysłała niechciany komunikat bezpośrednio do klienta MQTT . Tabela 12 na stronie 135 opisuje sposób ustawiania statycznych nagłówek komunikatów w komunikacie wysydanym do klienta MQTT . Wszystkie inne dane w nagłówku komunikatu IBM MQ lub inne nagłówki są usuwane. Dane komunikatu w komunikacie IBM MQ są wysyłane jako ładunek komunikatu w komunikacie MQTT bez żadnych zmian. Komunikat MQTT jest wysyłany do klienta MQTT przez usługę telemetryczną (MQXR).

Tabela 12. Sposób ustawiania statycznych nagłówek komunikatów w komunikacie IBM MQ wysydanym do klienta MQTT

MQTT field (pole)	Typ	Wartość
DUP	boolean (boolowskie)	Należy ustawić wartość QoS = 1 lub 2, jeśli komunikat został wysłany do tego klienta w poprzedniej transmisji, a komunikat nie został potwierdzony po pewnym czasie.

Tabela 12. Sposób ustawiania stałych nagłówków komunikatów w komunikacie IBM MQ wysyłanym do klienta MQTT (kontynuacja)

MQTT field (pole)	Typ	Wartość
QoS	int	<p>Sposób ustawiania wartości parametru QoS w publikacji wychodzącej z brokera publikowania/subskrypcji w produkcie IBM MQ zależy od publikacji przychodzącej. Zależy to od tego, czy publikacja przychodząca została wysłana z klienta MQTT, czy z aplikacji IBM MQ.</p> <p>MQTT Niższa wartość parametru QoS w przychodzącej publikacji oraz w polu QoS żądanym przez subskrybenta.</p> <p>IBM MQ Niższa wartość parametru QoS pochodzącego z publikacji przychodzącej: MQPER_NOT_PERSISTENT→QoS=0 MQPER_PERSISTENT→QoS=2</p> <p>i QoS żądanej przez subskrybenta. Jeśli komunikat jest wysyłany do klienta bez subskrypcji, wartość QoS jest domyślnie ustawiona na 2. Klient może zmienić tę wartość, subskrybując usługę DEFAULT. QoS za pomocą innej usługi QoS.</p>
RETAIN	boolean (boolowskie)	Ustaw, jeśli publikacja przychodząca ma ustawioną zachowaną właściwość.

Tabela 13 na stronie 136 opisuje sposób ustawiania nagłówków komunikatów zmiennych w komunikacie MQTT wysyłanym do klienta MQTT.

Tabela 13. Sposób ustawiania właściwości nagłówka zmiennej MQTT w komunikacie MQTT wysyłanym do klienta MQTT

MQTT field (pole)	Typ	Wartość
Topic name	łańcuch	łańcuch tematu, z którym został opublikowany komunikat.
Message ID	łańcuch	Ostatnie 2 bajty deskryptora MQMD produktu MQMD. MsgId właściwość publikacji umieszczonej w pliku SYSTEM.MQTT.TRANSMIT.QUEUE.
Payload	byte[]	Bezpośrednia kopia bajtów z publikacji przychodzącej do brokera publikowania/subskrybowania. Długość może wynosić zero.

Windows

Linux

AIX

Sesje bezstanowe i stanowe MQTT

Klienci MQTT mogą utworzyć sesję stanową z menedżerem kolejek. Po rozłączeniu się stanowego klienta MQTT menedżer kolejek przechowuje subskrypcje utworzone przez klienta i komunikaty w trakcie przetwarzania. Po ponownym nawiązaniu połączenia przez klient rozstrzyga on komunikat w trakcie przetwarzania. Wysyła on wszystkie komunikaty, które zostały umieszczone w kolejce w celu dostarczenia, i odbiera wszystkie komunikaty opublikowane dla swoich subskrypcji w czasie, gdy był rozłączony.

Gdy klient MQTT nawiązuje połączenie z kanałem pomiarowym, rozpoczyna nową sesję lub wznawia starą sesję. Nowa sesja nie zawiera żadnych zaległych komunikatów, które nie zostały potwierdzone, nie ma subskrypcji ani publikacji oczekujących na dostarczenie. Gdy klient nawiązuje połączenie, określa, czy ma

zostać uruchomiona czysta sesja, czy wznowiona istniejąca sesja. Więcej informacji na ten temat zawiera sekcja [Czyszczenie sesji](#).

Jeśli klient wznowia istniejącą sesję, kontynuuje działanie tak, jakby połączenie nie zostało zerwane. Publikacje oczekujące na dostarczenie są wysyłane do klienta, a wszystkie przesyłania komunikatów, które nie zostały zatwierdzone, są zakończone. Gdy klient w sesji trwałej rozłącza się z usługą telemetryczną (MQXR), wszystkie subskrypcje utworzone przez klienta pozostają. Publikacje dla subskrypcji są wysyłane do klienta po ponownym nawiązaniu połączenia. Jeśli nastąpi ponowne nawiązanie połączenia bez wznowiania starej sesji, publikacje zostaną odrzucone przez usługę telemetryczną (MQXR).

Informacje o stanie sesji są zapisywane przez menedżer kolejek w kolejce SYSTEM.MQTT.PERSISTENT.STATE.

Administrator IBM MQ może rozłączyć i wyczyścić sesję.

Windows

Linux

AIX

Gdy klient MQTT nie jest połączony

Jeśli klient nie jest połączony, menedżer kolejek może nadal odbierać publikacje w swoim imieniu. Są one przekazywane do klienta po ponownym nawiązaniu połączenia. Klient może utworzyć "testament i testament" publikowany przez menedżer kolejek w imieniu klienta, jeśli klient nieoczekiwanie rozłączy się.

Aby otrzymywać powiadomienia o nieoczekiwanym rozłączeniu klienta, można zarejestrować ostatnią publikację testamentu i testamentu; patrz sekcja [Ostatnia publikacja testamentu i testamentu](#). Jest on wysyłany przez usługę telemetryczną (MQXR), jeśli wykryje zerwane połączenie z klientem bez żądania klienta.

Klient może opublikować zachowaną publikację w dowolnym momencie. Więcej informacji na ten temat zawiera sekcja [Zachowane publikacje i klienci MQTT](#). Nowa subskrypcja tematu może zażądać wysłania dowolnej zachowanej publikacji powiązanej z tematem. Jeśli zostanie utworzona ostatnia wola i testament jako zachowana publikacja, można jej użyć do monitorowania statusu klienta.

Na przykład klient publikuje zachowaną publikację, gdy się z nią łączy, reklamując swoją dostępność. Jednocześnie tworzy zachowaną ostatnią wolę i publikację testamentu, która informuje o niedostępności. Ponadto, tuż przed planowanym rozłączeniem, publikuje swoją niedostępność jako zachowaną publikację. Aby sprawdzić, czy klient jest dostępny, należy zasubskrybować temat zachowanej publikacji. Zawsze otrzymujesz jedną z trzech publikacji.

Jeśli klient ma odbierać komunikaty publikowane po rozłączeniu, ponownie połącz go z poprzednią sesją (patrz sekcja ["Sesje bezstanowe i stanowe MQTT" na stronie 136](#)). Jego subskrypcje są aktywne do czasu ich usunięcia lub do czasu, gdy klient utworzy czystą sesję.

Windows

Linux

AIX

Luźne powiązanie między klientami MQTT

i aplikacjami IBM MQ

Przepływ publikacji między klientami MQTT i aplikacjami IBM MQ jest luźno powiązany. Publikacje mogą pochodzić z klienta MQTT lub aplikacji IBM MQ i nie mogą być uporządkowane w określonej kolejności. Publikatory i subskrybenci są luźno powiązane. Współdziałają ze sobą pośrednio poprzez publikacje i subskrypcje. Komunikaty można również wysłać bezpośrednio do klienta MQTT z aplikacji IBM MQ.

Klienci MQTT i aplikacje IBM MQ są luźno powiązane w dwóch zmysłach:

1. Publikatory i subskrybenci są luźno powiązane przez powiązanie publikacji i subskrypcji z tematem. Publikatory i subskrybenci nie mają zwykle informacji o adresie lub tożsamości innego źródła publikacji lub subskrypcji.
2. Klienci MQTT publikują, subskrybują, odbierają publikacje i przetwarzają potwierdzenia dostarczenia w oddzielnych wątkach.

Aplikacja kliencka MQTT nie czeka na dostarczenie publikacji. Aplikacja przekazuje komunikat do klienta MQTT, a następnie kontynuuje działanie we własnym wątku. Znacznik dostarczania służy do

synchronizowania aplikacji z dostarczaniem publikacji. Więcej informacji na ten temat zawiera sekcja [Znaczniki dostarczania](#).

Po przekazaniu komunikatu do klienta MQTT aplikacja może oczekiwać na znacznik dostarczania. Zamiast czekać, klient może udostępnić metodę wywołania zwrotnego, która jest wywoływana, gdy publikacja jest dostarczana do produktu IBM MQ. Może również ignorować znacznik dostarczania.

W zależności od jakości usługi powiązanej z komunikatem znacznik dostarczania jest zwracany natychmiast do metody wywołania zwrotnego lub po pewnym czasie. Znacznik dostarczania może zostać zwrócony nawet po rozłączeniu i ponownym połączeniu klienta. Jeśli jakość usługi to *fire and forget*, znacznik dostawy jest zwracany natychmiast. W pozostałych dwóch przypadkach znacznik dostarczania jest zwracany tylko wtedy, gdy klient otrzyma potwierdzenie, że publikacja została wysłana do subskrybentów.

Publikacje wysyłane do klienta MQTT w wyniku subskrypcji klienta są dostarczane do metody wywołania zwrotnego `messageArrived`. Program `messageArrived` działa w innym wątku niż aplikacja główna.

Wysyłanie komunikatów bezpośrednio do klienta MQTT

Komunikat można wysłać do konkretnego klienta MQTT na jeden z dwóch sposobów.

1. Aplikacja IBM MQ może wysłać komunikat bezpośrednio do klienta MQTT bez subskrypcji. Więcej informacji na ten temat zawiera sekcja [Wysyłanie komunikatu bezpośrednio do klienta](#).
2. Alternatywnym podejściem jest użycie konwencji nazewnictwa `ClientIdentifier`. Wszyscy subskrybenci programu MQTT tworzą subskrypcje, korzystając z ich unikalnego identyfikatora klienta (`ClientIdentifier`) jako tematu. Publikuj w `ClientIdentifier`. Publikacja jest wysyłana do klienta, który zasubskrybował temat `ClientIdentifier`. Za pomocą tej techniki można wysłać publikację do określonego subskrybenta programu MQTT.

Windows

Linux

AIX

Zabezpieczenia produktu MQ Telemetry

Zabezpieczanie urządzeń telemetrycznych może być ważne, ponieważ urządzenia mogą być przenośne i używane w miejscach, które nie mogą być dokładnie kontrolowane. Połączenia z urządzeniami MQTT do usługi telemetrycznej (MQXR) można zabezpieczyć za pomocą sieci VPN. MQ Telemetry udostępnia dwa inne mechanizmy zabezpieczeń: TLS i JAAS.

Protokół TLS jest używany głównie do szyfrowania komunikacji między urządzeniem i kanałem pomiarowym oraz do uwierzytelniania urządzenia łączącego się z poprawnym serwerem. Więcej informacji na ten temat zawiera sekcja [Uwierzytelnianie kanału pomiarowego przy użyciu protokołu TLS](#). Można również użyć protokołu TLS, aby sprawdzić, czy urządzenie klienckie może nawiązywać połączenie z serwerem. Więcej informacji na ten temat zawiera sekcja [Uwierzytelnianie klientaMQTT przy użyciu protokołu TLS](#).

Usługa JAAS jest używana głównie do sprawdzania, czy użytkownik urządzenia może używać aplikacji serwera. Więcej informacji na ten temat zawiera sekcja [Uwierzytelnianie klientaMQTT przy użyciu hasła](#). Usługa JAAS może być używana z serwerem LDAP do sprawdzania hasła przy użyciu katalogu pojedynczego logowania.

Protokołów TLS i JAAS można używać w połączeniu z uwierzytelnianiem dwuskładniowym. Istnieje możliwość ograniczenia szyfrów używanych przez protokół TLS do szyfrów zgodnych ze standardami FIPS.

W przypadku co najmniej dziesiątek tysięcy użytkowników nie zawsze jest praktyczne udostępnianie indywidualnych profili bezpieczeństwa. Nie zawsze jest również praktyczne użycie profili do autoryzowania poszczególnych użytkowników do uzyskiwania dostępu do obiektów IBM MQ. Zamiast tego należy grupować użytkowników w klasy w celu autoryzowania publikowania i subskrybowania tematów oraz wysyłania publikacji do klientów.

Skonfiguruj każdy kanał pomiarowy, aby odwzorować klienty na wspólne identyfikatory użytkowników klienta. Dla każdego klienta łączącego się w konkretnym kanale należy użyć wspólnego identyfikatora użytkownika; patrz sekcja [Tożsamość i autoryzacja klientaMQTT](#).

Autoryzowanie grup użytkowników nie stanowi naruszenia ochrony uwierzytelniania każdej osoby. Każdy pojedynczy użytkownik może zostać uwierzytelniony na kliencie lub serwerze przy użyciu Nazwy użytkownika i hasła, a następnie autoryzowany na serwerze przy użyciu wspólnego identyfikatora użytkownika.

Windows

Linux

AIX

MQ Telemetry globalizacja

Ładunek komunikatu w protokole MQTT v3 jest zakodowany jako tablica bajtów. Ogólnie rzecz biorąc, aplikacje obsługujące tekst tworzą ładunek komunikatu w pliku UTF-8. Kanał pomiarowy opisuje ładunek komunikatu jako UTF-8, ale nie wykonuje żadnych konwersji stron kodowych. Łańcuch tematu publikacji musi mieć postać UTF-8.

Aplikacja jest odpowiedzialna za przekształcanie danych alfabetycznych na poprawną stronę kodową i danych liczbowych na poprawne kodowanie liczb.

Klient MQTT Java ma wygodną metodę `MqttMessage.toString`. Metoda traktuje ładunek komunikatu jako zakodowany w domyślnym zestawie znaków platformy lokalnej, którym jest zazwyczaj UTF-8. Przekształca ładunek w łańcuch Java. W systemie Java używana jest metoda łańcuchowa `getBytes`, która przekształca łańcuch w tablicę bajtów zakodowaną przy użyciu domyślnego zestawu znaków platformy lokalnej. Dwa programy MQTT Java wymieniające tekst w ładunku komunikatu między platformami z tym samym domyślnym zestawem znaków mogą to zrobić w prosty i wydajny sposób w systemie UTF-8.

Jeśli domyślnym zestawem znaków na jednej z platform nie jest UTF-8, aplikacje muszą ustanowić konwencję wymiany komunikatów. Na przykład publikator określa konwersję z łańcucha do UTF-8 przy użyciu metody `getBytes("UTF8")`. Aby odbierać tekst komunikatu, subskrybent przyjmuje, że komunikat jest zakodowany w zestawie znaków UTF-8.

Usługa telemetryczna (MQXR) opisuje kodowanie wszystkich publikacji przychodzących z komunikatów klientów MQTT jako UTF-8. Ustawia on strukturę MQMD produktu MQMD. `CodedCharSetId` do UTF-8 i RFH2. `CodedCharSetId` do MQCCSI_INHERIT; zawiera sekcja ["Integracja programu MQ Telemetry z menedżerami kolejek"](#) na stronie 133. Format publikacji jest ustawiony na MQFMT_NONE, dlatego konwersja nie może być wykonywana przez kanały ani przez MQGET.

Windows

Linux

AIX

Wydajność i skalowalność produktu MQ

Telemetry

Podczas zarządzania dużą liczbą klientów i zwiększania skalowalności produktu MQ Telemetry należy wziąć pod uwagę następujące czynniki.

Planowania mocy obliczeniowej

Informacje na temat raportów wydajności produktu MQ Telemetry zawiera dokument [MQ Performance documents](#) (Dokumenty dotyczące wydajności produktu WebSphere MQ).

Połączenia

Koszty związane z połączeniami obejmują

- Koszt skonfigurowania samego połączenia pod względem wykorzystania procesora i czasu.
- Koszty sieci.
- Pamięć używana, gdy połączenie jest otwarte, ale nie jest używane.

Gdy klienci pozostają połączone, występuje dodatkowe obciążenie. Jeśli połączenie jest otwarte, przepływy TCP/IP i komunikaty MQTT używają sieci do sprawdzenia, czy połączenie nadal istnieje. Dodatkowo pamięć jest używana na serwerze dla każdego połączenia klienckiego, które jest otwarte.

Jeśli wysyłasz wiadomości więcej niż jeden na minutę, pozostaw połączenie otwarte, aby uniknąć kosztów inicjowania nowego połączenia. Jeśli wysyłasz wiadomości rzadziej niż raz na 10-15 minut, rozważ

zerwanie połączenia, aby uniknąć kosztów utrzymania go w stanie otwartym. Połączenie TLS może być otwarte, ale beczynne przez dłuższy czas, ponieważ konfiguracja jest bardziej kosztowna.

Dodatkowo należy wziąć pod uwagę możliwości klienta. Jeśli na kliencie jest dostępna funkcja przechowywania i przekazywania, można umieścić komunikaty w zadaniu wsadowym i usunąć połączenie między wysyłaniem zadań wsadowych. Jeśli jednak klient jest odłączony, nie jest możliwe, aby klient odebrał komunikat z serwera. W związku z tym cel wniosku ma znaczenie dla decyzji.

Jeśli w systemie jest jeden klient, który wysyła wiele komunikatów, na przykład przesyłanie plików, nie należy czekać na odpowiedź serwera dla każdego komunikatu. Zamiast tego należy wysłać wszystkie komunikaty i sprawdzić na końcu, czy wszystkie zostały odebrane. Alternatywnie można użyć opcji Jakość usługi (QoS).

Wartość QoS można zmieniać w zależności od komunikatu, dostarczając nieistotne komunikaty przy użyciu wartości QoS 0 i ważne komunikaty przy użyciu wartości QoS 2. Przepustowość komunikatów może być około dwukrotnie wyższa przy QoS 0 niż przy QoS 2.

Konwencje nazewnictwa

Jeśli aplikacja jest projektowana dla wielu klientów, należy zaimplementować efektywną konwencję nazewnictwa. Aby odwzorować każdego klienta na poprawny identyfikator `ClientIdentifier`, należy nadać mu znaczenie `ClientIdentifier`. Dobra konwencja nazewnictwa ułatwia administratorowi określanie, które klienty są uruchomione. Konwencja nazewnictwa pomaga administratorowi filtrować długą listę klientów w Eksploratorze IBM MQ i ułatwia określenie problemu; patrz Identyfikator klienta.

Moce produkcyjne

Długość nazw tematów wpływa na liczbę bajtów przepływających przez sieć. Podczas publikowania lub subskrybowania ważna może być liczba bajtów w komunikacie. Dlatego należy ograniczyć liczbę znaków w nazwie tematu. Gdy klient MQTT subskrybuje temat IBM MQ, nadaje mu nazwę w postaci:

```
ClientIdentifier: TopicName
```

Aby wyświetlić wszystkie subskrypcje dla klienta MQTT, można użyć komendy IBM MQ MQSC **DISPLAY**:

```
DISPLAY SUB(' ClientID1:*')
```

Definiowanie zasobów w produkcie IBM MQ do użycia przez klienty MQTT

Klient MQTT łączy się ze zdalnym menedżerem kolejek w systemie IBM MQ. Istnieją dwie podstawowe metody wysyłania komunikatów przez aplikację IBM MQ do klienta MQTT: ustawienie domyślnej kolejki transmisji na wartość `SYSTEM.MQTT.TRANSMIT.QUEUE` lub użycie aliasów menedżera kolejek. Zdefiniuj domyślną kolejkę transmisji menedżera kolejek, jeśli istnieje duża liczba klientów MQTT. Użycie domyślnego ustawienia kolejki transmisji upraszcza administrowanie. Więcej informacji na ten temat zawiera sekcja Konfigurowanie kolejkowania rozproszonego w celu wysyłania komunikatów do klientów MQTT.

Zwiększanie skalowalności przez unikanie subskrypcji.

Gdy klient produktu MQTT V3 subskrybuje temat, subskrypcja jest tworzona przez usługę telemetryczną (MQXR) w produkcie IBM MQ. Subskrypcja kieruje publikację dla klienta do programu `SYSTEM.MQTT.TRANSMIT.QUEUE`. Nazwa zdalnego menedżera kolejek w nagłówku transmisji każdej publikacji jest ustawiana na wartość `ClientIdentifier` klienta MQTT, który dokonał subskrypcji. Jeśli istnieje wiele klientów, z których każdy ma własne subskrypcje, powoduje to, że wiele subskrypcji proxy jest utrzymywanych w obrębie klastra lub hierarchii publikowania/subskrypcji produktu IBM MQ. Więcej informacji na temat nieużywania publikowania/subskrypcji, ale korzystania z rozwiązania opartego na trybie punkt z punktem, zawiera sekcja Wysyłanie komunikatu bezpośrednio do klienta.

Zarządzanie dużą liczbą klientów

Aby obsłużyć wiele jednocześnie połączonych klientów, należy zwiększyć ilość pamięci dostępnej dla usługi telemetrycznej (MQXR), ustawiając parametry maszyny JVM **-Xms** i **-Xmx**. Wykonaj następujące kroki:

1. Znajdź plik `java.properties` w katalogu konfiguracji usługi telemetrycznej. Więcej informacji na ten temat zawiera sekcja [Katalog konfiguracji usługi telemetrycznej \(MQXR\) w produkcie Windows](#) lub [Katalog konfiguracji usługi telemetrycznej w produkcie Linux](#).
2. Postępuj zgodnie ze wskazówkami w pliku; stertera 1 GB jest wystarczająca dla 50 000 jednocześnie połączonych klientów.

```
# Heap sizing options - uncomment the following lines to set the heap to 1G
#-Xmx1024m
#-Xms1024m
```

3. Dodaj inne argumenty wiersza komend, aby przekazać je do maszyny JVM, w której działa usługa telemetryczna (MQXR), w pliku `java.properties`. Więcej informacji na ten temat zawiera sekcja [Przekazywanie parametrów maszyny JVM do usługi telemetrycznej \(MQXR\)](#).

Aby zwiększyć liczbę otwartych deskryptorów plików w systemie Linux, należy dodać następujące wiersze do pliku `/etc/security/limits.conf`/i zalogować się ponownie.

```
@mqm soft nofile 65000
@mqm hard nofile 65000
```

Każde gniazdo wymaga jednego deskryptora pliku. Usługa telemetryczna wymaga dodatkowych deskryptorów plików, dlatego liczba ta musi być większa niż liczba wymaganych otwartych gniazd.

Menedżer kolejek używa uchwytu obiektu dla każdej nietrwałej subskrypcji. Aby obsłużyć wiele aktywnych, nietrwałych subskrypcji, należy zwiększyć maksymalną liczbę aktywnych uchwytów w menedżerze kolejek, na przykład:

```
echo ALTER QMGR MAXHANDS(99999999) | runmqsc qMgrName
```

Rysunek 48. Zmień maksymalną liczbę uchwytów w systemie Windows

```
echo "ALTER QMGR MAXHANDS(99999999)" | runmqsc qMgrName
```

Rysunek 49. Zmień maksymalną liczbę uchwytów w systemie Linux

Inne zagadnienia

Podczas planowania wymagań systemowych należy wziąć pod uwagę czas potrzebny na zrestartowanie systemu. Planowany czas przestoju może mieć wpływ na liczbę komunikatów oczekujących w kolejce na przetworzenie. Skonfiguruj system w taki sposób, aby komunikaty mogły być pomyślnie przetworzone w akceptowalnym czasie. Przejrzyj pamięć dyskową, pamięć i moc obliczeniową. W przypadku niektórych aplikacji klienckich może być możliwe usunięcie komunikatów, gdy klient ponownie nawiąże połączenie. Aby usunąć komunikaty, należy ustawić opcję `CleanSession` w parametrach połączenia klienta. Więcej informacji na ten temat zawiera sekcja [Czyszczenie sesji](#). Alternatywnie publikuj i subskrybuj przy użyciu najlepszej jakości usługi (0) w kliencie MQTT. Więcej informacji na ten temat zawiera sekcja [Jakość usługi](#). Podczas wysyłania komunikatów z produktu IBM MQ należy używać komunikatów [nietrwałych](#). Komunikaty o tych cechach jakości usługi nie są odtwarzane po restarcie systemu lub połączenia.

Windows

Linux

AIX

Urządzenia obsługiwane przez MQ Telemetry

Klienci MQTT mogą działać na wielu urządzeniach, od czujników i siłowników, po ręczne urządzenia i systemy pojazdów.

Klienty MQTT są małe i działają na urządzeniach ograniczonych przez niewielką ilość pamięci i małą moc obliczeniową. Produkt MQTT protocol jest niezawodny i ma małe nagłówki, które odpowiadają sieciom ograniczonym przez niską przepustowość, wysokie koszty i nieregularną dostępność.

Produkt MQ Telemetry komunikuje się z urządzeniami telemetrycznymi za pośrednictwem aplikacji klienckich MQTT. Aplikacje te korzystają z następujących zasobów, z których wszystkie implementują protokół MQTT v3:

- Następujące biblioteki klienckie:
 - *MQTT client for Java*, który jest używany do budowania aplikacji rodzimych dla (na przykład) urządzeń z systemem Android, OS X, Linux lub Windows. Aplikacje, które korzystają z tej biblioteki klienckiej, mogą działać na wszystkich wariantach Java, od najmniejszej konfiguracji CLDC (Connected Limited Device Configuration) / MIDP (Mobile Information Device Profile) poprzez CDC (Connected Device Configuration) / Foundation, J2SE (Java Platform, Standard Edition) i J2EE (Java Platform, Enterprise Edition). Obsługiwana jest również dostosowana biblioteka klas IBM jclRM. Platforma Java ME jest zwykle używana na małych urządzeniach, takich jak siłowniki, czujniki, telefony komórkowe i inne urządzenia wbudowane. Platforma Java SE jest zwykle instalowana na urządzeniach wbudowanych wyższego poziomu, takich jak komputery i serwery.
 - *MQTT client for C*, który jest używany do budowania aplikacji rodzimych dla (na przykład) urządzeń z systemem iOS, OS X, Linux lub Windows. Ta biblioteka kliencka udostępnia implementację referencyjną w języku C wraz z wbudowanym rodzimym klientem dla systemów Windows i Linux. Implementacja referencyjna w języku C umożliwiła przeportowanie produktu MQTT na szeroką gamę urządzeń i platform. Niektóre systemy Windows w systemie Intel, w tym Windows 7, RedHat, Ubuntu i niektóre systemy Linux na platformach ARM, takie jak Eurotech Viper, implementują wersje systemu Linux, w których działa klient C, ale produkt IBM nie zapewnia obsługi serwisowej dla tych platform. Jeśli planowane jest połączenie z centrum wsparcia IBM, należy odtworzyć problemy z klientem na obsługiwanej platformie.
 - Produkt *MQTT client for Java*, który jest używany do budowania aplikacji WWW opartych na przeglądarce.

Biblioteki klienta MQTT są dostępne bezpłatnie w serwisie Eclipse Paho i MQTT.org. Patrz [programy przykładowe IBM MQ Telemetry Transport](#).

Zabezpieczenia w produkcie IBM MQ

W produkcie IBM MQ istnieje kilka metod udostępniania zabezpieczeń: interfejs usługi autoryzacji, wyjścia kanału napisane przez użytkownika lub inne firmy, zabezpieczenia kanału przy użyciu protokołu TLS (Transport Layer Security), rekordy uwierzytelniania kanału i zabezpieczenia komunikatów.

Interfejs usługi autoryzacji

Autoryzacja do używania wywołań MQI, komend i dostępu do obiektów jest zapewniana przez **menedżer uprawnień do obiektów** (object authority manager-OAM), który domyślnie jest włączony. Dostęp do obiektów IBM MQ jest kontrolowany przez grupy użytkowników IBM MQ i OAM. Administratorzy mogą używać interfejsu wiersza komend do nadawania lub odbierania autoryzacji zgodnie z wymaganiami.

Więcej informacji na temat tworzenia komponentów usługi autoryzacji zawiera sekcja [Konfigurowanie zabezpieczeń w systemach AIX, Linux, and Windows](#).

Wyjścia kanału napisane przez użytkownika lub inne firmy

Kanały mogą korzystać z napisanych przez użytkownika lub zewnętrznych wyjść kanałów. Więcej informacji na ten temat zawiera sekcja [Programy obsługi wyjścia kanału dla kanałów przesyłania komunikatów](#).

Ochrona kanału przy użyciu protokołu TLS

Protokół TLS (Transport Layer Security) zapewnia bezpieczeństwo kanałów zgodne ze standardami branżowymi, zapewniając ochronę przed podsłuchiwaniami, manipulowaniem i imitowaniem.

Protokół TLS używa technik klucza publicznego i symetrycznych w celu zapewnienia poufności i integralności komunikatów oraz wzajemnego uwierzytelniania.

Obszerny przegląd zabezpieczeń w produkcie IBM MQ, w tym szczegółowe informacje o protokole TLS, zawiera sekcja [Zabezpieczanie](#). Przegląd protokołu TLS, w tym wskaźniki do komend opisanych w tej sekcji, zawiera sekcja [Cryptographic security protocols: TLS](#).

Rekordy uwierzytelniania kanału

Rekordy uwierzytelniania kanału umożliwiają uzyskanie precyzyjnej kontroli nad dostępem przyznanym systemom łączącym się na poziomie kanału. Więcej informacji na ten temat zawiera sekcja [Rekordy uwierzytelniania kanału](#).

Bezpieczeństwo komunikatu

Należy użyć produktu Advanced Message Security, który jest oddzielnie instalowanym i licencjonowanym komponentem produktu IBM MQ, aby zapewnić ochronę kryptograficzną dla komunikatów wysyłanych i odbieranych przy użyciu produktu IBM MQ. Patrz [Advanced Message Security](#).

Zadania pokrewne

[Zabezpieczanie](#)

[Planowanie wymagań dotyczących bezpieczeństwa](#)

Obsługa protokołu TLS zarządzanego klienta IBM MQ.NET

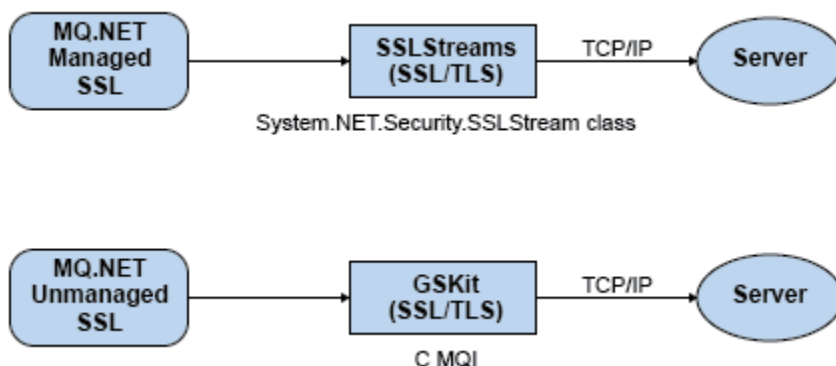
W pełni zarządzany klient IBM MQ.NET obsługuje protokół TLS (Transport Layer Security), który jest oparty na pakiecie Microsoft.NET SSLStreams. Różni się to od innych klientów IBM MQ, które wykorzystują IBM Global Security Kit (GSKit).

Aplikacje produktu IBM MQ.NET można projektować tak, aby działały w trybie zarządzanym lub w trybie niezarządzanym.

- W trybie zarządzanym aplikacje .NET działają w środowisku .NET CLR (Common Language Runtime) bez wywoływania wieloplatformowego, na przykład wywoływania interfejsu MQI języka C.
- W trybie niezarządzanym wywołanie interfejsu MQI języka C jest wykonywane dla bazowych operacji MQI. Zasadniczo interfejs trybu niezarządzanego składa się z klas opakowujących języka .NET znajdujących się w górnej części interfejsu MQI języka C.

Zarządzany klient IBM MQ.NET używa bibliotek środowiska Microsoft.NET do implementowania protokołów SSL TLS. Bezpieczeństwo Systemu.NET.Security.SSLStream SSLStream z Microsoft jest używana do implementowania zabezpieczeń (TLS) w produkcie IBM MQ.NET.

Niezarządzany tryb klienta IBM MQ.NET obsługuje już funkcję TLS, która jest oparta na MQI C (i GSKit). Oznacza to, że operacje TLS są obsługiwane przez MQI języka C. W tym przypadku GSKit implementuje protokoły TLS Secure Socket.



Rysunek 50. Porównanie zarządzanych i niez zarządzanych protokołów TLS w produkcie IBM MQ.NET

Poniższa tabela zawiera podsumowanie różnic między implementacjami zarządzanymi i niez zarządzanymi:

Tabela 14. Różnice między implementacjami zarządzanymi i niez zarządzanymi

Tryb	Protocols (Protokoły)	Koszt	Komentarze
Zarządzany protokół SSL produktu IBM MQ.NET	TLS	System.NET.Security.SSLStream Klasa SSLStream działa jako strumień w podłączonym gnieździe TCP	TLS 1.0 TLS 1.2 (tylko w środowisku Microsoft.NET Framework v4.5)
Niez zarządzany protokół SSL produktu IBM MQ.NET	TLS	GSKIT i C-MQI	Protokoły SSL TLS

Pojęcia pokrewne

Obsługa protokołów SSL (Secure Sockets Layer) i TLS (Transport Layer Security) w produkcie .NET

IBM MQ MQI clients

IBM MQ MQI client to komponent produktu IBM MQ, który można zainstalować w systemie, w którym nie jest uruchomiony żaden menedżer kolejek.

IBM MQ MQI Klient to komponent, który umożliwia aplikacji działającej w systemie wydawanie wywołań MQI do menedżera kolejek działającego w innym systemie. Dane wyjściowe wywołania są wysyłane z powrotem do klienta, który przekazuje je z powrotem do aplikacji.

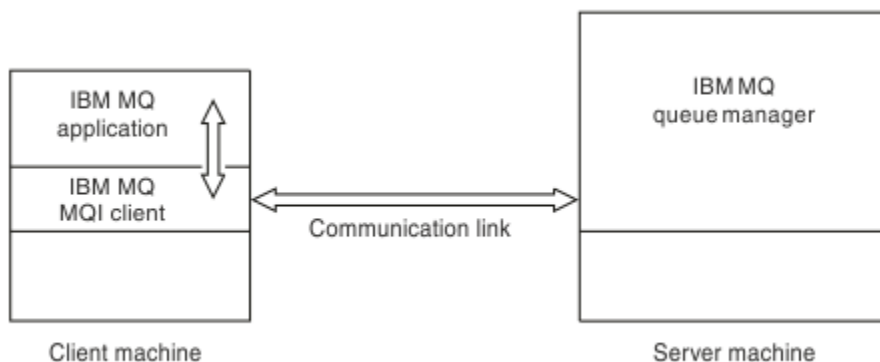
Za pomocą programu IBM MQ MQI client aplikacja działająca w tym samym systemie co klient może nawiązać połączenie z menedżerem kolejek działającym w innym systemie. Aplikacja może wydawać wywołania MQI do tego menedżera kolejek. Taka aplikacja jest nazywana aplikacją IBM MQ MQI client, a menedżer kolejek jest nazywany *menedżerem kolejek serwera*.

IBM MQ serwer jest menedżerem kolejek, który udostępnia usługi kolejkowania jednemu lub większej liczbie klientów. Wszystkie obiekty IBM MQ, na przykład kolejki, istnieją tylko na komputerze menedżera kolejek (na komputerze serwera IBM MQ), a nie na kliencie. Serwer IBM MQ może również obsługiwać lokalne aplikacje IBM MQ.

Różnica między serwerem IBM MQ a zwykłym menedżerem kolejek polega na tym, że serwer ma dedykowane łącze komunikacyjne z każdym klientem. Więcej informacji na temat tworzenia kanałów dla klientów i serwerów zawiera sekcja [Konfigurowanie rozproszonego kolejkowania](#).

Aplikacja IBM MQ MQI client i menedżer kolejek serwera komunikują się ze sobą za pomocą *kanatu MQI*. Kanał MQI jest uruchamiany, gdy aplikacja kliencka wysyła wywołanie **MQCONN** lub **MQCONNX**

w celu nawiązania połączenia z menedżerem kolejek, a kończy działanie, gdy aplikacja kliencka wysyła wywołanie **MQDISC** w celu rozłączenia się z menedżerem kolejek. Parametry wejściowe przepływu wywołania MQI w jednym kierunku w kanale MQI i parametry wyjściowe w przeciwnym kierunku.



Rysunek 51. Połączenie między klientem a serwerem

Można użyć następujących platform. Kombinacje zależą od używanego produktu IBM MQ i są opisane w sekcji [“Obsługa platform dla klientów IBM MQ”](#) na stronie 146.

IBM MQ MQI client

AIX and Linux
Windows
IBM i

Serwer IBM MQ

AIX and Linux
Windows
IBM i
z/OS

MQI jest dostępna dla aplikacji działających na platformie klienckiej; kolejki i inne obiekty IBM MQ są wstrzymane w menedżerze kolejek, który został zainstalowany na serwerze.

Aplikacja, która ma być uruchamiana w środowisku IBM MQ MQI client, musi być najpierw połączona z odpowiednią biblioteką klienta. Gdy aplikacja wysyła wywołanie MQI, IBM MQ MQI client kieruje żądanie do menedżera kolejek, w którym jest ono przetwarzane i z którego odpowiedź jest wysyłana z powrotem do IBM MQ MQI client.

Połączenie między aplikacją i serwerem IBM MQ MQI client jest ustanawiane dynamicznie w czasie wykonywania.

Aplikacje klienckie można również tworzyć za pomocą IBM MQ classes for .NET, IBM MQ classes for Java lub IBM MQ classes for Java Message Service (JMS). Klientów dla systemów Java i JMS można używać na następujących platformach:

-  IBM i
-  AIX
-  Linux
-  Windows

Użycie opcji Java i JMS nie zostało opisane w tej sekcji. Szczegółowe informacje na temat instalowania, konfigurowania i używania produktów IBM MQ classes for Java i IBM MQ classes for JMS zawiera sekcja [Korzystanie z produktów IBM MQ classes for Java i Korzystanie z produktu IBM MQ classes for JMS](#).

Aplikacje IBM MQ w środowisku klient-serwer

Po połączeniu z serwerem aplikacje klienckie IBM MQ mogą wywołać większość wywołań MQI w taki sam sposób, jak aplikacje lokalne. Aplikacja kliencka wysyła wywołanie MQCONN w celu nawiązania

połączenia z określonym menedżerem kolejek. Wszystkie dodatkowe wywołania MQI, które określają uchwyt połączenia zwrócony z żądania połączenia, są następnie przetwarzane przez ten menedżer kolejek.

Należy połączyć aplikacje z odpowiednimi bibliotekami klienta. Patrz sekcja [Budowanie aplikacji dla produktu IBM MQ MQI clients](#).

Pojęcia pokrewne

“Dlaczego warto korzystać z klientów IBM MQ ?” na stronie 146

Używanie klientów IBM MQ jest wydajnym sposobem implementowania mechanizmu przesyłania komunikatów i kolejkowania produktu IBM MQ .

“Co to jest rozszerzony klient transakcyjny?” na stronie 148

Rozszerzony klient transakcyjny IBM MQ może aktualizować zasoby zarządzane przez inny menedżer zasobów pod kontrolą zewnętrznego menedżera transakcji.

“W jaki sposób klient łączy się z serwerem” na stronie 149

Klient łączy się z serwerem za pomocą MQCONN lub MQCONNX i komunikuje się za pośrednictwem kanału.

“Zarządzanie transakcjami i wsparcie” na stronie 150

Wprowadzenie do zarządzania transakcjami i sposób, w jaki program IBM MQ obsługuje transakcje.

“Rozszerzanie narzędzi menedżera kolejek” na stronie 152

Narzędzia menedżera kolejek można rozszerzać za pomocą programów zewnętrznych, programów zewnętrznych funkcji API lub usług, które można zainstalować.

Informacje pokrewne

[Jak skonfigurować IBM MQ MQI client](#)

Dlaczego warto korzystać z klientów IBM MQ ?

Używanie klientów IBM MQ jest wydajnym sposobem implementowania mechanizmu przesyłania komunikatów i kolejkowania produktu IBM MQ .

Może istnieć aplikacja używająca interfejsu MQI działającego na jednym komputerze i menedżera kolejek działającego na innym komputerze (fizycznym lub wirtualnym). Korzyści płynące z tego są następujące:

- Nie ma potrzeby stosowania pełnej implementacji IBM MQ na komputerze klienckim.
- Wymagania sprzętowe w systemie klienta zostały ograniczone.
- Wymagania dotyczące administrowania systemem zostały ograniczone.
- Aplikacja IBM MQ działająca na kliencie może łączyć się z wieloma menedżerami kolejek w różnych systemach.
- Można użyć alternatywnych kanałów wykorzystujących różne protokoły transmisji.

Obsługa platform dla klientów IBM MQ

IBM MQ na wszystkich obsługiwanych platformach serwerów akceptuje połączenia klienckie z programu IBM MQ MQI clients na wielu platformach.

Program IBM MQ zainstalowany jako *produkt podstawowy i serwer* na wszystkich obsługiwanych platformach serwerów może akceptować połączenia programu IBM MQ MQI clients na następujących platformach:

-  IBM i
-  AIX
-  Linux
-  Windows

Połączenia klienckie różnią się identyfikatorem kodowanego zestawu znaków (CCSID) i protokołem komunikacyjnym.

Jakie aplikacje działają w systemie IBM MQ MQI client?

Pełna funkcja MQI jest obsługiwana w środowisku klienta. Dzięki temu prawie każda aplikacja produktu IBM MQ może zostać skonfigurowana do działania w systemie IBM MQ MQI client przez połączenie aplikacji w produkcie IBM MQ MQI client z biblioteką MQIC, a nie z biblioteką MQI. Wyjątki są następujące:

- MQGET z sygnałem
- Aplikacja, która wymaga koordynacji punktów synchronizacji z innymi menedżerami zasobów, musi używać rozszerzonego klienta transakcyjnego.

Jeśli funkcja odczytu z wyprzedzeniem jest włączona, w celu zwiększenia wydajności przesyłania komunikatów nietrwałych nie są dostępne wszystkie opcje MQGET. W tabeli przedstawiono dozwolone opcje oraz informacje o tym, czy można je zmieniać między wywołaniami MQGET.

Tabela 15. Opcje MQGET są dozwolone, gdy włączona jest funkcja odczytu z wyprzedzeniem

Wartości	Dozwolone, gdy odczyt z wyprzedzeniem jest włączony i można go zmieniać między wywołaniami MQGET	Dozwolone, gdy odczyt z wyprzedzeniem jest włączony, ale nie można go zmieniać między wywołaniami MQGET ¹	Opcje MQGET, które nie są dozwolone, gdy włączony jest odczyt z wyprzedzeniem ²
Wartości deskryptora MQGET MD	MsgId ³ CorrelId ³	Kodowanie CodedCharSetId	
Opcje MQGET MQGMO	MQGMO_WAIT MQGMO_NO_WAIT MQGMO_FAIL_IF QUIESCING, MQGMO_BROWSE_FIRST ⁴ MQGMO_BROWSE_NEXT ⁴ MQGMO_BROWSE_MESSAGE_UNDER_CURSOR ⁴	MQGMO_SYNCPOINT_IF_PERSISTENT, MQGMO_NO_SYNCPOINT MQGMO_ACCEPT_TRUNCATED_MSG MQGMO_CONVERT MQGMO_LOGICAL_ORDER (KOLEJNA_KOLEJNA_STRUKTURA) MQGMO_COMPLETE_MSG MQGMO_ALL_MSGS_AVAILABLE MQGMO_ALL_SEGMENTS_AVAILABLE MQGMO_MARK_BROWSE_HANDLE MQGMO_MARK_BROWSE_CO_OP MQGMO_UNMARK_BROWSE_CO_OP MQGMO_UNMARK_BROWSE_HANDLE MQGMO_UNMARKED_BROWSE_MSG MQGMO_PROPERTIES_FORCE_MQRFH2 MQGMO_NO_PROPERTIES MQGMO_PROPERTIES_IN_HANDLE MQGMO_PROPERTIES_COMPATIBILITY	MQGMO_SET_SIGNAL MQGMO_SYNCPOINT MQGMO_MARK_SKIP_BACKOUT MQGMO_MSG_UNDER_CURSOR ⁴ BLOKADA MQGMO_LOCK MQGMO_UNLOCK (odblokowanie MQGMO)
Wartości MQGMO		MsgHandle	

1. Jeśli te opcje zostaną zmienione między wywołaniami MQGET, zostanie zwrócony kod przyczyny MQRC_OPTIONS_CHANGED.
2. Jeśli te opcje zostaną podane podczas pierwszego wywołania MQGET, odczyt z wyprzedzeniem zostanie wyłączony. Jeśli te opcje zostaną podane w kolejnym wywołaniu MQGET, zostanie zwrócony kod przyczyny MQRC_OPTIONS_ERROR.
3. Aplikacje klienckie muszą uwzględniać fakt, że jeśli wartości MsgId i CorrelId zostały zmienione między wywołaniami MQGET, komunikaty z poprzednimi wartościami mogły już zostać wysłane do klienta i pozostają w buforze odczytu z wyprzedzeniem na kliencie, dopóki nie zostaną wykorzystane (lub automatycznie usunięte).

4. Pierwsze wywołanie MQGET określa, czy komunikaty mają być przeglądane lub pobierane z kolejki, gdy włączony jest odczyt z wyprzedzeniem. Jeśli w aplikacji zostanie podjęta próba użycia zarówno operacji przeglądania, jak i pobierania, zostanie zwrócony kod przyczyny MQRC_OPTIONS_CHANGED.
5. Opcja MQGMO_MSG_UNDER_CURSOR nie jest dostępna, jeśli włączony jest odczyt z wyprzedzeniem. Komunikaty można przeglądać albo odbierać, gdy włączony jest odczyt z wyprzedzeniem. Nie można jednak jednocześnie korzystać z obu tych funkcji.

Aplikacja działająca w systemie IBM MQ MQI client może połączyć się z więcej niż jednym menedżerem kolejek współbieżnie lub użyć nazwy menedżera kolejek z gwiazdką (*) w wywołaniu MQCONN lub MQCONNX (patrz przykłady w sekcji [Łączenie aplikacji IBM MQ MQI client z menedżerami kolejek](#)).

Co to jest rozszerzony klient transakcyjny?

Rozszerzony klient transakcyjny IBM MQ może aktualizować zasoby zarządzane przez inny menedżer zasobów pod kontrolą zewnętrznego menedżera transakcji.

Jeśli użytkownik nie zna pojęć związanych z zarządzaniem transakcjami, należy zapoznać się z sekcją [“Zarządzanie transakcjami i wsparcie”](#) na stronie 150.

Należy zauważyć, że klient transakcyjny XA jest teraz dostarczany jako część produktu IBM MQ.

Aplikacja kliencka może uczestniczyć w jednostce pracy zarządzanej przez menedżer kolejek, z którym jest połączona. W ramach jednostki pracy aplikacja kliencka może umieszczać komunikaty w kolejkach należących do tego menedżera kolejek i je z nich pobrać. Aplikacja kliencka może następnie użyć wywołania **MQCMIT**, aby zatwierdzić jednostkę pracy, lub wywołania **MQBACK**, aby wycofać jednostkę pracy. Jednak w ramach tej samej jednostki pracy aplikacja kliencka nie może aktualizować zasobów innego menedżera zasobów, na przykład tabel bazy danych Db2. Użycie rozszerzonego klienta transakcyjnego IBM MQ usuwa to ograniczenie.


Rozszerzony klient transakcyjny IBM MQ to IBM MQ MQI client z dodatkową funkcją. Za pomocą tej funkcji aplikacja kliencka w ramach tej samej jednostki pracy może wykonywać następujące zadania:

- Umieszczanie komunikatów w kolejkach, których właścicielem jest menedżer kolejek, z którym jest połączony, oraz pobieranie komunikatów z tych kolejek
- Aktualizowanie zasobów menedżera zasobów innego niż menedżer kolejek produktu IBM MQ

Ta jednostka pracy musi być zarządzana przez zewnętrznego menedżera transakcji, który działa w tym samym systemie, co aplikacja kliencka. Jednostka pracy nie może być zarządzana przez menedżer kolejek, z którym połączona jest aplikacja kliencka. Oznacza to, że menedżer kolejek może działać tylko jako menedżer zasobów, a nie jako menedżer transakcji. Oznacza to również, że aplikacja kliencka może zatwierdzić lub wycofać jednostkę pracy tylko za pomocą aplikacyjnego interfejsu programistycznego (API) udostępnianego przez zewnętrznego menedżera transakcji. Dlatego aplikacja kliencka nie może używać wywołań MQI, **MQBEGIN**, **MQCMIT** i **MQBACK**.

Zewnętrzny menedżer transakcji komunikuje się z menedżerem kolejek jako menedżerem zasobów przy użyciu tego samego kanału MQI, który jest używany przez aplikację kliencką połączoną z menedżerem kolejek. Jednak w sytuacji odtwarzania po awarii, gdy żadna aplikacja nie jest uruchomiona, menedżer transakcji może użyć dedykowanego kanału MQI w celu odtworzenia wszystkich niekompletnych jednostek pracy, w których menedżer kolejek brał udział w momencie awarii.

W tej sekcji IBM MQ MQI client, który nie ma rozszerzonej funkcji transakcyjnej, jest nazywany podstawowym klientem IBM MQ. Dlatego można rozważyć, aby rozszerzony klient transakcyjny IBM MQ składał się z klienta podstawowego IBM MQ z dodatkiem rozszerzonej funkcji transakcyjnej.





Uwaga:  System IBM MQ MQI client w systemie IBM i nie obsługuje rozszerzonej funkcji transakcyjnej IBM MQ.

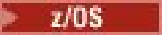
Obsługa platform dla rozszerzonych klientów transakcyjnych



Rozszerzone klienty transakcyjne są dostępne dla wszystkich platform, które obsługują klienta podstawowego. Klienty nie są dostępne dla systemu z/OS.

Aplikacja kliencka używająca rozszerzonego klienta transakcyjnego może nawiązać połączenie tylko z menedżerem kolejek następujących produktów IBM MQ 9.0 lub nowszych:

-  IBM MQ for AIX
-  IBM MQ for IBM i
-  IBM MQ dla Linux
-  IBM MQ for Windows

 Chociaż nie ma żadnych rozszerzonych klientów transakcyjnych, które działają w systemie z/OS, aplikacja kliencka używająca rozszerzonego klienta transakcyjnego może nawiązać połączenie z menedżerem kolejek działającym w systemie z/OS.

Dla każdej platformy wymagania sprzętowe i programowe dla rozszerzonego klienta transakcyjnego są takie same, jak dla klienta podstawowego IBM MQ. Język programowania jest obsługiwany przez rozszerzonego klienta transakcyjnego, jeśli jest obsługiwany przez klienta podstawowego IBM MQ i przez używany menedżer transakcji.

Informacje na temat zewnętrznych menedżerów transakcji dla wszystkich platform zawiera sekcja [Wymagania systemowe produktu IBM MQ](#).

W jaki sposób klient łączy się z serwerem

Klient łączy się z serwerem za pomocą MQCONN lub MQCONNX i komunikuje się za pośrednictwem kanału.

Aplikacja działająca w środowisku klienta IBM MQ musi utrzymywać aktywne połączenie między klientem a serwerem.

Połączenie jest nawiązywane przez aplikację, która wysyła wywołanie MQCONN lub MQCONNX. Klienty i serwery komunikują się za pośrednictwem *kanatów MQI* lub, w przypadku korzystania z konwersacji współużytkowanych, konwersacji, z których każda współużytkuje instancję kanału MQI. Gdy wywołanie powiedzie się, instancja kanału MQI lub konwersacja pozostaje połączona, dopóki aplikacja nie wyśle wywołania MQDISC. Tak jest w przypadku każdego menedżera kolejek, z którym aplikacja musi nawiązać połączenie.

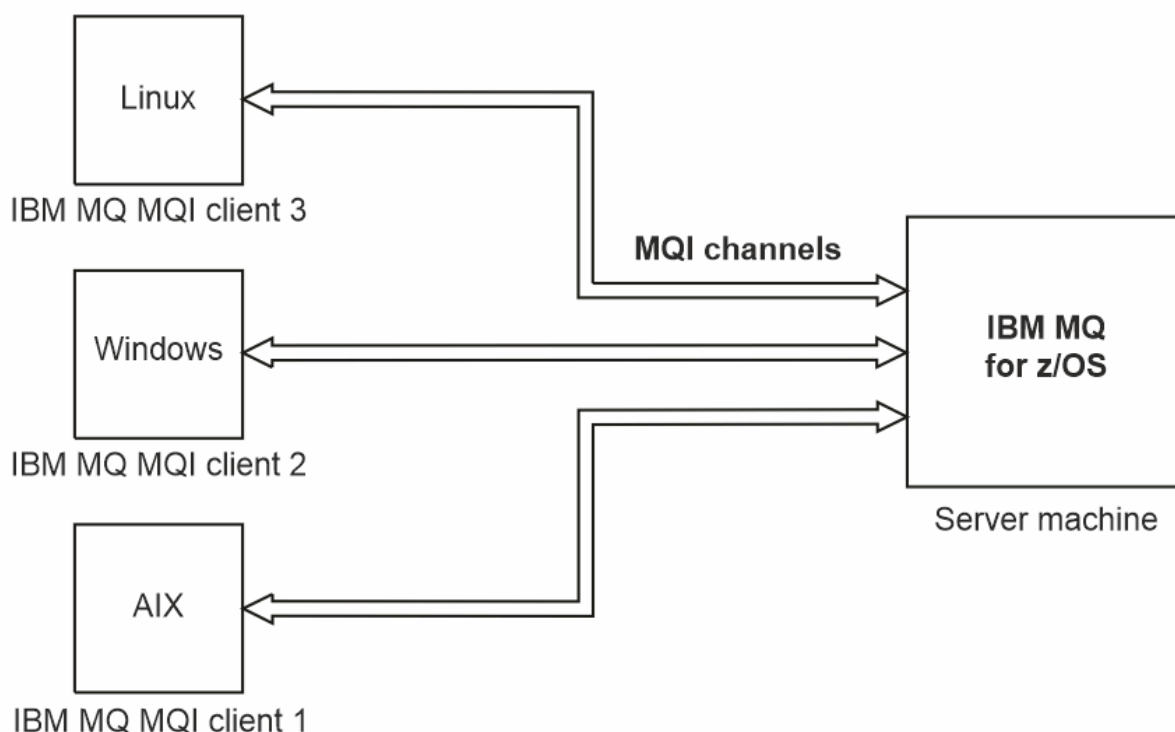
Klient i menedżer kolejek na tym samym komputerze

Aplikację można również uruchomić w środowisku IBM MQ MQI client, jeśli na komputerze jest również zainstalowany menedżer kolejek.

W takiej sytuacji można utworzyć dowiązanie do bibliotek menedżera kolejek lub bibliotek klienta, ale należy pamiętać, że w przypadku dowiązania do bibliotek klienta nadal konieczne jest zdefiniowanie połączeń kanału. Może to być przydatne podczas fazy projektowania aplikacji. Program można przetestować na własnym komputerze bez zależności od innych osób i mieć pewność, że nadal będzie działał po przeniesieniu go do niezależnego środowiska IBM MQ MQI client.

Klienty na różnych platformach

W tym przykładzie serwer komunikuje się z trzema serwerami IBM MQ MQI clients na różnych platformach.



Rysunek 52. Serwer IBM MQ połączony z klientami na różnych platformach

Inne bardziej złożone środowiska są możliwe. Na przykład klient produktu IBM MQ może nawiązać połączenie z więcej niż jednym menedżerem kolejek lub dowolną liczbą menedżerów kolejek połączonych w ramach grupy współużytkowania kolejek.

Korzystanie z różnych wersji oprogramowania klienta i serwera

Jeśli używane są wcześniejsze wersje produktów IBM MQ, należy upewnić się, że konwersja kodu z identyfikatora CCSID klienta jest obsługiwana przez serwer.

Klient produktu IBM MQ może nawiązać połączenie ze wszystkimi obsługiwanymi wersjami menedżera kolejek. W przypadku nawiązywania połączenia z menedżerem kolejek we wcześniejszej wersji nie można używać funkcji i struktur z nowszej wersji produktu w aplikacji IBM MQ na kliencie.

Menedżer kolejek systemu IBM MQ może komunikować się z klientami w różnych wersjach, prowadząc negocjacje na najwyższym wzajemnie obsługiwanym poziomie protokołu. Oznacza to, że starsze klienty mogą być używane z nowszymi poziomami menedżera kolejek. Zaleca się, aby zarówno klient, jak i serwer były w wersji produktu IBM MQ, która jest obecnie obsługiwana, w celu ułatwienia diagnozowania problemów i włączenia obsługi przez IBM.

Więcej informacji na ten temat zawiera sekcja dotycząca języków programowania obsługiwanych w sekcji [Tworzenie aplikacji](#).

Zarządzanie transakcjami i wsparcie

Wprowadzenie do zarządzania transakcjami i sposób, w jaki program IBM MQ obsługuje transakcje.

Menedżer zasobów jest podsystemem komputerowym, który jest właścicielem i zarządza zasobami, do których aplikacje mogą uzyskiwać dostęp i które mogą być aktualizowane. Poniżej przedstawiono przykłady menedżerów zasobów:

- Menedżer kolejek systemu IBM MQ z zasobami, które są jego kolejkami
- Baza danych Db2 z zasobami, które są jej tabelami

Gdy aplikacja aktualizuje zasoby jednego lub większej liczby menedżerów zasobów, może zaistnieć potrzeba biznesowa, która zapewni, że niektóre aktualizacje zostaną zakończone pomyślnie jako grupa lub żadna z nich nie zostanie zakończona. Tego rodzaju wymaganie jest spowodowane tym, że dane biznesowe pozostałyby w stanie niespójnym, gdyby niektóre z tych aktualizacji zakończyły się pomyślnie, ale inne nie.

Aktualizacje zasobów, które są zarządzane w ten sposób, występują w *jednostce pracy* lub *transakcji*. Aplikacja może zgrupować zestaw aktualizacji w jednostkę pracy.

Podczas wykonywania jednostki pracy aplikacja wysyła do menedżerów zasobów żądania aktualizacji swoich zasobów. Jednostka pracy kończy się, gdy aplikacja wysyła żądanie zatwierdzenia wszystkich aktualizacji. Dopóki aktualizacje nie zostaną zatwierdzone, żadna z nich nie będzie widoczna dla innych aplikacji, które uzyskują dostęp do tych samych zasobów. Alternatywnie, jeśli aplikacja zdecyduje, że nie może zakończyć jednostki pracy z jakiegokolwiek powodu, może wysłać żądanie, aby wycofać wszystkie żądane aktualizacje do tego momentu. W takim przypadku żadna z aktualizacji nie będzie widoczna dla innych aplikacji. Te aktualizacje są zwykle powiązane logicznie i muszą być pomyślne, aby zachować integralność danych. Jeśli jedna aktualizacja powiedzie się, a inna nie, integralność danych zostanie utracona.

Po pomyślnym zakończeniu jednostki pracy jest ona *zatwierdzana*. Po zatwierdzeniu wszystkie aktualizacje wprowadzone w ramach tej jednostki pracy stają się trwałe i nieodwracalne. Jeśli jednak jednostka pracy nie powiedzie się, wszystkie aktualizacje są *wycofywane*. Ten proces, w którym jednostki pracy są zatwierdzane lub wycofywane z zachowaniem integralności, jest nazywany *koordynacją punktu synchronizacji*.

Moment, w którym wszystkie aktualizacje w ramach jednostki pracy są zatwierdzane lub wycofywane, jest nazywany *punktem synchronizacji*. Mówi się, że aktualizacja w jednostce pracy ma miejsce w *ramach kontroli punktu synchronizacji*. Jeśli aplikacja żąda aktualizacji *poza kontrolą punktu synchronizacji*, menedżer zasobów natychmiast zatwierdza aktualizację, nawet jeśli jednostka pracy jest w toku, a aktualizacji nie można później wycofywać.

Podsystem komputerowy, który zarządza jednostkami pracy, jest nazywany *menedżerem transakcji* lub *koordynatorem punktowym*.

Lokalna jednostka pracy to jednostka, w której aktualizowane są tylko zasoby menedżera kolejek produktu IBM MQ. W tym przypadku koordynacja punktów synchronizacji jest udostępniana przez sam menedżer kolejek przy użyciu procesu zatwierdzania jednofazowego.

Globalna jednostka pracy to jednostka, w której aktualizowane są również zasoby należące do innych menedżerów zasobów, takich jak bazy danych zgodne z XA. W tym przypadku należy użyć procedury zatwierdzania dwufazowego, a jednostkę pracy może koordynować sam menedżer kolejek lub zewnętrznie inny menedżer transakcji zgodny z interfejsem XA, taki jak IBM TXSeries lub BEA Tuxedo.

Menedżer transakcji jest odpowiedzialny za zapewnienie, że wszystkie aktualizacje zasobów w jednostce pracy zostały zakończone pomyślnie lub że żadna z nich nie została zakończona. Do menedżera transakcji aplikacja wysyła żądanie zatwierdzenia lub wyrejestrowania jednostki pracy. Przykładami menedżerów transakcji są CICS i WebSphere Application Server, chociaż oba te menedżery mają również inne funkcje.

Niektóre menedżery zasobów udostępniają własne funkcje zarządzania transakcjami. Na przykład menedżer kolejek produktu IBM MQ może zarządzać jednostkami pracy, w tym aktualizacjami własnych zasobów i aktualizacjami tabel produktu Db2. Do wykonania tej funkcji menedżer kolejek nie wymaga osobnego menedżera transakcji, ale można go użyć, jeśli jest to wymagane przez użytkownika. Jeśli używany jest oddzielny menedżer transakcji, jest on nazywany *zewnętrznym menedżerem transakcji*.

Aby zewnętrzny menedżer transakcji mógł zarządzać jednostką pracy, musi istnieć standardowy interfejs między menedżerem transakcji a każdym menedżerem zasobów, który uczestniczy w tej jednostce pracy. Ten interfejs umożliwia komunikację między menedżerem transakcji i menedżerem zasobów. Jednym z tych interfejsów jest *interfejs XA*, który jest standardowym interfejsem obsługiwany przez wiele menedżerów transakcji i menedżerów zasobów. Interfejs XA jest publikowany przez grupę Open Group w sekcji *Distributed Transaction Processing: The XA Specification* (Przetwarzanie rozproszonej transakcji: specyfikacja interfejsu XA).

Jeśli w jednostce pracy uczestniczy więcej niż jeden menedżer zasobów, menedżer transakcji musi użyć protokołu *zatwierdzania dwufazowego*, aby zapewnić pomyślne zakończenie wszystkich aktualizacji w obrębie jednostki pracy lub aby żaden z nich nie został zakończony, nawet w przypadku awarii systemu. Gdy aplikacja wysyła do menedżera transakcji żądanie zatwierdzenia jednostki pracy, menedżer transakcji wykonuje następujące czynności:

Faza 1 (Przygotowanie do zatwierdzenia)

Menedżer transakcji prosi każdego menedżera zasobów uczestniczącego w jednostce pracy o sprawdzenie, czy wszystkie informacje o planowanych aktualizacjach jego zasobów są w stanie odtwarzalnym. Menedżer zasobów zwykle wykonuje tę czynność, zapisując informacje w dzienniku i upewniając się, że informacje są zapisywane na dysku twardym. Faza 1 kończy się, gdy menedżer transakcji otrzymuje od każdego menedżera zasobów powiadomienie o tym, że informacje o planowanych aktualizacjach jego zasobów są w stanie odtwarzalnym.

Faza 2 (zatwierdzenie)

Po zakończeniu fazy 1 menedżer transakcji podejmuje nieodwołalną decyzję o zatwierdzeniu jednostki pracy. Prosi on każdego menedżera zasobów uczestniczącego w jednostce pracy o zatwierdzenie aktualizacji jego zasobów. Gdy menedżer zasobów otrzyma to żądanie, musi zatwierdzić aktualizacje. Nie ma opcji, aby je wycofać na tym etapie. Faza 2 kończy się, gdy menedżer transakcji otrzymuje powiadomienie od każdego menedżera zasobów o tym, że zatwierdził aktualizacje swoich zasobów.

Interfejs XA używa protokołu zatwierdzania dwufazowego.

Więcej informacji na ten temat zawiera sekcja [Scenariusze obsługi transakcji](#).

Produkt IBM MQ zapewnia również obsługę serwera Microsoft Transaction Server (COM +). Sekcja [Korzystanie z produktu Microsoft Transaction Server \(COM +\)](#) zawiera informacje na temat konfigurowania produktu IBM MQ w celu korzystania z obsługi modelu COM +.

Rozszerzanie narzędzi menedżera kolejek

Narzędzia menedżera kolejek można rozszerzać za pomocą programów zewnętrznych, programów zewnętrznych funkcji API lub usług, które można zainstalować.

Procedury zewnętrzne

Procedury zewnętrzne udostępniają mechanizm umożliwiający wstawienie własnego kodu do funkcji menedżera kolejek. Obsługiwane są następujące procedury zewnętrzne:

Wyjścia kanału

Te wyjścia zmieniają sposób działania kanałów. Wyjścia kanału są opisane w sekcji [Programy obsługi wyjścia kanału dla kanałów przesyłania komunikatów](#).

Wyjścia konwersji danych

Te wyjścia tworzą fragmenty kodu źródłowego, które mogą być umieszczane w aplikacjach w celu konwersji danych z jednego formatu do innego. Wyjścia konwersji danych są opisane w sekcji [Zapisywanie wyjść konwersji danych](#).

Wyjście obciążenia klastra

Funkcja wykonywana przez to wyjście jest definiowana przez dostawcę wyjścia. Informacje o definicji wywołania są podane w sekcji [MQ_CLUSTER_WORKLOAD_EXIT-opis wywołania](#).

Wyjścia funkcji API

Wyjścia funkcji API umożliwiają napisanie kodu, który zmienia zachowanie wywołań funkcji API języka IBM MQ, takich jak MQPUT i MQGET, a następnie wstawia ten kod bezpośrednio przed tymi wywołaniami lub bezpośrednio po nich. Wstawianie jest automatyczne; menedżer kolejek steruje kodem wyjścia w zarejestrowanych punktach. Więcej informacji na temat wyjść funkcji API zawiera sekcja [Używanie i zapisywanie wyjść funkcji API](#).

Usługi instalowalne

Usługi instalowalne mają sformalizowane interfejsy (interfejs API) z wieloma punktami wejścia.

Implementacja instalowalnej usługi jest nazywana *komponentem usługi*. Można użyć komponentów dostarczonych z produktem IBM MQ lub napisać własny komponent w celu wykonania wymaganych funkcji.

Obecnie dostarczane są następujące usługi instalowalne:

Usługa autoryzacji

Usługa autoryzacji umożliwia utworzenie własnego narzędzia zabezpieczeń.

Domyślnym komponentem usługi, który implementuje usługę, jest menedżer uprawnień do obiektów (object authority manager-OAM). Domyślnie moduł OAM jest aktywny i nie trzeba wykonywać żadnych czynności, aby go skonfigurować. Za pomocą interfejsu usługi autoryzacji można utworzyć inne komponenty w celu zastąpienia lub rozszerzenia OAM. Więcej informacji na temat OAM zawiera sekcja [Konfigurowanie zabezpieczeń w systemach AIX, Linux, and Windows](#).

usługa nazw

Usługa nazw umożliwia aplikacjom współużytkowanie kolejek przez identyfikowanie kolejek zdalnych tak, jakby były kolejkami lokalnymi.

Istnieje możliwość napisania własnego komponentu usługi nazw. Można to zrobić, jeśli usługa nazw ma być używana na przykład z produktem IBM MQ. Aby można było korzystać z usługi nazw, musi istnieć komponent napisany przez użytkownika lub dostarczony przez innego dostawcę oprogramowania. Domyślnie usługa nazw jest nieaktywna.



Pojęcia pokrewne

[Wyjścia użytkownika, wyjścia funkcji API i instalowalne usługi systemu IBM MQ](#)

Interfejsy języka IBM MQ Java

IBM MQ udostępnia trzy aplikacyjne interfejsy programistyczne (API) do użytku w aplikacjach Java : IBM MQ classes for Jakarta Messaging, IBM MQ classes for JMS i IBM MQ classes for Java.

IBM obsługuje i jest aktywnym uczestnikiem otwartych standardów.

- Począwszy od wersji IBM MQ 8.0 produkt implementuje standard JMS 2.0 , który wprowadza nowy uproszczony interfejs API wraz z funkcjami, takimi jak subskrypcje współużytkowane.
-   Od wersji IBM MQ 9.3.0 produkt [Jakarta Messaging 3.0](#) jest również obsługiwany.
- Ponadto WebSphere Liberty obsługuje JMS 2.0 i Jakarta Messaging 3.0 z IBM MQ.

W programie IBM MQ istnieją trzy interfejsy API do użycia w aplikacjach Java :

IBM MQ classes for Jakarta Messaging

IBM MQ classes for Jakarta Messaging jest dostawcą Jakarta Messaging , który implementuje interfejsy Jakarta Messaging dla produktu IBM MQ jako system przesyłania komunikatów. Produkt Jakarta Connectors Architecture udostępnia standardowy sposób łączenia aplikacji działających w środowisku Jakarta EE z systemem informacyjnym przedsiębiorstwa (Enterprise Information System-EIS), takim jak IBM MQ lub Db2.

IBM MQ classes for JMS

IBM MQ classes for JMS jest dostawcą JMS , który implementuje interfejsy JMS dla produktu IBM MQ jako system przesyłania komunikatów. Java Platform, Enterprise Edition Connector Architecture (JCA) udostępnia standardowy sposób łączenia aplikacji działających w środowisku Java EE z systemem EIS (Enterprise Information System), takim jak IBM MQ lub Db2.

IBM MQ classes for Java

IBM MQ classes for Java umożliwia korzystanie z produktu IBM MQ w środowisku Java . IBM MQ classes for Java Zezwala aplikacje Java na nawiązywanie połączenia z produktem IBM MQ jako

klientem IBM MQ lub nawiązywanie połączenia bezpośrednio z menedżerem kolejek produktu IBM MQ .

Uwaga:

- **V 9.3.0** **V 9.3.0** JMS 2.0 został zastąpiony przez Jakarta Messaging. Produkt IBM MQ classes for JMS nadal obsługuje standard JMS 2.0 , ale przyszłe rozszerzenia przesyłania komunikatów produktu Java będą pojawiać się tylko w produkcie Jakarta Messaging, a więc w produkcie IBM MQ classes for Jakarta Messaging. IBM MQ classes for JMS są zalecane tylko w przypadku konserwacji i rozszerzania istniejących aplikacji JMS 2.0 . IBM MQ classes for Jakarta Messaging powinna być preferowaną technologią dla nowego programowania.
- **Stabilized** IBM MQ classes for Java są funkcjonalnie ustabilizowane na poziomie dostarczonym z produktem IBM MQ 8.0. Istniejące aplikacje korzystające z produktu IBM MQ classes for Java będą nadal w pełni obsługiwane, ale ten interfejs API jest ustabilizowany, więc nowe funkcje nie będą dodawane, a żądania dotyczące rozszerzeń będą odrzucane. W pełni obsługiwane oznacza, że defekty zostaną usunięte wraz ze wszystkimi zmianami wymaganymi przez zmiany w wymaganiach systemowych produktu IBM MQ .

V 9.3.0 **JM 3.0** **V 9.3.0** Począwszy od wersji IBM MQ 9.3, IBM MQ classes for Java, IBM MQ classes for JMS i IBM MQ classes for Jakarta Messaging są budowane przy użyciu języka Java 8. Środowiska wykonawcze Java na tych lub wyższych poziomach muszą być używane do uruchamiania aplikacji za pomocą tych interfejsów.

Pojęcia pokrewne

Uzyskiwanie dostępu do produktu IBM MQ z poziomu produktu Java -wybór interfejsu API

V 9.3.0 **V 9.3.0** Dlaczego należy używać klas IBM MQ na potrzeby usługi Jakarta Messaging?

Dlaczego należy używać IBM MQ classes for JMS?

Dlaczego należy używać IBM MQ classes for Java?

IBM MQ classes for JMS/Jakarta Messaging

IBM MQ classes for JMS i IBM MQ classes for Jakarta Messaging są dostawcami przesyłania komunikatów, które są dostarczane z produktem IBM MQ. Każdy z tych dostawców udostępnia również dwa zestawy rozszerzeń interfejsu API przesyłania komunikatów. Zarówno aplikacje Java Platform, Standard Edition (Java SE), jak i Java Platform, Enterprise Edition (Java EE) mogą korzystać z tych dostawców przesyłania komunikatów.

V 9.3.0 **JM 3.0** **V 9.3.0** IBM MQ 9.3.0 wprowadzono obsługę produktu Jakarta Messaging 3.0. Produkt JMS 2.0 jest nadal w pełni obsługiwany.

Specyfikacje JMS i Jakarta Messaging definiują zestaw interfejsów, które mogą być używane przez aplikacje do wykonywania operacji przesyłania komunikatów. Począwszy od wersji IBM MQ 8.0, produkt obsługuje wersję JMS 2.0 standardu JMS . Ta implementacja oferuje wszystkie funkcje klasycznego interfejsu API, ale wymaga mniejszej liczby interfejsów i jest prostsza w użyciu. Więcej informacji na ten temat zawiera sekcja “Model JMS i Jakarta Messaging” na stronie 158 oraz specyfikacja JMS

2.0 w serwisie Java.net. **JM 3.0** Od wersji IBM MQ 9.3.0 obsługiwany jest również język Jakarta Messaging .

Pakiet jakarta.jms (Jakarta Messaging 3.0) lub javax.jms (JMS 2.0) określa szczegóły interfejsów przesyłania komunikatów, a dostawca przesyłania komunikatów implementuje te interfejsy dla konkretnego produktu przesyłania komunikatów. Na przykład:

- IBM MQ classes for JMS jest dostawcą JMS , który implementuje interfejsy JMS dla IBM MQ , a także udostępnia dwa następujące zestawy rozszerzeń dla interfejsu API JMS :
 - Rozszerzenia IBM MQ JMS
 - Rozszerzenia IBM JMS

- Fabryka połączeń, kolejka lub obiekt tematu utworzony za pomocą interfejsów `javax.jms` lub `jakarta.jms` dowolnego zestawu rozszerzeń JMS mogą być adresowane przy użyciu dowolnej z tych interfejsów API, co oznacza, że mogą być rzutowane na dowolny z tych interfejsów. Aby zachować przenośność aplikacji na najwyższym poziomie, należy użyć najbardziej ogólnego interfejsu API, który jest odpowiedni dla potrzeb użytkownika.

Ponieważ JMS i Jakarta Messaging współużytkują wiele wspólnych elementów, dalsze odwołania do JMS w tym temacie mogą być traktowane jako odwołania do obu tych elementów. Wszelkie różnice są podświetlane w razie potrzeby.

Rozszerzenia IBM MQ JMS

IBM MQ classes for JMS udostępnia również rozszerzenia interfejsu API języka JMS. IBM MQ classes for JMS zawiera rozszerzenia zaimplementowane w obiektach `MQConnectionFactory`, `MQQueue` i `MQTopic`. Te obiekty mają właściwości i metody specyficzne dla produktu IBM MQ. Obiekty mogą być obiektami administrowanymi lub aplikacja może tworzyć obiekty dynamicznie w czasie wykonywania. Te rozszerzenia są określane jako rozszerzenia produktu IBM MQ JMS. Należy zauważyć, że w tej dokumentacji obiekty, które są tworzone dynamicznie przez aplikację w czasie wykonywania, nie są traktowane jako obiekty administrowane.

Rozszerzenia IBM JMS

Oprócz rozszerzeń IBM MQ JMS produkt IBM MQ classes for JMS udostępnia bardziej ogólny zestaw rozszerzeń dla interfejsu API JMS lub Java, które są używane w języku programowania. Te rozszerzenia są określane jako rozszerzenia produktu IBM JMS i mają następujące ogólne cele:

- Zapewnienie wyższego poziomu spójności między dostawcami IBM JMS.
- Ułatwienie pisania aplikacji pomostowej między dwoma systemami przesyłania komunikatów IBM.
- Ułatwia przeniesienie aplikacji z jednego dostawcy IBM JMS do innego.

Głównym celem tych rozszerzeń jest dynamiczne tworzenie i konfigurowanie fabryk połączeń i miejsc docelowych w czasie wykonywania, ale udostępniają one również funkcje, które nie są bezpośrednio związane z przesyłaniem komunikatów, takie jak funkcje służące do określania problemów.

Zadania pokrewne

[Używanie klas IBM MQ dla przesyłania komunikatów JMS/Jakarta](#)

[Konfigurowanie zasobów JMS i Jakarta Messaging](#)

IBM MQ classes for Jakarta Messaging: przegląd

W produkcie IBM MQ 9.3.0 wprowadzono obsługę języka Jakarta Messaging. W przypadku produktu Jakarta Messaging 3.0 sterowanie specyfikacją JMS przeniesiono z produktu Oracle do produktu Java Community Process. Jednak Oracle zachowuje kontrolę nad nazwą "javax", która jest używana w innych technologiach Java. Mimo że Jakarta Messaging 3.0 jest funkcjonalnie odpowiednikiem JMS 2.0, istnieją pewne różnice w nazewnictwie. Oficjalną nazwą wersji 3.0 jest Jakarta Messaging, a nie Java Message Service, a nazwy pakietów i stałych są poprzedzone przedrostkiem `jakarta`, a nie `javax`.

Tło

Od wielu lat platforma Java jest w dwóch formach - Standard Edition i Enterprise Edition.

Java Platform, Standard Edition (czasami skrócony jako Java SE) jest podstawowym językiem i bibliotekami klas, które mogą być uruchamiane w kontekście autonomicznym. Większość pakietów Java w pliku Java SE ma nazwy rozpoczynające się od łańcucha "java".

Java Platform, Enterprise Edition (Java EE) rozszerza tę funkcjonalność, dodając takie funkcje, jak przesyłanie komunikatów, różne komponenty bean, transakcyjność itd. Niektóre z tych technologii mogą być również używane w kontekście Java SE. Większość pakietów Java w produkcie Java EE historycznie

ma nazwy rozpoczynające się od łańcucha "javax."-istnieją jednak powiązania, więc niektóre pakiety Java SE mają łańcuch "javax". jako przedrostek ich nazwy.

Java Message Service (JMS) jest częścią systemu Java Platform, Enterprise Edition. Java EE 7 zawiera JMS 2.0.

Do Java EE 7, technologie były pod zarządem Oracle.

Technologie Java EE zostały ostatnio przeniesione z aplikacji Oracle do procesu społeczności, który jest nadzorowany przez produkt Eclipse Foundation.

Jako "javax." Nazwa nie mogła zostać przeniesiona do nowego projektu, nowe nazewnictwo zostało przyjęte-wszystkie pakiety i nazwy właściwości są teraz poprzedzone przedrostkiem "jakarta." a w przyszłości Java Platform, Enterprise Edition będzie miał nazwę "Jakarta EE". Numeracja wersji jest kontynuowana: wersja 8 była wersją tymczasową, która może być w dużej mierze ignorowana, a Jakarta EE 9 jest punktem, w którym "jakarta". prefiks zaczyna obowiązywać.

Główną technologią Jakarta EE , która ma zastosowanie w kontekście IBM MQ , jest Jakarta Messaging 3.0 -następca technologii Java Message Service (JMS) 2.0. Tak więc Jakarta EE 9 zawiera Jakarta Messaging 3.0.

IBM MQ nadal obsługuje systemy Java EE 7 i JMS 2.0, wprowadzając jednocześnie obsługę systemów Jakarta EE 9 i Jakarta Messaging 3.0.

Co jest dostarczane: Java SE

Dla Java Platform, Standard Edition, oprócz IBM MQ classes for JMS (które obsługują JMS 2.0 operacje z IBM MQ) IBM MQ 9.3.0 udostępnia IBM MQ classes for Jakarta Messaging. Klasy te udostępniają dostawcę Jakarta Messaging 3.0 , który integruje się z produktem IBM MQ, umożliwiając użycie menedżerów kolejek systemu IBM MQ w celu ułatwienia wykonywania operacji Jakarta Messaging .

Są one dostarczane jako standardowy plik JAR, `com.ibm.mq.jakarta.client.jar`, w podkatalogu `java/lib` instalacji IBM MQ .

Do użycia w kontenerach OSGi, takich jak Apache Felix lub Eclipse Equinox, IBM MQ udostępnia również parę pakunków OSGi:

- `com.ibm.mq.osgi.jms30.clientprereqs_V.R.M.F.jar`
- `com.ibm.mq.osgi.jms30.client_V.R.M.F.jar`

gdzie *V.R.M.F* reprezentuje wersję produktu IBM MQ, na przykład 9.3.0.0. Pakunki te można znaleźć w podkatalogu `java/lib/OSGi` instalacji produktu IBM MQ .

Co jest dostarczane: Jakarta EE 9

Aby zapewnić obsługę przesyłania komunikatów opartego na systemie IBM MQna kompatybilnym serwerze aplikacji Jakarta EE 9 , produkt IBM MQ udostępnia adapter zasobów zgodny z produktem Jakarta EE 9: `wmq.jakarta.jmsra.rar`. Można go znaleźć w podkatalogu `java/lib/jca` instalacji produktu IBM MQ .

IBM MQ nadal udostępnia kompatybilny z Java EE 7 adapter zasobów, `wmq.jmsra.rar`, w podkatalogu `java/lib/jca` instalacji IBM MQ .

Sposób dostarczania tych artefaktów

Te pliki JAR i plik RAR adaptera zasobów są spakowane z istniejącymi wcześniej artefaktami na zwykłym nośniku instalacyjnym produktu IBM MQ -zarówno na nośniku instalacyjnym specyficznym dla platformy, takim jak pliki ".rpm", jak i na nośniku podlegającym redystrybucji, takim jak samorozpakowujące się pliki JAR klienta podlegającego redystrybucji.

Co się zmieniło między produktami JMS 2.0 i Jakarta Messaging 3.0

Jakarta EE 9 i Jakarta Messaging 3.0 nie wprowadzają żadnych nowych funkcji. Wszystko, co się zmienia, to nazwy. Na przykład w przypadku użycia klasy "javax.jms.Connection" w pliku JMS 2.0 należy użyć klasy "jakarta.jms.Connection" w pliku Jakarta Messaging 3.0.

Ponieważ platforma Eclipse Foundation przeniesie platformę Jakarta EE do przodu, zostanie zbudowana na tej podstawie, a ta konwencja nazewnictwa będzie używana dla nowych funkcji wprowadzonych w przyszłości.

Co się zmieniło między produktami IBM MQ classes for JMS i IBM MQ classes for Jakarta Messaging

Podsumowanie

Produkt IBM MQ classes for JMS, który zapewnia obsługę produktu JMS 2.0, pozostaje dostępny i jest zalecany przede wszystkim do obsługi i rozszerzania istniejących aplikacji. Są one w pełni wspierane.

Produkt IBM MQ classes for Jakarta Messaging, który zapewnia wsparcie dla produktu Jakarta Messaging 3.0, jest zalecany do nowego programowania.

W systemie IBM MQ 9.3.0 te dwie oferty są równoważne pod względem funkcjonalnym. Różni się tylko nazewnictwem. Jednak nowe funkcje przesyłania komunikatów są bardziej prawdopodobne w produkcie IBM MQ classes for Jakarta Messaging niż w produkcie IBM MQ classes for JMS.

Obie oferty są współdziałające. Komunikaty generowane przez program IBM MQ classes for JMS mogą być używane przez program IBM MQ classes for Jakarta Messaging i odwrotnie. Te dwie oferty nie mogą jednak współistnieć w jednej aplikacji.

Zmiany nazw

IBM MQ classes for JMS nazwa pakietu	IBM MQ classes for Jakarta Messaging nazwa pakietu
com.ibm.mq.jms[*]	com.ibm.mq.jakarta.jms[*]
com.ibm.jms	com.ibm.jakarta.jms
com.ibm.msg.client.jms.*	com.ibm.msg.client.jakarta.jms.*
com.ibm.msg.client.wmq.*	com.ibm.msg.client.jakarta.wmq.*

Pakiety związane ze wspólnymi usługami (śledzenie, rejestrowanie, obsługa języków narodowych itp.) i implementacjami interfejsu JMQI (lokalne i zdalne) są wspólne zarówno dla produktów IBM MQ classes for JMS, jak i IBM MQ classes for Jakarta Messaging, dlatego w tych obszarach nie są wymagane żadne zmiany.

Należy zauważyć, że nazwy właściwości również uległy zmianie. Na przykład właściwość do włączenia rozszerzeń IBM MQ w pliku IBM MQ classes for Jakarta Messaging to **com.ibm.mq.jakarta.jms.SupportMQExtensions**.

Nazwy właściwości, które są niezależne od IBM MQ classes for JMS lub IBM MQ classes for Jakarta Messaging, takie jak różne właściwości **com.ibm.msg.client.commonservices.trace.***, mają zastosowanie jednakowo do obu ofert.

Narzędzia administracyjne

Programy narzędziowe **crtmqenv** i **setmqenv** akceptują teraz opcję określającą, czy ścieżka klasy powinna być skonfigurowana dla IBM MQ classes for JMS (-j 2.0), czy IBM MQ classes for Jakarta Messaging (-j 3.0), a istnieją IBM MQ classes for Jakarta Messaging warianty programów narzędziowych **runjms** o nazwie **runjms30** i podobnych nazwach.

Program narzędziowy **dspmqver**, na żądanie raportowania komponentów Java, dołącza do danych wyjściowych łańcuch IBM MQ classes for Jakarta Messaging.

Aby skonfigurować pobieranie obiektów IBM MQ classes for Jakarta Messaging za pośrednictwem interfejsu JNDI, nowy program narzędziowy **JMS30Admin** jest równoważny programowi narzędziowemu **JMSAdmin** dla IBM MQ classes for JMS.

Należy zauważyć, że ponieważ obiekty bazowe pochodzą z różnych pakietów. JNDI definitions created by **JMSAdmin** cannot be used by IBM MQ classes for Jakarta Messaging, nor can those created by **JMS30Admin** be used by IBM MQ classes for JMS.

Uwaga: Obiekty IBM MQ classes for Jakarta Messaging nie są obsługiwane przez produkt IBM MQ Explorer; jego integracja JNDI jest przeznaczona tylko dla produktu IBM MQ classes for JMS.


Pojęcia pokrewne

Dlaczego należy używać klas IBM MQ na potrzeby usługi Jakarta Messaging?

Model JMS i Jakarta Messaging

Model JMS i Jakarta Messaging definiują zestaw interfejsów, które mogą być używane przez aplikacje Java do wykonywania operacji przesyłania komunikatów. IBM MQ classes for JMS i IBM MQ classes for Jakarta Messaging są dostawcami przesyłania komunikatów, które definiują sposób, w jaki obiekty przesyłania komunikatów produktu Java są powiązane z pojęciami związanymi z produktem IBM MQ. Specyfikacje JMS i Jakarta Messaging oczekują, że niektóre obiekty przesyłania komunikatów będą administrowane.

Począwszy od wersji IBM MQ 8.0, produkt obsługuje wersję JMS 2.0 standardu JMS, która wprowadziła uproszczony interfejs API, zachowując jednocześnie klasyczny interfejs API w produkcie JMS 1.1.

 IBM MQ 9.3.0 wprowadzono obsługę produktu Jakarta Messaging 3.0. Produkt JMS 2.0 jest nadal w pełni obsługiwany. Ponieważ JMS i Jakarta Messaging współużytkują wiele wspólnych elementów, dalsze odwołania do JMS w tym temacie mogą być traktowane jako odwołania do obu tych elementów. Wszelkie różnice są podświetlane w razie potrzeby.

Uproszczony interfejs API

Firma JMS 2.0 wprowadziła uproszczony interfejs API, zachowując jednocześnie specyficzne dla domeny i niezależne od domeny interfejsy produktu JMS 1.1. Uproszczony interfejs API zmniejsza liczbę obiektów potrzebnych do wysyłania i odbierania komunikatów i składa się z następujących interfejsów:

ConnectionFactory

ConnectionFactory to obiekt administrowany, który jest używany przez klienta JMS do tworzenia połączenia. Ten interfejs jest również używany w klasycznym interfejsie API.

JMSKontekst

Ten obiekt łączy obiekty połączenia i sesji klasycznego interfejsu API. JMSObiekty kontekstowe można tworzyć na podstawie innych obiektów JMSContext, przy czym połączenie bazowe jest duplikowane.

JMSProducent

Producent JMS jest tworzony przez kontekst JMSi jest używany do wysyłania komunikatów do kolejki lub tematu. Obiekt producenta JMS powoduje utworzenie obiektów wymaganych do wysłania komunikatu.

JMSKonsument

Konsument JMS jest tworzony przez kontekst JMSi używany do odbierania komunikatów z tematu lub kolejki.

Uproszczony interfejs API ma wiele efektów:

- Obiekt kontekstu JMS zawsze automatycznie uruchamia połączenie bazowe.
- JMSProducent i JMSKonsument mogą teraz pracować bezpośrednio z treścią komunikatu bez konieczności pobierania całego obiektu komunikatu przy użyciu metody `getBody` komunikatu.
- Właściwości komunikatu można ustawić w obiekcie producenta JMS przy użyciu łańcucha metod przed wysłaniem treści komunikatu. Producent JMS będzie obsługiwać tworzenie wszystkich obiektów, które

są potrzebne do wystania komunikatu. Za pomocą parametru JMS 2.0 można ustawić właściwości i wysłać komunikat w następujący sposób:

```
context.createProducer().
setProperty("foo", "bar").
setTimeToLive(10000).
setDeliveryMode(NON_PERSISTENT).
setDisableMessageTimestamp(true).
send(dataQueue, body);
```

W produkcie JMS 2.0 wprowadzono również subskrypcje współużytkowane, w których komunikaty mogą być współużytkowane przez wielu konsumentów. Wszystkie subskrypcje programu JMS 1.1 są traktowane jako subskrypcje niewspółużytkowane.

Klasyczny interfejs API

Poniższa lista zawiera podsumowanie głównych interfejsów JMS klasycznego interfejsu API:

Miejsce docelowe

Miejsce docelowe to miejsce, w którym aplikacja wysyła komunikaty lub jest źródłem, z którego aplikacja odbiera komunikaty, lub oba te elementy.

ConnectionFactory

Obiekt ConnectionFactory hermetyzuje zestaw właściwości konfiguracyjnych połączenia. Aplikacja używa fabryki połączeń do utworzenia połączenia.

Połączenie

Obiekt połączenia hermetyzuje aktywne połączenie aplikacji z serwerem przesyłania komunikatów. Aplikacja używa połączenia do tworzenia sesji.

Sesja

Sesja jest jednowątkowym kontekstem do wysyłania i odbierania komunikatów. Aplikacja używa sesji do tworzenia komunikatów, producentów komunikatów i konsumentów komunikatów. Sesja jest transakcją lub nie jest transakcją.

Komunikat

Obiekt Message hermetyzuje komunikat wysyłany lub odbierany przez aplikację.

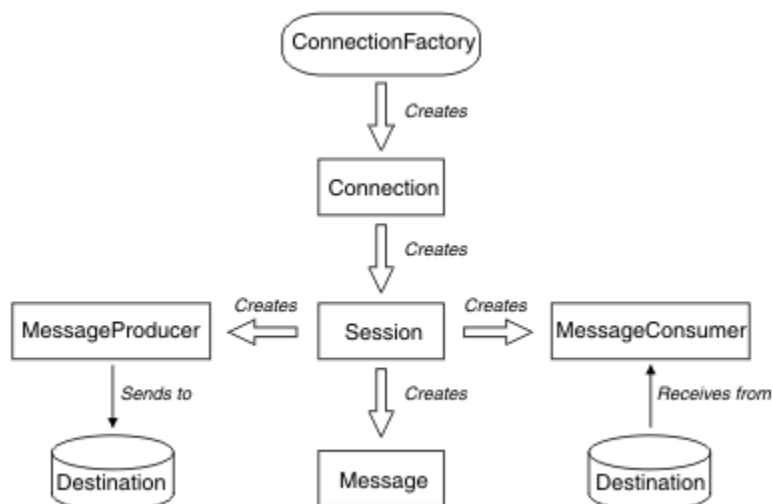
MessageProducer

Aplikacja używa producenta komunikatów do wysyłania komunikatów do miejsca docelowego.

MessageConsumer

Aplikacja używa konsumenta komunikatów do odbierania komunikatów wysyłanych do miejsca docelowego.

Rysunek 53 na stronie 159 przedstawia te obiekty i ich relacje.



Rysunek 53. Obiekty JMS i ich relacje

Diagram przedstawia główne interfejsy: ConnectionFactory, Connection, Session, MessageProducer, MessageConsumer, Message i Destination. Aplikacja używa fabryki połączeń do utworzenia połączenia i używa połączenia do utworzenia sesji. Aplikacja może następnie użyć sesji do utworzenia komunikatów, producentów komunikatów i konsumentów komunikatów. Aplikacja używa producenta komunikatów do wysyłania komunikatów do miejsca docelowego, a konsumenta komunikatów do odbierania komunikatów wysyłanych do miejsca docelowego.

Obiekt docelowy, obiekt ConnectionFactory lub obiekt Connection mogą być używane wspólnie przez różne wątki aplikacji wielowątkowej, ale obiekt Session, MessageProducer lub MessageConsumer nie może być używany wspólnie przez różne wątki. Najprostszym sposobem zapewnienia, że obiekt Session, MessageProducer lub MessageConsumer nie jest używany wspólnie, jest utworzenie osobnego obiektu Session dla każdego wątku.

Produkt JMS obsługuje dwa style przesyłania komunikatów:

- Przesyłanie komunikatów w modelu punkt-punkt
- Przesyłanie komunikatów w trybie publikowania/subskrypcji

Te style przesyłania komunikatów są również nazywane *domenami przesyłania komunikatów* można łączyć oba style przesyłania komunikatów w aplikacji. W domenie typu punkt z punktem miejsce docelowe jest kolejką, a w domenie typu publikowanie/subskrypcja miejscem docelowym jest temat.

W wersjach JMS wcześniejszych niż JMS 1.1 programowanie w domenie typu punkt z punktem używa jednego zestawu interfejsów i metod, a programowanie w domenie publikowania/subskrypcji używa innego zestawu. Dwa zestawy są podobne, ale oddzielne. W produkcie JMS 1.1 można używać wspólnego zestawu interfejsów i metod, które obsługują obie domeny przesyłania komunikatów. Wspólne interfejsy udostępniają niezależny od domeny widok każdej domeny przesyłania komunikatów. [Tabela 17 na stronie 160](#) zawiera listę niezależnych interfejsów domeny JMS i odpowiadających im interfejsów specyficznych dla domeny.

<i>Tabela 17. Niezależna domena JMS i odpowiadające jej interfejsy specyficzne dla domeny</i>		
Interfejsy niezależne od domeny	Interfejsy specyficzne dla domeny typu punkt z punktem	Interfejsy specyficzne dla domeny publikowania/subskrypcji
ConnectionFactory	Fabryka QueueConnection	Fabryka TopicConnection
Połączenie	QueueConnection	TopicConnection
Miejsce docelowe	Kolejka	Temat
Sesja	QueueSession	TopicSession
MessageProducer	QueueSender	TopicPublisher
MessageConsumer	QueueReceiver QueueBrowser	TopicSubscriber

JMS 2.0 Produkt IBM MQ classes for JMS 2.0 obsługuje zarówno wcześniejsze interfejsy specyficzne dla domeny JMS 1.1, jak i uproszczony interfejs API produktu JMS 2.0. Produkt IBM MQ classes for JMS 2.0 może być używany do obsługi istniejących aplikacji, w tym do tworzenia nowych funkcji w istniejących aplikacjach.

JM 3.0 Produkt IBM MQ classes for Jakarta Messaging 3.0 obsługuje wersje Jakarta Messaging tych samych interfejsów i jest zalecany do tworzenia nowych aplikacji.

W systemach IBM MQ classes for JMS i IBM MQ classes for Jakarta Messaging obiekty JMS są powiązane z pojęciami IBM MQ w następujący sposób:

- Obiekt połączenia ma właściwości, które pochodzą z właściwości fabryki połączeń użytej do utworzenia połączenia. Te właściwości sterują sposobem, w jaki aplikacja nawiązuje połączenie z menedżerem kolejek. Przykładami tych właściwości są nazwa menedżera kolejek oraz, w przypadku aplikacji

nawiązującej połączenie z menedżerem kolejek w trybie klienta, nazwa hosta lub adres IP systemu, w którym działa menedżer kolejek.

- Obiekt Session hermetyzuje uchwyt połączenia IBM MQ , który definiuje zasięg transakcyjny sesji.
- Każdy z obiektów MessageProducer i MessageConsumer hermetyzuje uchwyt obiektu IBM MQ .

W przypadku korzystania z systemu IBM MQ classes for JMS lub IBM MQ classes for Jakarta Messaging mają zastosowanie wszystkie normalne reguły IBM MQ . W szczególności należy zauważyć, że aplikacja może wysłać komunikat do kolejki zdalnej, ale może odebrać komunikat tylko z kolejki, której właścicielem jest menedżer kolejek, z którym połączona jest aplikacja.

Specyfikacja JMS oczekuje, że obiekty ConnectionFactory i Destination będą administrowane. Administrator tworzy i obsługuje obiekty administrowane w centralnym repozytorium, a aplikacja JMS pobiera te obiekty za pomocą interfejsu JNDI (Java Naming and Directory Interface).


W systemach IBM MQ classes for JMS i IBM MQ classes for Jakarta Messaging implementacja interfejsu miejsca docelowego jest abstrakcyjną nadklasą klasy Queue i Topic, a więc instancją klasy Destination jest obiekt kolejki lub obiekt tematu. Niezależnie od domeny interfejsy traktują kolejkę lub temat jako miejsce docelowe. Domena przesyłania komunikatów dla obiektu MessageProducer lub MessageConsumer jest określana na podstawie tego, czy miejsce docelowe jest kolejką, czy tematem.

Dlatego w systemach IBM MQ classes for JMS i IBM MQ classes for Jakarta Messaging można administrować obiektami następujących typów:

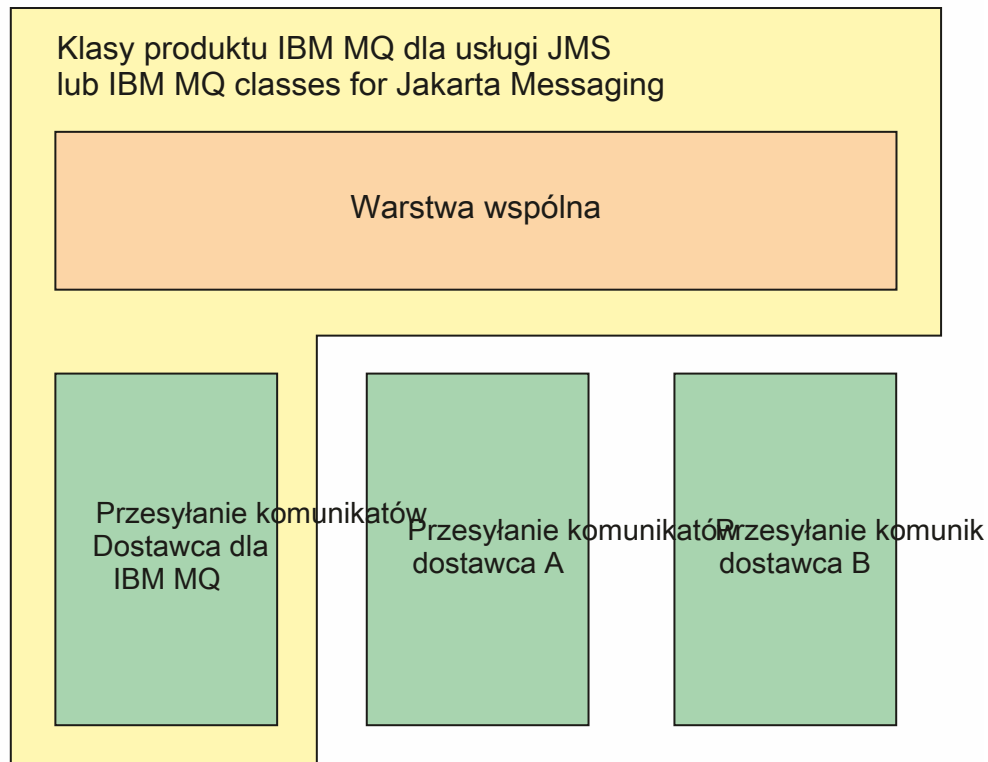
- ConnectionFactory
- Fabryka QueueConnection
- Fabryka TopicConnection
- Kolejka
- Temat
- XAConnectionFactory
- Fabryka XAQueueConnection
- Fabryka XATopicConnection

Architektura IBM MQ classes for JMS/Jakarta Messaging

IBM MQ classes for JMS i IBM MQ classes for Jakarta Messaging mają architekturę warstwową. Warstwa najwyższego poziomu kodu jest wspólną warstwą, z której może korzystać każdy dostawca przesyłania komunikatów produktu IBM Java .

 IBM MQ 9.3.0 wprowadzono obsługę produktu [Jakarta Messaging 3.0](#). Produkt JMS 2.0 jest nadal w pełni obsługiwany.

IBM MQ classes for JMS i IBM MQ classes for Jakarta Messaging mają architekturę warstwową przedstawioną na diagramie [Rysunek 54 na stronie 162](#). Warstwa najwyższego poziomu kodu jest wspólną warstwą, która może być używana przez dowolnego dostawcę IBM JMS lub Jakarta Messaging. Gdy aplikacja wywołuje metodę JMS lub Jakarta Messaging, przetwarzanie wywołania, które nie jest specyficzne dla systemu przesyłania komunikatów, jest wykonywane przez wspólną warstwę, która również zapewnia spójną odpowiedź na wywołanie. Przetwarzanie wywołania, które jest specyficzne dla systemu przesyłania komunikatów, jest delegowane do niższej warstwy. Na poniższym diagramie przedstawiono dostawcę przesyłania komunikatów produktu IBM MQ w niższej warstwie wraz z dwoma innymi dostawcami przesyłania komunikatów (dostawca przesyłania komunikatów A i dostawca przesyłania komunikatów B.).



Rysunek 54. Architektura warstwowa dla dostawców IBM JMS i Jakarta Messaging

Architektura warstwowa spełnia następujące cele:

- W celu zwiększenia spójności zachowania różnych dostawców IBM JMS i Jakarta Messaging
- Aby ułatwić pisanie aplikacji pomostowej między dwoma systemami przesyłania komunikatów produktu IBM .
- Aby ułatwić przeniesienie aplikacji z jednego dostawcy IBM JMS lub Jakarta Messaging do innego

Zadania pokrewne

[Używanie klas IBM MQ dla przesyłania komunikatów JMS/Jakarta](#)

Obsługa obiektów administrowanych

Systemy IBM MQ classes for JMS i IBM MQ classes for Jakarta Messaging obsługują użycie obiektów administrowanych.

V 9.3.0 **JM 3.0** **V 9.3.0** W produkcie IBM MQ 9.3.0 produkt Jakarta Messaging 3.0 jest obsługiwany na potrzeby tworzenia nowych aplikacji. Produkt IBM MQ 9.3.0 nadal obsługuje produkt JMS 2.0 dla istniejących aplikacji. Nie jest obsługiwane używanie zarówno interfejsu API Jakarta Messaging 3.0 , jak i interfejsu API JMS 2.0 w tej samej aplikacji. Więcej informacji na ten temat zawiera sekcja [Używanie klas IBM MQ dla przesyłania komunikatów JMS/Jakarta](#).

Przeptyw logiki w aplikacji JMS lub IBM MQ classes for Jakarta Messaging rozpoczyna się od obiektów ConnectionFactory i Destination. Aplikacja używa obiektu ConnectionFactory do utworzenia obiektu Connection, który reprezentuje aktywne połączenie między aplikacją a serwerem przesyłania komunikatów. Aplikacja używa obiektu połączenia do utworzenia obiektu sesji, który jest jednowątkowym kontekstem do generowania i odbierania komunikatów. Aplikacja może następnie użyć obiektu Session i obiektu Destination do utworzenia obiektu MessageProducer , który będzie używany przez aplikację do wysyłania komunikatów do określonego miejsca docelowego. Miejsce docelowe jest kolejką lub tematem w systemie przesyłania komunikatów i jest hermetyzowane przez obiekt docelowy. Aplikacja może również użyć obiektu Session i obiektu Destination do utworzenia obiektu MessageConsumer ,

który będzie używany przez aplikację do odbierania komunikatów wysłanych do określonego miejsca docelowego.

Specyfikacje JMS i Jakarta Messaging oczekują, że obiekty `ConnectionFactory` i `Destination` będą administrowane. Administrator tworzy i obsługuje obiekty administrowane w centralnym repozytorium, a aplikacja JMS lub Jakarta Messaging pobiera te obiekty za pomocą programu Java Naming Directory Interface (JNDI). Repozytorium obiektów administrowanych może mieć zakres od prostego pliku do katalogu LDAP (Lightweight Directory Access Protocol).

Systemy IBM MQ classes for JMS i IBM MQ classes for Jakarta Messaging obsługują użycie obiektów administrowanych. Aplikacja może korzystać ze wszystkich funkcji produktu IBM MQ classes for JMS lub IBM MQ classes for Jakarta Messaging, które są udostępniane za pośrednictwem produktu IBM MQ, bez konieczności wprowadzania do aplikacji jakichkolwiek informacji specyficznych dla produktu IBM MQ. Dzięki temu aplikacja będzie w pewnym stopniu niezależna od bazowej konfiguracji produktu IBM MQ.

Aby uzyskać tę niezależność, aplikacja może używać interfejsu JNDI do pobierania fabryk połączeń i miejsc docelowych, które są przechowywane jako obiekty administrowane, oraz do wykonywania operacji przesyłania komunikatów używać tylko interfejsów zdefiniowanych w pakiecie `javax.jms` (JMS 2.0) lub `jakarta.jms` (Jakarta Messaging 3.0).

JMS 2.0 W przypadku systemu JMS 2.0 administrator może użyć narzędzia administracyjnego IBM MQ JMS **JMSAdmin** lub IBM MQ Explorer, aby utworzyć i obsługiwać obiekty administrowane w centralnym repozytorium.

JM 3.0 W przypadku systemu Jakarta Messaging 3.0 nie można administrować interfejsem JNDI przy użyciu programu IBM MQ Explorer. Administrowanie interfejsem JNDI jest obsługiwane przez Jakarta Messaging 3.0 wariant **JMSAdmin**, który ma wartość **JMS30Admin**.

Serwer aplikacji zwykle udostępnia własne repozytorium dla obiektów administrowanych oraz własne narzędzia do tworzenia i obsługi obiektów. Z tego powodu aplikacja Java EE **JM 3.0** lub Jakarta EE może używać języka JNDI do pobierania obiektów administrowanych z repozytorium serwera aplikacji lub z centralnego repozytorium.

Zadania pokrewne

[Konfigurowanie zasobów JMS i Jakarta Messaging](#)

Obsługiwane typy komunikacji na platformach Java EE i Jakarta EE

Na platformach Java EE i Jakarta EE produkty IBM MQ classes for JMS i IBM MQ classes for Jakarta Messaging obsługują dwa typy komunikacji między komponentem aplikacji i menedżerem kolejek produktu IBM MQ.

V 9.3.0 **JM 3.0** **V 9.3.0** IBM MQ 9.3.0 wprowadzono obsługę produktu Jakarta Messaging 3.0. Produkt JMS 2.0 jest nadal w pełni obsługiwany. Ponieważ JMS i Jakarta Messaging współużytkują wiele wspólnych elementów, dalsze odwołania do JMS w tym temacie mogą być traktowane jako odwołania do obu tych elementów. Wszelkie różnice są podświetlane w razie potrzeby.

Obsługiwane są następujące dwa typy komunikacji między komponentem aplikacji i menedżerem kolejek produktu IBM MQ:

- Komunikacja wychodząca
- Komunikacja przychodząca

Komunikacja wychodząca

Bezpośrednio przy użyciu funkcji API języka JMS lub Jakarta Messaging komponent aplikacji tworzy połączenie z menedżerem kolejek, a następnie wysyła i odbiera komunikaty.

Na przykład komponentem aplikacji może być klient aplikacji, serwlet, strona JSP (Java Server Page), komponent EJB (Enterprise Java Bean) lub komponent MDB (Message Driven Bean). W tym typie

komunikacji kontener serwera aplikacji udostępnia tylko funkcje niskiego poziomu w zakresie obsługi operacji przesyłania komunikatów, takich jak zestawianie połączeń i zarządzanie wątkami.

Komunikacja przychodząca

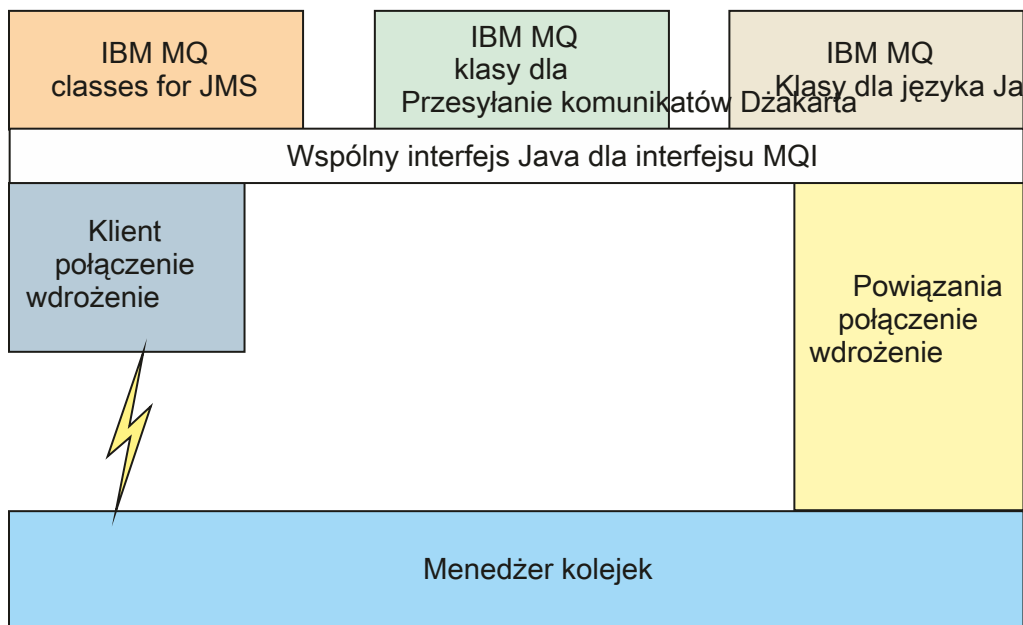
W przypadku komunikacji przychodzącej komunikat przychodzący do miejsca docelowego jest dostarczany do komponentu MDB, który następnie przetwarza komunikat.

Aplikacje Java EE **JM 3.0** i Jakarta EE używają komponentów MDB do asynchronicznego przetwarzania komunikatów. Komponent MDB działa jako obiekt nasłuchiwanie komunikatów JMS i jest implementowany przy użyciu metody `onMessage()`, która definiuje sposób przetwarzania komunikatu. Komponent MDB jest wdrażany w kontenerze EJB serwera aplikacji. Dokładny sposób konfigurowania komponentu MDB zależy od używanego serwera aplikacji, ale informacje o konfiguracji muszą określać, z którym menedżerem kolejek ma zostać nawiązane połączenie, jak nawiązać połączenie z menedżerem kolejek, które miejsce docelowe ma być monitorowane pod kątem komunikatów oraz zachowanie transakcyjne komponentu MDB. Te informacje są następnie używane przez kontener EJB. Gdy komunikat spełniający kryteria wyboru komponentu MDB dociera do określonego miejsca docelowego, kontener EJB używa metody IBM MQ classes for JMS lub IBM MQ classes for Jakarta Messaging do pobrania komunikatu z menedżera kolejek, a następnie dostarcza komunikat do komponentu MDB, wywołując metodę `onMessage()`.

Relacja z IBM MQ classes for Java

Systemy IBM MQ classes for Java, IBM MQ classes for Jakarta Messaging i IBM MQ classes for JMS są węzłami sieci, które używają wspólnego interfejsu Java dla interfejsu MQI.

Rysunek 55 na stronie 164 przedstawia relację między IBM MQ classes for JMS, IBM MQ classes for Jakarta Messaging i IBM MQ classes for Java.



Rysunek 55. Relacja między IBM MQ classes for JMS, IBM MQ classes for Jakarta Messaging i IBM MQ classes for Java

Ogólnie rzecz biorąc, programy Java powinny używać tylko jednego interfejsu do komunikowania się z systemem IBM MQ - IBM MQ classes for Java, IBM MQ classes for Jakarta Messaging lub IBM MQ classes for JMS. Mieszanie interfejsów nie jest obsługiwane, z jednym wyjątkiem. Aby zachować zgodność z wersjami wcześniejszymi niż IBM WebSphere MQ 7.0, klasy wyjścia kanału napisane w języku Java nadal mogą używać interfejsów IBM MQ classes for Java, nawet jeśli klasy wyjścia kanału są wywoływane z pliku IBM MQ classes for JMS. Jednak użycie interfejsów IBM MQ classes for Java oznacza, że aplikacje nadal są zależne od jednego z następujących elementów:

- **JMS 2.0** Plik JAR IBM MQ classes for Java : `com.ibm.mq.jar`. Jeśli zmienna `com.ibm.mq.jar` nie ma być używana w ścieżce klasy, można użyć zestawu interfejsów w pakiecie `com.ibm.mq.exits`.
- **V 9.3.0** **JM 3.0** **V 9.3.0** Użycie `com.ibm.mq.jakarta.client.jar` podczas współpracy z produktem IBM MQ classes for Jakarta Messaging.

Pojęcia pokrewne

V 9.3.0 **V 9.3.0** [Dlaczego należy używać klas IBM MQ na potrzeby usługi Jakarta Messaging?](#)

[Dlaczego należy używać klas IBM MQ dla usługi JMS?](#)

[Dlaczego należy używać klas IBM MQ dla systemu Java?](#)

IBM MQ Dostawca komunikatów

Dostawca usługi przesyłania komunikatów produktu IBM MQ ma trzy tryby działania: tryb normalny, tryb normalny z ograniczeniami i tryb migracji.

Dostawca przesyłania komunikatów produktu IBM MQ ma trzy tryby działania:

- Tryb normalny dostawcy usługi przesyłania komunikatów produktu IBM MQ
- Tryb normalny dostawcy usługi przesyłania komunikatów produktu IBM MQ z ograniczeniami
- Tryb migracji dostawcy usługi przesyłania komunikatów produktu IBM MQ

W trybie normalnym dostawcy usługi przesyłania komunikatów produktu IBM MQ wszystkie funkcje menedżera kolejek produktu IBM MQ są używane do implementowania interfejsu JMS. Ten tryb jest zoptymalizowany pod kątem używania funkcji i interfejsu API JMS

2.0 **V 9.3.0** **JM 3.0** **V 9.3.0** lub [Jakarta Messaging 3.0](#).

Jeżeli:

- Klient określa wersję 6 dostawcy na serwerze **ConnectionFactory**. Klient zachowuje się w sposób zgodny z klientem dostarczonym z produktem IBM WebSphere MQ 6.0. Obsługiwane są tylko interfejsy JMS 1.1 i JMS 2, ale niektóre funkcje JMS 2, takie jak subskrypcje współużytkowane, opóźnienie dostarczania i wysyłanie asynchroniczne, są wyłączone. Brak współużytkowania połączenia.
- Klient określa wersję dostawcy 7 w systemie **ConnectionFactory**, zarówno interfejs JMS 1.1, jak i interfejs JMS 2 są w pełni obsługiwane.
- Nie określono wersji dostawcy. Podjęto próbę nawiązania połączenia z dostawcą w wersji 7. Jeśli to się nie powiedzie, zostanie podjęta kolejna próba z dostawcą w wersji 6.

Aby nawiązać połączenie z produktem IBM Integration Bus przy użyciu transportu IBM MQ Enterprise Transport, należy użyć trybu migracji. Jeśli używany jest transport w czasie rzeczywistym produktu IBM MQ, tryb migracji jest wybierany automatycznie, ponieważ w obiekcie fabryki połączeń zostały jawnie wybrane właściwości. Połączenie z produktem IBM Integration Bus przy użyciu transportu IBM MQ Enterprise Transport jest zgodne z ogólnymi regułami wyboru trybu, które są opisane w sekcji [Konfigurowanie właściwości JMS PROVIDERVERSION](#).

Zadania pokrewne

[Konfigurowanie zasobów JMS](#)

z/OS IBM MQ for z/OS pojęcia

Niektóre pojęcia używane przez system IBM MQ for z/OS są unikalne dla platformy z/OS. Na przykład mechanizm rejestrowania, techniki zarządzania pamięcią masową, dyspozycja jednostki odzyskiwania i grupy współużytkowania kolejek są udostępniane tylko z produktem IBM MQ for z/OS. Ten temat stanowi wprowadzenie do dalszych informacji na temat tych pojęć.

Pojęcia te obejmują przegląd obiektów używanych przez program IBM MQ for z/OS, w tym:

- Menedżer kolejek
- Inicjator kanału

- Kolejki współużytkowane i grupy współużytkowania kolejek
- Kolejowanie wewnątrz grupy

Poniższe tematy obejmują również różne potrzebne procedury, w tym:

- Definicje systemu w systemie z/OS
- Zarządzanie pamięcią masową
- Odtwarzanie i restartowanie
- Pojęcia związane z bezpieczeństwem w produkcie IBM MQ for z/OS

Pojęcia pokrewne

“Menedżer kolejek w systemie z/OS” na stronie 167

Zanim będzie można pozwolić aplikacjom korzystającym z produktu IBM MQ w systemie z/OS, należy zainstalować produkt IBM MQ for z/OS i uruchomić menedżer kolejek. Menedżer kolejek jest właścicielem zestawu zasobów używanych przez program IBM MQi zarządza nim.

“Inicjator kanału w systemie z/OS” na stronie 168

Inicjator kanału udostępnia zasoby umożliwiające kolejowanie rozproszone IBM MQ i zarządza nimi. Program IBM MQ używa *agentów kanału komunikatów* (MCA) do wysyłania komunikatów z jednego menedżera kolejek do innego.

“Terminy i zadania związane z zarządzaniem produktem IBM MQ for z/OS” na stronie 170

Ten temat stanowi wprowadzenie do terminologii i zadań specyficznych dla produktu IBM MQ for z/OS.

“Kolejki współużytkowane i grupy współużytkowania kolejek” na stronie 173

Kolejek współużytkowanych i grup współużytkowania kolejek można używać do implementowania wysokiej dostępności zasobów IBM MQ. Kolejki współużytkowane i grupy współużytkowania kolejek są funkcjami unikalnymi dla systemu IBM MQ for z/OS na platformie z/OS.

“Kolejkowanie wewnątrz grupy” na stronie 221

W tej sekcji opisano kolejowanie wewnątrz grupy, funkcję IBM MQ for z/OS unikalną dla platformy z/OS. Ta funkcja jest dostępna tylko dla menedżerów kolejek zdefiniowanych dla grupy współużytkowania kolejek.

“Zarządzanie pamięcią masową w systemie z/OS” na stronie 234

Produkt IBM MQ for z/OS wymaga trwałych i tymczasowych struktur danych oraz używa zestawów stron i buforów pamięci do przechowywania tych danych. W tych tematach podano więcej szczegółowych informacji na temat sposobu, w jaki produkt IBM MQ używa tych zestawów stron i buforów.

“logowanie IBM MQ for z/OS” na stronie 239

IBM MQ przechowuje *dzienniki* zmian danych i istotnych zdarzeń w momencie ich wystąpienia. Te dzienniki mogą być używane do odtwarzania danych do poprzedniego stanu, jeśli jest to wymagane.

“Odtwarzanie i restartowanie w systemie z/OS” na stronie 262

Skorzystaj z odsyłaczy w tym temacie, aby dowiedzieć się więcej o funkcjach produktu IBM MQ for z/OS służących do restartowania i odtwarzania.

“Pojęcia związane z bezpieczeństwem w produkcie IBM MQ for z/OS” na stronie 280

Ten temat zawiera informacje o znaczeniu zabezpieczeń dla systemu IBM MQ oraz o konsekwencjach braku odpowiednich ustawień zabezpieczeń w systemie.

“Dostępność w systemie z/OS” na stronie 287

IBM MQ for z/OS oferuje wiele funkcji wysokiej dostępności. W tej sekcji opisano niektóre zagadnienia związane z dostępnością.

“Dyspozycja jednostki odzyskiwania w systemie z/OS” na stronie 291

W przypadku połączenia z menedżerem kolejek w grupie współużytkowania kolejek (QSG) niektóre aplikacje transakcyjne mogą używać dyspozycji jednostki odzyskiwania GROUP, a nie QMGR, przez określenie nazwy QSG podczas nawiązywania połączenia zamiast nazwy menedżera kolejek. Dzięki temu odtwarzanie transakcji będzie bardziej elastyczne i stabilne, jeśli zostanie usunięte wymaganie ponownego nawiązania połączenia z tym samym menedżerem kolejek w grupie QSG.

Odsyłacze pokrewne

“Definicja systemu w systemie z/OS” na stronie 250

Produkt IBM MQ for z/OS używa wielu domyślnych definicji obiektów i udostępnia przykładowy kod JCL do tworzenia tych domyślnych obiektów. W tym temacie opisano te domyślne obiekty i przykładowy kod JCL.

[“Monitorowanie i statystyki w systemie IBM MQ for z/OS” na stronie 290](#)

Produkt IBM MQ for z/OS zawiera zestaw narzędzi do monitorowania menedżera kolejek i zbierania statystyk.

z/OS

Menedżer kolejek w systemie z/OS

Zanim będzie można pozwolić aplikacjom korzystającym z produktu IBM MQ w systemie z/OS, należy zainstalować produkt IBM MQ for z/OS i uruchomić menedżer kolejek. Menedżer kolejek jest właścicielem zestawu zasobów używanych przez program IBM MQi zarządza nim.

Menedżer kolejek

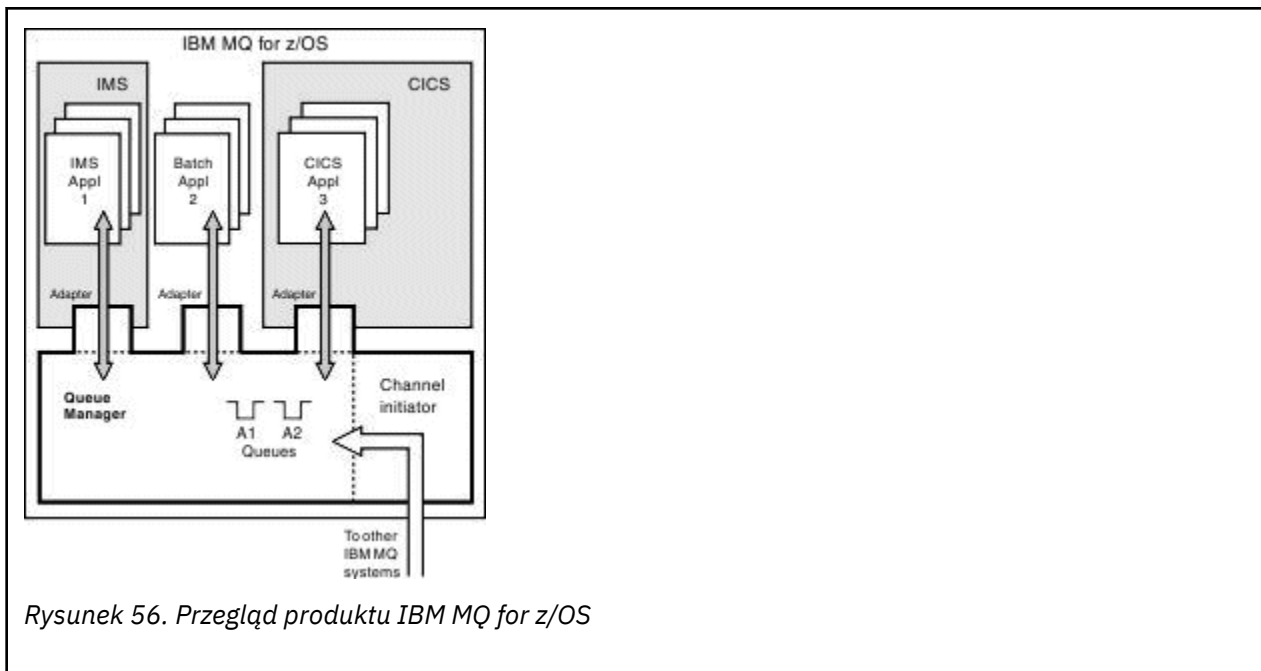
Menedżer kolejek jest programem udostępniającym aplikacjom usługi przesyłania komunikatów. Aplikacje, które używają interfejsu kolejek komunikatów (Message Queue Interface - MQI) mogą umieszczać komunikaty w kolejkach i pobierać komunikaty z kolejek. Menedżer kolejek sprawdza, czy komunikaty są wysyłane do poprawnych kolejek lub czy są kierowane do innego menedżera kolejek. Menedżer kolejek przetwarza wywołania interfejsu MQI, które są do niego wysyłane oraz komendy, które są wprowadzane do menedżera kolejek (niezależnie od źródła, z którego pochodzą). Menedżer kolejek generuje odpowiednie kody zakończenia dla każdego wywołania lub komendy.

Zasoby zarządzane przez menedżer kolejek obejmują:

- Zestawy stron zawierające definicje obiektów IBM MQ i dane komunikatów
- Dzienniki używane do odtwarzania komunikatów i obiektów w przypadku awarii menedżera kolejek
- Pamięć masowa procesora
- Połączenia, za pośrednictwem których różne środowiska aplikacji (CICS, IMSi Batch) mogą uzyskiwać dostęp do interfejsu API języka IBM MQ
- Inicjator kanału IBM MQ, który umożliwia komunikację między programem IBM MQ w systemie z/OS i innymi systemami.

Menedżer kolejek ma nazwę i aplikacje mogą nawiązywać z nim połączenie przy użyciu tej nazwy.

Na ilustracji [Rysunek 56 na stronie 168](#) przedstawiono menedżer kolejek z połączeniami do różnych środowisk aplikacji i inicjatora kanału.



Rysunek 56. Przegląd produktu IBM MQ for z/OS

Podsystem menedżera kolejek w systemie z/OS

W systemie z/OS system IBM MQ działa jako podsystem z/OS, który jest uruchamiany podczas IPL. W podsystemie menedżer kolejek jest uruchamiany przez wykonanie procedury JCL, która określa zestawy danych z/OS zawierające informacje o dziennikach oraz przechowujące definicje obiektów i dane komunikatów (zestawy stron). Podsystem i menedżer kolejek mają taką samą nazwę o długości do czterech znaków. Wszystkie menedżery kolejek w sieci muszą mieć unikalne nazwy, nawet jeśli znajdują się w różnych systemach, systemach sysplex lub platformach.

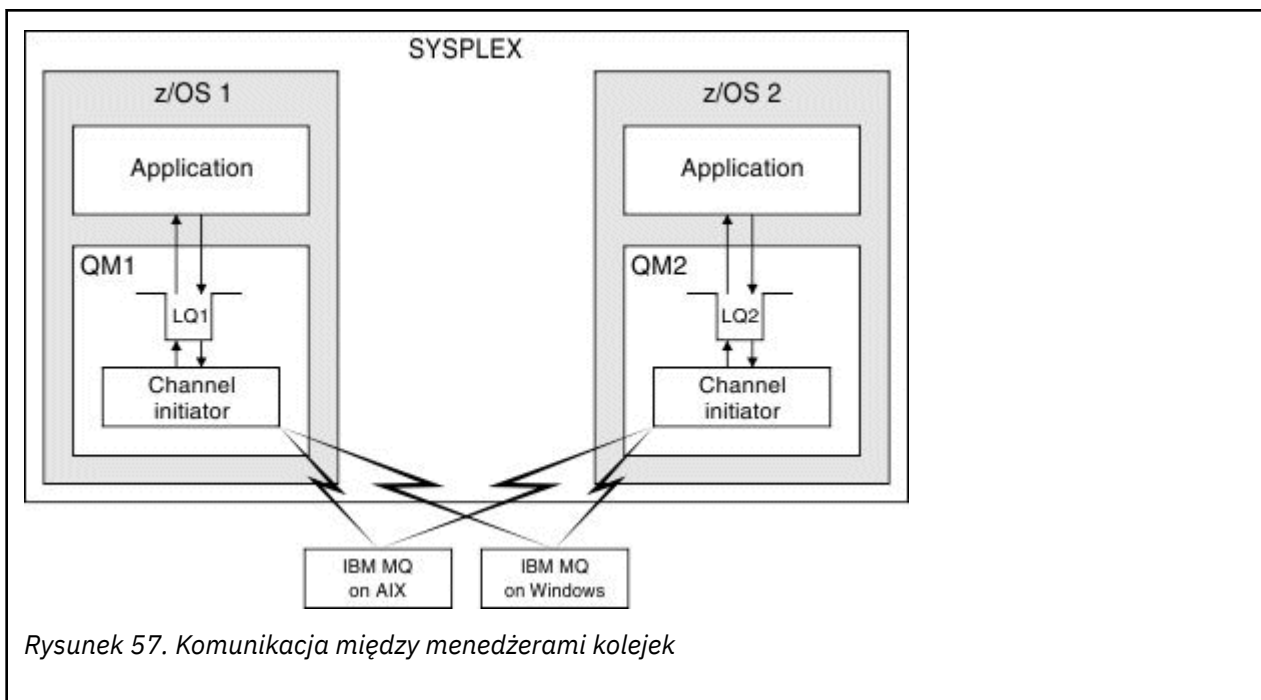
z/OS Inicjator kanału w systemie z/OS

Inicjator kanału udostępnia zasoby umożliwiające kolejkowe rozproszone IBM MQ i zarządza nimi. Program IBM MQ używa *agentów kanału komunikatów* (MCA) do wysyłania komunikatów z jednego menedżera kolejek do innego.

Aby wysłać komunikaty z menedżera kolejek A do menedżera kolejek B, *wysyłający* agent MCA w menedżerze kolejek A musi skonfigurować łącze komunikacyjne z menedżerem kolejek B. Aby odbierać komunikaty z łącza komunikacyjnego, w menedżerze kolejek B musi być uruchomiony *odbierający* agent MCA. Ta jednokierunkowa ścieżka składająca się z wysyłającego agenta MCA, łącza komunikacyjnego i odbierającego agenta MCA jest nazywana *kanalem*. Wysyłający agent MCA pobiera komunikaty z kolejki transmisji i wysyła je w dół kanału do odbierającego agenta MCA. Odbierający agent MCA odbiera komunikaty i umieszcza je w kolejkach docelowych.

W systemie IBM MQ for z/OS wszystkie wysyłające i odbierające adaptory MCA są uruchamiane wewnątrz inicjatora kanału (inicjator kanału jest również nazywany *narzędziem przenoszenia*). Inicjator kanału działa jako przestrzeń adresowa z/OS pod kontrolą menedżera kolejek. Z menedżerem kolejek może być połączony tylko jeden inicjator kanału i jest on uruchamiany w tym samym obrazie systemu z/OS co menedżer kolejek. W inicjatorze kanału mogą być uruchomione współbieżnie tysiące procesów MCA.

Rysunek 57 na stronie 169 przedstawia dwa menedżery kolejek w syspleksie. Każdy menedżer kolejek ma inicjatora kanału i kolejkę lokalną. Komunikaty wysyłane przez menedżery kolejek w systemach AIX i Windows są umieszczane w kolejce lokalnej, z której są pobierane przez aplikację. Komunikaty odpowiedzi są zwracane przez podobną trasę.



Inicjator kanału zawiera również inne procesy związane z zarządzaniem kanałami. Procesy te obejmują:

Procesy nasłuchujące

Procesy te nasłuchują żądań kanału przychodzącego w podsystemie komunikacyjnym, takim jak TCP, i uruchamiają nazwany agent MCA po odebraniu żądania przychodzącego.

Nadzorca

Zarządza to przestrzenią adresową inicjatora kanału, na przykład odpowiada za restartowanie kanałów po awarii.

Serwer nazw

Służy do tłumaczenia nazw TCP na adresy.

Zadania TLS

Są one używane do szyfrowania i deszyfrowania oraz sprawdzania list odwołań certyfikatów.

z/OS Rekordy SMF dla inicjatora kanału

Inicjator kanału (CHINIT) może tworzyć rekordy statystyk SMF i rekordy rozliczeniowe z informacjami o zadaniach i kanałach.

Komenda CHINIT może generować rekordy statystyk SMF i rekordy rozliczeniowe z następującymi typami informacji:

- Zadania: program rozsyłający, adapter, serwer nazw domen (DNS) i protokół SSL. Zadania te tworzą tzw. statystykę CHINIT.
- Kanały: udostępnia informacje rozliczeniowe podobne do tych, które są dostępne w komendzie DIS CHSTATUS. Jest to nazywane rozliczanie kanału.

IBM MQ for Multiplatforms udostępnia podobne informacje, zapisując komunikaty PCF w systemie SYSTEM.ADMIN.STATISTICS.QUEUE. Więcej informacji na temat sposobu rejestrowania informacji statystycznych w systemie IBM MQ for Multiplatforms zawiera sekcja [Dane komunikatu statystyki kanału](#).

Dane statystyczne

Informacje te można wykorzystać do uzyskania następujących informacji:

- Określa, czy wymagane jest więcej zadań CHINIT, takich jak liczba rekordów wykonania instrukcji testowania SSL i ilość zasobów procesora wykorzystywanych przez te zadania.

- Średni czas żądań dla tych zadań.
- Najdłuższy czas trwania żądania w danym przedziale czasu oraz godzina wystąpienia tego zdarzenia dla zadań DNS i SSL. Tę porą dnia można skorelować z problemami, które mogą wystąpić w kanale.

Dane księgowo

Informacji tych można użyć do monitorowania wykorzystania kanału i uzyskania następujących informacji:

- Kanały o najwyższej przepustowości.
- Szybkość wysyłania komunikatów i szybkość wysyłania danych w MB/s.
- Osiągnięta wielkość zadania wsadowego. Jeśli osiągnięta wielkość partii jest zbliżona do wielkości partii określonej dla kanału, kanał może być zbliżony do limitu wysyłania komunikatów.

Komendy `START TRACE` i `STOP TRACE` służą do sterowania gromadzeniem danych śledzenia rozliczania i śledzenia statystyk. Za pomocą opcji `STATCHL` i `STATACLS` w kanale i menedżerze kolejek można sterować tym, czy kanały generują dane SMF.

Terminy i zadania związane z zarządzaniem produktem IBM MQ for z/OS

Ten temat stanowi wprowadzenie do terminologii i zadań specyficznych dla produktu IBM MQ for z/OS.

Niektóre terminy i zadania wymagane do zarządzania produktem IBM MQ for z/OS są specyficzne dla platformy z/OS. Poniższa lista zawiera niektóre z tych terminów i zadań.

- [Kolejki współużytkowane](#)
- [Zestawy stron i pule buforów](#)
- [Rejestrowanie](#)
- [dostosowywanie środowiska menedżera kolejek](#)
- [Restartowanie i odtwarzanie](#)
- [Bezpieczeństwo](#)
- [Dostępność](#)
- [Manipulowanie obiektami](#)
- [Monitorowanie i statystyka](#)
- [Środowiska aplikacji](#)

Kolejki współużytkowane

Kolejki mogą być *niewspółużytkowane*, których właścicielem i dostępem jest tylko jeden menedżer kolejek lub *współużytkowane*, których właścicielem jest *grupa współużytkowania kolejek*. Grupa współużytkowania kolejek składa się z kilku menedżerów kolejek działających w ramach pojedynczego syspleksu z/OS, które mogą współbieżnie uzyskiwać dostęp do tych samych definicji obiektów IBM MQ i danych komunikatów. W obrębie grupy współużytkowania kolejek definicje obiektów współużytkowanych są przechowywane we współużytkowanej bazie danych Db2. Komunikaty kolejki współużytkowanej są przechowywane w jednej lub kilku strukturach narzędzia CF. Jeśli dane komunikatu są zbyt duże, aby można je było zapisać bezpośrednio w strukturze (wielkość przekracza 63 kB) lub jeśli komunikat jest wystarczająco duży, aby reguły zdefiniowane przez instalację wybrały go do odciążania, informacje sterujące komunikatem są nadal przechowywane w pozycji narzędzia CF, ale dane komunikatu są przenoszone do współużytkowanego zestawu danych komunikatu (SMDS) lub do współużytkowanej bazy danych Db2. Współużytkowane zestawy danych komunikatów, współużytkowana baza danych Db2 oraz struktury narzędzia CF są zasobami, które są wspólnie zarządzane przez wszystkie menedżery kolejek w grupie.

Zestawy stron i pule buforów

Gdy komunikat jest umieszczany w kolejce niewspółużytkowanej, menedżer kolejek zapisuje dane w zestawie stron w taki sposób, że można je pobrać, gdy kolejna operacja pobiera komunikat z tej samej kolejki. Jeśli komunikat zostanie usunięty z kolejki, miejsce w zestawie stron, w którym znajdują się dane, zostanie później zwolnione do ponownego wykorzystania. Wraz ze wzrostem liczby komunikatów przechowywanych w kolejce zwiększa się ilość miejsca używanego w zestawie stron, a wraz ze zmniejszaniem się liczby komunikatów w kolejce zmniejsza się ilość miejsca używanego w zestawie stron.

Aby zmniejszyć koszt związany z wydajnością związaną z zapisywaniem danych do zestawów stron i odczytem danych z zestawów stron, menedżer kolejek buforuje aktualizacje w pamięci masowej procesora. Ilość pamięci masowej używanej do buforowania dostępu do zestawu stron jest kontrolowana przez IBM MQ obiekty nazywane *pulami buforów*.

Więcej informacji na temat zestawów stron i pul buforów zawiera sekcja [Zarządzanie pamięcią masową](#).

Rejestrowanie

Wszelkie zmiany obiektów wstrzymanych w zestawach stron oraz operacje na trwałych komunikatach są rejestrowane jako rekordy dziennika. Te rekordy dziennika są zapisywane w zestawie danych dziennika nazywanym *aktywnym dziennikiem*. Nazwa i wielkość aktywnego zestawu danych dziennika jest przechowywana w zestawie danych nazywanym *zestawem danych programu startowego* (BSDS).

Po zapełnieniu aktywnego zestawu danych dziennika menedżer kolejek przełącza się na inny zestaw danych dziennika, aby rejestrowanie było kontynuowane, i kopiuje zawartość pełnego zestawu danych dziennika aktywnego do zestawu danych *dziennika archiwalnego*. Informacje o tych działaniach, w tym nazwa zestawu danych dziennika archiwalnego, są przechowywane w zestawie danych programu startowego. Konceptyjnie istnieją zestawy danych aktywnego dziennika, przez które przechodzi menedżer kolejek; po zapełnieniu aktywnego dziennika dane dziennika są przenoszone do dziennika archiwalnego, a zestaw danych aktywnego dziennika jest dostępny do ponownego wykorzystania.

Więcej informacji na temat zestawów danych dziennika i programu startowego zawiera sekcja [“logowanie IBM MQ for z/OS” na stronie 239](#).

Dostosowywanie środowiska menedżera kolejek

Po uruchomieniu menedżera kolejek odczytywany jest zestaw parametrów inicjowania, które sterują sposobem działania menedżera kolejek. Ponadto odczytywane są zestawy danych zawierające komendy IBM MQ i wykonywane są komendy, które zawierają. Zwykle te zestawy danych zawierają definicje obiektów systemowych wymaganych do uruchomienia programu IBM MQ i można je dostosować w celu zdefiniowania lub zainicjowania obiektów IBM MQ wymaganych w używanym środowisku operacyjnym. Po odczytaniu tych zestawów danych wszystkie zdefiniowane przez nie obiekty są zapisywane w zestawie stron lub w programie Db2.

Więcej informacji na temat parametrów inicjowania i obiektów systemowych zawiera sekcja [“Definicja systemu w systemie z/OS” na stronie 250](#).

Odtwarzanie i restartowanie

W dowolnym momencie podczas operacji IBM MQ mogą wystąpić zmiany w pamięci procesora, które nie zostały jeszcze zapisane w zestawie stron. Te zmiany są zapisywane w zestawie stron, który jest najmniej ostatnio używany przez zadanie w tle w menedżerze kolejek.

Jeśli menedżer kolejek zakończy działanie nieprawidłowo, faza odtwarzania restartu menedżera kolejek może odzyskać utracone zmiany zestawu stron, ponieważ dane komunikatu trwałego są przechowywane w rekordach dziennika. Oznacza to, że program IBM MQ może odtworzyć trwałe dane komunikatów i zmiany obiektów aż do punktu awarii.

Jeśli menedżer kolejek należący do grupy współużytkowania kolejek napotka awarię narzędzia CF, komunikaty trwałe w tej kolejce mogą zostać odtworzone tylko wtedy, gdy utworzona została kopia zapasowa struktury narzędzia CF.

Więcej informacji na temat odtwarzania i restartowania zawiera sekcja [“Odtwarzanie i restartowanie w systemie z/OS”](#) na stronie 262.

Zabezpieczenia

Można użyć zewnętrznego menedżera zabezpieczeń, takiego jak Security Server (wcześniej znanego pod nazwą RACF) . w celu ochrony zasobów, do których IBM MQ jest właścicielem i którymi zarządza, przed dostępem nieautoryzowanych użytkowników. Można również użyć protokołu TLS (Transport Layer Security) do ochrony kanałów. Protokół TLS jest częścią produktu IBM MQ .

Więcej informacji na temat zabezpieczeń systemu IBM MQ zawiera sekcja [“Pojęcia związane z bezpieczeństwem w produkcie IBM MQ for z/OS”](#) na stronie 280.

Dostępność

Istnieje kilka funkcji produktu IBM MQ , które zostały zaprojektowane w celu zwiększenia dostępności systemu w przypadku awarii menedżera kolejek lub podsystemu komunikacyjnego. Więcej informacji o tych funkcjach zawiera temat [“Dostępność w systemie z/OS”](#) na stronie 287.

Manipulowanie obiektami

Gdy menedżer kolejek jest uruchomiony, można manipulować obiektami IBM MQ za pomocą interfejsu konsoli z/OS lub za pomocą administracyjnego programu narzędziowego, który używa usług ISPF w środowisku TSO. Oba mechanizmy umożliwiają definiowanie, modyfikowanie lub usuwanie obiektów IBM MQ . Można również sterować i wyświetlać status różnych funkcji programu IBM MQ i menedżera kolejek.

Więcej informacji na temat tych narzędzi zawiera sekcja [Źródła](#), z których można wydawać komendy MQSC i PCF w systemie IBM MQ for z/OS.

Obiektami programu IBM MQ można również manipulować przy użyciu graficznego interfejsu użytkownika programu IBM MQ Explorer, który udostępnia wizualny sposób pracy z kolejkami, menedżerami kolejek i innymi obiektami.

Monitorowanie i statystyka

Dostępnych jest kilka narzędzi do monitorowania menedżerów kolejek i inicjatorów kanałów. Można również gromadzić dane statystyczne na potrzeby oceny wydajności i rozliczania.

Więcej informacji na temat tych narzędzi zawiera sekcja [“Monitorowanie i statystyki w systemie IBM MQ for z/OS”](#) na stronie 290.

Środowiska aplikacji

Po uruchomieniu menedżera kolejek aplikacje mogą nawiązywać z nim połączenie i uruchamiać za pomocą funkcji API języka IBM MQ . Mogą to być aplikacje CICS, IMS, wsadowe lub WebSphere Application Server . Aplikacje IBM MQ mogą również uzyskiwać dostęp do aplikacji w systemach CICS i IMS , które nie mają informacji o produkcie IBM MQ, za pomocą mostów CICS i IMS .

Więcej informacji na temat tych narzędzi zawiera sekcja [“IBM MQ i inne produkty z/OS”](#) na stronie 294.

Informacje na temat pisania aplikacji IBM MQ można znaleźć w następującej dokumentacji:

- [Projektowanie aplikacji](#)
- [Używanie języka C++](#)
- [Korzystanie z produktu IBM MQ classes for Java](#)

Kolejki współużytkowane i grupy współużytkowania kolejek

Kolejek współużytkowanych i grup współużytkowania kolejek można używać do implementowania wysokiej dostępności zasobów IBM MQ. Kolejki współużytkowane i grupy współużytkowania kolejek są funkcjami unikalnymi dla systemu IBM MQ for z/OS na platformie z/OS.

W tej sekcji opisano atrybuty i korzyści oraz przedstawiono informacje o tym, w jaki sposób kilka menedżerów kolejek może współużytkować te same kolejki i komunikaty w tych kolejkach.

Co to jest kolejka współużytkowana?

Kolejka współużytkowana jest typem kolejki lokalnej. Dostęp do komunikatów w tej kolejce może uzyskać jeden lub więcej menedżerów kolejek znajdujących się w syspleksie.

Grupa współużytkowania kolejek

Menedżery kolejek, które mogą uzyskać dostęp do tego samego zestawu kolejek współużytkowanych, tworzą grupę nazywaną *grupą współużytkowania kolejek*.

Dostęp do komunikatów może uzyskać dowolny menedżer kolejek

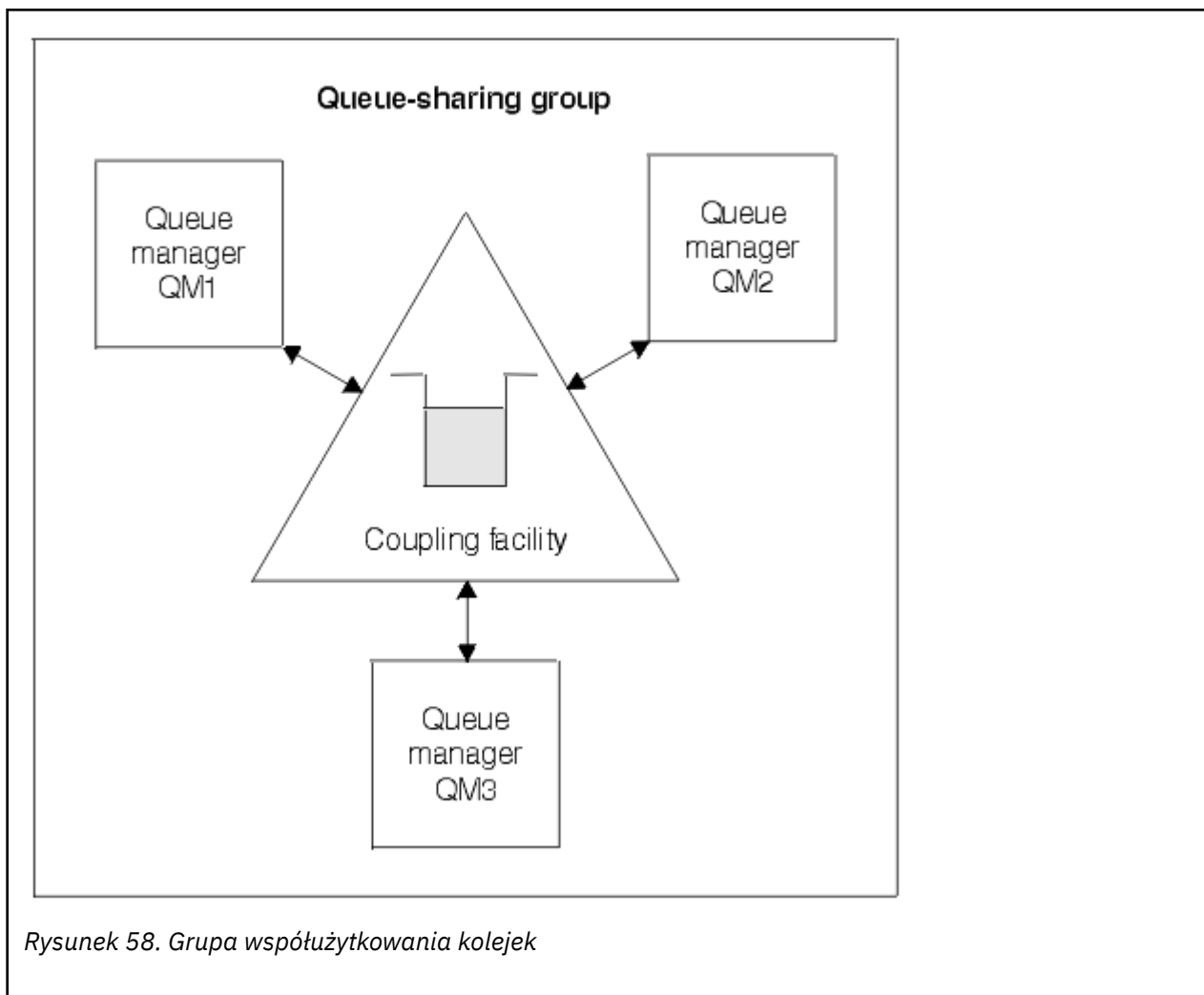
Każdy menedżer kolejek w grupie współużytkowania kolejek może uzyskać dostęp do kolejki współużytkowanej. Oznacza to, że można umieścić komunikat w kolejce współużytkowanej w jednym menedżerze kolejek i pobrać ten sam komunikat z kolejki z innego menedżera kolejek. Zapewnia to szybki mechanizm komunikacji w grupie współużytkowania kolejek, który nie wymaga, aby kanały były aktywne między menedżerami kolejek.

Produkt IBM MQ obsługuje przenoszenie komunikatów do systemu Db2 lub do współużytkowanego zestawu danych komunikatów (SMDS). Przenoszenie komunikatów o dowolnej wielkości jest konfigurowalne.

Rysunek 58 na stronie 174 przedstawia trzy menedżery kolejek i narzędzie CF tworzące grupę współużytkowania kolejek. Wszystkie trzy menedżery kolejek mogą uzyskać dostęp do kolejki współużytkowanej w narzędziu CF.

Aplikacja może nawiązać połączenie z dowolnym menedżerem kolejek w grupie współużytkowania kolejek. Ponieważ wszystkie menedżery kolejek w grupie współużytkowania kolejek mają dostęp do wszystkich kolejek współużytkowanych, aplikacja nie zależy od dostępności konkretnego menedżera kolejek. Wszystkie menedżery kolejek w grupie współużytkowania kolejek mogą obsługiwać kolejkę.

Zapewnia to większą dostępność, ponieważ wszystkie inne menedżery kolejek w grupie współużytkowania kolejek mogą kontynuować przetwarzanie kolejki, jeśli wystąpi problem z jednym z menedżerów kolejek.



Rysunek 58. Grupa współużytkowania kolejek

Definicja kolejki jest współużytkowana przez wszystkie menedżery kolejek

Definicje kolejek współużytkowanych są przechowywane w tabeli bazy danych Db2 OBJ_B_QUEUE. Z tego powodu należy zdefiniować kolejkę tylko raz, a następnie wszyscy menedżery kolejek w grupie współużytkowania kolejek będą mogli uzyskać do niej dostęp. Oznacza to, że istnieje mniej definicji do zdefiniowania.

Natomiast definicja kolejki niewspółużytkowanej jest składowana w zerowym zestawie stron menedżera kolejek, który jest właścicielem kolejki (zgodnie z opisem w sekcji [Zestawy stron](#)).

Nie można zdefiniować kolejki współużytkowanej, jeśli kolejka o tej nazwie została już zdefiniowana w zestawach stron definiującego menedżera kolejek. Podobnie nie można zdefiniować lokalnej wersji kolejki w zestawach stron menedżera kolejek, jeśli istnieje kolejka współużytkowana o takiej samej nazwie.

Co to jest grupa współużytkowania kolejek?

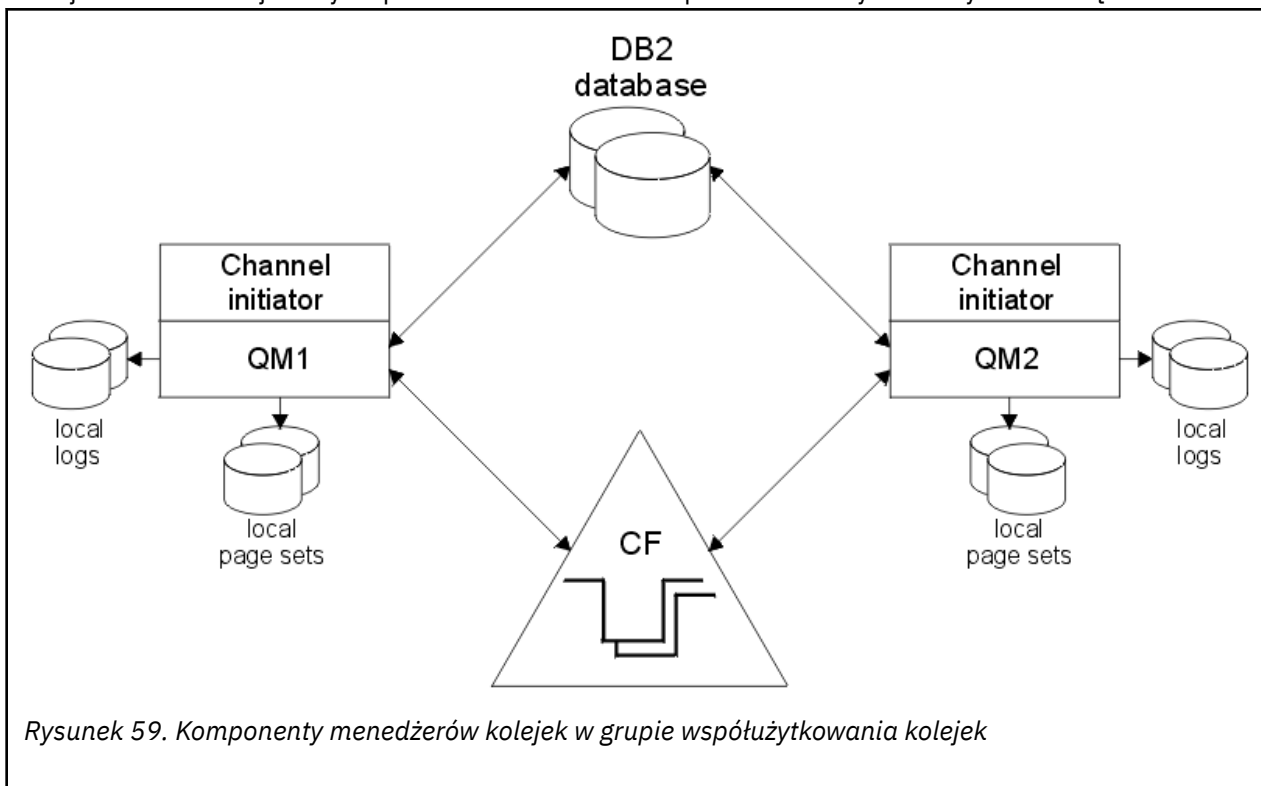
Grupa menedżerów kolejek, które mogą uzyskać dostęp do tych samych kolejek współużytkowanych, jest nazywana grupą współużytkowania kolejek. Dla każdego podzbioru grupy współużytkowania kolejek udostępniony jest ten sam zbiór kolejek współużytkowanych.

Nazwy grup współużytkujących kolejki składają się z maksymalnie czterech znaków. Nazwa taka musi być unikalna w danej sieci i nie może być identyczna z nazwą menedżera kolejek.

Rysunek 59 na stronie 175 przedstawia grupę współużytkowania kolejek, która zawiera dwa menedżery kolejek. Każdy menedżer kolejek ma inicjator kanału oraz własne lokalne zestawy stron i zestawy danych dziennika.

Każdy element grupy współużytkowania kolejek musi również łączyć się z systemem Db2. Wszystkie systemy Db2 muszą należeć do tej samej grupy współużytkowania danych Db2, aby menedżery kolejek miały dostęp do współużytkowanego repozytorium Db2 używanego do przechowywania definicji obiektów współużytkowanych. Są to definicje dowolnego typu obiektu IBM MQ (na przykład kolejek i kanałów), które są zdefiniowane tylko raz, a następnie dowolny menedżer kolejek w grupie może z nich korzystać. Są one nazywane *globalnymi* definicjami i są opisane w sekcji Definicje prywatne i globalne.

Do konkretnej grupy współużytkowania danych może odwoływać się więcej niż jedna grupa współużytkowania kolejek. Należy określić nazwę podsystemu Db2 i grupę współużytkowania danych, której menedżer kolejek używa podczas uruchamiania w parametrach systemowych IBM MQ.



Gdy menedżer kolejek dołączył do grupy współużytkowania kolejek, ma dostęp do obiektów współużytkowanych zdefiniowanych dla tej grupy i można użyć tego menedżera kolejek do zdefiniowania nowych obiektów współużytkowanych w grupie. Jeśli w grupie zdefiniowano kolejki współużytkowane, można użyć tego menedżera kolejek do umieszczania komunikatów w tych kolejkach współużytkowanych i pobierania komunikatów z tych kolejek. Każdy menedżer kolejek w grupie może pobrać komunikaty wstrzymane we współużytkowanej kolejce.

Komendę MQSC można wprowadzić raz, a następnie uruchomić ją we wszystkich menedżerach kolejek w grupie współużytkowania kolejek, tak jakby została wprowadzona osobno dla każdego menedżera kolejek. W tym celu używany jest atrybut *command scope*. Ten atrybut jest opisany w sekcji Kierowanie komend do różnych menedżerów kolejek.

Gdy menedżer kolejek działa jako element grupy współużytkowania kolejek, musi być możliwe rozróżnienie między obiektami produktu IBM MQ zdefiniowanymi prywatnie dla tego menedżera kolejek i obiektami produktu IBM MQ zdefiniowanymi globalnie, które są dostępne dla wszystkich menedżerów kolejek w grupie współużytkowania kolejek. W tym celu używany jest atrybut *dyspozycja grupy współużytkowania kolejek*. Ten atrybut jest opisany w sekcji Definicje prywatne i globalne.

Można zdefiniować pojedynczy zestaw profili zabezpieczeń, które kontrolują dostęp do obiektów IBM MQ w dowolnym miejscu w grupie. Oznacza to, że liczba profili, które należy zdefiniować, jest znacznie mniejsza.

Menedżer kolejek może należeć tylko do jednej grupy współużytkowania kolejek, a wszystkie menedżery kolejek w grupie muszą znajdować się w tym samym syspleksie. Należy określić grupę współużytkowania kolejek, do której należy menedżer kolejek, w parametrach systemowych podczas uruchamiania.

Pojęcia pokrewne

[“Gdzie przechowywane są komunikaty kolejki współużytkowanej?”](#) na stronie 176

Każdy komunikat w kolejce współużytkowanej jest reprezentowany przez pozycję w strukturze listy narzędzia CF (Coupling Facility) systemu z/OS . Jeśli dane komunikatu są zbyt duże, aby zmieścić się w tej samej pozycji, są przenoszone do współużytkowanego zestawu danych komunikatu (SMDS) lub do produktu Db2.

[“Zalety korzystania z kolejek współużytkowanych”](#) na stronie 194

Kolejka współużytkowana umożliwia skalowalność i wysoką dostępność aplikacji IBM MQ oraz umożliwia zaimplementowanie równoważenia obciążenia.

[“Rozproszone kolejkowanie i grupy współużytkowania kolejek”](#) na stronie 215

Rozproszone kolejkowanie i grupy współużytkowania kolejek to dwie techniki, których można użyć w celu zwiększenia dostępności systemów aplikacji. Więcej informacji na temat tych technik można znaleźć w tej sekcji.

[“Wpływanie na dystrybucję obciążenia za pomocą kolejek współużytkowanych”](#) na stronie 218

W tym temacie opisano czynniki, które mają wpływ na dystrybucję obciążenia za pomocą kolejek współużytkowanych w grupie współużytkowania kolejek.

Odsyłacze pokrewne

[“Gdzie można znaleźć więcej informacji o współużytkowanych kolejkach i grupach współużytkowania kolejek”](#) na stronie 220

Tabela w tym temacie zawiera więcej informacji na temat sposobu, w jaki produkt IBM MQ for z/OS używa współużytkowanych kolejek i grup współużytkowania kolejek.

Gdzie przechowywane są komunikaty kolejki współużytkowanej?

Każdy komunikat w kolejce współużytkowanej jest reprezentowany przez pozycję w strukturze listy narzędzia CF (Coupling Facility) systemu z/OS . Jeśli dane komunikatu są zbyt duże, aby zmieścić się w tej samej pozycji, są przenoszone do współużytkowanego zestawu danych komunikatu (SMDS) lub do produktu Db2.

Jeśli struktura CF została skonfigurowana do używania pamięci SCM (System Class Memory), produkt IBM MQ może użyć tej opcji bez dodatkowej konfiguracji.

Ważne: Planowane jest, że IBM z16 będzie ostatnią generacją IBM Z[®] obsługującej użycie wirtualnej pamięci Flash (znanej również jako pamięć klasy pamięci masowej lub SCM) dla obrazów narzędzia CF. Więcej informacji na ten temat zawierają: [IBM Z i IBM LinuxONE 4Q 2023 Oświadczenia o kierunku](#).

Alternatywnie należy użyć większych struktur lub przenieść komunikaty do SMDS.

Pamięć komunikatów kolejki współużytkowanej

Komunikaty umieszczane w kolejkach współużytkowanych nie są przechowywane w zestawach stron i nie korzystają z pul buforów.

Komunikaty w kolejkach współużytkowanych mają pozycje w strukturach listy w narzędziu CF (Coupling Facility) systemu z/OS . Wiele menedżerów kolejek w tym samym syspleksie może uzyskać dostęp do tych komunikatów przy użyciu struktury listy CF.

Dane komunikatu dla małych komunikatów kolejki współużytkowanej są zwykle uwzględniane w pozycji narzędzia CF. W przypadku większych komunikatów dane komunikatu mogą być przechowywane we współużytkowanym zestawie danych komunikatu (SMDS) lub jako jeden lub więcej dużych obiektów binarnych (BLOB) w tabeli Db2 , która jest współużytkowana przez grupę współużytkowania danych Db2 . Dane komunikatu przekraczające 63 kB są zawsze przenoszone do SMDS lub Db2. Mniejsze komunikaty mogą być również opcjonalnie przenoszone w ten sam sposób, aby zaoszczędzić miejsce w strukturze narzędzia CF. Więcej informacji na temat zawiera sekcja [“Określanie opcji przenoszenia dla komunikatów współużytkowanych”](#) na stronie 178.

Komunikaty umieszczane w kolejce współużytkowanej są przywoływane w strukturze narzędzia CF, dopóki nie zostaną pobrane za pomocą wywołania MQGET. Operacje narzędzia CF służą do:

- Wyszukaj następny możliwy do pobrania komunikat
- Zablokuj niezatwierdzone komunikaty w kolejkach współużytkowanych
- Powiadomianie zainteresowanych menedżerów kolejek o nadejściu zatwierdzonych komunikatów

Operacje MQPUT i MQGET na trwałych komunikatach są zapisywane w dzienniku menedżera kolejek wykonującego tę operację. Minimalizuje to ryzyko utraty danych w przypadku awarii narzędzia CF.

Narzędzie CF

Komunikaty przechowywane w kolejkach współużytkowanych są przywoływane w narzędziu CF. Narzędzie CF znajduje się poza dowolnym obrazem z/OS w syspleksie i jest zwykle skonfigurowane do działania na innym zasilaczu. Narzędzie CF jest zatem odporne na awarie oprogramowania i można je skonfigurować w taki sposób, aby były odporne na awarie sprzętu lub przerwy w zasilaniu. Oznacza to, że komunikaty przechowywane w narzędziu CF mają wysoką dostępność.

Każda struktura listy narzędzia CF używana przez produkt IBM MQ jest dedykowana dla konkretnej grupy współużytkowania kolejek, ale narzędzie CF może przechowywać struktury dla więcej niż jednej grupy współużytkowania kolejek. Menedżery kolejek w różnych grupach współużytkowania kolejek nie mogą współużytkować danych. Maksymalnie 32 menedżery kolejek w grupie współużytkowania kolejek mogą jednocześnie łączyć się ze strukturą listy narzędzia CF.

Struktura listy pojedynczego narzędzia CF może zawierać maksymalnie 512 współużytkowanych kolejek. Łączna ilość danych komunikatu przechowywanych w strukturze jest ograniczona przez pojemność struktury. Jednak w systemie **CFLEVEL (5)** można użyć parametrów odciążania w celu odciążenia danych dla komunikatów o wielkości mniejszej niż 63 kB, co spowoduje zwiększenie liczby komunikatów, które mogą być przechowywane w strukturze, chociaż każdy komunikat nadal wymaga co najmniej pozycji narzędzia CF plus co najmniej 768 bajtów danych, składających się z 256 bajtów dla pozycji i 512 bajtów dla dwóch elementów nagłówka i deskryptora.

Wielkość struktury listy jest ograniczona przez następujące czynniki:

- Musi leżeć w obrębie jednego narzędzia sprzęgającego.
- Może współużytkować dostępną pamięć masową narzędzia CF z innymi strukturami produktu IBM MQ i innych produktów.

Struktury listy narzędzia CF mogą mieć powiązaną pamięć klasy pamięci. W pewnych sytuacjach ta pamięć klasy pamięci może być przydatna, gdy jest używana z kolejkami współużytkowanymi. Więcej informacji zawiera sekcja [“Użycie pamięci klasy pamięci masowej z kolejkami współużytkowanymi”](#) na stronie 195.

Planowanie wielkości struktury CF

Jeśli potrzebne są wskazówki dotyczące określania wielkości struktur CF, można użyć narzędzia [MP16: IBM MQ for z/OS Planowanie i strojenie mocy obliczeniowej](#). Można również użyć narzędzia z interfejsem [WWW CFSizer](#), które jest dostarczane przez IBM w celu ułatwienia obsługi wielkości systemów CF.

Obiekt struktury CF

Użycie struktury narzędzia CF przez menedżer kolejek jest określone w obiekcie IBM MQ struktury CF (CFSTRUCT).

Te obiekty struktury są przechowywane w katalogu Db2.

W przypadku używania komend lub definicji systemu z/OS związanych ze strukturą narzędzia CF wymagane są pierwsze cztery znaki nazwy grupy współużytkowania kolejek. Jednak obiekt IBM MQ

CFSTRUCT zawsze istnieje w pojedynczej grupie współużytkowania kolejek i dlatego jego nazwa nie zawiera pierwszych czterech znaków nazwy grupy współużytkowania kolejek. Na przykład CFSTRUCT (MYDATA) zdefiniowany w grupie współużytkowania kolejek rozpoczynającej się od SQ03 używa struktury listy narzędzia CF SQ03MYDATA.

Struktury CF mają atrybut CFLEVEL określający ich możliwości funkcjonalne:

- 1, 2-mogą być używane dla nietrwałych komunikatów o wielkości mniejszej niż 63 kB
- 3-może być używany dla trwałych i nietrwałych komunikatów o wielkości mniejszej niż 63 kB
- 4-może być używany dla trwałych i nietrwałych komunikatów o wielkości do 100 MB
- 5-może być używane dla trwałych i nietrwałych komunikatów o wielkości do 100 MB i selektywnie przenoszone do współużytkowanych zestawów danych komunikatów (SMDS) lub Db2.

Uwaga: W przypadku użycia IBM MQ można zaszyfrować strukturę narzędzia CF. Więcej informacji na ten temat zawiera sekcja [Szyfrowanie danych struktury narzędzia CF](#).

Tworzenie i odtwarzanie kopii zapasowych narzędzia CF

Za pomocą komendy IBM MQ BACKUP CFSTRUCT można tworzyć kopie zapasowe struktur listy narzędzia CF. Spowoduje to umieszczenie kopii komunikatów trwałych znajdujących się obecnie w strukturze systemu CF w zestawie danych aktywnego dziennika menedżera kolejek, który tworzy kopię zapasową, i zapisanie rekordu kopii zapasowej w pliku Db2.

Jeśli narzędzie CF ulegnie awarii, można użyć komendy IBM MQ RECOVER CFSTRUCT. W tym celu używany jest rekord kopii zapasowej z systemu Db2 w celu znalezienia i odtworzenia trwałych komunikatów z kopii zapasowej struktury CF. Wszystkie działania od czasu utworzenia ostatniej kopii zapasowej są powtarzane przy użyciu dzienników wszystkich menedżerów kolejek w grupie współużytkowania kolejek, a struktura CF jest odtwarzana aż do momentu wystąpienia awarii.

Więcej szczegółów zawierają komendy [BACKUP CFSTRUCT](#) i [RECOVER CFSTRUCT](#).

Pojęcia pokrewne

“Określanie opcji przenoszenia dla komunikatów współużytkowanych” na stronie 178

Można wybrać miejsce przechowywania danych komunikatu kolejki współużytkowanej w tabeli Db2 lub w zestawie SMDS (Shared Message Data set). Można również wybrać, które komunikaty mają być przenoszone, na podstawie wielkości komunikatu i bieżącego użycia struktury CF.

“Zarządzanie środowiskiem współużytkowanego zestawu danych komunikatów (SMDS)” na stronie 181

Jeśli wybrano współużytkowane zestawy danych komunikatów w celu przenoszenia dużych komunikatów, należy również pamiętać o informacjach używanych przez produkt IBM MQ do zarządzania tymi zestawami danych i komendach używanych do pracy z tymi informacjami. W tym temacie opisano sposób zarządzania współużytkowanymi zestawami danych komunikatów.

Określanie opcji przenoszenia dla komunikatów współużytkowanych

Można wybrać miejsce przechowywania danych komunikatu kolejki współużytkowanej w tabeli Db2 lub w zestawie SMDS (Shared Message Data set). Można również wybrać, które komunikaty mają być przenoszone, na podstawie wielkości komunikatu i bieżącego użycia struktury CF.

Dane komunikatów dla kolejek współużytkowanych mogą być przenoszone z narzędzia CF i przechowywane w tabeli Db2 lub w zarządzanym zestawie danych IBM MQ nazywanym *współużytkowanym zestawem danych komunikatów* (SMDS).

W przypadku komunikatów większych niż wielkość pozycji narzędzia CF (Coupling Facility) o wielkości 63 kB przenoszenie danych komunikatu do SMDS może spowodować znaczne zwiększenie wydajności w porównaniu z przenoszeniem do Db2.

Każdy komunikat kolejki współużytkowanej jest nadal zarządzany przy użyciu pozycji listy w strukturze narzędzia CF, ale gdy dane komunikatu są przenoszone do SMDS, pozycja narzędzia CF zawiera tylko niektóre informacje sterujące i listę odwołań do odpowiednich bloków dyskowych, w których komunikat

jest przechowywany. Użycie tego mechanizmu oznacza, że ilość pamięci masowej elementów narzędzia CF wymaganej dla każdego komunikatu jest tylko ułamkiem rzeczywistej wielkości komunikatu.

Wybieranie miejsca, w którym przechowywane są komunikaty kolejki współużytkowanej

Do sterowania wyborem współużytkowanej pamięci masowej komunikatów SMDS lub Db2 służy parametr **OFFLOAD (SMDS | DB2)** w definicji **CFSTRUCT**. **OFFLOAD (SMDS)** jest wartością domyślną.

Ten parametr wymaga również, aby parametr **CFSTRUCT** używał wartości **CFLEVEL (5)** lub większej.

Parametr **OFFLOAD** jest poprawny tylko od **CFLEVEL (5)**. Więcej szczegółów zawiera sekcja DEFINE CFSTRUCT.

Produkt **OFFLOAD (DB2)** jest obsługiwany przede wszystkim na potrzeby migracji.

Wybieranie komunikatów kolejki współużytkowanej, które mają być przenoszone

Dane komunikatu są przenoszone do SMDS lub Db2 na podstawie wielkości danych komunikatu i bieżącego użycia struktury narzędzia CF. Istnieją trzy reguły, a każda reguła określa zgodną parę parametrów. Te parametry są odpowiednim procentem progu użycia struktury narzędzia CF (**OFFLDnTH**) i limit wielkości komunikatu (**OFFLDnSZ**).

Bieżąca implementacja trzech reguł jest określana przy użyciu następujących par słów kluczowych:

- OFFLD1TH i OFFLD1SZ
- OFFLD2TH i OFFLD2SZ
- OFFLD3TH i OFFLD3SZ

Para reguł	Wartość domyślna	Opis
Para reguł 1	OFFLD1TH(70) i OFFLD1SZ(32K)	Jeśli struktura narzędzia CF jest więcej niż 70% pełnych danych odciażających dla komunikatów o wielkości przekraczającej 32 KB
Para reguł 2	OFFLD2TH(80) i OFFLD2SZ(4K)	Jeśli struktura narzędzia CF jest więcej niż 80% pełnych danych odciażających dla komunikatów o wielkości przekraczającej 4 kB
Para reguł 3	OFFLD3TH(90) i OFFLD3SZ(0K)	Jeśli struktura narzędzia CF jest więcej niż 90% pełnych danych odciażających dla komunikatów o wielkości przekraczającej 0 KB (wszystkie komunikaty)

Jeśli reguła odciażania ma wartość OFFLD x SZ równą 64K, oznacza to, że reguła nie obowiązuje. W takim przypadku komunikaty będą przenoszone tylko wtedy, gdy obowiązuje inna reguła przenoszenia lub gdy komunikat jest większy niż 63.75 kB i jest zbyt duży, aby można go było zapisać w strukturze.

Każdy komunikat, który jest nieobciążony, nadal wymaga 0.75 kB pamięci masowej w narzędziu CF.

Trzy reguły odciażania, które można określić dla każdej struktury, mają być używane w następujący sposób.

- Wydajność
 - Jeśli w strukturze aplikacji jest dużo miejsca, dane komunikatu powinny być przenoszone tylko wtedy, gdy są zbyt duże, aby można je było zapisać w strukturze, lub gdy przekraczają one próg wielkości komunikatu, tak że wartość wydajności przechowywania ich w strukturze nie jest warta takiej ilości miejsca w strukturze, jaka byłaby mu potrzebna.

- Jeśli wymagany jest konkretny próg wielkości komunikatu, jest on standardowo określany przy użyciu pierwszej reguły przenoszenia.
- pojemność
 - Jeśli w strukturze aplikacji jest bardzo mało miejsca, należy przenieść maksymalną ilość danych komunikatu, aby jak najlepiej wykorzystała pozostałą przestrzeń.
 - Trzecia reguła przenoszenia jest zwykle używana w celu wskazania, że gdy struktura jest prawie pełna, większość komunikatów powinna być odciążana, więc pozycje w strukturze aplikacji będą zwykle miały minimalną wielkość (około 0.75K bajtów).
 - Parametr progu użycia należy wybrać na podstawie wielkości struktury aplikacji i maksymalnej przewidywanej liczby zaległości. Na przykład, jeśli maksymalna oczekiwana liczba zaległych komunikatów wynosi 1M, ilość pamięci masowej struktury wymaganej dla tej liczby komunikatów wynosi około 0.75G bajtów. Oznacza to na przykład, że jeśli struktura ma wielkość około 10G bajtów, próg użycia dla odciążania wszystkich komunikatów musi być ustawiony na 92% lub mniej.
 - Przestrzeń struktury jest podzielona na elementy i pozycje, a mimo tego, że ilość miejsca może być wystarczająca, jedna z nich może się wyczerpać przed drugą. System udostępnia funkcję AUTOALTER w celu dostosowania współczynnika, gdy jest to konieczne, ale nie jest to zbyt wrażliwe, więc ilość dostępnego miejsca może być nieco mniejsza. W związku z tym lepszym rozwiązaniem może być wykorzystanie nie więcej niż 90% maksymalnego obszaru struktury, dlatego w poprzednim przykładzie próg użycia dla przenoszenia wszystkich komunikatów byłby lepiej ustawiony na około 80%.
- Amortyzowane przejście:
 - Ponieważ ilość miejsca w strukturze narzędzia CF zmniejsza się, nie jest wskazane nagłe nagłe zmiany parametrów wydajności. Nie jest również pożądane, aby zarządzanie narzędziem CF miało nagłą zmianę progu w typowym stosunku pozycji do używanych elementów.
 - Druga reguła odciążania jest zwykle używana w celu zapewnienia pewnej pośredniej poduszki między regułami odciążania wydajności i obciążenia. Można ją ustawić w taki sposób, aby spowodować znaczny wzrost aktywności przenoszenia, gdy przestrzeń używana w strukturze narzędzia CF przekroczy próg pośredni. Oznacza to, że pozostałe miejsce jest używane wolniej i daje narzędziu CF więcej czasu na automatyczne przetwarzanie zmian w celu dostosowania się do wyższych poziomów wykorzystania.

Jeśli struktura narzędzia CF nie może zostać rozwinięta i istnieje potrzeba zapisania co najmniej pewnej wstępnie określonej liczby komunikatów, trzecią regułą można w razie potrzeby zmodyfikować w celu zapewnienia, że przenoszenie danych dla wszystkich komunikatów rozpoczyna się od odpowiedniego progu, aby zapewnić zarezerwowanie miejsca na tę wstępnie określoną liczbę komunikatów.

Na przykład, jeśli struktura narzędzia CF ma wielkość 4 GB, a wstępnie określona liczba komunikatów wynosi 1 milion, to potrzebne jest $1\ 000\ 000 * 0.75\ \text{kB}$, czyli 768 MB, 18.75% z 4 GB. W takim przypadku próg przenoszenia wszystkich komunikatów musi być ustawiony na około 80%, a nie na 90%. Daje to parametry OFFLD3TH(80) i OFFLD3SZ(0K). Należy również dostosować inne parametry odciążania.

Jeśli okaże się, że przenoszenie bardzo małych komunikatów ma znaczący wpływ na wydajność, ale względny wpływ jest mniejszy w przypadku większych komunikatów, można zmniejszyć progi użycia dla innych reguł, aby przenieść większe komunikaty wcześniej, pozostawiając więcej miejsca w strukturze dla małych komunikatów, zanim będzie trzeba je przenieść.

Na przykład, jeśli komunikaty o wielkości przekraczającej 32KB występują często, ale wydajność ich odciążania (określona na podstawie statystyk RMF lub wydajności aplikacji) jest bardzo podobna do wydajności utrzymania ich w narzędziu CF, próg dla pierwszej reguły może być ustawiony na 0% w celu odciążenia wszystkich takich komunikatów. Daje to parametry OFFLD1TH(0) i OFFLD1SZ(32K). Ponownie należy dostosować inne parametry odciążania.

Jeśli istnieje wiele komunikatów o określonych wielkościach pośrednich, takich jak 16 KB i 6 KB, wówczas może być przydatna zmiana opcji wielkości komunikatu dla drugiej reguły, tak aby większe

komunikaty były przenoszone przy dość niskim progu użycia, co pozwoli zaoszczędzić znaczną ilość miejsca, ale mniejsze komunikaty są nadal przechowywane tylko w narzędziu CF.

Zarządzanie środowiskiem współużytkowanego zestawu danych komunikatów (SMDS)

Jeśli wybrano współużytkowane zestawy danych komunikatów w celu przenoszenia dużych komunikatów, należy również pamiętać o informacjach używanych przez produkt IBM MQ do zarządzania tymi zestawami danych i komendach używanych do pracy z tymi informacjami. W tym temacie opisano sposób zarządzania współużytkowanymi zestawami danych komunikatów.

Obiekty SMDS

Właściwości i status każdego zestawu danych komunikatów współużytkowanych są śledzone we współużytkowanym obiekcie SMDS, który może być aktualizowany za pośrednictwem dowolnego menedżera kolejek w grupie współużytkowania kolejek.

Dla każdego menedżera kolejek istnieje jeden współużytkowany zestaw danych komunikatu, który może uzyskać dostęp do każdej struktury aplikacji CF. Współużytkowany zestaw danych komunikatu jest identyfikowany przez nazwę menedżera kolejek będącego właścicielem określoną za pomocą słowa kluczowego SMDS oraz przez nazwę struktury aplikacji określoną za pomocą słowa kluczowego CFSTRUCT.

Uwaga: Podczas definiowania zestawów danych SMDS dla struktury należy mieć po jednym dla każdego menedżera kolejek.

Obiekt SMDS jest przechowywany w tablicy (z jedną pozycją na menedżera kolejek w grupie), która tworzy rozszerzenie odpowiedniego obiektu CFSTRUCT przechowywanego w programie Db2.

Nie ma komendy DEFINE lub DELETE dla obiektu SMDS, ponieważ jest on tworzony lub usuwany jako część obiektu CFSTRUCT, ale istnieje komenda ALTER umożliwiająca zmianę ustawień dla pojedynczego menedżera kolejek będącego właścicielem.

Więcej informacji na temat komend SMDS zawiera sekcja [“Komendy związane z SMDS”](#) na stronie 193

Informacje SMDSCONN

Możliwe, że współużytkowany zestaw danych komunikatów jest w stanie normalnym, ale co najmniej jeden menedżer kolejek nie może nawiązać z nim połączenia, na przykład z powodu problemu z definicją zabezpieczeń lub z połączeniem z urządzeniem o dostępie bezpośrednim. Dlatego konieczne jest, aby każdy menedżer kolejek śledził status połączenia i informacje o dostępności dla każdego zestawu danych komunikatów współużytkowanych, wskazując na przykład, czy może on obecnie nawiązać z nim połączenie, a jeśli nie, to dlaczego nie.

Informacje SMDSCONN reprezentują połączenie menedżera kolejek ze współużytkowanym zestawem danych komunikatów. Podobnie jak w przypadku samego współużytkowanego zestawu danych komunikatu, jest on identyfikowany przez menedżera kolejek, który jest właścicielem współużytkowanego zestawu danych komunikatu (określonego przez słowo kluczowe SMDS dla samego obiektu współużytkowanego) w połączeniu z nazwą CFSTRUCT.

Nie ma parametru identyfikującego połączony menedżer kolejek, ponieważ komendy adresowane do konkretnego menedżera kolejek mogą odwoływać się tylko do informacji SMDSCONN dla tego samego menedżera kolejek.

Pozycje informacji SMDSCONN są przechowywane w pamięci głównej w menedżerze kolejek będącym właścicielem i są ponownie tworzone po zrestartowaniu menedżera kolejek. Jeśli jednak połączenie z pojedynczego menedżera kolejek zostało jawnie zatrzymane, informacje te są również przechowywane jako flaga w tablicy połączeń w odpowiednim obiekcie CFSTRUCT lub SMDS, tak aby były zachowywane po restarcie menedżera kolejek.

Informacje o statusie i dostępności

Informacje o statusie wskazują stan zasobu lub połączenia (na przykład, czy zasób nie jest jeszcze używany, jest używany normalnie lub wymaga odtwarzania). Jest ona zwykle opisywana za pomocą słowa kluczowego STATUS. Możliwe wartości zależą od typu obiektu.

Informacje o statusie są zwykle aktualizowane automatycznie, na przykład w przypadku wykrycia błędu podczas korzystania z zasobu lub połączenia. Jednak w niektórych przypadkach można również użyć komendy w celu zaktualizowania statusu, aby umożliwić automatyczne określenie poprawnego statusu przez menedżera kolejek w przypadku, gdy nie jest to możliwe.

Informacje o dostępności wskazują, czy zasób lub połączenie mogą być używane i są zwykle określane głównie przez informacje o statusie. W przypadku typów zasobów lub połączeń używanych w obsłudze współużytkowanego zestawu danych komunikatów zaimplementowane są trzy poziomy dostępności:

Dostępny

Oznacza to, że zasób jest dostępny do normalnego użycia. Nie musi to oznaczać, że jest ona obecnie używana (można ją określić na podstawie wartości STATUS). W przypadku zestawu danych, jeśli wymaga on ponownego uruchomienia, pozwala to menedżerowi kolejek, do którego należy, na jego otwarcie, ale inne menedżery kolejek muszą czekać, aż zestaw danych znajdzie się ponownie w stanie ACTIVE.

Niedostępne z powodu błędu

Oznacza to, że zasób został automatycznie zablokowany z powodu błędu i nie powinien być ponownie dostępny, dopóki nie zostanie wykonana pewna forma naprawy lub odtwarzania. Jednakże próby ponownego udostępnienia są dozwolone bez interwencji operatora. Taka próba może być również wyzwalana przez komendę oznaczającą zasób jako włączony lub przez komendę, która zmienia status w taki sposób, aby wskazać, że przetwarzanie odtwarzania zostało zakończone.

Przyczyna, dla której zasób stał się niedostępny, jest zwykle oczywista w stosunku do powiązanej wartości STATUS, ale w niektórych przypadkach mogą istnieć inne powody, dla których zasób jest niedostępny. W takim przypadku podawana jest oddzielna wartość REASON wskazująca przyczynę.

Niedostępne z powodu komendy operatora

Oznacza to, że dostęp do zasobu został jawnie wyłączony przez komendę. Można go udostępnić tylko za pomocą komendy włączonej ponownie.

Dostępność SMDS

W przypadku współużytkowanego obiektu SMDS dostępność jest opisana przez słowo kluczowe ACCESS z możliwymi wartościami ENABLED, SUSPENDED i DISABLED.

Dostępność można zaktualizować przy użyciu komendy **RESET SMDS** dla odpowiedniego obiektu współużytkowanego z dowolnego menedżera kolejek w grupie, aby ustawić parametr ACCESS (ENABLED) lub ACCESS (DISABLED).

Jeśli dostępność była wcześniej określona jako ACCESS (SUSPENDED), zmiana na ACCESS (ENABLED) spowoduje wyzwolenie nowej próby użycia współużytkowanego zestawu danych komunikatów, ale jeśli poprzedni błąd nadal istnieje, dostępność zostanie przywrócona do ACCESS (SUSPENDED).

Dostępność SMDSCONN

W przypadku lokalnej pozycji informacji SMDSCONN dostępność jest opisana przez słowo kluczowe AVAIL z możliwymi wartościami NORMAL, ERROR lub STOPPED. Dostępność można zaktualizować przy użyciu komendy **START SMDSCONN** lub **STOP SMDSCONN** adresowanej do konkretnego menedżera kolejek w celu włączenia lub wyłączenia jego połączenia.

Jeśli wcześniej była dostępna usługa AVAIL (ERROR), zmiana jej na AVAIL (NORMAL) spowoduje nową próbę użycia współużytkowanego zestawu danych komunikatów, ale jeśli poprzedni błąd nadal istnieje, dostępność zostanie przywrócona do AVAIL (ERROR).

Współużytkowany status i dostępność zestawu danych komunikatów

Dostępność każdego współużytkowanego zestawu danych komunikatów jest zarządzana w ramach grupy przy użyciu współużytkowanych informacji o statusie, które można wyświetlić za pomocą komendy

DISPLAY CFSTATUS z parametrem TYPE (SMDS). Spowoduje to wyświetlenie informacji o statusie dla każdego menedżera kolejek, który aktywował zestaw danych dla każdej struktury. Każdy zestaw danych może być w jednym z następujących stanów:

NIE ZNALEZIONO

Oznacza to, że odpowiedni zestaw danych nie został jeszcze aktywowany. Ten status jest wyświetlany tylko wtedy, gdy określono konkretny menedżer kolejek, ponieważ zestawy danych, które nie zostały aktywowane, są pomijane po wybraniu wszystkich menedżerów kolejek.

NOWOŚĆ

Zestaw danych jest otwierany i inicjowany po raz pierwszy, gotowy do aktywowania.

AKTYWNE

Oznacza to, że zestaw danych jest w pełni dostępny i powinien zostać przydzielony i otwarty przez wszystkie aktywne menedżery kolejek dla struktury.

Niepowodzenie

Oznacza to, że zestaw danych nie jest w ogóle dostępny (z wyjątkiem przetwarzania odtwarzania) i musi być zamknięty i zwalniany przez wszystkie menedżery kolejek.

NIEODZYSKAJ

Oznacza to, że dla tego zestawu danych trwa odzyskiwanie nośników (za pomocą komendy RECOVER CFSTRUCT).

Odtworzono

Oznacza to, że została wydana komenda przełączenia zestawu danych zakończonego niepowodzeniem z powrotem do stanu aktywnego, ale wymagane jest dalsze przetwarzanie restartu, które nie jest jeszcze zakończone, dlatego zestaw danych może zostać otwarty tylko przez menedżera kolejek będącego właścicielem na potrzeby przetwarzania restartu.

PUSTE

Zestaw danych nie zawiera komunikatów. Zestaw danych jest wprowadzany w ten stan, jeśli jest zamykany normalnie przez menedżera kolejek będącego właścicielem w momencie, gdy nie zawiera on żadnych komunikatów. Może być również w stanie EMPTY, gdy poprzednia zawartość zestawu danych ma zostać usunięta, ponieważ struktura aplikacji została opróżniona (przy użyciu parametru **RECOVER CFSTRUCT** z typem PURGE lub, tylko w przypadku struktury nienaprawialnej, przez usunięcie poprzedniej instancji struktury). Przy następnym otwarciu zestawu danych przez menedżera kolejek będącego właścicielem mapa obszaru zostanie zresetowana do wartości pustej, a status zostanie zmieniony na AKTYWNY. Ponieważ poprzednia zawartość zestawu danych nie jest już wymagana, zestaw danych w tym stanie można zastąpić nowo przydzielonym zestawem danych, na przykład w celu zmiany przydziału miejsca lub przeniesienia go na inny wolumin.

Dane wyjściowe komendy zawierają datę i godzinę włączenia rejestrowania odtwarzania, jeśli istnieje, oraz datę i godzinę niepowodzenia zestawu danych, jeśli nie jest on aktualnie aktywny.

Współużytkowany zestaw danych komunikatu może zostać przełączony w stan NIEPOWODZENIE za pomocą komendy **RESET SMDS** lub automatycznie po wykryciu dowolnego z następujących typów błędów:

- Zestaw danych nie może zostać przydzielony lub otwarty przez menedżera kolejek będącego właścicielem.
- Sprawdzanie poprawności nagłówka zestawu danych kończy się niepowodzeniem po pomyślnym otwarciu nagłówka przez dowolny menedżer kolejek.
- Trwały błąd we/wy występuje, gdy menedżer kolejek będący właścicielem odczytuje lub zapisuje dane.
- Trwały błąd we/wy występuje, gdy inny menedżer kolejek odczytuje dane z zestawu danych, który pomyślnie zakończył otwarte przetwarzanie i sprawdzanie poprawności.

Gdy zestaw danych jest w stanie NIEPOWODZENIE lub INRECOVER, nie jest on dostępny do normalnego użytku, więc jeśli stan dostępności to ACCESS (ENABLED), jest on zmieniany na ACCESS (SUSPENDED).

Jeśli zestaw danych został przełączony w stan NIEPOWODZENIE, ale nie jest wymagane odtwarzanie nośnika, na przykład dlatego, że dane były nadal poprawne, ale urządzenie pamięci masowej było tymczasowo w trybie bez połączenia, można użyć komendy **RESET SMDS**, aby zażądać zmiany statusu bezpośrednio na ODTWORZONY.

Po przejściu zestawu danych w stan ODTWORZONE, po zakończeniu przetwarzania odtwarzania lub w wyniku działania komendy **RESET SMDS**, zestaw danych jest gotowy do ponownego użycia po zakończeniu przetwarzania restartu. Jeśli był w stanie ACCESS (SUSPENDED), jest on automatycznie przelączany z powrotem do stanu ACCESS (ENABLED), co umożliwia menedżerowi kolejek, do którego należy, wykonanie przetwarzania restartu. Po zakończeniu przetwarzania restartu stan jest zmieniany na AKTYWNY, a wszystkie inne menedżery kolejek mogą ponownie nawiązać połączenie z zestawem danych.

Status i dostępność połączenia zestawu danych komunikatów współużytkowanych

Każdy menedżer kolejek przechowuje informacje o statusie lokalnym i dostępności na potrzeby połączenia z każdym współużytkowanym zestawem danych komunikatów, którego właścicielem jest ten sam menedżer kolejek oraz z innymi menedżerami kolejek w grupie. Te informacje można wyświetlić za pomocą komendy **DISPLAY SMDSCONN**.

Jeśli nie można uzyskać dostępu do współużytkowanego zestawu danych komunikatów w stanie ACTIVE, który należy do innego menedżera kolejek, oznacza to, że połączenie jest niedostępne z jego własnego punktu widzenia.

Jeśli błąd zdecydowanie wskazuje na problem z samym zestawem danych, menedżer kolejek również automatycznie zmienia status współużytkowania, aby wskazać, że zestaw danych jest teraz w stanie NIEPOWODZENIE. Jeśli jednak błąd może być spowodowany przez problem ze środowiskiem, na przykład brak uprawnień do otwarcia zestawu danych, menedżer kolejek wysyła komunikaty o błędach i traktuje zestaw danych jako niedostępny, ale nie modyfikuje statusu współużytkowanego zestawu danych. Jeśli błąd środowiskowy okaże się mimo to problemem z zestawem danych (na przykład został on przydzielony do urządzenia, do którego niektóre menedżery kolejek nie mają dostępu), operator może użyć komendy **RESET SMDS** z parametrem STATUS (FAILED), aby umożliwić odtworzenie lub naprawę zestawu danych w razie potrzeby.

Jeśli nie można nawiązać połączenia ze współużytkowanym zestawem danych komunikatu, ale zestaw danych wydaje się poprawny, nowa próba użycia tego zestawu danych może zostać wyzwolona przez wydanie komendy **START SMDSCONN** dla menedżera kolejek będącego właścicielem.

Jeśli istnieje potrzeba działania w celu tymczasowego przerwania połączenia między konkretnym menedżerem kolejek i zestawem danych, ale sam zestaw danych nie jest uszkodzony, to zestaw danych można zamknąć i przydzielić za pomocą komendy **STOP SMDSCONN**. Jeśli zestaw danych jest używany, menedżer kolejek zamknie go normalnie (choć wszystkie żądania dotyczące danych w tym zestawie danych będą odrzucane z kodem powrotu). Jeśli jest to posiadany zestaw danych, menedżer kolejek zapisze mapę obszaru podczas przetwarzania CLOSE, unikając konieczności restartowania przetwarzania.

Jeśli zestaw danych musi zostać tymczasowo wycofany z użytku ze wszystkich menedżerów kolejek (na przykład w celu przeniesienia go), ale nie jest uszkodzony, zaleca się użycie parametru **STOP SMDSCONN** dla odpowiedniego zestawu danych z opcją **CMDSCOPE (*)**, aby zatrzymać menedżery kolejek używające go jako pierwszego, ponieważ pozwoli to uniknąć konieczności restartowania przetwarzania po przywróceniu zestawu danych do eksploatacji. Dla porównania, jeśli zestaw danych jest oznaczony jako USZKODZONY, informuje to menedżery kolejek, że muszą natychmiast zaprzestać korzystania z tego zestawu, co oznacza, że mapa obszarów nie zostanie zapisana i będzie musiała zostać odbudowana przez ponowne uruchomienie przetwarzania.

Dostęp do wszystkich współużytkowanych zestawów danych komunikatów, które znajdowały się wcześniej w stanie ACCESS (SUSPENDED), zostanie ponowiony, jeśli menedżer kolejek zostanie zrestartowany.

Rejestrowanie odtwarzania współużytkowanego zestawu danych komunikatów

Trwałe komunikaty współużytkowane są rejestrowane na potrzeby odtwarzania nośników. Oznacza to, że komunikaty można odzyskać po awarii struktur narzędzia CF lub współużytkowanych zestawów danych komunikatów, pod warunkiem, że dzienniki odtwarzania są nadal nienaruszone. Trwałe komunikaty mogą być również ponownie tworzone z dzienników odtwarzania w innym ośrodku na potrzeby odtwarzania po awarii.

Gdy dane komunikatu są zapisywane we współużytkowanym zestawie danych komunikatu, każdy blok zapisywany w zestawie danych jest rejestrowany oddzielnie, po którym następuje pozycja komunikatu (wraz z odwzorowaniem danych) zapisana w narzędziu CF. Proces odtwarzania zawsze odtwarza strukturę narzędzia CF, ale nie musi odtwarzać pojedynczych współużytkowanych zestawów danych komunikatów, z wyjątkiem sytuacji, gdy status zestawu danych ma wartość NIEPOWODZENIE lub gdy status ma wartość AKTYWNE, ale rekord nagłówek zestawu danych nie jest już poprawny, co oznacza, że zestaw danych został ponownie utworzony. Zestaw danych nie jest wybierany do odtwarzania, jeśli jego status to ACTIVE, a nagłówek zestawu danych jest nadal poprawny, lub jeśli jego status to EMPTY, co oznacza, że w momencie niepowodzenia nie były w nim przechowywane żadne komunikaty.

Kopie zapasowe współużytkowanych zestawów danych komunikatów

Jeśli komenda BACKUP CFSTRUCT jest używana do tworzenia kopii zapasowej komunikatów współużytkowanych w strukturze aplikacji, wszystkie dane dla komunikatów trwałych zapisanych we współużytkowanych zestawach danych komunikatów są kopiowane w tym samym czasie, tak jak w przypadku trwałych komunikatów współużytkowanych poprzednio przechowywanych w bazie danych.

Odtwarzanie współużytkowanego zestawu danych komunikatów

Jeśli współużytkowany zestaw danych komunikatu jest uszkodzony lub utracony, należy go wprowadzić w stan NIEPOWODZENIE, aby uniemożliwić używanie tego zestawu przez menedżery kolejek do czasu, aż zostanie on naprawiony. Zwykle odbywa się to automatycznie, ale można to również zrobić za pomocą komendy **RESET SMDS** z parametrem STATUS (FAILED).

Jeśli współużytkowany zestaw danych komunikatu zawierał jakiegokolwiek komunikaty trwałe, można je odzyskać za pomocą komendy RECOVER CFSTRUCT. Ta komenda najpierw odtwarza dane komunikatów trwałych dla tego zestawu danych komunikatów współużytkowanych z ostatniej komendy BACKUP CFSTRUCT, a następnie stosuje wszystkie zarejestrowane zmiany od tego czasu. Jeśli od czasu pierwszej aktywacji zestawu danych nie została wykonana żadna komenda **BACKUP CFSTRUCT**, zostanie ona zresetowana do wartości pustej, a następnie zostaną zastosowane wszystkie zmiany od momentu aktywacji.

Jeśli treść CFSTRUCT i wszystkie współużytkowane zestawy danych komunikatów są niedostępne, na przykład w sytuacji odtwarzania po awarii, wszystkie mogą zostać odzyskane za pomocą jednej komendy **RECOVER CFSTRUCT**.

Jeśli współużytkowany zestaw danych komunikatów jest uszkodzony, ale odzyskiwanie nie było aktywne dla CFSTRUCT lub dziennik zawierający najnowszą kopię zapasową BACKUP CFSTRUCT jest niedostępny lub nie można go użyć, to komunikaty przenoszone do tego zestawu danych nie mogą zostać odzyskane. W takim przypadku można użyć komendy **RECOVER CFSTRUCT** z parametrem TYPE (PURGE), aby oznaczyć współużytkowany zestaw danych komunikatu jako pusty, a następnie usunąć wszystkie komunikaty ze struktury, w której znajdują się dane zapisane w tym zestawie danych.

Po wydaniu komendy **RECOVER CFSTRUCT** status zestawu danych komunikatów współużytkowanych jest zmieniany z NIEPOWODZENIE na INRECOVER. Jeśli odtwarzanie zakończy się pomyślnie, status zostanie automatycznie zmieniony na ODZYSKANY, w przeciwnym razie zostanie zmieniony z powrotem na NIEPOWODZENIE.

Po zmianie stanu zestawu danych na ODTWORZONY informuje on menedżera kolejek będącego właścicielem, że może on teraz podjąć próbę otwarcia zestawu danych i wykonania przetwarzania restartu.

Odtwarzanie współużytkowanego zestawu danych komunikatów i punkty synchronizacji

Proces odtwarzania współużytkowanego zestawu danych komunikatów ponownie stosuje zmiany dla wszystkich pełnych rekordów dziennika do końca dziennika, niezależnie od punktów synchronizacji.

Jeśli zmiany zostały wprowadzone w punkcie synchronizacji, przetwarzanie operacji restartowania lub odtwarzania dla CFSTRUCT może spowodować wycofanie niezatwierdzonych żądań, dlatego niektóre

z odtworzonych zmian mogą nie być w rzeczywistości używane, ale mimo to nie ma żadnej szkody w ich odzyskiwaniu.

Możliwe jest również, że niezatwierdzony komunikat MQPUT został zapisany w strukturze, ale odpowiednie dane mogły nie zostać zapisane w zestawie danych lub w dzienniku (ponieważ zakończenie operacji we/wy jest wymuszane tylko na początku przetwarzania punktu synchronizacji). Jest to nieszkodliwe, ponieważ proces restartowania wycofa pozycję komunikatu w strukturze, więc fakt, że odnosi się ona do nieodzyskanych danych, nie ma znaczenia.

Przetwarzanie restartu współużytkowanego zestawu danych komunikatu

Jeśli połączenie menedżera kolejek z CFSTRUCT zostanie zakończone normalnie, menedżer kolejek zapisuje mapę wolnego obszaru bloków dla każdego współużytkowanego zestawu danych komunikatu w obszarze punktu kontrolnego w zestawie danych, tuż przed zamknięciem zestawu danych. Odwzorowanie obszaru można następnie wczytać ponownie podczas restartu połączenia, pod warunkiem, że ani CFSTRUCT, ani współużytkowany zestaw danych komunikatu nie wymagają przetwarzania odtwarzania przed następnym restartem.

Jeśli jednak menedżer kolejek zostanie zakończony nieprawidłowo lub struktura lub zestaw danych wymagają przetwarzania odtwarzania, w celu dynamicznego odbudowania mapy obszarów po zrestartowaniu połączenia menedżera kolejek ze strukturą wymagane jest dodatkowe przetwarzanie.

Pod warunkiem, że sam zestaw danych nie wymaga odzyskiwania, menedżer kolejek restartuje po prostu skanuje bieżącą zawartość struktury, aby znaleźć odwołania do danych komunikatu, których właścicielem jest bieżący menedżer kolejek, i oznacza odpowiednie bloki danych jako należące do odwzorowania obszaru. Inne menedżery kolejek mogą nadal używać tej struktury i odczytywać dane, których właścicielem jest restartowany menedżer kolejek, podczas odbudowywania mapy obszarów.

Restart współużytkowanego zestawu danych komunikatu po odtworzeniu

Jeśli współużytkowany zestaw danych komunikatu musiał zostać odzyskany z kopii zapasowej, wszystkie nietrwale komunikaty zapisane w zestawie danych zostaną utracone, a jeśli zestaw danych został odzyskany przy użyciu komendy TYPE (PURGE), wszystkie komunikaty zapisane w zestawie danych zostaną utracone. Do czasu zakończenia odtwarzania zestaw danych zostanie oznaczony jako NIEPOWODZENIE lub INRECOVER, więc każda próba odczytania jednego z komunikatów z innego menedżera kolejek powoduje zwrócenie kodu błędu wskazującego, że zestaw danych jest tymczasowo niedostępny.

Po odtworzeniu zestawu danych jego status jest zmieniany na ODZYSKANY, co umożliwia menedżerowi kolejek, który jest właścicielem, otwarcie go w celu zrestartowania, ale zestaw danych pozostaje niedostępny dla innych menedżerów kolejek. Restart menedżera kolejek skanuje strukturę w celu odbudowania mapy obszarów dla pozostałych komunikatów. Skanowanie sprawdza również komunikaty, dla których dane zostały utracone, i usuwa je ze struktury (lub, jeśli to konieczne, oznacza je jako utracone, aby usunąć je później).

Po zakończeniu skanowania restartu status zestawu danych jest automatycznie zmieniany z ODZYSKANE na AKTYWNE, w którym to momencie inne menedżery kolejek mogą ponownie zacząć z niego korzystać.

Informacje o użyciu współużytkowanego zestawu danych komunikatów

Komenda DISPLAY USAGE wyświetla teraz również informacje o wykorzystaniu miejsca w współużytkowanym zestawie danych komunikatów i puli buforów dla wszystkich obecnie otwartych współużytkowanych zestawów danych komunikatów. Ta informacja jest wyświetlana, jeśli podano nową opcję TYPE (SMDS) lub istniejącą opcję TYPE (ALL).

Uwagi dotyczące wydajności i pojemności współużytkowanych danych komunikatów

Monitorowanie użycia zestawu danych

Bieżący procent zapewnienia każdego posiadanego współużytkowanego zestawu danych komunikatów można wyświetlić za pomocą komendy **DISPLAY USAGE** z opcją **TYPE (SMDS)**.

Menedżer kolejek zwykle automatycznie rozszerzy współużytkowany zestaw danych komunikatu, gdy osiągnie 90% zapewnienia, pod warunkiem, że dla definicji SMDS włączona jest opcja **DSEXPAND(YES)**. Ma to zastosowanie, gdy opcja SMDS ma wartość **DSEXPAND(YES)** lub opcja SMDS ma wartość **DSEXPAND(DEFAULT)**, a opcja domyślna CFSTRUCT ma wartość **DSEXPAND(YES)**.

Jeśli próba rozwinięcia nie powiedzie się, ponieważ podczas tworzenia zestawu danych nie określono dodatkowej wielkości przydziału (komunikat IEC070I z kodem przyczyny 203) Menedżer kolejek powtarza żądanie rozszerzenia przy użyciu dodatkowego przydziału przestąpienia wynoszącego około 20% bieżącej wielkości.

Po rozwinięciu zestawu danych nowe przydziały zestawu danych są formatowane jako część procesu rozszerzania, co może zająć dziesiątki sekund, a nawet minuty w przypadku bardzo dużych przydziałów. Nowy obszar staje się dostępny do użycia po zakończeniu formatowania i zaktualizowaniu katalogu w celu wyświetlenia nowego okresu kontroli wysokiego użycia.

Jeśli nowe komunikaty są tworzone bardzo szybko, istniejący zestaw danych może zostać wypełniony przed zakończeniem przetwarzania rozszerzenia. W takim przypadku każde żądanie, które nie mogło przydzielić obszaru, jest tymczasowo zawieszane do czasu zakończenia próby rozszerzenia i udostępnienia nowego obszaru do użycia. Jeśli rozszerzenie powiodło się, żądanie jest automatycznie ponawiane.

Jeśli próba rozszerzenia nie powiedzie się z powodu braku dostępnego miejsca lub z powodu osiągnięcia maksymalnej liczby przydziałów, zostanie wyświetlony komunikat z informacją o przyczynie niepowodzenia, wówczas opcja nadpisania dla danego zestawu SMDS zostanie automatycznie zmieniona na **DSEXPAND(NO)**, aby zapobiec kolejnym próbom rozszerzenia. W takim przypadku istnieje ryzyko, że zestaw danych zostanie wypełniony. W takim przypadku może być konieczne podjęcie dalszych działań zgodnie z opisem w sekcji [Zestaw danych zostanie wypełniony](#).

Monitorowanie wykorzystania struktury aplikacji

Poziom użycia struktury aplikacji można wyświetlić za pomocą komendy MVS **DISPLAY XCF, STRUCTURE**, określając pełną nazwę struktury aplikacji (w tym przedrostek grupy współużytkowania kolejek). Komunikat odpowiedzi IXC360I przedstawia bieżące użycie elementów i pozycji.

Jeśli użycie struktury przekracza wartość **FULLTHRESHOLD** określoną w strategii CFRM, system wysyła komunikat IXC585E i może wykonać automatyczne **ALTER** działania, jeśli zostały określone, które mogą zmienić współczynnik pozycji do elementu lub zwiększyć wielkość struktury.

Optymalizowanie wielkości puli buforów

Każdy bufor w puli buforów współużytkowanych jest używany do odczytu lub zapisu ciągłego zakresu stron dla jednego komunikatu aż do wielkości bloku logicznego. Jeśli komunikat jest zapisywany w kolejnych blokach, każdy zakres stron w oddzielnym bloku wymaga osobnego buforu.

Bufory zawierające dane komunikatów po operacji zapisu lub odczytu są zachowywane w pamięci masowej i ponownie wykorzystywane przy użyciu schematu pamięci podręcznej LRU (least-recently-used), tak aby żądanie ponownego odczytania tych samych danych wkrótce potem nie było konieczne. Zapewnia to znaczną optymalizację, gdy komunikaty współużytkowane są zapisywane, a następnie odczytywane z powrotem przez aplikacje działające w tym samym systemie. Jeśli komunikaty należące do innego menedżera kolejek są przeglądane w celu dokonania wyboru, a następnie pobierane, pozwala to uniknąć konieczności ponownego odczytania komunikatu z dysku.

Oznacza to, że liczba buforów wymaganych dla każdej struktury aplikacji wynosi jeden dla każdego współbieżnego żądania API, które odczytuje lub zapisuje duże komunikaty dla tej struktury aplikacji

oraz pewną liczbę dodatkowych buforów, które zostaną użyte do zapisania ostatnio używanych danych w celu zoptymalizowania kolejnych operacji odczytu.

W przypadku pul buforów współużytkowanych, jeśli liczba buforów jest niewystarczająca, żądania API będą po prostu oczekiwać, jeśli bufor nie jest natychmiast dostępny. Należy jednak unikać tej sytuacji, ponieważ może ona spowodować znaczne obniżenie wydajności.

Statystyki z komendy **DISPLAY USAGE** dla pul buforów współużytkowanych pokazują, czy w bieżącym przedziale czasu statystyk istniały jakiegokolwiek oczekiwania na bufor, a także przedstawiają najniższą liczbę wolnych buforów (lub wartość ujemną wskazującą maksymalną liczbę wątków, które oczekiwały na bufor w dowolnym momencie), liczbę buforów, które zapisały dane, oraz procent czasu, przez jaki żądanie buforu pomyślnie znalazło zapisane dane w łańcuchu LRU ("trafienia LRU") zamiast czytać ("LRU misses")¹.

- Jeśli były jakieś oczekiwania, należy zwiększyć liczbę buforów.
- Jeśli istnieje wiele nieużywanych buforów, liczba buforów może zostać zmniejszona w celu udostępnienia większej ilości pamięci masowej w regionie do innych celów.
- Jeśli istnieje wiele buforów zawierających zapisane dane, ale odsetek odczytów, które były trafieniami w stosunku do tych zapisanych danych, jest bardzo mały, liczba buforów może zostać zmniejszona, jeśli pamięć masowa może być lepiej wykorzystywana do innych celów. Liczba buforów nie powinna być jednak zmniejszana o więcej niż najniższą liczbę wolnych buforów, ponieważ mogłoby to wyzwolić oczekiwania, i powinna być na tyle wysoka, aby najniższa liczba wolnych buforów była zwykle znacznie większa od zera.

Usuwanie współużytkowanych zestawów danych komunikatów

Komenda DELETE CFSTRUCT (która jest dozwolona tylko wtedy, gdy wszystkie kolejki współużytkowane w strukturze są puste i zamknięte) nie usuwa samych współużytkowanych zestawów danych komunikatów, ale można je usunąć w zwykły sposób po zakończeniu działania tej komendy. Jeśli ten sam zestaw danych ma być ponownie wykorzystywany jako współużytkowany zestaw danych komunikatu, należy go najpierw ponownie sformatować, aby przywrócić pusty stan.

Sytuacje wyjątków dla współużytkowanych zestawów danych komunikatów

Istnieje wiele sytuacji wyjątkowych, które mogą wystąpić podczas normalnego używania, nawet jeśli nie występuje błąd programowy lub sprzętowy.

Zestaw danych staje się pełny

Jeśli zestaw danych zostanie zapełniony, ale nie można go rozwinąć, lub jeśli próba rozwinięcia nie powiedzie się, aplikacje używające odpowiedniego menedżera kolejek do zapisywania dużych komunikatów w odpowiedniej strukturze aplikacji otrzymają błąd 2192, MQRC_STORAGE_MEDIUM_FULL (znany również jako MQRC_PAGESET_FULL).

Zestaw danych może zostać zapełniony z powodu awarii aplikacji, która ma przetwarzać dane, co powoduje gromadzenie się dużej liczby komunikatów. Jeśli tak, dalsze rozszerzenie zestawu danych będzie tylko rozwiązaniem tymczasowym i ważne jest jak najszybsze ponowne działanie aplikacji przetwarzającej.

Jeśli można udostępnić więcej miejsca, można użyć komendy **ALTER SMDS** do ustawienia **DSEXPAND(YES)** lub **DSEXPAND(DEFAULT)** (przy założeniu, że wartość YES została ustawiona lub przyjęta jako wartość domyślna **DSEXPAND** dla definicji CFSTRUCT) w celu wyzwolenia ponowienia. Jeśli jednak przyczyną niepowodzenia było osiągnięcie maksymalnej liczby przydziałów, nowa próba rozwinięcia zostanie odrzucona z komunikatem i ponownie zostanie ustawiona wartość **DSEXPAND(NO)**. W tym przypadku jedynym sposobem na jego dalsze rozszerzenie jest ponowne przydzielenie, co oznacza, że jest on tymczasowo niedostępny, zgodnie z opisem poniżej.

¹ (Hits / (Hits+Misses))* 100

Zestaw danych musi zostać przeniesiony lub ponownie przydzielony

Jeśli zestaw danych wymaga przeniesienia lub rozwinięcia, ale jest używany w inny sposób, może zostać tymczasowo wyjęty z użycia, aby umożliwić jego przeniesienie lub ponowne przydzielenie. Każde żądanie API, które próbuje użyć zestawu danych, gdy jest on niedostępny, otrzyma kod przyczyny MQRC_DATA_SET_NOT_AVAILABLE .

1. Użyj komendy **RESET SMDS** , aby oznaczyć zestaw danych jako **ACCESS (DISABLED)** . Spowoduje to, że zostanie ona normalnie zamknięta i zwolnione przez wszystkie aktualnie podłączone menedżery kolejek.
2. Przenieś lub zmień przydział zestawu danych zgodnie z potrzebami, kopiując starą zawartość do nowo przydzielonego zestawu danych, na przykład za pomocą komendy AMS (Access Method Services) **REPRO** .

Nie należy próbować wstępnie formatować nowego zestawu danych przed skopiowaniem do niego starych danych, ponieważ spowodowałoby to dopisanie skopiowanych danych na końcu sformatowanego zestawu danych.

3. Użyj komendy **RESET SMDS** , aby ponownie oznaczyć zestaw danych jako **ACCESS (ENABLED)** , aby przywrócić go do użycia.

Jeśli stara zawartość jest mniejsza niż wielkość nowego zestawu danych, reszta obszaru zostanie wstępnie sformatowana automatycznie po otwarciu nowego zestawu danych.

Jeśli stara zawartość była większa niż wielkość nowego zestawu danych, menedżer kolejek musi przeskanować komunikaty w strukturze narzędzia CF i odbudować mapę obszaru, aby upewnić się, że żadne aktywne dane nie zostały utracone. Jeśli zostanie znalezione odwołanie do bloku danych, który znajduje się poza nowymi przydziałami, zestaw danych jest oznaczany jako **STATUS (FAILED)** i musi zostać naprawiony przez zastąpienie zestawu danych jedną z poprawnych wielkości i ponowne skopiowanie do niego starego zestawu danych lub użycie funkcji **RECOVER CFSTRUCT** w celu odtworzenia wszystkich trwałych komunikatów.

Struktura narzędzia CF ma mało miejsca

Jeśli w strukturze narzędzia CF kończy się miejsce, co powoduje wyświetlenie komunikatu IXC585E , warto sprawdzić, czy reguły przenoszenia zostały ustawione w celu zapewnienia, że w tym przypadku maksymalna ilość danych jest odciążana. Jeśli nie, reguły przenoszenia mogą być modyfikowane za pomocą komendy **ALTER CFSTRUCT** .

Sytuacje błędów dla współużytkowanych zestawów danych komunikatów

Istnieje wiele problemów, o których należy pamiętać, które mogą być spowodowane tylko przez błędy i nie występują w normalnych sytuacjach operacyjnych.

Nie można otworzyć posiadanego zestawu danych

Jeśli menedżer kolejek, który jest właścicielem współużytkowanego zestawu danych komunikatów, nie może go przydzielić lub otworzyć, albo atrybuty zestawu danych nie są obsługiwane, menedżer kolejek ustawia odpowiednią wartość statusu **SMDSCONN** na **ALLOCFAIL** lub **OPENFAIL** i ustawia dostępność **SMDSCONN** na **AVAIL (ERROR)** . Ustawia również dostępność SMDS na **ACCESS (SUSPENDED)** . Po usunięciu błędu użyj komendy **RESET SMDS** , aby ustawić parametr **ACCESS (ENABLED)** w celu wyzwolenia ponowienia, lub wprowadź komendę **START SMDSCONN** do menedżera kolejek będącego właścicielem.

Nie można otworzyć zestawu danych tylko do odczytu

Jeśli menedżer kolejek nie może przydzielić lub otworzyć współużytkowanego zestawu danych komunikatów, którego właścicielem jest inny menedżer kolejek i który jest oznaczony jako **STATUS (ACTIVE)** , przyjmuje się, że przyczyną jest prawdopodobnie konkretny problem z połączeniem z zestawem danych (reprezentowany przez obiekt **SMDSCONN**), a nie problem z samym zestawem danych.

Oznacza on **SMDSCONN** jako **STATUS(ALLOCFAIL)** lub **STATUS(OPENFAIL)** jako odpowiednie i oznacza dostępność **SMDSCONN** jako **AVAIL (ERROR)**, aby zapobiec dalszym próbom jego użycia.

Jeśli problem można rozwiązać bez wpływu na status samego zestawu danych, należy użyć komendy **START SMDSCONN**, aby wyzwolić ponowienie.

Jeśli okaże się, że problem dotyczy samego zestawu danych, można użyć komendy **RESET SMDS**, aby oznaczyć zestaw danych jako **STATUS(FAILED)**, dopóki nie zostanie on odtworzony. Po odtworzeniu zestawu danych zmiana statusu z powrotem na **STATUS(ACTIVE)** spowoduje powiadomienie innych menedżerów kolejek. Jeśli właściwość **SMDSCONN** jest oznaczona jako **AVAIL (ERROR)**, zostanie ona automatycznie zmieniona z powrotem na **AVAIL (NORMAL)**, aby wyzwolić nową próbę otwarcia zestawu danych.

Nagłówek zestawu danych jest uszkodzony

Jeśli zestaw danych został pomyślnie otwarty, ale format informacji nagłówka jest niepoprawny, menedżer kolejek zamyka i zwalnia zestaw danych i ustawia status na **STATUS(FAILED)**, a dostępność na **ACCESS(SUSPENDED)**. Pozwala to na użycie komendy **RECOVER CFSTRUCT** do odtworzenia zawartości.

Jeśli błąd wystąpił, ponieważ zestaw danych zawierał dane rezydujące z innego użycia i nie został później wstępnie sformatowany, należy wstępnie sformatować zestaw danych i użyć komendy **RESET SMDS**, aby zmienić status na **STATUS(RECOVERED)**.

W przeciwnym razie zestaw danych musi zostać odtworzony.

Zestaw danych jest nieoczekiwanie pusty

Jeśli menedżer kolejek otwiera zestaw danych, który jest oznaczony jako **STATUS(ACTIVE)**, ale stwierdza, że nie został zainicjowany lub jest nowo sformatowany, ale jest poprawny w inny sposób, menedżer kolejek zamyka i zwalnia współużytkowany zestaw danych komunikatów, a następnie ustawia status **STATUS(FAILED)** (Niezainicjowany) i dostępność na wartość **ACCESS(SUSPENDED)** (Dostępność).

Zestaw danych zawiera trwałe błędy we/wy

Jeśli w zestawie danych występują trwałe błędy we/wy po pomyślnym przetworzeniu systemu **OPEN**, prawdopodobnie wymaga to odtworzenia. Menedżer kolejek oznaczy zestaw danych jako **STATUS(FAILED)**, aby wszystkie obecnie połączone menedżery kolejek mogły go zamknąć i zwolnić.

Zestaw danych zawiera naprawialne błędy we/wy

Jeśli występują problemy sprzętowe z zestawem danych, może to spowodować naprawialne błędy we/wy, które nie są odzwierciedlane w menedżerze kolejek, ale powodują znaczne obniżenie wydajności, a także wskazuje na ryzyko wystąpienia trwałych błędów we/wy w niedalekiej przyszłości.

W takim przypadku zestaw danych może zostać wyłączony w celu odtworzenia za pomocą komendy **RESET SMDS** w celu oznaczenia go jako **STATUS(FAILED)**. Spowoduje to zamknięcie menedżera kolejek i jego ponowne przydzielenie przez wszystkie menedżery kolejek, na przykład można go przenieść do nowego woluminu przed ponownym udostępnieniem.

Jeśli zestaw danych jest w ten sposób niedostępny, odwzorowanie obszaru nie zostanie zapisane, więc przetwarzanie restartu połączenia menedżera kolejek będzie wymagało skanowania struktury narzędzia CF w celu znalezienia komunikatów w zestawie danych i odbudowania odwzorowania obszaru przed ponownym udostępnieniem zestawu danych. Jeśli współużytkowany zestaw danych komunikatu jest nadal użyteczny, można go ustawić jako niedostępny. W tym celu należy użyć komendy **RESET SMDS**, aby oznaczyć zestaw danych **ACCESS(DISABLED)** do czasu, aż będzie gotowy do ponownego udostępnienia.

Zawartość zestawu danych jest niepoprawna

Menedżer kolejek nie może bezpośrednio wykryć, że zestaw danych zawiera niepoprawne dane lub nie jest aktualny, na przykład dlatego, że wolumin zawierający ten zestaw danych musiał zostać odtworzony z kopii zapasowych. Sprawdza jednak integralność, co sprawia, że jest bardzo mało prawdopodobne, aby takie błędy mogły spowodować, że aplikacje będą widzieć niepoprawne dane komunikatów.

Na potrzeby sprawdzania integralności każdy blok komunikatu w zestawie danych jest poprzedzony kopią odpowiedniego identyfikatora pozycji narzędzia CF, w tym unikalnym znacznikiem czasu, który jest sprawdzany przy każdym odczycie bloku komunikatu przed przekazaniem danych komunikatu do programu użytkownika. Jeśli przedrostek bloku komunikatów nie jest zgodny z identyfikatorem pozycji (a pozycja narzędzia CF nie została usunięta w czasie średnim), zakłada się, że blok komunikatów jest uszkodzony i nie nadaje się do użycia.

Jeśli uszkodzony komunikat był trwały, zestaw danych jest oznaczany jako **STATUS (FAILED)**, a zawartość struktury musi zostać odzyskana za pomocą komendy **RECOVER CFSTRUCT**. Jeśli uszkodzony komunikat był nietrwały, nie ma możliwości jego odzyskania, dlatego wysyłany jest komunikat diagnostyczny i usuwana jest odpowiednia pozycja komunikatu narzędzia CF.

Jeśli podczas otwierania zestawu danych nie jest dostępne żadne zapisane odwzorowanie obszaru, jest ono odbudowywane przez skanowanie struktury narzędzia CF w poszukiwaniu odwołań do danych w zestawie danych. Podczas tego skanowania menedżer kolejek wykonuje kilka działań:

1. Menedżer kolejek określa położenie najnowszego komunikatu (jeśli istnieje), który aktualnie pozostaje w zestawie danych.
2. Następnie menedżer kolejek odczytuje ten komunikat z zestawu danych, aby upewnić się, że przedrostek bloku jest zgodny z identyfikatorem pozycji komunikatu.

Te działania gwarantują, że menedżer kolejek wykryje wszystkie przypadki, w których poziom zestawu danych jest niższy, i oznaczy zestaw danych jako NIEPOWODZENIE. To sprawdzenie jest jednak tolerowane w przypadku, gdy zestaw danych został odtworzony z poprzedniej kopii i od tego czasu nie dodano żadnych nowych wiadomości lub wszystkie wiadomości dodane od tej kopii zostały później odczytane i usunięte.

Aby zabezpieczyć się przed danymi niższego poziomu w przypadku normalnego zamknięcia zestawu danych, menedżer kolejek wykonuje kilka działań:

1. Menedżer kolejek zapisuje kopię znacznika czasu mapy obszaru w obiekcie SMDS w programie Db2, gdy zestaw danych jest normalnie zamknięty.
2. Następnie menedżer kolejek sprawdza, czy znacznik czasu mapy obszaru jest taki sam, gdy zestaw danych jest ponownie otwierany.

Jeśli znacznik czasu nie jest zgodny, oznacza to, że mogła zostać użyta kopia zestawu danych o niższym poziomie, dlatego menedżer kolejek ignoruje istniejące odwzorowanie obszaru i odbudowuje je, co powiedzie się tylko wtedy, gdy nie utracono żadnych danych komunikatu.

Uwaga: Te sprawdzenia integralności nie gwarantują wykrycia uszkodzonego lub niekompletnego zestawu danych we wszystkich teoretycznie możliwych przypadkach. Na przykład nie wykryją przypadku, w którym początek bloku komunikatów jest poprawny, ale reszta danych została częściowo nadpisana.

Scenariusze odtwarzania dla współużytkowanych zestawów danych komunikatów

W tej sekcji opisano scenariusze odtwarzania współużytkowanego zestawu danych komunikatów.

Odtwarzanie zestawu danych w przypadku, gdy nie utracono żadnych danych

W niektórych przypadkach poprawna zawartość zestawu danych, który uległ awarii, może zostać odtworzona bez konieczności rzeczywistego odtwarzania. Przykładem może być sytuacja, w której zbiór danych zawiera dane rezydujące z poprzedniego użycia i nie został ponownie wstępnie sformatowany, co można naprawić przez wstępne formatowanie. Innym przypadkiem jest przeniesienie zestawu danych, ale wystąpił błąd w procesie kopiowania danych, który można naprawić, kopiując dane ponownie poprawnie.

W takich przypadkach poprawiony zestaw danych można ponownie udostępnić za pomocą komendy **RESET SMDS** w celu ustawienia wartości **STATUS (RECOVERED)**. Jeśli aktualnie dostępna jest opcja **ACCESS (SUSPENDED)**, zostanie ona automatycznie ustawiona z powrotem na wartość **ACCESS (ENABLED)**.

Gdy menedżer kolejek będący właścicielem zostanie powiadomiony o odzyskaniu zestawu danych, skanuje zawartość struktury, aby zrekonstruować mapę obszaru, a następnie zmienia status na **STATUS (ACTIVE)**. Inne menedżery kolejek mogą następnie ponownie rozpocząć odczytywanie zestawu danych.

Odzyskiwanie zestawu danych z **TYPE (NORMAL)**

Jeśli treść zestawu danych została utracona, ale struktura aplikacji została zdefiniowana z produktem **RECOVER (YES)** i dostępne są odpowiednie dzienniki odtwarzania, można użyć komendy **RECOVER CFSTRUCT** do odtworzenia wszystkich trwałych komunikatów zapisanych w strukturze, w tym danych komunikatów trwałych, które zostały załadowane do współużytkowanych zestawów danych komunikatów. Ta komenda odtwarza bieżący stan przy użyciu informacji rejestrowanych przez komendę **BACKUP CFSTRUCT** oraz wszystkich zarejestrowanych zmian w komunikatach trwałych od czasu utworzenia kopii zapasowej.

Komenda **RECOVER CFSTRUCT** zawsze odtwarza wszystkie trwałe komunikaty w strukturze narzędzia CF wraz z odciążonymi danymi komunikatów zapisanymi w pliku Db2. W przypadku odciążonych danych przechowywanych we współużytkowanych zestawach danych komunikatów każdy zestaw danych jest wybierany do przetwarzania odtwarzania tylko wtedy, gdy jest już oznaczony jako **STATUS (FAILED)** lub gdy zostanie nieoczekiwanie pusty albo niepoprawny, gdy zostanie otwarty podczas przetwarzania odtwarzania. Każdy współużytkowany zestaw danych komunikatu, który jest oznaczony jako aktywny i który spełnia kryteria sprawdzania poprawności, nie musi być odtwarzany, ponieważ istniejące dane komunikatu są już poprawne, ale nagłówek jest aktualizowany w celu wskazania, że wszystkie zapisane odwzorowania obszaru będą musiały zostać odbudowane po odtworzeniu.

Przetwarzanie odtwarzania jest możliwe tylko wtedy, gdy struktura została oznaczona jako zakończona niepowodzeniem, ponieważ cała zawartość struktury musi zostać zrekonstruowana przez przetwarzanie odtwarzania. Jeśli jednak co najmniej jeden współużytkowany zestaw danych komunikatu został oznaczony jako zakończony niepowodzeniem, komenda **RECOVER CFSTRUCT** automatycznie oznaczy strukturę jako zakończoną niepowodzeniem, jeśli jest to konieczne, aby umożliwić kontynuowanie przetwarzania odtwarzania.

Odtwarzanie można wykonać z dowolnego menedżera kolejek w grupie współużytkowania kolejek, pod warunkiem, że ma on prawo do zapisu w odpowiednich zestawach danych.

Składowane i rejestrowane są tylko komunikaty trwałe, dlatego podczas normalnego przetwarzania odtwarzania zostaną odtworzone wszystkie komunikaty trwałe, ale wszystkie komunikaty nietrwałe w strukturze zostaną utracone.

Po zakończeniu odtwarzania każdy zestaw danych, który został wybrany do odtwarzania, jest automatycznie zmieniany na **STATUS (RECOVERED)**, a jeśli dostępność była **ACCESS (SUSPENDED)**, jest zmieniany na **ACCESS (ENABLED)**. Menedżer kolejek odbudowuje odwzorowanie obszaru dla każdego zestawu danych, skanując komunikaty w narzędziu CF, a następnie oznacza zestaw danych jako **STATUS (ACTIVE)**, aby można go było użyć ponownie.

Odzyskiwanie zestawu danych z **TYPE (PURGE)**

W przypadku struktury odtwarzalnej, jeśli treść zestawu danych została utracona, ale z jakiegoś powodu odtwarzanie nie jest możliwe, na przykład z powodu braku dostępnych dzienników odtwarzania lub zbyt długiego czasu odtwarzania, można użyć komendy **RECOVER CFSTRUCT** z komendą **TYPE (PURGE)**, aby przywrócić strukturę do stanu używalnego. Spowoduje to przywrócenie struktury do stanu pustego i oznaczenie wszystkich powiązanych zestawów danych jako **STATUS (EMPTY)**.

Usuwanie struktury aplikacji

Jeśli nienaprawialna struktura aplikacji zostanie usunięta za pomocą komendy MVS **SETXCF FORCE** lub w wyniku awarii struktury, przy następnym połączeniu struktury zostanie wyświetlony komunikat CSQE028I informujący, że struktura została zresetowana, a wszystkie istniejące komunikaty zostały usunięte, a wszystkie istniejące zestawy danych również zostaną automatycznie zresetowane do wartości **STATUS (EMPTY)**. Ta czynność powoduje, że struktura nienaprawialna staje się ponownie użyteczna po utracie danych w strukturze lub w dowolnym z powiązanych zestawów danych.

Jeśli odtwarzalna struktura aplikacji zostanie usunięta, będzie traktowana w taki sam sposób, jak w przypadku niepowodzenia struktury.

Odtwarzanie zestawu danych nie powiodło się

Jeśli z jakiegoś powodu program **RECOVER CFSTRUCT** nie może zakończyć działania, na przykład dlatego, że zestaw danych dziennika nie jest już dostępny lub ponieważ menedżer kolejek został przerwany w trakcie odtwarzania, wszystkie zestawy danych, dla których odtwarzanie zostało przynajmniej uruchomione, zostaną oznaczone w nagłówku, aby pokazać, że podjęto próbę częściowego odtwarzania, a zestaw danych pozostanie w stanie **STATUS (FAILED)**.

W takim przypadku można powtórzyć oryginalne żądanie odtwarzania lub odzyskać dane za pomocą komendy **TYPE (PURGE)**, usuwając istniejące dane.

Jeśli zostanie podjęta próba oznaczenia zestawu danych jako **STATUS (RECOVERED)** bez rzeczywistego odtwarzania, przy następnym otwarciu menedżera kolejek menedżer kolejek zobaczy, że nagłówek wskazuje na niepełne odtwarzanie, a następnie ponownie oznaczy go jako **STATUS (FAILED)**.

Odtwarzanie po awarii poza siedzibą przedsiębiorstwa

W przypadku odtwarzania po awarii poza siedzibą przedsiębiorstwa trwałe komunikaty współużytkowane mogą być ponownie tworzone tylko przy użyciu dzienników i obiektów współużytkowanych Db2 zawierających definicje CFSTRUCT i powiązane informacje o statusie SMDS.

Po skonfigurowaniu tabel Db2 zawierających definicje struktury aplikacji i współużytkowane zestawy danych komunikatów mogą być ustawione jako puste. Gdy menedżer kolejek połączy się z nimi i stwierdzi, że są one nieoczekiwanie puste, oznaczy je jako zakończone niepowodzeniem, po czym będzie można użyć jednej komendy **RECOVER CFSTRUCT** do odtworzenia wszystkich trwałych komunikatów dla wszystkich struktur, których to dotyczy.

Komendy związane z SMDS

W tej sekcji opisano i udostępniono komendy dotyczące współużytkowanych zestawów danych komunikatów.

Wyświetl i zmień opcje **CFSTRUCT** dotyczące przenoszenia dużych komunikatów (**OFFLOAD** i reguły przenoszenia) oraz współużytkowanych zestawów danych komunikatów (**DSGROUP**, **DSBLOCK**, **DSBUFS**, **DSEXPAND**):

- [WYŚWIETL CFSTRUCT](#)
- [DEFINIOWANIE STRUKTURY CFSTRUCT](#)
- [ALTER CFSTRUCT \(ZMIANA STRUKTURY\)](#)
- [USUŃ CFSTRUCT](#)

Wyświetl **CFSTRUCT** status związany z przenoszeniem dużych komunikatów (**OFFLDUSE**):

- [WYŚWIETLANIE STATUSU CFSTATUS](#)

Wyświetlanie i zmienianie opcji nadpisywania zestawu danych (**DSEXPAND** i **DSBUFS**) dla poszczególnych menedżerów kolejek:

- [Wyświetlanie zestawu SMDS](#)
- [MODYFIKUJ SMDS](#)

Wyświetl lub zmodyfikuj status i dostępność zestawów danych w grupie współużytkowania kolejek:

- [WYŚWIETLENIE TYPU CFSTATUS \(DISPLAY CFSTATUS TYPE-SMDS\)](#)
- [RESETOWANIE ZESTAWÓW SMDS](#)

Wyświetl informacje o użyciu miejsca w zestawie danych SMDS i o użyciu buforu dla menedżera kolejek:

- [WYŚWIETL TYP UŻYCIA \(DISPLAY USAGE TYPE-SMDS\)](#)

Wyświetlanie lub modyfikowanie statusu i dostępności połączeń (**SMDSCONN**) do zestawów danych z pojedynczego menedżera kolejek:

- WYŚWIETL SMDSCONN
- URUCHOM SMDSCONN
- ZATRZYMAJ SMDSCONN

Tworzenie i odtwarzanie kopii zapasowych komunikatów współużytkowanych, w tym dużych danych komunikatów w SMDS, jeśli jest to konieczne:

- KOPIA ZAPASOWA CFSTRUCT
- ODZYSKIWANIE CFSTRUCT

Zalety korzystania z kolejek współużytkowanych

Kolejka współużytkowana umożliwia skalowalność i wysoką dostępność aplikacji IBM MQ oraz umożliwia zaimplementowanie równoważenia obciążenia.

Zalety kolejek współużytkowanych

Architektura kolejki współużytkowanej, w której sklonowane serwery pobierają pracę z pojedynczej kolejki współużytkowanej, ma kilka przydatnych właściwości:

- Można go skalować, dodając nowe instancje aplikacji serwera lub nawet dodając nowy obraz z/OS z menedżerem kolejek (w grupie współużytkowania kolejek) i kopią aplikacji.
- Jest on wysoce dostępny.
- Wykonuje on naturalnie równoważenie obciążenia *pobierania* na podstawie dostępnej mocy przetwarzania każdego menedżera kolejek w grupie współużytkowania kolejek.

Korzystanie ze współużytkowanych kolejek w celu zapewnienia wysokiej dostępności

Poniższe przykłady ilustrują sposób użycia kolejki współużytkowanej w celu zwiększenia dostępności aplikacji.

Rozważmy scenariusz IBM MQ, w którym aplikacje klienckie działające w sieci chcą wysyłać żądania do aplikacji serwera działających w systemie z/OS. Aplikacja kliencka tworzy komunikat żądania i umieszcza go w kolejce żądań. Następnie klient oczekuje na odpowiedź z serwera, wysłaną do kolejki odpowiedzi określonej w deskrypcji komunikatu żądania.

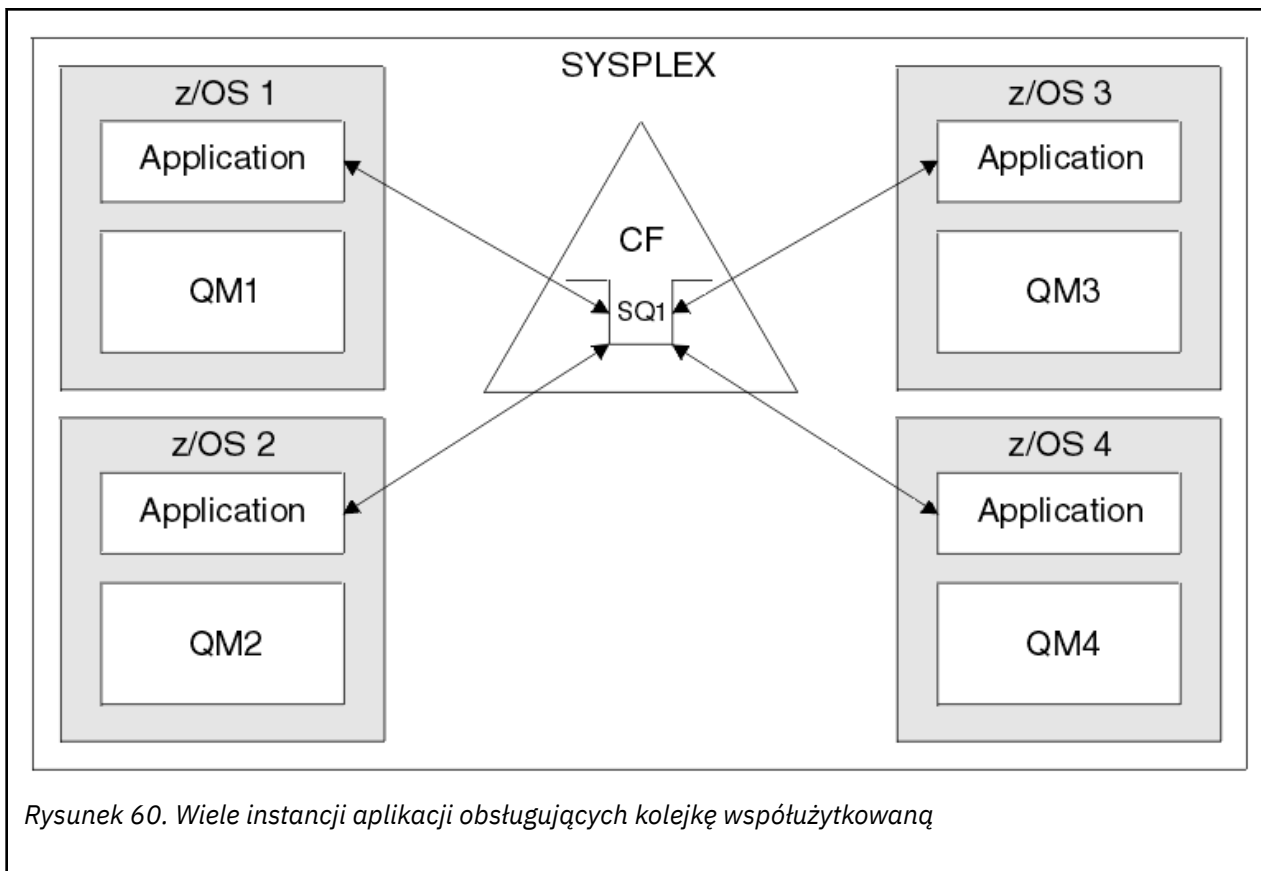
Program IBM MQ zarządza transportem komunikatu żądania z komputera klienta do kolejki wejściowej serwera w systemie z/OS oraz odpowiedzią z serwera z powrotem do klienta. Definiując kolejkę wejściową serwera jako kolejkę współużytkowaną, wszystkie komunikaty umieszczane w kolejce mogą być pobierane w dowolnym menedżerze kolejek w grupie współużytkowania kolejek. Oznacza to, że można skonfigurować menedżer kolejek w każdym obrazie z/OS w syspleksie i, łącząc je wszystkie z tą samą grupą współużytkowania kolejek, każdy z nich może uzyskać dostęp do komunikatów w kolejce wejściowej serwera.

Komunikaty w kolejce wejściowej serwera są nadal dostępne, nawet jeśli jeden z menedżerów kolejek zostanie zakończony nieprawidłowo lub należy go zatrzymać z przyczyn administracyjnych. Można przełączyć cały obraz z/OS w tryb bez połączenia, a wiadomości będą nadal dostępne.

Aby skorzystać z tej dostępności komunikatów w kolejce współużytkowanej, należy uruchomić instancję aplikacji serwera na każdym obrazie z/OS w syspleksie, aby zapewnić większą moc obliczeniową i dostępność aplikacji serwera (patrz [Rysunek 60](#) na stronie 195).

Jedna instancja aplikacji serwera pobiera komunikat żądania z kolejki współużytkowanej i na podstawie treści wykonuje przetwarzanie, generując wynik, który jest wysyłany z powrotem do klienta jako komunikat IBM MQ. Komunikat odpowiedzi jest przeznaczony dla kolejki odpowiedzi i menedżera kolejek odpowiedzi określonego w deskrypcji komunikatu żądania.

Istnieje wiele opcji, których można użyć do skonfigurowania ścieżki zwrotu. Więcej informacji o tych opcjach zawiera sekcja “Rozproszone kolejkowanie i grupy współużytkowania kolejek” na stronie 215.



Odtwarzanie równorzędne

W celu dalszego zwiększenia dostępności komunikatów w grupie współużytkowania kolejek produkt IBM MQ wykrywa, czy inny menedżer kolejek w grupie odłącza się od narzędzia CF nieprawidłowo i w miarę możliwości wykonuje oczekujące jednostki pracy dla tego menedżera kolejek. Ta funkcja jest nazywana *odtworzeniem równorzędnym*.

Założmy, że menedżer kolejek został zakończony nieprawidłowo w punkcie, w którym aplikacja pobrała komunikat żądania z kolejki w punkcie synchronizacji, ale nie umieściła jeszcze komunikatu odpowiedzi ani nie zatwierdziła jednostki pracy. Inny menedżer kolejek w grupie współużytkowania kolejek wykrywa awarię i wycofuje jednostki pracy wykonywane w trakcie przetwarzania dla menedżera kolejek, który uległ awarii. Oznacza to, że komunikat żądania jest umieszczony z powrotem w kolejce żądań i jest dostępny dla jednej z innych instancji serwera do przetworzenia, bez oczekiwania na zrestartowanie menedżera kolejek, który uległ awarii.

Jeśli produkt IBM MQ nie może rozstrzygnąć jednostki pracy automatycznie, można rozstrzygnąć część współużytkowaną ręcznie, aby umożliwić kontynuowanie przetwarzania tej pracy innemu menedżerowi kolejek w grupie współużytkowania kolejek.

z/OS Użycie pamięci klasy pamięci masowej z kolejkami współużytkowanymi

Użycie pamięci klasy pamięci masowej (SCM) może być korzystne w przypadku używania z kolejkami współużytkowanymi IBM MQ for z/OS.

Ważne: Planowane jest, że IBM z16 będzie ostatnią generacją IBM Z[®] obsługującej użycie wirtualnej pamięci Flash (znanej również jako pamięć klasy pamięci masowej lub SCM) dla obrazów narzędzia CF. Więcej informacji na ten temat zawierają: [IBM Z i IBM LinuxONE 4Q 2023 Oświadczenia o kierunku](#).

Alternatywnie należy użyć większych struktur lub przenieść komunikaty do SMDS.

Komputery z13, zEC12i zBC12 umożliwiają instalowanie kart Flash Express. Karty te zawierają dyski SSD. Po instalacji pamięć masowa flash z kart może zostać przydzielona do jednej lub kilku partycji LPAR, gdzie jest zwykle znana jako SCM.

System SCM znajduje się między rzeczywistą pamięcią masową a urządzeniem pamięci masowej o dostępie bezpośrednim (DASD) zarówno pod względem opóźnienia we/wy, jak i kosztu. Ponieważ system SCM nie ma ruchomych części, charakteryzuje się znacznie mniejszymi opóźnieniami we/wy niż DASD.

SCM jest również znacznie tańsze niż prawdziwej pamięci masowej. W związku z tym można zainstalować dużą ilość pamięci masowej przy stosunkowo niskich kosztach, na przykład para kart Flash Express zawiera 1424 GB użytecznej pamięci masowej.

Te cechy oznaczają, że SCM jest przydatne, gdy duża ilość danych musi być pobrana z rzeczywistej pamięci masowej w krótkim okresie czasu, ponieważ dane te mogą być zapisane w systemie SCM znacznie szybciej niż mogą być zapisane w DASD. Ten konkretny punkt może być bardzo przydatny podczas korzystania ze struktur listy narzędzia CF zawierających kolejki współużytkowane w systemie IBM MQ .

Dlaczego struktury list wypełniają się

Po zdefiniowaniu struktury CF jest ona konfigurowana z atrybutem SIZE opisującym maksymalną wielkość struktury. Ponieważ struktury CF są zawsze trwale rezydentne w pamięci rzeczywistej, suma atrybutów SIZE struktur zdefiniowanych w systemie CF powinna być mniejsza niż ilość pamięci rzeczywistej przydzielonej do systemu CF.

W związku z tym istnieje stałe ciśnienie, które pozwala zachować wartość SIZE dowolnej struktury do wartości minimalnej, tak aby większa liczba struktur mogła zmieścić się w systemie CF. Jednak zapewnienie, że struktury są wystarczająco duże, aby osiągnąć ich cel, może spowodować powstanie presji powodującej konflikt, ponieważ zbyt mała struktura oznacza, że może się wypełnić, zakłócając aplikacje lub podsystemy korzystające z niej.

Istnieje silna potrzeba dokładnego określenia wielkości struktury na podstawie jej oczekiwanego użycia. Zadanie to jest jednak trudne do wykonania, ponieważ obciążenia mogą się zmieniać w czasie, a rozliczanie się z ich fluktuacji nie jest łatwe.

Kolejki współużytkowane systemu IBM MQ używają struktur list systemu CF do przechowywania komunikatów. Produkt IBM MQ wywołuje struktury CF, które zawierają komunikaty i struktury aplikacji.

Struktury aplikacji są przywoływane przy użyciu informacji przechowywanych w obiektach IBM MQ CFSTRUCT. Jeśli komunikat o wielkości mniejszej niż 63 kB jest umieszczany w kolejce współużytkowanej, komunikat jest przechowywany w całości w strukturze aplikacji jako pojedyncza pozycja listy i zero lub więcej elementów listy.

Ponieważ kolejki współużytkowane w systemie IBM MQ używają struktur list, opisane ciśnienia mają również wpływ na kolejki współużytkowane. W takim przypadku maksymalna liczba komunikatów, które mogą być przechowywane w kolejce współużytkowanej, jest funkcją:

- Wielkość komunikatów w kolejce
- Maksymalna wielkość struktury
- Liczba wpisów i elementów dostępnych w strukturze

Ponieważ do 512 współużytkowanych kolejek może korzystać z tej samej struktury i skutecznie konkurować o pozycje i elementy, to jeszcze bardziej komplikuje sprawę.

Kolejki produktu IBM MQ są używane do przesyłania danych między aplikacjami, dlatego typową sytuacją jest aplikacja umieszczająca komunikaty w kolejce, gdy aplikacja partnerska, która powinna otrzymać te komunikaty, nie jest uruchomiona.

W takim przypadku liczba komunikatów w kolejce zwiększa się z upływem czasu, aż wystąpi co najmniej jedna z następujących sytuacji:

- Aplikacja umieszczania zatrzymuje umieszczanie komunikatów.
- Aplikacja pobierująca rozpoczyna pobieranie komunikatów.
- Istniejące komunikaty w kolejce tracą ważność i są usuwane z kolejki.
- Kolejka osiąga maksymalne zapętnienie, w którym to przypadku kod przyczyny MQRC_Q_FULL jest zwracany do aplikacji umieszczającej.
- Struktura zawierająca kolejkę współużytkowaną osiąga maksymalną wielkość lub w systemie CF zawierającym strukturę zabraknie dostępnej pamięci. W obu przypadkach do aplikacji umieszczającej zwracany jest kod przyczyny MQRC_STORAGE_MEDIUM_FULL.

W ostatnich trzech przypadkach kolejka jest pełna. W tym momencie aplikacja odkładania ma problem, ponieważ nie ma dokąd pójść jej komunikat. Aplikacja aplikująca zwykle rozwiązuje ten problem, korzystając z jednego lub kilku z następujących rozwiązań:

- Powtarzające się próby umieszczenia komunikatu, opcjonalnie z opóźnieniem między ponownymi próbami.
- Umieść komunikaty w innym miejscu, na przykład w bazie danych lub w pliku. Dostęp do komunikatów można uzyskać później i umieścić je w kolejce w normalny sposób.
- Usuń komunikat, jeśli jest nietrwały.

Jednak w przypadku niektórych klas aplikacji, na przykład z dużą liczbą komunikatów przychodzących lub bez dostępu do systemu plików, rozwiązania te nie są praktyczne. Istnieje realna potrzeba zapewnienia, że kolejki nigdy nie zostaną zapętnione lub jest mało prawdopodobne, aby zostały zapętnione w pierwszej kolejności, co jest szczególnie istotne dla kolejek współużytkowanych.

Reguły SMDS i odciążania

Reguły przenoszenia wprowadzone w produkcie IBM WebSphere MQ 7.1 umożliwiają zmniejszenie prawdopodobieństwa zapętnienia struktury aplikacji.

Z każdą strukturą aplikacji są powiązane trzy reguły, określone za pomocą trzech par słów kluczowych:

- OFFLD1SZ i OFFLD1TH
- OFFLD2SZ i OFFLD2TH
- OFFLD3SZ i OFFLD3TH

Każda reguła określa warunki, które muszą być spełnione, aby dane komunikatu były przenoszone do mechanizmu pamięci masowej powiązanego ze strukturą aplikacji. Obecnie dostępne są dwa typy mechanizmów pamięci masowej:

- Db2
- Grupa liniowych zestawów danych VSAM (Virtual Storage Access Method), które IBM MQ wywołuje współużytkowany zestaw danych komunikatów (SMDS).

Poniższy przykład przedstawia komendę MQSC służącą do tworzenia struktury aplikacji o nazwie LIST1 przy użyciu komendy DEFINE CFSTRUCT.

Ta struktura zawiera domyślne reguły przenoszenia i używa SMDS jako mechanizmu przenoszenia. Oznacza to, że jeśli struktura jest zapętniona w 70% (OFFLD1TH), wszystkie komunikaty o wielkości 32 kB lub większe (OFFLD1SZ) są przenoszone do SMDS.


Podobnie, jeśli struktura jest zapętniona w 80% (OFFLD2TH), wszystkie komunikaty o wielkości 4 kB lub większe (OFFLD2SZ) są przenoszone. Gdy struktura jest zapętniona w 90% (OFFLD3TH), wszystkie komunikaty (OFFLD3SZ) są przenoszone.

```
DEFINE CFSTRUCT(LIST1)
CFLEVEL(5)
OFFLOAD(SMDS)
OFFLD1SZ(32K) OFFLD1TH(70)
OFFLD2SZ(4K) OFFLD2TH(80)
OFFLD3SZ(0K) OFFLD3TH(90)
```

Odciążony komunikat jest przechowywany na nośniku odciażającym, a wskaźnik do komunikatu jest przechowywany w strukturze. Podczas gdy reguły przenoszenia zmniejszają prawdopodobieństwo zapelnienia struktury, przez umieszczenie mniejszej ilości danych komunikatu w strukturze po wyczerpaniu pamięci, niektóre dane są nadal zapisywane w strukturze dla każdego komunikatu. Oznacza to, że wskaźnik do odciażonego komunikatu.

Ponadto reguły odciażania mają wpływ na wydajność. Zapisywanie komunikatu do struktury jest stosunkowo szybkie i w dużym stopniu zdominowane przez czas potrzebny na wysłanie żądania zapisu do systemu CF. Faktyczny zapis do struktury jest szybki, dzieje się to z prawdziwą szybkością pamięci masowej.

Zapisywanie komunikatu do SMDS jest znacznie wolniejsze, ponieważ obejmuje zapis do struktury dla wskaźnika komunikatu i zapis danych komunikatu do SMDS. Ta druga operacja zapisu jest wykonywana z szybkością DASD i może spowodować dodatkowe opóźnienie. Jeśli jako mechanizm przenoszenia używany jest parametr Db2, koszt wydajności jest znacznie większy.

 *Sposób działania pamięci klasy pamięci masowej z produktem IBM MQ for z/OS*

Przegląd użycia pamięci klasy pamięci masowej (SCM) z kolejkami współużytkowanymi produktu IBM MQ for z/OS.

Ważne: Planowane jest, że IBM z16 będzie ostatnią generacją IBM Z[®] obsługującej użycie wirtualnej pamięci Flash (znanej również jako pamięć klasy pamięci masowej lub SCM) dla obrazów narzędzia CF. Więcej informacji na ten temat zawierają: [IBM Z i IBM LinuxONE 4Q 2023 Oświadczenia o kierunku](#).

Alternatywnie należy użyć większych struktur lub przenieść komunikaty do SMDS.

Do narzędzia CF, które jest na poziomie CFLEVEL 19 lub wyższym, może być przydzielony system SCM. Struktury zdefiniowane w tym systemie CF można następnie skonfigurować w taki sposób, aby korzystały z systemu SCM w celu zmniejszenia ryzyka zapelnienia struktur (znanego jako pełny warunek struktury). Gdy struktura skonfigurowana do korzystania z systemu SCM zapelnia się po punkcie określonym przez system, system CF rozpoczyna przenoszenie danych ze struktury do systemu SCM, co zwalnia miejsce w strukturze dla nowych danych.

Uwaga: Ponieważ sam komponent SCM może zostać zapelniony, przydzielenie komponentu SCM do struktury zmniejsza tylko prawdopodobieństwo spełnienia pełnego warunku struktury, ale nie całkowicie usuwa możliwości wystąpienia takiego stanu.

Struktura jest skonfigurowana do korzystania z systemu SCM przez określenie zarówno słów kluczowych **SCMALGORITHM**, jak i **SCMMAXSIZE** w strategii menedżera zasobów narzędzia CF (Coupling Facility Resource Manager-CFRM) zawierającej definicję tej struktury.

Należy zauważyć, że po podaniu tych słów kluczowych i zastosowaniu strategii CFRM struktura musi zostać odbudowana lub zwolona, aby mogły one zostać zastosowane.

Słowo kluczowe SCMALGORITHM

Ponieważ szybkość wejścia/wyjścia systemu SCM jest wolniejsza niż rzeczywista pamięć masowa, system CF używa algorytmu dostosowanego do oczekiwanego użycia struktury w celu zmniejszenia wpływu zapisu lub odczytu z systemu SCM.

Algorytm jest konfigurowany za pomocą słowa kluczowego **SCMALGORITHM** w strategii CFRM dla struktury przy użyciu wartości *KEYPRIORITY1*. Należy pamiętać, że wartości *KEYPRIORITY1* należy używać tylko ze strukturami listy używanymi przez kolejki współużytkowane systemu IBM MQ.

Algorytm *KEYPRIORITY1* działa, zakładając, że większość aplikacji będzie pobierała komunikaty z kolejki współużytkowanej w kolejności priorytetów. Oznacza to, że gdy aplikacja pobiera komunikat, otrzymuje on najstarszy komunikat o najwyższym priorytecie.

Gdy struktura zaczyna zapelniać się po przekroczeniu progu zdefiniowanego przez system (90%), system CF rozpoczyna asynchronicznie migrowanie komunikatów, które są najmniej prawdopodobne, że będą następne. Są to komunikaty o niższych priorytetach, które zostały ostatnio umieszczone w kolejce.

Ta asynchroniczna migracja komunikatów ze struktury do systemu SCM jest nazywana migracją wstępną.

Wstępne przemieszczanie zmniejsza koszty wydajności związane z używaniem systemu SCM, ponieważ zmniejsza ryzyko zablokowania aplikacji podczas występowania synchronicznego wejścia/wyjścia do systemu SCM.

Oprócz wstępnego przemieszczania algorytm *KEYPRIORITY1* także asynchronicznie przywołuje komunikaty z systemu SCM i do struktury, gdy dostępna jest wystarczająca ilość wolnego miejsca. Dla algorytmu *KEYPRIORITY1* oznacza to, że gdy struktura jest mniejsza lub równa 70% pełnej.

Działanie pobierania komunikatów z systemu SCM do struktury jest określane jako "preselekcja".

Preselekcja zmniejsza prawdopodobieństwo, że aplikacja próbuje pobrać komunikat, który został wstępnie przemieszczony do systemu SCM, i musi czekać, aż system CF synchronicznie przywróci komunikat do struktury.

Słowo kluczowe SCMMAXSIZE

Słowo kluczowe **SCMMAXSIZE** definiuje maksymalną ilość SCM, która może być używana przez strukturę. Ponieważ system SCM jest przydzielany do struktury przez system CF, gdy jest to wymagane, można określić parametr **SCMMAXSIZE**, który jest większy niż łączna ilość dostępnego wolnego systemu SCM. Jest to nazywane "nadmiernym zatwierdzeniem".

Ważne: Nigdy nie należy nadzatwierdzać SCM. W takim przypadku aplikacje, które na nim polegają, nie uzyskają oczekiwanego zachowania. Na przykład aplikacje IBM MQ korzystające z kolejek współużytkowanych mogą otrzymać nieoczekiwane kody przyczyny MQRD_STORAGE_MEDIUM_FULL.

System CF używa różnych struktur danych do śledzenia użycia systemu SCM. Te struktury danych rezydują w pamięci rzeczywistej, która jest przydzielona do systemu CF i w rezultacie zmniejszają ilość pamięci rzeczywistej, która może być używana przez struktury. Pamięć masowa używana przez te struktury danych jest nazywana "przestrzenią rozszerzoną".

Jeśli struktura jest skonfigurowana z systemem SCM, niewielka ilość rzeczywistej pamięci masowej jest przydzielana z systemu CF do struktury zwanej stałym obszarem rozszerzonym. Jest ona przydzielana nawet wtedy, gdy struktura nigdy nie używa żadnego systemu SCM. Ponieważ dane ze struktury są przechowywane w systemie SCM, dodatkowy dynamiczny obszar rozszerzony zostanie przydzielony z zapasowej rzeczywistej pamięci masowej w systemie CF.

Po usunięciu danych z systemu SCM dynamiczny obszar rozszerzony jest zwracany do systemu CF. Obszar rozszerzony, stały lub dynamiczny, nigdy nie jest pobierany z rzeczywistej pamięci masowej, która jest przydzielona do struktury.

Oprócz rozszerzonej pamięci masowej, gdy struktura jest skonfigurowana do korzystania z systemu SCM, zwiększa się ilość pamięci masowej używanej przez tę strukturę. Oznacza to, że struktura listy skonfigurowana z użyciem systemu SCM może zawierać mniej pozycji i elementów niż struktura o tej samej wielkości bez konfigurowania systemu SCM.

Aby zrozumieć wpływ systemu SCM na nowe lub istniejące struktury, należy użyć narzędzia [CFSizer](#).

Należy zauważyć, że po przeniesieniu danych ze struktury do systemu SCM i użyciu dynamicznego, rozszerzonego obszaru, struktura nie może być zmieniana ani ręcznie, ani automatycznie.

Oznacza to, że nie można zwiększyć ani zmniejszyć ilości pamięci masowej przydzielonej dla struktury, nie można zmienić współczynnika pozycji do elementu, który jest używany przez strukturę itd. Aby można było ponownie zmienić strukturę, struktura nie może mieć żadnych danych przechowywanych w systemie SCM i nie może korzystać z dynamicznej rozszerzonej pamięci masowej.

Dlaczego należy używać systemu SCM

Awaryjna pamięć masowa i zwiększona wydajność to dwa przypadki użycia systemu SCM z produktem IBM MQ for z/OS.

Ważne: Planowane jest, że IBM z16 będzie ostatnią generacją IBM Z[®] obsługującej użycie wirtualnej pamięci Flash (znanej również jako pamięć klasy pamięci masowej lub SCM) dla obrazów narzędzia CF. Więcej informacji na ten temat zawierają: [IBM Z i IBM LinuxONE 4Q 2023 Oświadczenia o kierunku](#).

Alternatywnie należy użyć większych struktur lub przenieść komunikaty do SMDS.

W tej sekcji przedstawiono teorię leżącą u podstaw dwóch możliwych scenariuszy. Więcej informacji na temat konfigurowania scenariuszy można znaleźć w następujących sekcjach:

- [“Pamięć awaryjna-podstawowa konfiguracja” na stronie 203](#)
- [“Poprawiona wydajność-podstawowa konfiguracja” na stronie 209](#)

Ważne: Użycie systemu SCM ze strukturami narzędzia CF nie jest zależne od żadnej konkretnej wersji produktu IBM MQ. Jednak scenariusz awaryjnej pamięci masowej działa tylko z systemem IBM WebSphere MQ 7.1 i nowszym, ponieważ wymaga on SMDS i reguł przenoszenia.

Składowanie awaryjne

Aby zmniejszyć prawdopodobieństwo zwrócenia kodu przyczyny MQRD_STORAGE_MEDIUM_FULL do aplikacji IBM MQ podczas rozszerzonego wyłączenia, można użyć funkcji SMDS i przenoszenia komunikatów w połączeniu z systemem SCM.

Przegląd

Pojedyncza kolejka współużytkowana jest skonfigurowana w strukturze aplikacji. Aplikacja umieszcza komunikaty w kolejce współużytkowanej, a aplikacja pobiera komunikaty z kolejki współużytkowanej.

Podczas normalnego działania głębokość kolejki powinna być bliska zeru, ale wymaganie biznesowe wskazuje, że system musi być w stanie tolerować dwugodzinne wyłączenie aplikacji pobierającej. Oznacza to, że kolejka współużytkowana musi zawierać dwie godziny komunikatów z aplikacji umieszczającej.

Obecnie ten proces jest realizowany przy użyciu domyślnych reguł przenoszenia i SMDS, dzięki czemu wielkość struktury jest minimalizowana, przy jednoczesnym obniżeniu kosztów wydajności związanych z przenoszeniem.

Oczekuje się, że szybkość wysyłania komunikatów do kolejki współużytkowanej podwoi się w perspektywie krótko- i średnioterminowej. Mimo że wymaganie, aby system mógł tolerować dwugodzinne wyłączenie, nadal istnieje, w systemie CF nie ma wystarczającej ilości pamięci rzeczywistej, aby podwoić wielkość struktury.

Ponieważ system CF, który zawiera strukturę aplikacji, znajduje się na komputerze z systemem zEC12, istnieje możliwość powiązania wystarczającego systemu SCM ze strukturą w celu zapisania wystarczającej liczby komunikatów, aby można było tolerować dwugodzinne wyłączenie.

Zastanów się, co się dzieje w danym okresie:

1. Początkowo system jest w stanie ustalonym. Zarówno aplikacja umieszczana, jak i pobierana działają normalnie, a głębokość kolejki jest zbliżona do zera lub wynosi zero. W rezultacie struktura aplikacji jest w dużej mierze pusta.
2. W pewnym momencie aplikacja pobierająca ulega nieoczekiwanym awariom i zatrzymuje się. Aplikacja umieszczająca nadal umieszcza komunikaty w kolejce i zaczyna wypełniać strukturę aplikacji.
3. Gdy struktura osiągnie 70% zapętnienia, warunki pierwszej reguły przenoszenia są spełnione, a wszystkie komunikaty o wielkości większej lub równej 32 kB są przenoszone do SMDS.

Przegląd reguł przenoszenia zawiera sekcja [“Reguły SMDS i odciążania” na stronie 197](#).

4. Ponieważ komunikaty są nadal umieszczane w kolejce współużytkowanej, struktura jest nadal wypełniona (z powodu danych komunikatu przechowywanych w strukturze lub w wyniku wskaźników do odciążonych komunikatów przechowywanych w strukturze).

Gdy struktura osiągnie 80% zapętnienia, zaczyna obowiązywać druga reguła przenoszenia, a komunikaty o wielkości 4 kB lub większe są przenoszone do SMDS.

5. Jeśli struktura jest zapętniona w 90%, wszystkie komunikaty są przenoszone do SMDS i tylko wskaźniki komunikatów są umieszczane w strukturze.

W tym czasie algorytm wstępnego przemieszczania rozpoczyna działanie i rozpoczyna przenoszenie danych ze struktury do systemu SCM. Zakładając, że wszystkie komunikaty w kolejce mają ten sam priorytet, najnowsze komunikaty są wstępnie przemieszczane.

Ponieważ wszystkie komunikaty są teraz przenoszone do SMDS, dane przenoszone do systemu SCM nie są rzeczywistymi danymi komunikatu, ale wskaźnikami do komunikatów w SMDS.

W związku z tym liczba komunikatów, które można zapisać w kombinacji struktury oraz SCM i SMDS powiązanych ze strukturą, jest bardzo duża.

Wydajność: Na tym etapie wyłączenia aplikacja może być w pewnym stopniu narażona na spadek wydajności z powodu konieczności zapisu w SMDS. W tym przypadku użycie SCM nie powinno być czynnikiem ograniczającym umieszczanie aplikacji pod względem wydajności. System SCM udostępnia dodatkowe miejsce, aby zapobiec zapełnieniu struktury.

6. W końcu aplikacja uzyskująca dostęp jest ponownie dostępna i wyłączenie jest skończone.

Jednak system SCM jest nadal używany przez strukturę. Aplikacja pobierająca rozpoczyna odczytywanie komunikatów z kolejki, pobierając najpierw najstarsze komunikaty o najwyższym priorytecie.

Ponieważ te komunikaty zostały napisane zanim struktura zaczęła się zapełniać, pochodzą one całkowicie z rzeczywistej części struktury przechowywanej w pamięci.

7. Gdy struktura zaczyna być pusta, spada poniżej wartości progowej, przy której jest aktywne wstępne przemieszczanie, a więc wstępne przemieszczanie zostaje zatrzymane.
8. Użycie struktury zmniejsza się poniżej punktu, w którym reguły przenoszenia są stosowane, więc komunikaty nie są już przenoszone do SMDS, chyba że mają więcej niż 63 kB.

W tym momencie algorytm preselekcji rozpoczyna przenoszenie danych z systemu SCM do struktury. Ponieważ aplikacja pobiera komunikaty z kolejki w kolejności oczekiwanej przez algorytm SCM, komunikaty są wprowadzane przed aplikacją pobierającą, która ich potrzebuje.

W rezultacie aplikacja pobierająca nigdy nie musi czekać, aż komunikaty będą wprowadzane synchronicznie z systemu SCM.

9. Ponieważ aplikacja pobierająca kontynuuje przenoszenie w dół kolejki, rozpoczyna pobieranie komunikatów, które zostały przeniesione do SMDS.
10. Wreszcie, system jest w stanie stabilnym. W systemie SCM lub SMDS nie są przechowywane żadne komunikaty, a głębokość kolejki jest bliska zeru.

Większa wydajność

W tym scenariuszu opisano korzystanie z systemu SCM w celu zwiększenia liczby komunikatów, które mogą być przechowywane w kolejce współużytkowanej bez ponoszenia kosztów związanych z wydajnością związanych z używaniem SMDS.

Opis

W tym scenariuszu aplikacja umieszczająca i pobierająca komunikuje się za pośrednictwem kolejki współużytkowanej, która jest przechowywana w strukturze aplikacji.

Aplikacja umieszczająca ma tendencję do uruchamiania w porcjach, gdy umieszcza dużą liczbę komunikatów w krótkim czasie. Następnie, w dłuższym okresie czasu, nie generuje żadnych komunikatów.

Aplikacja pobierająca przetwarza sekwencyjnie każdy komunikat i wykonuje złożone przetwarzanie każdego z nich. Oznacza to, że w większości przypadków zapełnienie kolejki wynosi zero, z wyjątkiem sytuacji, gdy aplikacja umieszczająca rozpoczyna działanie, w której zapełnienie kolejki zaczyna rosnąć, ponieważ komunikaty są umieszczane szybciej niż są odbierane.

Zapełnienie kolejki zwiększa się do momentu zatrzymania aplikacji umieszczającej, a aplikacja pobierającej komunikaty ma wystarczająco dużo czasu na przetworzenie wszystkich komunikatów w kolejce.

Uwagi:

1. W tym scenariuszu kluczowym czynnikiem jest wydajność. Komunikaty wysyłane do kolejki są zawsze mniejsze niż 63 kB i dlatego nigdy nie muszą być przenoszone do SMDS.

2. Struktura aplikacji została tak dobrana, aby była wystarczająco duża, aby zawierała wszystkie komunikaty, które zostaną na niej umieszczone przez aplikację umieszczającą w pojedynczym rozdzielaniu.
3. Wszystkie reguły przenoszenia muszą być wyłączone, aby nawet po wypełnieniu struktury komunikaty nie były przenoszone do SMDS. Wynika to z faktu, że koszty związane z wydajnością związane z zapisywaniem komunikatów do SMDS i odczytem komunikatów z SMDS są uważane za niedopuszczalne.

Z biegiem czasu liczba komunikatów wysyłanych przez aplikację umieszczającą w porcji musi wzrosnąć o kilka rzędów wielkości. Ponieważ aplikacja pobierająca musi przetwarzać każdy komunikat sekwencyjnie, liczba komunikatów w kolejce zwiększa się do punktu, w którym struktura wypełnia się.

W tym momencie aplikacja umieszczająca otrzymuje kod przyczyny (MQRC_STORAGE_MEDIUM_FULL) podczas umieszczania komunikatu i operacja umieszczania kończy się niepowodzeniem. Aplikacja umieszczana może chwilowo tolerować okresy tylko wtedy, gdy nie jest w stanie umieścić komunikatów w kolejce. Jeśli okres jest zbyt długi, aplikacja kończy działanie.

Zakładając, że nie masz dostępnego czasu lub umiejętności na przepisanie aplikacji lub aplikacji, ten problem ma trzy możliwe rozwiązania:

1. Zwiększ wielkość struktury aplikacji.
2. Dodaj reguły przenoszenia do struktury aplikacji, aby komunikaty były przenoszone do SMDS w momencie rozpoczęcia wypełniania kolejki.
3. Powiąż system SCM ze strukturą.

Pierwsze rozwiązanie jest szybkie w implementacji, ale w systemie CF nie ma wystarczającej ilości rzeczywistej pamięci masowej.

Drugie rozwiązanie może być również szybkie w implementacji, ale wpływ na wydajność przenoszenia do SMDS jest uważany za zbyt istotny, aby można było użyć tej opcji.

Trzecie rozwiązanie, które wiąże system SCM ze strukturą, zapewnia akceptowalne równoważenie kosztów i wydajności.

Powiązanie systemu SCM ze strukturą powoduje większe użycie rzeczywistej pamięci masowej w systemie CF z powodu rozszerzonej pamięci masowej, która jest używana do wykonywania operacji. Jednak rzeczywista ilość pamięci rzeczywistej będzie mniejsza niż ilość użyta w pierwszej opcji.

Innym czynnikiem jest koszt SCM. Jednak koszt ten jest znacznie tańszy niż rzeczywisty magazyn. Czynniki te łączą się, aby trzecia opcja była tańsza niż pierwsza opcja.

Chociaż trzecia opcja, potencjalnie, może nie działać tak dobrze, jak pierwsza, algorytmy preselekcji i wstępnego przemieszczania używane przez system CF mogą się łączyć, aby różnice w wydajności mogły być akceptowalne lub w niektórych przypadkach nieistotne.

Z pewnością wydajność może być znacznie lepsza niż użycie SMDS do przenoszenia komunikatów.

Zastanów się, co się dzieje w danym okresie:

1. Początkowo aplikacja pobierająca jest aktywna i oczekuje na dostarczenie komunikatów do kolejki współużytkowanej. Aplikacja umieszczana nie jest aktywna, a kolejka współużytkowana jest pusta.
2. W określonym momencie aplikacja umieszczająca komunikaty staje się aktywna i rozpoczyna umieszczanie dużej liczby komunikatów we współużytkowanej kolejce. Aplikacja pobierająca rozpoczyna pobieranie komunikatów, ale wypełnienie kolejki szybko zaczyna rosnąć, ponieważ aplikacja pobierająca jest wolniejsza od aplikacji umieszczającej.

W rezultacie struktura aplikacji zaczyna się wypełniać.

3. Wraz ze wzrostem czasu aplikacja umieszczana jest nadal aktywna. Struktura aplikacji wypełnia się do około 90%.

Jest to moment, w którym algorytm wstępnego przemieszczania SCM rozpoczyna przenoszenie komunikatów ze struktury do systemu SCM, zwalniając miejsce w strukturze.

Ponieważ aplikacja pobierająca pobiera najpierw najstarsze komunikaty o najwyższym priorytecie z kolejki, zawsze pobiera komunikaty ze struktury i nie musi czekać, aż komunikaty zostaną wprowadzone synchronicznie z systemu SCM do struktury.

4. Aplikacja umieszczana jest nadal aktywna i umieszcza komunikaty we współużytkowanej kolejce. Jednak aplikacja nigdy nie otrzymuje kodu przyczyny MQR_STORAGE_MEDIUM_FULL, ponieważ w systemie SCM istnieje wystarczająca ilość miejsca do przechowywania wszystkich komunikatów, które nie mieszczą się w strukturze.
5. W końcu aplikacja wstawiania zatrzymuje się, ponieważ nie ma więcej komunikatów do umieszczenia.

Algorytm wstępnego przemieszczania zostaje zatrzymany, ponieważ struktura jest poniżej 90% w użyciu, a aplikacja pobierająca kontuuje przetwarzanie komunikatów w kolejce.

6. Gdy aplikacja pobierająca rozpoczyna zwalnianie miejsca w strukturze, algorytm preselekcji rozpoczyna pobieranie komunikatów z systemu SCM do struktury.

Ponieważ aplikacja pobierająca przetwarza komunikaty w kolejności oczekiwanej przez algorytm preselekcji, aplikacja pobierająca nigdy nie jest blokowana, oczekując na synchroniczne wprowadzenie danych komunikatu z systemu SCM do struktury.

7. Na koniec aplikacja pobierająca przetwarza wszystkie komunikaty w kolejce współużytkowanej i czeka na udostępnienie następnego komunikatu. Struktura i system SCM są puste dla komunikatów.

Pamięć awaryjna-podstawowa konfiguracja

W jaki sposób można skonfigurować podstawowy scenariusz awaryjnej pamięci masowej w systemie IBM MQ.

O tym zadaniu

Ważne: Planowane jest, że IBM z16 będzie ostatnią generacją IBM Z[®] obsługującej użycie wirtualnej pamięci Flash (znanej również jako pamięć klasy pamięci masowej lub SCM) dla obrazów narzędzia CF. Więcej informacji na ten temat zawierają: [IBM Z i IBM LinuxONE 4Q 2023 Oświadczenia o kierunku](#).

Alternatywnie należy użyć większych struktur lub przenieść komunikaty do SMDS.

Aby zmniejszyć prawdopodobieństwo zwrócenia kodu przyczyny MQR_STORAGE_MEDIUM_FULL do aplikacji IBM MQ podczas rozszerzonego wyłączenia, można użyć funkcji SMDS i przenoszenia komunikatów w połączeniu z systemem SCM.

Na przykład w przedsiębiorstwie znajduje się aplikacja, która umieszcza komunikaty w kolejce i aplikacja, która pobiera komunikaty z kolejki. Podczas normalnego działania oczekiwane jest, że głębokość kolejki będzie bliska zeru, ale wymaganie biznesowe wskazuje, że system może tolerować dwugodzinne wyłączenie aplikacji, która pobiera komunikaty.

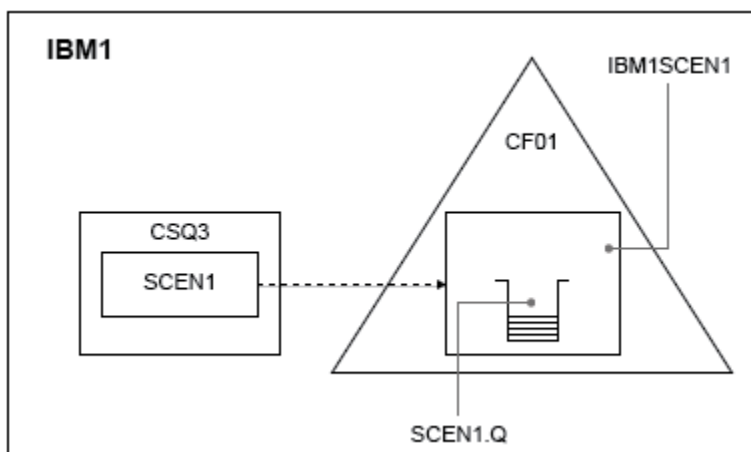
Oznacza to, że używana kolejka współużytkowana musi zawierać dwie godziny komunikatów z aplikacji umieszczającej. Obecnie można to osiągnąć za pomocą domyślnych reguł przenoszenia i SMDS.

Oczekuje się, że szybkość wysyłania komunikatów do kolejki współużytkowanej podwoi się w perspektywie krótko- i średnioterminowej. Pomimo tego, że system nadal musi tolerować dwugodzinne wyłączenie, w systemie CF nie ma wystarczającej ilości pamięci rzeczywistej, aby podwoić wielkość struktury. Ponieważ system CF zawierający strukturę aplikacji rezyduje na komputerze z systemem zEC12, istnieje możliwość powiązania wystarczającego systemu SCM ze strukturą do przechowywania wystarczającej liczby komunikatów, co pozwala na tolerowanie dwugodzinnego wyłączenia.

W tym scenariuszu początkowym używane są:

- Grupa współużytkowania kolejek IBM1, która zawiera pojedynczy menedżer kolejek CSQ3. Oprócz struktury administracyjnej grupa współużytkowania kolejek zdefiniuje pojedynczą strukturę aplikacji SCEN1.
- Narzędzie CF CF01, w którym struktura aplikacji SCEN1 jest przechowywana jako struktura IBM1SCEN1. Maksymalna wielkość tej struktury wynosi 1 GB.
- Pojedyncza kolejka współużytkowana SCEN1. Q używana przez strukturę aplikacji.

Ta konfiguracja została zilustrowana na ilustracji Rysunek 61 na stronie 204.



Rysunek 61. Konfiguracja podstawowa

Ponadto założymy, że menedżer kolejek CSQ3 jest już jedynym elementem grupy współużytkowania kolejek IBM1.

Należy dodać definicję struktury IBM1SCEN1 do strategii menedżera zasobów CF (Coupling Facility) menedżera zasobów CFRM. Dla uproszczenia struktura jest definiowana w taki sposób, aby można ją było utworzyć tylko w pojedynczym narzędziu CF CF01, podając parametr PREFLIST (CF01).



Ostrzeżenie: Aby zapewnić wysoką dostępność w systemie produkcyjnym, należy dołączyć co najmniej dwa systemy CF do pliku PREFLIST dla wszystkich struktur używanych przez produkt IBM MQ.

Procedura

1. Odśwież strategię CFRM za pomocą następującej komendy:

```
SETXCF START,POLICY,TYPE=CFRM,POLNAME=IBM1SCEN1
```

Przykładowa strategia CFRM dla struktury IBM1SCEN1:

```
STRUCTURE  
NAME(IBM1SCEN1)  
SIZE(1024M)  
INITSIZE(512M)  
ALLOWAUTOALT(YES)  
FULLTHRESHOLD(85)  
PREFLIST(CF01)  
ALLOWREALLOCATE(YES)  
DUPLEX(DISABLED)  
ENFORCEORDER(NO)
```

2. Sprawdź, czy struktura została poprawnie utworzona, używając następującej komendy:

```
D XCF,STR,STRNAME=IBM1SCEN1
```

W tym momencie struktura nie została przydzielona (wyświetlana w wierszu STATUS) do grupy współużytkowania kolejek.

3. Skonfiguruj IBM MQ, aby użyć struktury zdefiniowanej w strategii CFRM.

- a. Użyj komendy DEFINE CFSTRUCT z nazwą struktury SCEN1, aby utworzyć obiekt IBM MQ CFSTRUCT:

```
DEFINE CFSTRUCT(SCEN1)  
CFCONLOS(TOLERATE)
```

```
CFLEVEL(5)
DESCR('Structure for SCM scenario 1')
RECOVER(NO)
RECAUTO(YES)
OFFLOAD(DB2)
OFFLD1SZ(64K) OFFLD1TH(70)
OFFLD2SZ(64K) OFFLD2TH(80)
OFFLD3SZ(64K) OFFLD3TH(90)
```

- b. Sprawdź poprawność struktury za pomocą komendy [DISPLAY CFSTRUCT](#) .
- c. Zdefiniuj współużytkowaną kolejkę SCEN1 . Q , aby użyć struktury SCEN1 , używając następującej komendy MQSC:

```
DEFINE QLOCAL(SCEN1.Q) QSGDISP(SHARED) CFSTRUCT(SCEN1) MAXDEPTH(999999999)
```

4. Użyj komendy IBM MQ Explorer , aby umieścić pojedynczy komunikat w kolejce SCEN1 . Q i ponownie go zdjąć.
5. Wydadź następującą komendę, aby sprawdzić, czy struktura jest teraz przydzielona:

```
D XCF,STR,STRNAME=IBM1SCEN1
```

Sprawdź dane wyjściowe komendy, w wierszu STATUS wpisz ALLOCATED.

Wyniki

Utworzono konfigurację podstawową. Teraz można uzyskać informacje na temat podstawowej wydajności konfiguracji przy użyciu dowolnej wybranej metody.

Co dalej

Dodaj SMDS i SCM do struktury początkowej

Pojęcia pokrewne

[“Użycie pamięci klasy pamięci masowej z kolejkami współużytkowanymi” na stronie 195](#)

Użycie pamięci klasy pamięci masowej (SCM) może być korzystne w przypadku używania z kolejkami współużytkowanymi IBM MQ for z/OS .

 *Dodawanie zestawów SMDS i SCM do struktury początkowej*

Sposób dodawania zestawów SMDS i SCM na potrzeby pamięci awaryjnej w systemie IBM MQ.

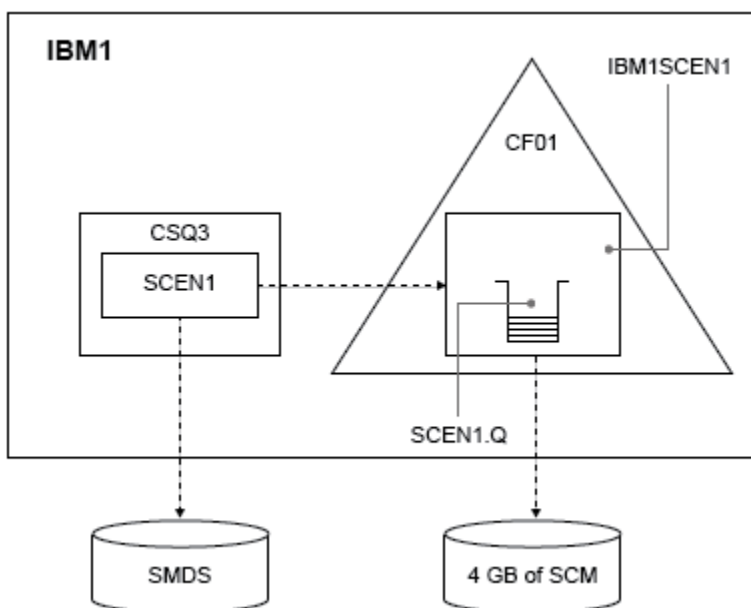
O tym zadaniu

Ważne: Planowane jest, że IBM z16 będzie ostatnią generacją IBM Z[®] obsługującej użycie wirtualnej pamięci Flash (znanej również jako pamięć klasy pamięci masowej lub SCM) dla obrazów narzędzia CF. Więcej informacji na ten temat zawierają: [IBM Z i IBM LinuxONE 4Q 2023 Oświadczenia o kierunku](#).

Alternatywnie należy użyć większych struktur lub przenieść komunikaty do SMDS.

Ta część zadania korzysta z konfiguracji podstawowej opisanej w sekcji [“Pamięć awaryjna-podstawowa konfiguracja” na stronie 203](#). Scenariusz opisuje dodawanie współużytkowanych zestawów danych komunikatów (SMDS), a następnie systemu SCM do struktury początkowej.

Ta ostateczna konfiguracja została zilustrowana w sekcji [Rysunek 62 na stronie 206](#).



Rysunek 62. Konfiguracja z dodaniem SMDS i SCM dla pamięci masowej awaryjnej

Procedura

1. Utwórz zestaw danych SMDS używany przez strukturę aplikacji SCEN1 , edytując przykładowy kod JCL **CSQ4SMDS** , jak pokazano poniżej:

```
//CSQ4SMDS JOB NOTIFY=&SYSUID
//*
/* Allocate SMDS
/*
//DEFINE EXEC PGM=IDCAMS,REGION=4M
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DEFINE CLUSTER          -
(NAME(CSQSMDS.SCEN1.CSQ3.SMDS) -
MEGABYTES(5000 3000)   -
LINEAR                 -
SHAREOPTIONS(2 3)     -
DATA                   -
(NAME(CSQSMDS.SCEN1.CSQ3.SMDS.DATA) )
/*
/*
/* Format the SMDS
/*
//FORM EXEC PGM=CSQJUFMT,COND=(0,NE),REGION=0M
//STEPLIB DD DSN=MQ800.SCSQANLE,DISP=SHR
// DD DSN=MQ800.SCSQAUTH,DISP=SHR
//SYSUT1 DD DISP=OLD,DSN=CSQSMDS.SCEN1.CSQ3.SMDS
//SYSPRINT DD SYSOUT=*
```

2. Wydadź komendę **ALTER CFSTRUCT** , aby zmienić strukturę aplikacji SCEN1 w celu użycia SMDS do odciążania, implementując domyślne reguły odciążania:

```
ALTER CFSTRUCT(SCEN1) OFFLOAD(SMDS) OFFLD1SZ(32K) OFFLD2SZ(4K) OFFLD3SZ(0K)
DSGROUP('CSQSMDS.SCEN1.*.SMDS') DSBLOCK(1M)
```

Na co zwrócić uwagę:

- Ponieważ SCEN1.Q jest jedyną kolejką współużytkowaną w strukturze aplikacji SCEN1 , wartość **DSBLOCK** została ustawiona na 1M, co jest największą możliwą wartością. To powinno być najbardziej efektywne ustawienie dla naszego scenariusza.

- Ponieważ komunikaty wysyłane przez aplikację umieszczającą mają wielkość 30 kB, przenoszenie do SMDS nie jest uruchamiane do momentu spełnienia drugiej reguły przenoszenia, gdy struktura jest zapelniona w 80%.
- Uruchom ponownie aplikację testową.
Zanotuj zwiększoną ilość pamięci komunikatów w kolejce.
 - Dodaj 4 GB SCM do struktury IBM1SCEN1 , wykonując następującą procedurę:
 - Sprawdź ilość zainstalowanego systemu SCM i przydzielonego do produktu CF01, wprowadzając następującą komendę:

```
D CF,CFNAME=CF01
```

- Sprawdź wartości STORAGE - CLASS MEMORY w sekcji STORAGE CONFIGURATION wyświetlanych danych wyjściowych, aby wyświetlić dostępną pamięć masową.
- Zaktualizuj strategię CFRM, używając słów kluczowych SCMMAXSIZE i SCMALGORITHM w następujący sposób:

```
STRUCTURE
NAME (IBM1SCEN1)
SIZE (1024M)
INITSIZE (512M)
ALLOWAUTOALT (YES)
FULLTHRESHOLD (85)
PREFLIST (CF01)
ALLOWREALLOCATE (YES)
DUPLEX (DISABLED)
ENFORCEORDER (NO)
SCMMAXSIZE (4G)
SCMALGORITHM (KEYPRIORITY1)
```

- Aktywuj strategię CFRM, wprowadzając następującą komendę:

```
SETXCF START,POLICY,TYPE=CFRM,POLNAME=polname
```

- Odbuduj strukturę IBM1SCEN1 .

Tę procedurę należy wykonać, ponieważ struktura została przydzielona podczas wprowadzania wcześniejszych zmian.

Wprowadź następującą komendę, aby odbudować strukturę:

```
SETXCF START,REBUILD,STRNM=IBM1SCEN1
```

Wyniki

System SCM został pomyślnie dodany do konfiguracji.

Co dalej

Optymalizacja wydajności systemu. Więcej informacji zawiera sekcja [“Optymalizowanie wykorzystania pamięci klasy pamięci masowej”](#) na stronie 207.

 *Optymalizowanie wykorzystania pamięci klasy pamięci masowej*

W jaki sposób można usprawnić korzystanie z pamięci klasy pamięci masowej (SCM).

Ważne: Planowane jest, że IBM z16 będzie ostatnią generacją IBM Z[®] obsługującej użycie wirtualnej pamięci Flash (znanej również jako pamięć klasy pamięci masowej lub SCM) dla obrazów narzędzia CF. Więcej informacji na ten temat zawierają: [IBM Z i IBM LinuxONE 4Q 2023 Oświadczenia o kierunku](#).

Alternatywnie należy użyć większych struktur lub przenieść komunikaty do SMDS.

Uruchom następującą komendę:

```
D XCF,STR,STRNAME=IBM1SCEN1
```

Ponieważ struktura była już pełna danymi komunikatu, z powodu poprzednich testów część odbudowywania obejmowała wstępne przemieszczanie niektórych komunikatów ze struktury do systemu SCM. Ten proces został zainicjowany przy użyciu poprzedniej komendy.

Dane wyjściowe tej komendy są generowane, na przykład:

```
ACTIVE STRUCTURE
-----
ALLOCATION TIME: 06/17/2014 09:28:50
CFNAME : CF01
COUPLING FACILITY: 002827.IBM.02.00000000B8D7
PARTITION: 3B CPCID: 00
STORAGE CONFIGURATION ALLOCATED MAXIMUM %
ACTUAL SIZE: 1024 M 1024 M 100
AUGMENTED SPACE: 3 M 142 M 2
STORAGE-CLASS MEMORY: 88 M 4096 M 2
ENTRIES: 120120 1089536 11
ELEMENTS: 240240 15664556 1
SPACE USAGE IN-USE TOTAL %
ENTRIES: 84921 219439 38
ELEMENTS: 2707678 3149050 85
EMCS: 2 282044 0
LOCKS: 1024
SCMHIGHTHRESHOLD : 90
SCMLOWTHRESHOLD : 70
ACTUAL SUBNOTIFYDELAY: 5000
PHYSICAL VERSION: CD5186A0 2BD8B85C
LOGICAL VERSION: CD515C50 CE2ED258
SYSTEM-MANAGED PROCESS LEVEL: 9
XCF GRPNAME : IXCLO053
DISPOSITION : KEEP
ACCESS TIME : NOLIMIT
MAX CONNECTIONS : 32
# CONNECTIONS : 1
CONNECTION NAME ID VERSION SYSNAME JOBNAME ASID STATE
-----
CSQEIBM1CSQ301 01 00010059 SC61 CSQ3MSTR 0091 ACTIVE
```

Należy zwrócić uwagę na następujące informacje z danych wyjściowych komendy:

- Program STORAGE_CLASS MEMORY potwierdza, że do struktury dodano system SCM **MAXIMUM** o wielkości 4096 MB.
- Wartość ALLOCATED określa ilość danych STORAGE-CLASS MEMORY używanych na potrzeby przemieszczania wstępnego. W strukturze jest teraz wolne miejsce, w którym nie było miejsca przed dodaniem komponentu SCM.
- Ilość AUGMENTED SPACE używana do śledzenia użycia SCM.
- Punkt, w którym algorytm wstępnego przemieszczania rozpoczyna przenoszenie danych ze struktury do systemu SCM, jest wtedy, gdy struktura jest zapełniona w 90%. Jest to wskazywane przez niekonfigurowalną właściwość **SCMHIGHTHRESHOLD**.
- Poniżej znajduje się punkt, w którym algorytm pobierania wstępnego rozpoczyna przenoszenie danych z systemu SCM do struktury, gdy struktura jest zapełniona w 70%. Jest to wskazywane przez niekonfigurowalną właściwość **SCMLOWTHRESHOLD**.

Teraz można przetestować różne sposoby optymalizacji użycia systemu SCM. Na co zwrócić uwagę:

- Po użyciu systemu SCM do przechowywania komunikatów nie można zmienić struktury, dopóki wszystkie dane nie zostaną usunięte z systemu SCM.

W tym przypadku oznacza to, że współczynnik pozycji do elementu jest zamrożony w wartości, która znajdowała się w miejscu, gdy system SCM był używany po raz pierwszy. Należy upewnić się, że struktura jest w pożądanym stanie, zanim algorytm wstępnego przemieszczania rozpocznie przenoszenie danych do systemu SCM.

- Czy bieżąca wielkość struktury jest poprawna przed użyciem SCM?

Na przykład, czy zwiększono **INITSIZE** z 512 MB do 1 GB?

Jeśli ta czynność nie zostanie włączona, możliwe jest, że pomimo włączenia automatycznej modyfikacji struktury, algorytm wstępnego przemieszczania rozpocznie przenoszenie danych do systemu SCM,

zanim będzie można rozpocząć tę zmianę. W wyniku tego struktura jest zamrożona przy użyciu 512 MB pamięci rzeczywistej.

- Czy współczynnik pozycji do elementu jest poprawny przed użyciem systemu SCM?

Celem tego scenariusza jest zwiększenie liczby odciążonych wskaźników komunikatów, które mogą być przechowywane w strukturze i systemie SCM jako całości, a także zachowanie jak największej liczby komunikatów w pamięci masowej struktury. Dostęp do tych komunikatów jest szybszy niż dostęp do komunikatów w SMDS.

Dlatego konieczne jest posiadanie struktury rozpoczynającej się współczynnikiem wejścia do elementu, który jest dobry do przechowywania komunikatów, a następnie przejście do współczynnika, który jest dobry do przechowywania wskaźników komunikatów przed pierwszym uruchomieniem algorytmu wstępnego etapu. To przejście można częściowo wykonać, korzystając z reguły przenoszenia IBM MQ .

Zmień reguły przenoszenia, wydając następującą komendę:

```
ALTER CFSTRUCT(SCEN1) OFFLD1SZ(0K)
```

Może być konieczne wykonanie kilku przebiegów w celu zoptymalizowania współczynników wejścia do elementu.

W poniższej tabeli przedstawiono możliwe ulepszenia dotyczące liczby komunikatów umieszczanych w kolejce w różnych fazach scenariusza pamięci awaryjnej.

Opis testu	Liczba komunikatów	Czas wypełniania kolejki (sekundy)
Konfiguracja podstawowa	27 850	3.2
SMDS z domyślnymi regułami przenoszenia	205 000	158
SCM z domyślnymi regułami przenoszenia	828 610	469
SCM z dopasowanymi regułami przenoszenia	1,135,775	679

Ostatni wiersz w tabeli pokazuje, że dopasowywanie reguł przenoszenia miało wymagany efekt.

Należy sprawdzić system, aby sprawdzić, czy można poprawić te dane liczbowe w jakikolwiek sposób. Na przykład może zabraknąć dostępnej pamięci masowej SMDS. Jeśli można przydzielić więcej pamięci SMDS, powinno być możliwe znaczne zwiększenie liczby komunikatów w kolejce.

Poprawiona wydajność-podstawowa konfiguracja

W jaki sposób można skonfigurować podstawowy scenariusz w celu zwiększenia wydajności przy użyciu kolejek współużytkowanych w systemie IBM MQ.

O tym zadaniu

Ważne: Planowane jest, że IBM z16 będzie ostatnią generacją IBM Z[®] obsługującej użycie wirtualnej pamięci Flash (znanej również jako pamięć klasy pamięci masowej lub SCM) dla obrazów narzędzia CF. Więcej informacji na ten temat zawierają: [IBM Z i IBM LinuxONE 4Q 2023 Oświadczenia o kierunku](#).

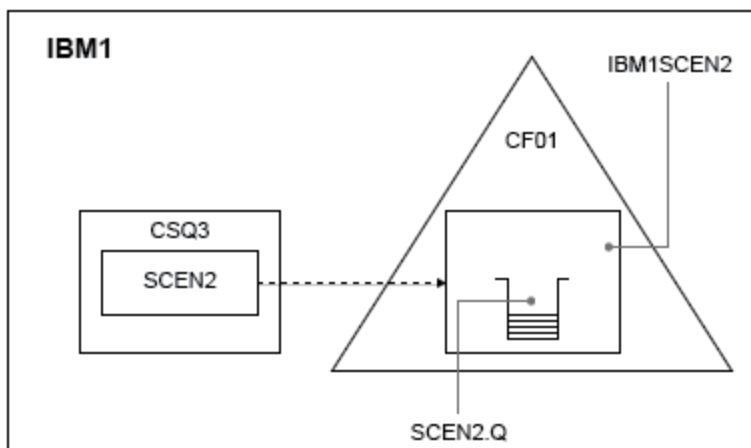
Alternatywnie należy użyć większych struktur lub przenieść komunikaty do SMDS.

W tym scenariuszu opisano użycie systemu SCM w celu zwiększenia liczby komunikatów, które mogą być przechowywane w kolejce współużytkowanej bez ponoszenia kosztów wydajności związanych z używaniem SMDS.

Ten początkowy scenariusz jest bardzo podobny do tego, który jest używany do przechowywania w nagłych wypadkach i używa:

- Grupa współużytkowania kolejek IBM1, która zawiera pojedynczy menedżer kolejek CSQ3. Oprócz struktury administracyjnej grupa współużytkowania kolejek zdefiniuje pojedynczą strukturę aplikacji SCEN2.
- Narzędzie CF CF01, w którym struktura aplikacji SCEN2 jest przechowywana jako struktura IBM1SCEN2. Maksymalna wielkość tej struktury wynosi 2 GB.
- Pojedyncza kolejka współużytkowana, SCEN2.Q, która jest skonfigurowana do korzystania ze struktury aplikacji.

Ta konfiguracja została zilustrowana na ilustracji [Rysunek 63](#) na stronie 210.



Rysunek 63. Konfiguracja podstawowa

Ponadto założmy, że menedżer kolejek CSQ3 jest już jedynym elementem grupy współużytkowania kolejek IBM1.

Należy dodać definicję struktury IBM1SCEN2 do strategii menedżera zasobów CF (Coupling Facility) menedżera zasobów CFRM. Dla uproszczenia struktura jest definiowana w taki sposób, aby można ją było utworzyć tylko w pojedynczym narzędziu CF CF01, podając parametr PREFLIST (CF01).

Przykładowa strategia CFRM dla struktury IBM1SCEN2:

```
STRUCTURE
NAME (IBM1SCEN2)
SIZE (2048M)
INITSIZE (2048M)
ALLOWAUTOALT (YES)
FULLTHRESHOLD (85)
PREFLIST (CF01)
ALLOWREALLOCATE (YES)
DUPLEX (DISABLED)
ENFORCEORDER (NO)
```

Zarówno słowa kluczowe **INITSIZE**, jak i **SIZE** mają wartość 2048M, dlatego nie można zmienić wielkości struktury.

Procedura

1. Odśwież strategię CFRM za pomocą następującej komendy:

```
SETXCF START,POLICY,TYPE=CFRM,POLNAME=IBM1SCEN2
```

2. Sprawdź, czy struktura została poprawnie utworzona, używając następującej komendy:

```
D XCF,STR,STRNAME=IBM1SCEN2
```

Wykonanie powyższej komendy powoduje wyświetlenie następujących danych wyjściowych:

```

RESPONSE=SC61
IXC360I 07.58.51 DISPLAY XCF 581
STRNAME: IBM1SCEN2
STATUS: NOT ALLOCATED
POLICY INFORMATION:
POLICY SIZE : 2048 M
POLICY INITSIZE: 2048 M
POLICY MINSIZE : 1536 M
FULLTHRESHOLD : 85
ALLOWAUTOALT : YES
REBUILD PERCENT: N/A
DUPLEX : DISABLED
ALLOWREALLOCATE: YES
PREFERENCE LIST: CF01
ENFORCEORDER : NO
EXCLUSION LIST IS EMPTY

EVENT MANAGEMENT: MESSAGE-BASED   MANAGER SYSTEM NAME: SC53
MANAGEMENT LEVEL : 01050107

```

W tym momencie struktura nie została przydzielona (wyświetlana w wierszu STATUS) do grupy współużytkowania kolejek.

3. Skonfiguruj IBM MQ , aby użyć struktury zdefiniowanej w strategii CFRM.

- a. Użyj komendy DEFINE CFSTRUCT z nazwą struktury SCEN2 , aby utworzyć obiekt IBM MQ CFSTRUCT.

```

DEFINE CFSTRUCT(SCEN2)
CFCONLOS(TOLERATE)
CFLEVEL(5)
DESCR('Structure for SCM scenario 2')
RECOVER(NO)
RECAUTO(YES)
OFFLOAD(DB2)
OFFLD1SZ(64K) OFFLD1TH(70)
OFFLD2SZ(64K) OFFLD2TH(80)
OFFLD3SZ(64K) OFFLD3TH(90)

```

- b. Sprawdź strukturę za pomocą komendy DISPLAY CFSTRUCT .
- c. Zdefiniuj współużytkowaną kolejkę SCEN2 . Q , aby użyć struktury SCEN2 , używając następującej komendy MQSC:

```

DEFINE QLOCAL(SCEN2.Q) QSGDISP(SHARED) CFSTRUCT(SCEN2) MAXDEPTH(999999999)

```

4. Użyj programu IBM MQ Explorer, aby umieścić pojedynczy komunikat w kolejce SCEN2 . Q i ponownie go zdjąć.
5. Wydadaj następującą komendę, aby sprawdzić, czy struktura jest teraz przydzielona:

```

D XCF,STR,STRNAME=IBM1SCEN2

```

Przejrzyj dane wyjściowe komendy, której część jest wyświetlana, i upewnij się, że w wierszu STATUS jest wyświetlana wartość ALLOCATED.

```

RESPONSE=SC61
IXC360I 08.31.27 DISPLAY XCF 703
STRNAME: IBM1SCEN2
STATUS: ALLOCATED
EVENT MANAGEMENT: MESSAGE-BASED
TYPE: SERIALIZED LIST
POLICY INFORMATION:
POLICY SIZE : 2048 M
POLICY INITSIZE: 2048 M
POLICY MINSIZE : 1536 M
FULLTHRESHOLD : 85
ALLOWAUTOALT : YES
REBUILD PERCENT: N/A
DUPLEX : DISABLED
ALLOWREALLOCATE: YES

```

```
PREFERENCE LIST: CF01
ENFORCEORDER : NO
EXCLUSION LIST IS EMPTY
```

Ponadto należy zwrócić uwagę na wartości pól w sekcji SPACE USAGE:

- Pozycje
- Elementy
- EMCS,
- BLOKADY

Poniżej przedstawiono przykładowe wartości:

SPACE USAGE	IN-USE	TOTAL	%
ENTRIES:	344686	345242	99
ELEMENTS:	6548455	6548467	99
EMCS:	2	780318	0
LOCKS:	1024		

Wyniki

Utworzono konfigurację podstawową. Teraz można uzyskać informacje na temat podstawowej wydajności konfiguracji przy użyciu dowolnej wybranej metody.

Co dalej

Należy przetestować podstawowy scenariusz. Na przykład można użyć następujących trzech aplikacji, uruchamiając aplikacje w podanej kolejności i uruchamiając je współbieżnie.

1. Użyj aplikacji PCF, aby zażądać bieżącej głębokości (**CURDEPTH**) wartość SCEN2 . Q co pięć sekund. Dane wyjściowe mogą być użyte do wykreślenia głębokości kolejki w czasie.
2. Aplikacja pobierająca w pojedynczym wątku wielokrotnie pobiera komunikaty z produktu SCEN2 . Q, korzystając z metody get z nieskończonym oczekiwaniem. Aby zasymulować przetwarzanie usuniętych komunikatów, aplikacja pobierająca zatrzyma się na cztery milisekundy na każde dziesięć usuniętych komunikatów.
3. Pojedyncza aplikacja umieszczania w wątku umieszcza w pliku SCEN2 . Q łącznie milion nietrwałych komunikatów o wielkości 4 kB. Ta aplikacja nie jest wstrzymywana między umieszczaniem poszczególnych komunikatów, więc komunikaty są umieszczane w produkcie SCEN2 . Q szybciej niż może je uzyskać aplikacja pobierająca.

W wyniku tego, gdy aplikacja umieszczana jest uruchomiona, głębokość SCEN2 . Q zwiększa się.

Gdy struktura IBM1SCEN2 jest wypełniona, a aplikacja umieszczająca odbiera kod przyczyny MQRC_STORAGE_MEDIUM_FULL, aplikacja umieszczająca jest uśpiona przez pięć sekund przed próbą umieszczenia następnego komunikatu w kolejce.

Wyniki aplikacji CURDEPTH można wykreślić w czasie. Otrzymujesz jakąś formę wyjścia fal saw-zęba, ponieważ aplikacja wstrzymuje się, aby umożliwić częściowe opróżnianie kolejki.

Przejdź do sekcji [“Dodawanie systemu SCM do struktury początkowej”](#) na stronie 212.

Pojęcia pokrewne

[“Użycie pamięci klasy pamięci masowej z kolejkami współużytkowanymi”](#) na stronie 195

Użycie pamięci klasy pamięci masowej (SCM) może być korzystne w przypadku używania z kolejkami współużytkowanymi IBM MQ for z/OS .

 *Dodawanie systemu SCM do struktury początkowej*

Sposób dodawania komponentu SCM w celu zwiększenia wydajności w produkcie IBM MQ.

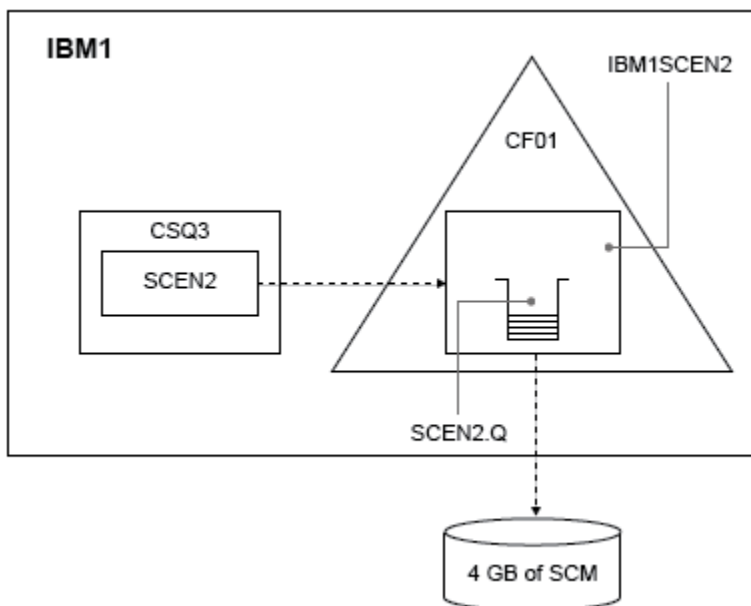
O tym zadaniu

Ważne: Planowane jest, że IBM z16 będzie ostatnią generacją IBM Z® obsługującej użycie wirtualnej pamięci Flash (znanej również jako pamięć klasy pamięci masowej lub SCM) dla obrazów narzędzia CF. Więcej informacji na ten temat zawierają: [IBM Z i IBM LinuxONE 4Q 2023 Oświadczenia o kierunku](#).

Alternatywnie należy użyć większych struktur lub przenieść komunikaty do SMDS.

Ta część zadania korzysta z konfiguracji podstawowej opisanej w sekcji [“Poprawiona wydajność- podstawowa konfiguracja”](#) na stronie 209. Scenariusz opisuje dodawanie systemu SCM do struktury początkowej.

Ta ostateczna konfiguracja została zilustrowana w sekcji [Rysunek 64](#) na stronie 213.



Rysunek 64. Dodawanie konfiguracji SCM w celu zwiększenia wydajności

Procedura

1. Dodaj 4 GB SCM do struktury IBM1SCEN2 , wykonując następującą procedurę:

- Sprawdź ilość zainstalowanego systemu SCM i przydzielonego do produktu CF01, wprowadzając następującą komendę:

```
D CF,CFNAME=CF01
```

- Sprawdź wartości STORAGE - CLASS MEMORY w sekcji STORAGE CONFIGURATION wyświetlanych danych wyjściowych, aby wyświetlić dostępną pamięć masową.
- Zaktualizuj strategię CFRM, używając słów kluczowych SCMMAXSIZE i SCMALGORITHM w następujący sposób:

```
STRUCTURE  
NAME(IBM1SCEN2)  
SIZE(2048M)  
INITSIZE(2048M)  
ALLOWAUTOALT(YES)  
FULLTHRESHOLD(85)  
PREFLIST(CF01)  
ALLOWREALLOCATE(YES)  
DUPLEX(DISABLED)  
ENFORCEORDER(NO)  
SCMMAXSIZE(4G)  
SCMALGORITHM(KEYPRIORITY1)
```

2. Aktywuj strategię CFRM, wprowadzając następującą komendę:

```
SETXCF START ,POLICY ,TYPE=CFRM ,POLNAME=IBM1SCEN2
```

3. Odbuduj strukturę IBM1SCEN2 .

Tę procedurę należy wykonać, ponieważ struktura została przydzielona podczas wprowadzania wcześniejszych zmian.

Wprowadź następującą komendę, aby odbudować strukturę:

```
SETXCF START ,REBUILD ,STRNM=IBM1SCEN2
```

4. Wydadź następującą komendę, aby potwierdzić nową konfigurację struktury:

```
D XCF ,STR ,STRNAME=IBM1SCEN2
```

Przejrzyj dane wyjściowe komendy, z których część jest następująca:

SPACE USAGE	IN-USE	TOTAL	%
ENTRIES:	33	342684	0
ELEMENTS:	48	6503697	0
EMCS:	2	575600	0
LOCKS:		1024	

Wyniki

Oblicz zmianę w użyciu rzeczywistej pamięci masowej przez zwiększenie pamięci masowej kontroli wymaganej do użycia systemu SCM.

- Przed dodaniem SCM do struktury, struktura ma następujące sumy, jak pokazano w [“Poprawiona wydajność-podstawowa konfiguracja”](#) na stronie 209:
 - 345,242 pozycje
 - 6,548,467 elementów
 - 780 318 EMCS
- Po dodaniu systemu SCM do struktury struktura zawiera następujące sumy:
 - 342,684 pozycji
 - 6,503,697 elementów
 - 575600 EMCS

Na podstawie tych danych liczbowych, po dodaniu systemu SCM, struktura jest zmniejszana o:

- 2558 wpisów
- 44 770 elementów
- 204,718 EMCS

Ilość pamięci masowej struktury, która jest używana do zarządzania systemem SCM, jest następująca dla struktury 2 GB z 4 GB przydzielonego systemu SCM:

$$(2558 + 44,770 + 204,718) * 256 = 61.5 \text{ MB}$$

Należy zauważyć, że dodanie większej liczby elementów SCM prawdopodobnie spowoduje tylko marginalne zmniejszenie wielkości struktury, ponieważ ilość pamięci masowej kontroli używanej do śledzenia wzrostów SCM, zarówno jako wielkość struktury, jak i ilość przydzielonych wzrostów SCM.

Co dalej

Powtórz testy opisane w końcowej sekcji sekcji [“Poprawiona wydajność-podstawowa konfiguracja”](#) na stronie 209.

Wyniki skorygowanej aplikacji można wykreślić w czasie. Porównując wykres z wcześniej uzyskanym, otrzymujesz teraz dane wyjściowe bez fali saw-tooth, ponieważ aplikacja nie musi już czekać na częściowe opróżnianie kolejki.

Więcej informacji na ten temat zawiera sekcja [MP16: WebSphere MQ for z/OS -planowanie i strojenie mocy obliczeniowej](#).

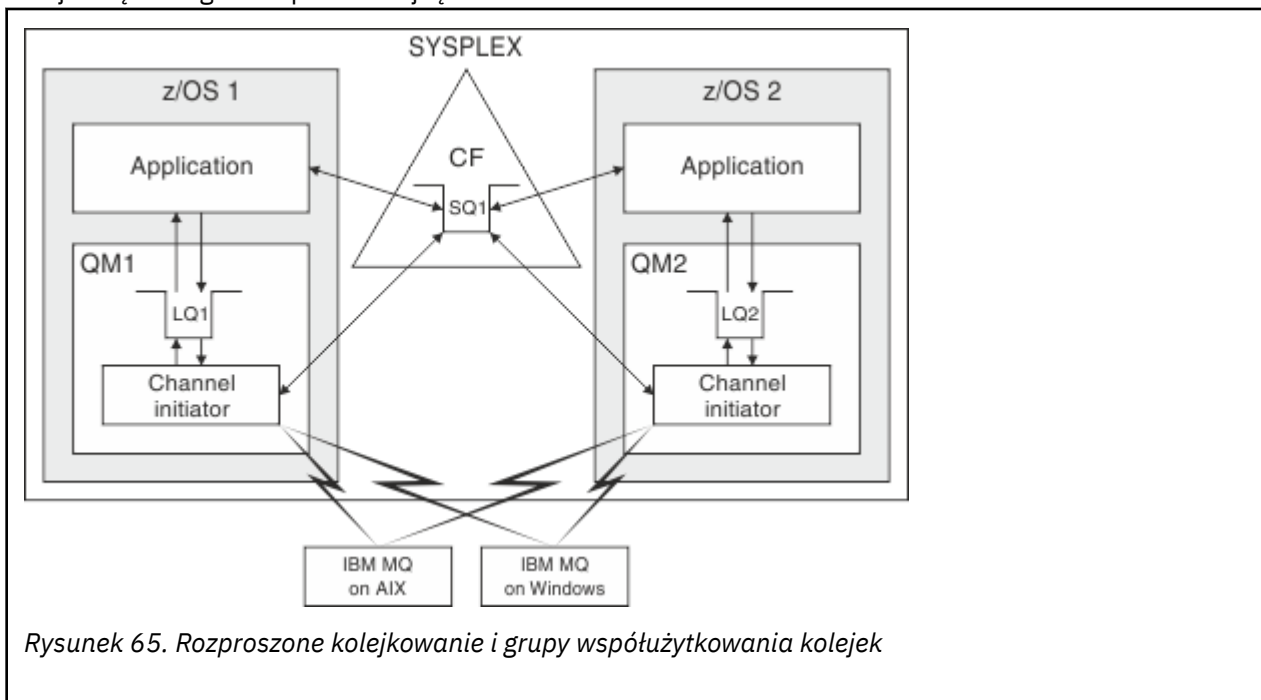
z/OS Rozproszone kolejkowanie i grupy współużytkowania kolejek

Rozproszone kolejkowanie i grupy współużytkowania kolejek to dwie techniki, których można użyć w celu zwiększenia dostępności systemów aplikacji. Więcej informacji na temat tych technik można znaleźć w tej sekcji.

Aby uzupełnić wysoką dostępność komunikatów w kolejkach współużytkowanych, komponent rozproszonego kolejkowania produktu IBM MQ zawiera dodatkowe funkcje umożliwiające:

- Wyższa dostępność dla sieci.
- Zwiększona pojemność przychodzących połączeń sieciowych z grupą współużytkowania kolejek.

Rysunek 65 na stronie 215 przedstawia rozproszone kolejkowanie i grupy współużytkowania kolejek. Wyświetlane są dwa menedżery kolejek w obrębie sysplexu, z których oba należą do tej samej grupy współużytkowania kolejek. Oba mają dostęp do współużytkowanej kolejki SQ1. Menedżery kolejek w sieci (na przykład w systemach AIX i Windows) mogą umieszczać komunikaty w tej kolejce za pośrednictwem inicjatora kanału dowolnego z tych menedżerów kolejek. Sklonowane aplikacje w obu menedżerach kolejek są obsługiwane przez kolejkę.



Rysunek 65. Rozproszone kolejkowanie i grupy współużytkowania kolejek

Pojęcia pokrewne

[“Kanały współużytkowane”](#) na stronie 216

Ten temat zawiera informacje o pojęciach związanych z kanałami współużytkowanymi i ich używaniem w produkcie IBM MQ for z/OS.

[“Kolejkowanie wewnątrz grupy”](#) na stronie 221

W tej sekcji opisano kolejkowanie wewnątrz grupy, funkcję IBM MQ for z/OS unikalną dla platformy z/OS. Ta funkcja jest dostępna tylko dla menedżerów kolejek zdefiniowanych dla grupy współużytkowania kolejek.

“Klastry i grupy współużytkowania kolejek” na stronie 218

W tym temacie opisano, w jaki sposób można używać grup współużytkowania kolejek z klastrami.

Kanały współużytkowane

Ten temat zawiera informacje o pojęciach związanych z kanałami współużytkowanymi i ich używaniem w produkcie IBM MQ for z/OS.

Wiele produktów sieciowych udostępnia mechanizm ukrywania awarii serwera w sieci lub równoważenia przychodzących żądań sieciowych między różnymi serwerami. Produkty sieciowe udostępniają *port ogólny* dla przychodzących żądań połączeń sieciowych, a żądanie przychodzące można spełnić, łącząc się z jednym z zakwalifikowanych serwerów.

Do tych produktów sieciowych należą:

- Zasoby ogólne VTAM
- Dystrybutor SYSPLEX

Inicjator kanału wykorzystuje te produkty do korzystania z możliwości kolejek współużytkowanych

Istnieją dwa typy kanałów współużytkowanych: *współużytkowany kanał przychodzący* i *współużytkowany kanał wychodzący*.

- [Współużytkowane kanały przychodzące](#)
- [Współużytkowane kanały wychodzące](#)

Aby uzyskać więcej informacji na temat kanałów, patrz

- [Podsumowanie kanału współużytkowanego](#)
- [Status kanału współużytkowanego](#)

Współużytkowane kanały przychodzące

Każdy inicjator kanału w grupie współużytkowania kolejek uruchamia dodatkowe zadanie nasłuchiwanie, które nasłuchuje na *porcie ogólnym*. Ten port ogólny jest udostępniany sieci przez jedną z technologii pomocniczych (VTAM, TCP/IP). Przychodzące żądania przyłączenia sieciowego do portu ogólnego są rozsyłane przez technologię sieciową do dowolnego programu nasłuchującego w grupie współużytkowania kolejek (QSG), który nasłuchuje na porcie ogólnym.

Kanał można uruchomić w inicjatorze kanału, do którego jest kierowane przyłączenie przychodzące, jeśli inicjator kanału ma dostęp do definicji kanału dla kanału o tej nazwie. Definicję kanału można zdefiniować jako prywatną dla menedżera kolejek lub jako składowaną we współużytkowanym repozytorium i dostępną w dowolnym miejscu (definicja globalna). Oznacza to, że można udostępnić definicję kanału w dowolnym inicjatorze kanału w grupie współużytkowania kolejek, definiując ją jako definicję globalną.

Istnieje dodatkowa różnica podczas uruchamiania kanału za pośrednictwem portu ogólnego. Synchronizacja kanału odbywa się z grupą współużytkowania kolejek, a nie z pojedynczym menedżerem kolejek. Załóżmy na przykład, że zdalny menedżer kolejek uruchamia kanał za pośrednictwem portu ogólnego. Przy pierwszym uruchomieniu kanału może on zostać uruchomiony w menedżerze kolejek QM1 i w przepływie komunikatów. Jeśli kanał zostanie zatrzymany i zrestartowany w menedżerze kolejek QM2, informacje o liczbie komunikatów, które przepłyły, są nadal poprawne, ponieważ synchronizacja jest wykonywana z grupą współużytkowania kolejek.

Aby umieścić komunikaty w dowolnej kolejce, można użyć kanału danych przychodzących uruchomionego za pośrednictwem portu ogólnego. Zdalny menedżer kolejek nie wie, czy kolejka docelowa jest współużytkowana, czy nie. Jeśli kolejka docelowa jest kolejką współużytkowaną, zdalny menedżer kolejek łączy się za pośrednictwem dowolnego dostępnego inicjatora kanału w sposób z równoważeniem obciążenia, a komunikaty są umieszczane w kolejce współużytkowanej.

Jeśli kolejka docelowa jest kolejką prywatną, komunikaty są umieszczane w kolejce prywatnej, której właścicielem jest menedżer kolejek, z którym jest połączona bieżąca instancja kanału. W tym środowisku, zwanym *replikowanymi kolejkami lokalnymi*, każdy menedżer kolejek musi mieć zdefiniowany ten sam zestaw kolejek prywatnych.

Konfigurowanie kanałów SVRCONN dla grupy współużytkowania kolejek

Optymalną konfiguracją dla kanałów SVRCONN w grupie współużytkowania kolejek jest skonfigurowanie prywatnych programów nasłuchujących w każdym CHINIT, które używają innego numeru portu od punktu do punktu. Te porty nasłuchiwanie są następnie używane jako zasoby zaplecza dla nowego mechanizmu dystrybucji obciążenia, takiego jak Sysplex Distributor korzystający z wirtualnych adresów IP (VIPA). Zewnętrzny adres VIPA jest następnie używany jako adres docelowy dla definicji CLNTCONN w sieci. Kanał SVRCONN można zdefiniować za pomocą komendy QSGDISP (GROUP), więc ta sama definicja jest dostępna dla wszystkich menedżerów kolejek w grupie QSG. Taka konfiguracja pozwala uniknąć korzystania ze współużytkowanego programu nasłuchującego, a tym samym zmniejsza wpływ na wydajność grupy współużytkowania kolejek utrzymującej stan współużytkowanego kanału, który nie jest wymagany dla kanałów klient/serwer.

Współużytkowane kanały wychodzące

Kanał wychodzący jest uważany za kanał współużytkowany, jeśli przyjmuje komunikaty ze współużytkowanej kolejki transmisji. Jeśli jest współużytkowana, przechowuje informacje o synchronizacji na poziomie grupy współużytkowania kolejki. Oznacza to, że kanał może zostać zrestartowany w innej instancji menedżera kolejek i inicjatora kanału w grupie współużytkowania kolejek, jeśli podsystem komunikacyjny, inicjator kanału lub menedżer kolejek ulegnie awarii. Restartowanie uszkodzonych kanałów w ten sposób jest funkcją współużytkowanych kanałów nazywaną *odtworzeniem kanału równorzędnego*.

Równoważenie obciążenia dla współużytkowanych kanałów wychodzących

Wychodzący kanał współużytkowany jest zakwalifikowany do uruchomienia w dowolnym inicjatorze kanału w grupie współużytkowania kolejek, jeśli nie określono, że ma on zostać uruchomiony w konkretnym inicjatorze kanału. Inicjator kanału wybrany przez program IBM MQ jest określany na podstawie następujących kryteriów:

- Czy podsystem komunikacyjny jest obecnie dostępny dla inicjatora kanału?
- Czy połączenie Db2 jest dostępne dla inicjatora kanału?
- Który inicjator kanału ma najniższe bieżące obciążenie? Obciążenie obejmuje kanały, które są aktywne i ponawiane.

Podsumowanie kanału współużytkowanego

Kanały współużytkowane różnią się od kanałów prywatnych w następujący sposób:

Kanał prywatny

Powiązany z pojedynczym inicjatorem kanału.

- Kanał wychodzący używa lokalnej kolejki transmisji.
- Kanał przychodzący został uruchomiony przez port lokalny.
- Informacje o synchronizacji przechowywane w systemie SYSTEM.CHANNEL.SYNCQ .

Kanał współużytkowany

Obciążenie zbalansowane z wysoką dostępnością.

- Kanał wychodzący używa współużytkowanej kolejki transmisji.
- Kanał przychodzący został uruchomiony przez port ogólny.
- Informacje o synchronizacji przechowywane w systemie SYSTEM.QSG.CHANNEL.SYNCQ .

Za pomocą opcji CHLDISP komendy START CHANNEL można określić, czy kanał jest prywatny, czy współużytkowany. Kanał współużytkowany można uruchomić, wyzwalając go w taki sam sposób, jak kanał prywatny. Jednak po uruchomieniu kanału współużytkowanego produkt IBM MQ wykonuje równoważenie obciążenia i uruchamia kanał w najbardziej odpowiednim inicjatorze kanału w grupie współużytkowania kolejek. (W razie potrzeby można określić, że kanał współużytkowany ma być uruchamiany w konkretnym inicjatorze kanału).

Status kanału współużytkowanego

Inicjatory kanałów w grupie współużytkowania kolejek zachowują tabelę współużytkowanych statusów kanałów w programie Db2. Zapisuje, które kanały są aktywne w których inicjatorach kanałów. Tabela statusu kanału współużytkowanego jest używana w przypadku awarii inicjatora kanału lub systemu komunikacyjnego. Wskazuje, które kanały muszą zostać zrestartowane w innym inicjatorze kanału w grupie współużytkowania kolejek.

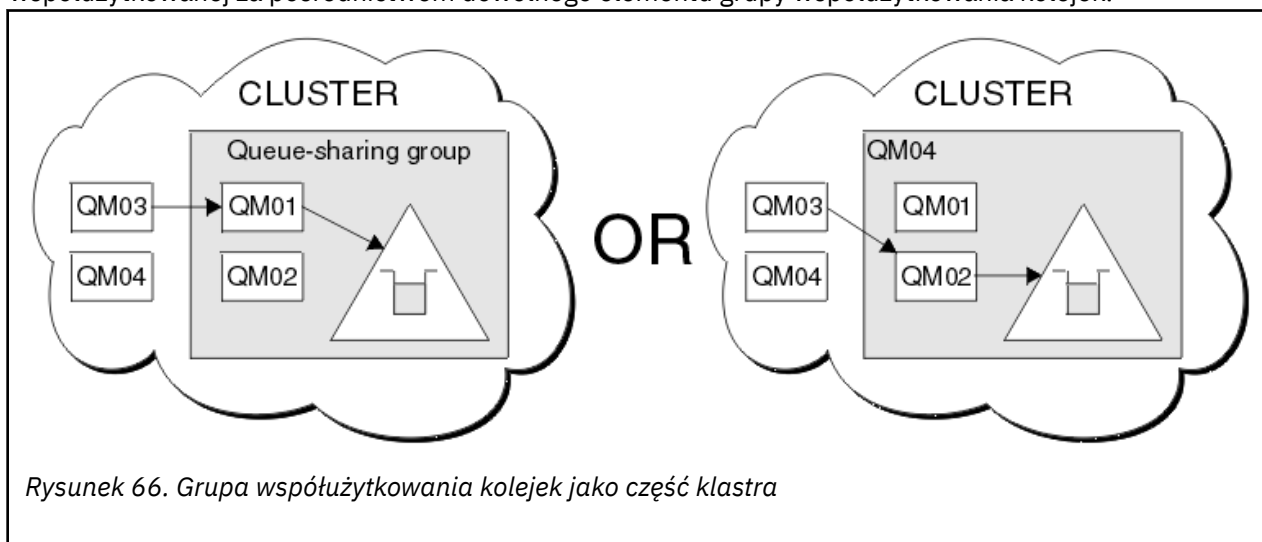
z/OS **Klasy i grupy współużytkowania kolejek**

W tym temacie opisano, w jaki sposób można używać grup współużytkowania kolejek z klastrami.

Kolejki współużytkowane można udostępnić dla klastra w pojedynczej definicji. W tym celu należy określić nazwę klastra podczas definiowania kolejki współużytkowanej.

Użytkownicy w sieci widzą współużytkowaną kolejkę jako udostępnianą przez każdego menedżera kolejek w grupie współużytkowania kolejek (kolejka współużytkowana nie jest ogłaszana jako udostępniana przez grupę współużytkowania kolejek). Klienci mogą uruchamiać sesje z dowolnym członkiem grupy współużytkowania kolejek w celu umieszczenia komunikatów w tej samej kolejce współużytkowanej.

Rysunek 66 na stronie 218 pokazuje, w jaki sposób elementy klastra mogą uzyskać dostęp do kolejki współużytkowanej za pośrednictwem dowolnego elementu grupy współużytkowania kolejek.



Rysunek 66. Grupa współużytkowania kolejek jako część klastra

z/OS **Wpływanie na dystrybucję obciążenia za pomocą kolejek współużytkowanych**

W tym temacie opisano czynniki, które mają wpływ na dystrybucję obciążenia za pomocą kolejek współużytkowanych w grupie współużytkowania kolejek.

Produkt IBM MQ nie zapewnia równoważenia obciążenia dla kolejek współużytkowanych. Jednak wpływ na dystrybucję obciążenia w grupie współużytkowania kolejek (QSG) może mieć *sposób pobierania*. Dostępne możliwości przetwarzania każdego menedżera kolejek w grupie współużytkowania kolejek i cele zarządzania obciążeniem zdefiniowane w syspleksie mają wpływ na wybór menedżera kolejek, który obsługuje kolejkę (odbiera komunikat zapisany w kolejce współużytkowanej).

Należy jednak pamiętać, że menedżer kolejek wykonujący operację MQPUT dla komunikatu może również mieć duży wpływ na wybór menedżera kolejek, który pobiera komunikat.

Istnieje większe prawdopodobieństwo, że lokalny menedżer kolejek wykona operację MQGET

W przypadku aplikacji wykonującej operację MQPUT lokalny menedżer kolejek jest to menedżer kolejek, z którym aplikacja jest połączona.

Poniższe uwagi mają wpływ na to, który menedżer kolejek obsługuje operację MQPUT komunikatu, wykonując operację MQGET w imieniu aplikacji pobierającej.

Gdy komunikat jest umieszczany w pustej kolejce współużytkowanej, menedżer kolejek lokalnych jest zwykle wysyłany przed powiadomieniem innego menedżera kolejek w grupie współużytkowania kolejek. Jeśli lokalny menedżer kolejek jest w stanie przetworzyć komunikat, otrzymuje powiadomienie o przejściu do listy z narzędzia CF przed jakimkolwiek innym menedżerem kolejek w grupie QSG. (Powiadomienie o przejściu do listy jest powiadomieniem o zmianie stanu kolejki współużytkowanej z pustego na niepusty).

W tym przypadku możliwe są następujące scenariusze:

1. MQPUT komunikatu nietrwałego poza punktem synchronizacji i *szybkie umieszczanie w oczekującej metodzie pobierającej*.

Jeśli w menedżerze kolejek lokalnych dla kolejki istnieje aplikacja z atrybutem *MQGET z oczekiwaniem*, to operacja MQPUT komunikatu jest przekazywana bezpośrednio do buforu aplikacji pobierającej i nie jest zapisywana w kolejce. Dotyczy to kolejek współużytkowanych i niewspółużytkowanych. Ta funkcja jest często nazywana *szybkim umieszczaniem w mechanizmie oczekujących procedur pobierających*. W przypadku kolejek współużytkowanych żaden inny menedżer kolejek w grupie QSG nie jest powiadamiany, ponieważ nie ma przejścia z pustej do niepustej kolejki. Oznacza to na przykład, że pod warunkiem, że ten menedżer kolejek może obsługiwać wszystkie operacje umieszczania z tej aplikacji i przy założeniu, że żadna inna aplikacja nie umieszcza komunikatów w kolejce, żaden inny menedżer kolejek w grupie współużytkowania kolejek nie pomaga w opróżnianiu tej kolejki. Jeśli jednak w lokalnym menedżerze kolejek nie ma wywołania *MQGET z oczekiwaniem*, a komunikat jest umieszczany w kolejce współużytkowanej, system CF powiadomi inne menedżery kolejek w grupie współużytkowania kolejek zgodnie z regułami powiadamiania o przejściach listy.

2. MQPUT komunikatu trwałego lub komunikatu w punkcie synchronizacji.

W takim przypadku, jeśli w lokalnym menedżerze kolejek istnieje aplikacja z atrybutem *MQGET z oczekiwaniem*, komunikat jest umieszczany w kolejce współużytkowanej, a system CF powiadamia inne menedżery kolejek w grupie współużytkowania kolejek zgodnie z regułami dotyczącymi powiadomień o przejściach listy. Jednak lokalny menedżer kolejek nie czeka na powiadomienie o przejściu z systemu CF, ale najpierw honoruje wszystkie lokalne wywołania *MQGET z oczekiwaniem* i zwykle pobiera ten komunikat w imieniu aplikacji, zanim dowolny inny menedżer kolejek w grupie współużytkowania kolejek będzie mógł odpowiedzieć na powiadomienie systemu CF. Jest to zależne od zajętości lokalnego menedżera kolejek. W przeciwnym razie każdy menedżer kolejek powiadamiany przez system CF z powodu pojawienia się komunikatu w pustej kolejce będzie próbował obsłużyć pobieranie jako pierwszy. Pierwszy menedżer kolejek, który odpowiada, przetwarza nowy komunikat.

3. Na koniec, jeśli kolejka nie jest zapełniona komunikatami, a system CF wysłał powiadomienie o zmianie stanu kolejki z pustego na niepusty, wszystkie połączone menedżery kolejek będą miały możliwość asysty podczas przetwarzania kolejki. W tym przypadku obciążenie jest określane jako *oparte na pobierze*.

Ten projekt pozwala na zwiększenie wydajności w porównaniu z czysto ciągiem opartym na rozkładzie obciążenia. Celem jest wykorzystanie wysokiej dostępności oferowanej przez kolejki utrzymywane w systemie CF przy jednoczesnym umożliwieniu menedżerowi kolejek, o ile to możliwe, wykonania operacji MQGET bez konieczności odwoływania się do systemu CF i w ten sposób jak najbardziej wydajnego przetwarzania obciążenia związanego z komunikatami.

Można zastosować alternatywne podejście, w którym nacisk na zrównoważenie obciążenia jest ważniejszy niż poprzednio opisane zwiększenie wydajności. Na przykład należy upewnić się, że żadna z aplikacji pobierających nie jest połączona z tym samym menedżerem kolejek, z którym aplikacja umieszczająca jest połączona. Przy użyciu tego projektu wszystkie komunikaty są umieszczane w kolejce, a wszystkie menedżery kolejek w grupie QSG są powiadamiane o przeniesieniu kolejki z pustego na niepusty, zgodnie z algorytmem CF do obsługi takich przejść. Ponadto mechanizm *fast put to waiting getter* nie ma zastosowania.

z/OS Gdzie można znaleźć więcej informacji o współużytkowanych kolejkach i grupach współużytkowania kolejek

Tabela w tym temacie zawiera więcej informacji na temat sposobu, w jaki produkt IBM MQ for z/OS używa współużytkowanych kolejek i grup współużytkowania kolejek.

<i>Tabela 19. Gdzie można znaleźć więcej informacji o współużytkowanych kolejkach i grupach współużytkowania kolejek</i>	
Temat	Gdzie szukać
Odtwarzanie grupy współużytkowania kolejek	“Odtwarzanie i restartowanie w systemie z/OS” na stronie 262
Zabezpieczenia grupy współużytkowania kolejek	“Pojęcia związane z bezpieczeństwem w produkcie IBM MQ for z/OS” na stronie 280
Definicje obiektów prywatnych i globalnych Kierowanie komend do innej kolejki menedżery	Źródła, z których można wydawać komendy w systemie z/OS
Planowanie narzędzia CF programowo	Definiowanie zasobów narzędzia CF
Planowanie środowiska SMDS	Planowanie środowiska współużytkowanego zestawu danych komunikatów (SMDS)
Planowanie Środowisko Db2	Planowanie środowiska Db2
Konfigurowanie kolejek współużytkowanych Parametry systemu	“Kolejki współużytkowane i grupy współużytkowania kolejek” na stronie 173
Programy narzędziowe Migrowanie kolejek	Programy narzędziowe produktu IBM MQ w serwisie z/OS -informacje dodatkowe
Komunikaty konsoli	Komunikaty produktu IBM MQ for z/OS
Komendy MQSC	Komendy MQSC
IBM MQKlastry	Konfigurowanie klastra menedżera kolejek
IBM MQ kolejkowanie rozproszone Nazwy kanałów	Wprowadzenie do rozproszonego zarządzania kolejkami
Pisanie aplikacji	Przegląd projektowania aplikacji

Tabela 19. Gdzie można znaleźć więcej informacji o współużytkowanych kolejkach i grupach współużytkowania kolejek (kontynuacja)

Temat	Gdzie szukać
Wywołanie MQCONN	MQCONN (usługa MQCONN)

z/OS Kolejowanie wewnątrz grupy

W tej sekcji opisano kolejowanie wewnątrz grupy, funkcję IBM MQ for z/OS unikalną dla platformy z/OS. Ta funkcja jest dostępna tylko dla menedżerów kolejek zdefiniowanych dla grupy współużytkowania kolejek.

Informacje na temat grup współużytkowania kolejek zawiera sekcja [“Kolejki współużytkowane i grupy współużytkowania kolejek”](#) na stronie 173.

Pojęcia związane z kolejowaniem wewnątrz grupy

Istnieje możliwość szybkiego przesyłania komunikatów między menedżerami kolejek w grupie współużytkowania kolejek bez definiowania kanałów. Używana jest kolejka systemowa o nazwie SYSTEM.QSG.TRANSMIT.QUEUE, która jest współużytkowaną kolejką transmisji. Każdy menedżer kolejek w grupie współużytkowania kolejek uruchamia zadanie nazywane wewnątrzgrupowym agentem kolejowania, które oczekuje na przybycie do tej kolejki komunikatów przeznaczonych dla menedżera kolejek. Po wykryciu takiego komunikatu jest on usuwany z kolejki i umieszczany w poprawnej kolejce docelowej.

Używane są standardowe reguły tłumaczenia nazw, ale jeśli kolejowanie wewnątrz grupy (IGQ) jest włączone, a docelowy menedżer kolejek znajduje się w grupie współużytkowania kolejek, system SYSTEM.QSG.TRANSMIT.QUEUE jest używana do przesyłania komunikatu do poprawnego docelowego menedżera kolejek zamiast do przesyłania kolejki transmisji i kanału.

Kolejowanie wewnątrz grupy można włączyć za pomocą atrybutu menedżera kolejek. Kolejowanie wewnątrz grupy przenosi komunikaty nietrwale poza punkt synchronizacji, a komunikaty trwałe w punkcie synchronizacji. Jeśli wystąpi problem z dostarczeniem komunikatów do kolejki docelowej, kolejowanie wewnątrz grupy próbuje umieścić je w kolejce niedostarczonych komunikatów. Jeśli kolejka niedostarczonych komunikatów jest pełna lub niezdefiniowana, komunikaty nietrwale są usuwane, ale komunikaty trwałe są wycofywane i zwracane do systemu SYSTEM.QSG.TRANSMIT.QUEUE, a agent IGQ będzie próbował dostarczyć komunikaty, dopóki nie zakończy się pomyślnie.

Przechodzący kanał współużytkowany, który odbiera komunikat przeznaczony dla kolejki w innym menedżerze kolejek w grupie współużytkowania kolejek, może używać kolejowania wewnątrz grupy do *przeskoku* komunikatu do poprawnego miejsca docelowego.

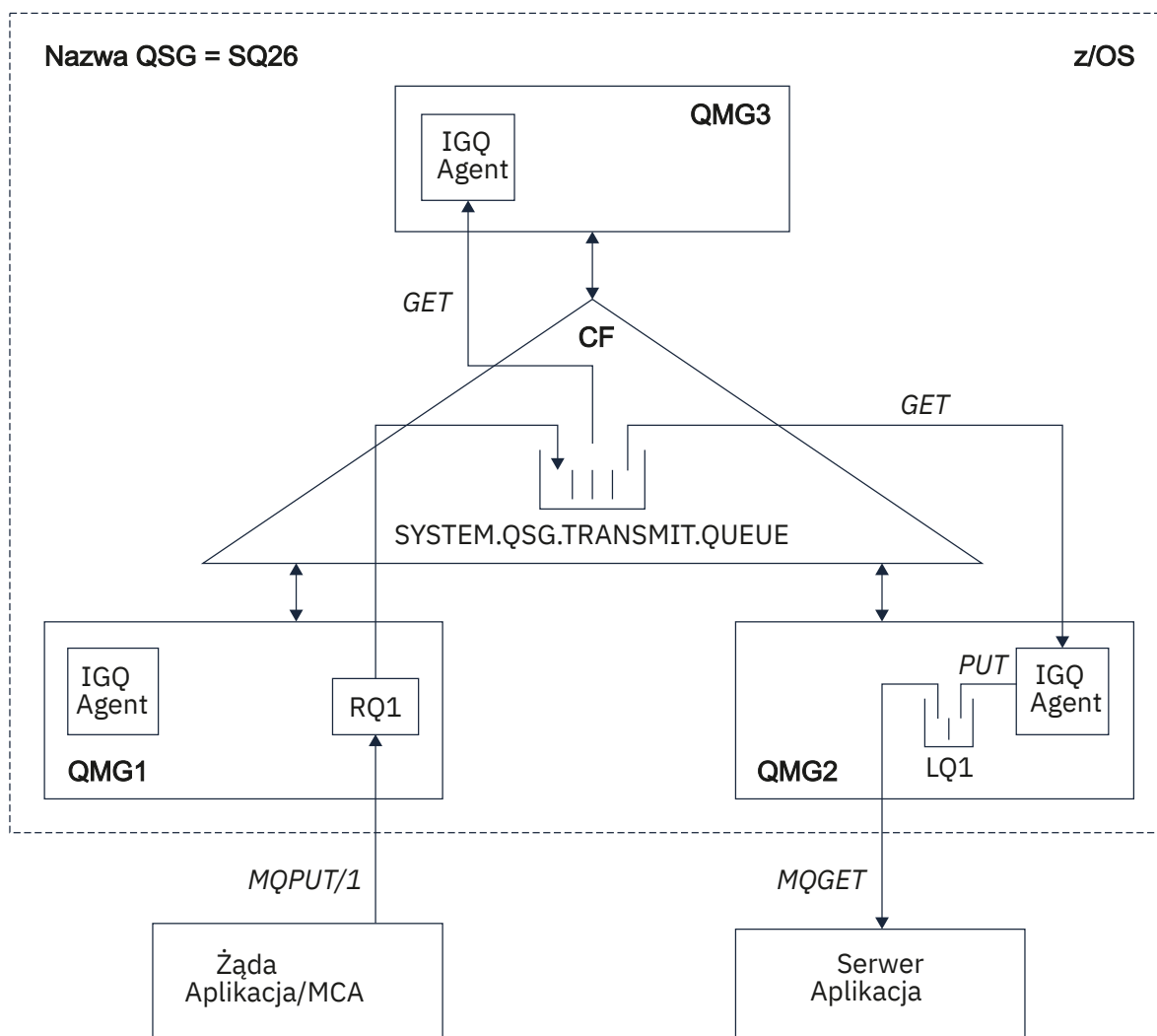
Mogą wystąpić sytuacje, w których lokalny menedżer kolejek ma umieścić komunikat bezpośrednio w kolejce docelowej, jeśli kolejka docelowa jest kolejką współużytkowaną, a nie komunikat przesyłany jako pierwszy do docelowego menedżera kolejek. Do sterowania tym elementem można użyć atrybutu SQQMNAME menedżera kolejek. Jeśli parametr SQQMNAME zostanie ustawiony na wartość USE, komenda MQOPEN zostanie wykonana w menedżerze kolejek określonym przez parametr **ObjectQMgrName**.

Jeśli jednak kolejka docelowa jest kolejką współużytkowaną i ustawiono wartość SQQMNAME na IGNORE, a **ObjectQMgrName** jest kolejką innego menedżera kolejek w grupie współużytkowania kolejek, kolejka współużytkowana jest otwierana w lokalnym menedżerze kolejek. Jeśli menedżer kolejek lokalnych nie może otworzyć kolejki docelowej lub umieścić komunikatu w kolejce, komunikat jest przesyłany do określonej kolejki **ObjectQMgrName** za pośrednictwem kanału IGQ lub kanału IBM MQ.

Kolejowanie wewnątrz grupy może być używane do bardziej wydajnego dostarczania małych komunikatów do kolejek rezydujących w zdalnych menedżerach kolejek w grupie współużytkowania kolejek. Kolejowanie wewnątrz grupy obsługuje również duże komunikaty, przy czym największe z nich to 100 MB *minus* długość nagłówka kolejki transmisji.

Uwaga: Jeśli ta opcja jest używana, użytkownicy muszą mieć takie same prawa dostępu do kolejek w każdym menedżerze kolejek w grupie współużytkowania kolejek.

Poniższy diagram przedstawia typowy przykład kolejkowania wewnątrz grupy.



Rysunek 67. Przykład kolejkowania wewnątrz grupy

Diagram przedstawia:

- Agenty IGQ działające w trzech menedżerach kolejek (QMG1, QMG2i QMG3), które są zdefiniowane w grupie współużytkowania kolejek o nazwie SQ26.
- Współużytkowana kolejka transmisji SYSTEM.QSG.TRANSMIT.QUEUE zdefiniowana w narzędziu CF.
- Definicja kolejki zdalnej zdefiniowana w menedżerze kolejek QMG1.
- Kolejka lokalna zdefiniowana w menedżerze kolejek QMG2.
- Aplikacja żądająca (może to być agent kanatu komunikatów (MCA)), która jest połączona z menedżerem kolejek QMG1.
- Aplikacja serwera połączona z menedżerem kolejek QMG2.
- Komunikat żądania jest umieszczany w systemie SYSTEM.QSG.TRANSMIT.QUEUE.

Kolejkowanie wewnątrz grupy i agent kolejkowania wewnątrz grupy

Agent IGQ jest uruchamiany podczas inicjowania menedżera kolejek. Gdy aplikacje otwierają i umieszczają komunikaty w kolejkach zdalnych, menedżer kolejek lokalnych określa, czy do przesyłania komunikatów jest używane kolejkowanie wewnątrz grupy. Jeśli ma być używane kolejkowanie wewnątrz grupy, menedżer kolejek lokalnych umieszcza komunikat w systemie SYSTEM.QSG.TRANSMIT.QUEUE. Agent IGQ w docelowym zdalnym menedżerze kolejek pobiera komunikat i umieszcza go w kolejce docelowej.

Terminologia kolejkowania wewnątrz grupy

Wyjaśnienie terminologii: kolejkowanie wewnątrz grupy, współużytkowana kolejka transmisji do użycia przez kolejkowanie wewnątrz grupy i wewnątrzgrupowy agent kolejkowania.

Kolejkowanie wewnątrz grupy

Kolejkowanie wewnątrz grupy może mieć wpływ na potencjalnie szybkie i mniej kosztowne przesyłanie komunikatów między menedżerami kolejek w grupie współużytkowania kolejek, bez konieczności definiowania kanałów.

Współużytkowana kolejka transmisji do użycia przez kolejkowanie wewnątrz grupy

Każda grupa współużytkowania kolejek ma współużytkowaną kolejkę transmisji o nazwie SYSTEM.QSG.TRANSMIT.QUEUE do użycia przez kolejkowanie wewnątrz grupy. Jeśli kolejkowanie wewnątrz grupy jest włączone, SYSTEM.QSG.TRANSMIT.QUEUE pojawia się w ścieżce tłumaczenia nazw podczas otwierania kolejek zdalnych. Gdy aplikacje (w tym agenty kanału komunikatów (Message Channel Agents-MCA)) umieszczają komunikaty w kolejce zdalnej, menedżer kolejek lokalnych określa kwalifikację komunikatów do szybkiego przesyłania i umieszcza je w systemie SYSTEM.QSG.TRANSMIT.QUEUE.

Agent kolejkowania wewnątrz grupy

Agent IGQ to zadanie uruchamiane podczas inicjowania menedżera kolejek, które oczekuje na nadejście odpowiednich komunikatów do systemu SYSTEM.QSG.TRANSMIT.QUEUE. Agent IGQ pobiera odpowiednie komunikaty z tej kolejki i dostarcza je do kolejek docelowych.

Agent IGQ dla każdego menedżera kolejek jest zawsze uruchamiany, ponieważ kolejkowanie wewnątrz grupy jest używane przez sam menedżer kolejek na potrzeby jego wewnętrznego przetwarzania.

Korzyści z kolejkowania wewnątrz grupy

Zalety kolejkowania wewnątrz grupy to: skrócenie definicji systemu, ograniczenie administrowania systemem, zwiększenie wydajności, obsługa migracji i dostarczanie komunikatów podczas wielokrotnego przeskakiwania między menedżerami kolejek w grupie współużytkowania kolejek.

Korzyści wynikające z kolejkowania wewnątrz grupy są następujące:

Zredukowane definicje systemu

Kolejkowanie wewnątrz grupy eliminuje konieczność definiowania kanałów między menedżerami kolejek w grupie współużytkowania kolejek.

Zredukowane administrowanie systemem

Ponieważ między menedżerami kolejek w grupie współużytkowania kolejek nie ma zdefiniowanych żadnych kanałów, administrowanie kanałami nie jest wymagane.

Większa wydajność

Ponieważ do dostarczenia komunikatu do kolejki docelowej potrzebny jest tylko jeden agent IGQ (zamiast dwóch pośrednich agentów wysyłających i odbierających), dostarczanie komunikatów przy użyciu kolejkowania wewnątrz grupy może być mniej kosztowne niż dostarczanie komunikatów przy użyciu kanałów. W kolejkowaniu wewnątrz grupy istnieje tylko komponent odbierający, ponieważ usunięto potrzebę użycia komponentu wysyłającego. Jest to spowodowane tym, że komunikat jest dostępny dla agenta IGQ w docelowym menedżerze kolejek w celu dostarczenia do kolejki docelowej

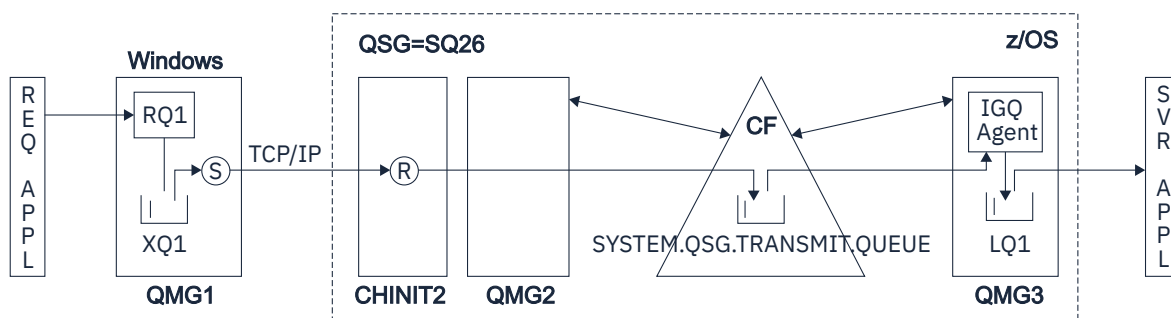
po zakończeniu operacji umieszczania w lokalnym menedżerze kolejek i zatwierdzeniu (w przypadku komunikatów umieszczonych w zasięgu punktu synchronizacji).

Obsługa migracji

Aplikacje zewnętrzne względem grupy współużytkowania kolejek mogą dostarczać komunikaty do kolejki rezydującej w dowolnym menedżerze kolejek w grupie współużytkowania kolejek, podczas gdy są połączone tylko z określonym menedżerem kolejek w grupie współużytkowania kolejek. Wynika to z faktu, że komunikaty przychodzące do kanału odbiorczego, przeznaczone do kolejki w zdalnym menedżerze kolejek, mogą być w sposób przezroczysty wysyłane do kolejki docelowej za pomocą kolejkowania wewnątrz grupy. Ta funkcja umożliwia wdrażanie aplikacji w grupie współużytkowania kolejek bez konieczności zmiany systemów zewnętrznych w stosunku do grupy współużytkowania kolejek.

Typową konfigurację ilustruje poniższy diagram, w którym:

- Aplikacja żądająca połączona z menedżerem kolejek QMG1 musi wysłać komunikat do kolejki lokalnej w menedżerze kolejek QMG3.
- Menedżer kolejek QMG1 jest połączony tylko z menedżerem kolejek QMG2.
- Menedżery kolejek QMG2 i QMG3, które były wcześniej połączone za pomocą kanałów, są teraz elementami grupy współużytkowania kolejek SQ26.



Rysunek 68. Przykład obsługi migracji

Przebieg operacji jest następujący:

1. Aplikacja żądająca umieszcza komunikat przeznaczony dla kolejki lokalnej LQ1 w zdalnym menedżerze kolejek QMG3 w definicji kolejki zdalnej RQ1.
2. Menedżer kolejek QMG1 działający na stacji roboczej z systemem Windows NT umieszcza komunikat w kolejce transmisji XQ1.
3. Wysyłający agent MCA (S) w QMG1 przesyła komunikat przy użyciu protokołu TCP/IP do odbiornika MCA (R) w inicjatorze kanału CHINIT2.
4. Odbierający agent MCA (R) w inicjatorze kanału CHINIT2 umieszcza komunikat we współużytkowanej kolejce transmisji SYSTEM.QSG.TRANSMIT.QUEUE.
5. Agent IGQ w menedżerze kolejek QMG3 pobiera komunikat z systemu SYSTEM.QSG.TRANSMIT.QUEUE i umieszcza ją w docelowej kolejce lokalnej LQ1.
6. Aplikacja serwera pobiera komunikat z docelowej kolejki lokalnej i przetwarza go.

Dostarczanie komunikatów w przypadku wielokrotnego przeskakiwania między menedżerami kolejek w grupie współużytkowania kolejek

Poprzedni diagram w sekcji Obsługa migracji ilustruje również dostarczanie komunikatów podczas wielokrotnego przeskakiwania między menedżerami kolejek w grupie współużytkowania kolejek. Komunikaty przychodzące do menedżera kolejek w grupie współużytkowania kolejek, ale przeznaczone do kolejki w innym menedżerze kolejek w grupie współużytkowania kolejek, mogą być łatwo przesyłane do kolejki docelowej w docelowym menedżerze kolejek przy użyciu kolejkowania wewnątrz grupy.

Ograniczenia dotyczące kolejkowania wewnątrz grupy

Ograniczenia dotyczące kolejkowania wewnątrz grupy to: komunikaty zakwalifikowane do przesyłania przy użyciu kolejkowania wewnątrz grupy, liczba agentów kolejkowania wewnątrz grupy dla każdego menedżera kolejek oraz uruchamianie i zatrzymywanie agenta kolejkowania wewnątrz grupy.

W tej sekcji opisano ograniczenia dotyczące kolejkowania wewnątrz grupy.

Komunikaty zakwalifikowane do przesyłania przy użyciu kolejkowania wewnątrz grupy

Ponieważ kolejkowanie wewnątrz grupy używa współużytkowanej kolejki transmisji zdefiniowanej w narzędziu CF, kolejkowanie wewnątrz grupy jest ograniczone do dostarczania komunikatów o maksymalnej obsługiwanej długości komunikatu dla kolejek współużytkowanych pomniejszonej o długość nagłówka kolejki transmisji (MQXQH).

Liczba agentów kolejkowania wewnątrz grupy dla każdego menedżera kolejek

Dla każdego menedżera kolejek w grupie współużytkowania kolejek uruchamiany jest tylko jeden agent IGQ.

Uruchamianie i zatrzymywanie agenta kolejkowania wewnątrz grupy

Agent IGQ jest uruchamiany podczas inicjowania menedżera kolejek i przerywany podczas zamykania menedżera kolejek. Jest to zadanie długotrwałe, samoistnie odzyskane (w przypadku nieprawidłowego zakończenia). W przypadku wystąpienia błędu w definicji SYSTEM.QSG.TRANSMIT.QUEUE (na przykład, jeśli ta kolejka jest zablokowana dla pobierania), agent IGQ będzie ponawiał próby. Jeśli agent IGQ napotka błąd, który spowoduje normalne zakończenie działania agenta, gdy menedżer kolejek jest nadal aktywny, można go zrestartować, wprowadzając komendę ALTER QMGR IGQ (ENABLED). Ta komenda pozwala uniknąć konieczności ponownego uruchamiania menedżera kolejek.

Ustawianie atrybutu IGQ menedżera kolejek na wartość ENABLED lub DISABLED

Jeśli atrybut IGQ menedżera kolejek ma wartość ENABLED lub DISABLED, istniejące uchwyty obiektów mogą zostać unieważnione z kodem przyczyny MQR_OBJECT_CHANGED. Więcej informacji na ten temat zawiera sekcja [Pierwsze kroki z kolejkowaniem wewnątrz grupy](#).

Pierwsze kroki z kolejkowaniem wewnątrz grupy

Kolejkowanie wewnątrz grupy można włączać, wyłączać i używać w sposób opisany w tym temacie.

Włączanie kolejkowania wewnątrz grupy

Aby włączyć kolejkowanie wewnątrz grupy w menedżerach kolejek, należy wykonać następujące czynności:

- Zdefiniuj współużytkowaną kolejkę transmisji o nazwie SYSTEM.QSG.TRANSMIT.QUEUE. Definicję tej kolejki można znaleźć w pliku thlqual.SCSQPROC(CSQ4INSS), w przykładzie CSQINP2 dla obiektów SYSTEM dla grup współużytkowania kolejek. Ta kolejka musi być zdefiniowana z poprawnymi atrybutami, zgodnie z opisem w sekcji thlqual.SCSQPROC(CSQ4INSS), aby kolejkowanie wewnątrz grupy działało poprawnie.
- Ponieważ agent IGQ jest zawsze uruchamiany podczas inicjowania menedżera kolejek, kolejkowanie wewnątrz grupy jest zawsze dostępne dla przetwarzania komunikatów przychodzących. Agent IGQ przetwarza wszystkie komunikaty umieszczone w systemie SYSTEM.QSG.TRANSMIT.QUEUE. Aby jednak włączyć kolejkowanie wewnątrz grupy dla przetwarzania danych wychodzących, atrybut IGQ menedżera kolejek musi być ustawiony na wartość ENABLED.

Ważne: Jeśli atrybut IGQ menedżera kolejek jest ustawiony na wartość ENABLED, istniejące uchwyty obiektów mogą zostać unieważnione z kodem przyczyny MQR_OBJECT_CHANGED. Więcej informacji zawiera sekcja "Konkretne właściwości kolejkowania wewnątrz grupy" na stronie [233](#). Jak opisano w sekcji 'Programmer response' dla tego kodu przyczyny, aplikacje muszą być zakodowane w celu obsługi tej sytuacji (więcej szczegółów zawiera sekcja [2041 \(07F9\) \(RC2041\): MQR_OBJECT_CHANGED](#)).

Dodatkowo, ponieważ IGQ zostało zaprojektowane jako zadanie długotrwałe i samoobsługowe, które rozpoczyna się podczas inicjowania i kończy z zamknięciem, należy zapoznać się z sekcją

“Ograniczenia dotyczące kolejkowania wewnątrz grupy” na stronie 225 , aby uzyskać więcej informacji.

Wyłączanie kolejkowania wewnątrz grupy

Aby wyłączyć kolejkowanie wewnątrz grupy dla przesyłania komunikatów wychodzących, należy ustawić atrybut IGQ menedżera kolejek na wartość DISABLED. Jeśli kolejkowanie wewnątrz grupy jest wyłączone dla konkretnego menedżera kolejek, agent IGQ w tym menedżerze kolejek może nadal przetwarzać komunikaty przychodzące, które zostały umieszczone w systemie SYSTEM.QSG.TRANSMIT.QUEUE menedżera kolejek z włączonym kolejkowaniem wewnątrzgrupowym dla przesyłania danych wychodzących.

Ważne: Jeśli atrybut IGQ menedżera kolejek jest ustawiony na wartość ENABLED, istniejące uchwytów obiektów mogą zostać unieważnione z kodem przyczyny MQRC_OBJECT_CHANGED. Więcej informacji zawiera sekcja “Konkretne właściwości kolejkowania wewnątrz grupy” na stronie 233. Jak opisano w sekcji 'Programmer response' dla tego kodu przyczyny, aplikacje muszą być zakodowane w celu obsługi tej sytuacji (więcej szczegółów zawiera sekcja 2041 (07F9) (RC2041): MQRC_OBJECT_CHANGED).

Dodatkowo, ponieważ IGQ zostało zaprojektowane jako zadanie długotrwałe i samoobsługowe, które rozpoczyna się podczas inicjowania i kończy z zamknięciem, należy zapoznać się z sekcją “Ograniczenia dotyczące kolejkowania wewnątrz grupy” na stronie 225 , aby uzyskać więcej informacji.

Korzystanie z kolejkowania wewnątrz grupy

Po włączeniu kolejkowania wewnątrz grupy jest ono dostępne do użycia, a menedżer kolejek używa go, gdy tylko jest to możliwe. Oznacza to, że gdy aplikacja umieszcza komunikat w definicji kolejki zdalnej, w pełni kwalifikowanej kolejce zdalnej lub w kolejce klastra, menedżer kolejek określa, czy komunikat kwalifikuje się do dostarczenia przy użyciu kolejkowania wewnątrz grupy, a jeśli tak, umieszcza komunikat w systemie SYSTEM.QSG.TRANSMIT.QUEUE. Nie ma potrzeby zmiany aplikacji użytkownika ani kolejek aplikacji, ponieważ w przypadku zakwalifikowanych komunikatów menedżer kolejek używa systemu SYSTEM.QSG.TRANSMIT.QUEUE, zamiast innych kolejek transmisji.

Konfiguracje kolejkowania wewnątrz grupy

Oprócz typowej konfiguracji kolejkowania wewnątrz grupy możliwe są także inne konfiguracje.

“Pojęcia związane z kolejkowaniem wewnątrz grupy” na stronie 221 opisuje typową konfigurację.

Pojęcia pokrewne

“Kolejkowanie rozproszone z kolejkowaniem wewnątrz grupy (wiele ścieżek dostarczania)” na stronie 226
W przypadku aplikacji, które przetwarzają krótkie komunikaty, możliwe może być skonfigurowanie kolejkowania wewnątrz grupy tylko na potrzeby dostarczania komunikatów między menedżerami kolejek w grupie współużytkowania kolejek.

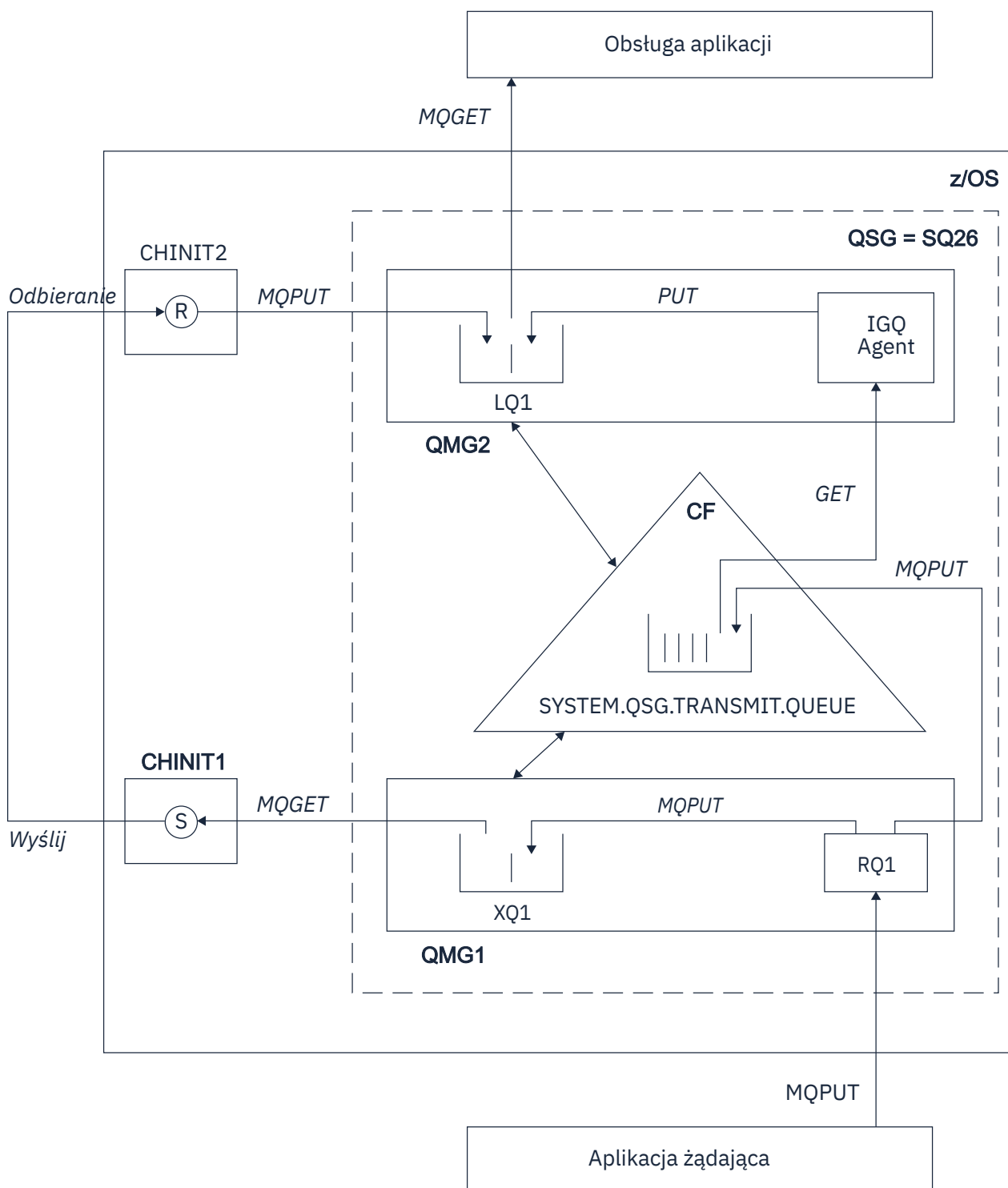
“Łączenie w klastry z kolejkowaniem wewnątrz grupy (wiele ścieżek dostarczania)” na stronie 228
Menedżery kolejek można skonfigurować w taki sposób, aby były w klastrze, a także w grupie współużytkowania kolejek.

“Grupowanie, kolejkowanie wewnątrz grupy i kolejkowanie rozproszone” na stronie 231
Istnieje możliwość skonfigurowania menedżera kolejek, który jest elementem klastra, a także grupy współużytkowania kolejek i jest połączony z rozproszonym menedżerem kolejek za pomocą pary kanału nadawcy/odbiorczego.

Kolejkowanie rozproszone z kolejkowaniem wewnątrz grupy (wiele ścieżek dostarczania)

W przypadku aplikacji, które przetwarzają krótkie komunikaty, możliwe może być skonfigurowanie kolejkowania wewnątrz grupy tylko na potrzeby dostarczania komunikatów między menedżerami kolejek w grupie współużytkowania kolejek.

Wybór kolejki wewnątrz grupy w komunikacji kanału może być kontrolowany przez poziom typu CFSTRUCT. (3 zamiast 4 lub 5). Maksymalna długość komunikatu ustawiona w systemie SYSTEM.QSG.TRANSMIT.QUEUE.



Rysunek 69. Przykładowa konfiguracja

Przetwarzanie otwierania/umieszczenia

1. Należy zauważyć, że gdy aplikacja żądająca otwiera zdalną kolejkę RQ1, tłumaczenie nazw jest wykonywane zarówno dla niewspółużytkowanej kolejki transmisji XQ1, jak i dla współużytkowanej kolejki transmisji SYSTEM.QSG.TRANSMIT.QUEUE.

2. Gdy aplikacja żądająca umieszcza komunikat w kolejce zdalnej, w zależności od tego, czy w menedżerze kolejek włączono kolejkowanie wewnątrz grupy dla przesyłania danych wychodzących i w charakterystyce komunikatu, komunikat jest umieszczany w kolejce transmisji XQ1 lub w kolejce transmisji SYSTEM.QSG.TRANSMIT.QUEUE. Menedżer kolejek umieszcza wszystkie duże komunikaty w kolejce transmisji XQ1, a wszystkie małe komunikaty w kolejce transmisji SYSTEM.QSG.TRANSMIT.QUEUE.
3. Jeśli kolejka transmisji XQ1 jest pełna lub niedostępna, żądania umieszczania dla dużych komunikatów kończą się synchronicznie z odpowiednim kodem powrotu i przyczyny. Jednak żądania umieszczania dla małych komunikatów nadal są wykonywane pomyślnie i są umieszczane w kolejce transmisji SYSTEM.QSG.TRANSMIT.QUEUE.
4. Jeśli kolejka transmisji SYSTEM.QSG.TRANSMIT.QUEUE jest pełna lub nie można jej umieścić w żądaniach umieszczania dla małych komunikatów, synchronicznie kończy się niepowodzeniem z odpowiednim kodem powrotu i przyczyny. Jednak żądania umieszczenia dla dużych komunikatów nadal są wykonywane pomyślnie i są umieszczane w kolejce transmisji XQ1. W takim przypadku nie jest podejmowana próba umieszczenia małych komunikatów w kolejce transmisji.

Przeptyw dla dużych komunikatów

1. Aplikacja żądająca umieszcza duże komunikaty w kolejce zdalnej RQ1.
2. Menedżer kolejek QMG1 umieszcza komunikaty w kolejce transmisji XQ1.
3. Wysyłający agent MCA (S) w menedżerze kolejek QMG1 pobiera komunikaty z kolejki transmisji XQ1 i wysyła je do menedżera kolejek QMG2.
4. Odbierający agent MCA (R) w menedżerze kolejek QMG2 odbiera komunikaty i umieszcza je w kolejce docelowej LQ1.
5. Aplikacja udostępniająca pobiera, a następnie przetwarza komunikaty z kolejki LQ1.

Przeptyw dla małych komunikatów

1. Aplikacja żądająca umieszcza małe komunikaty w kolejce zdalnej RQ1.
2. Menedżer kolejek QMG1 umieszcza komunikaty w kolejce transmisji SYSTEM.QSG.TRANSMIT.QUEUE.
3. IGQ w menedżerze kolejek QMG2 pobiera komunikaty i umieszcza je w kolejce docelowej LQ1.
4. Aplikacja udostępniająca pobiera komunikaty z kolejki LQ1.

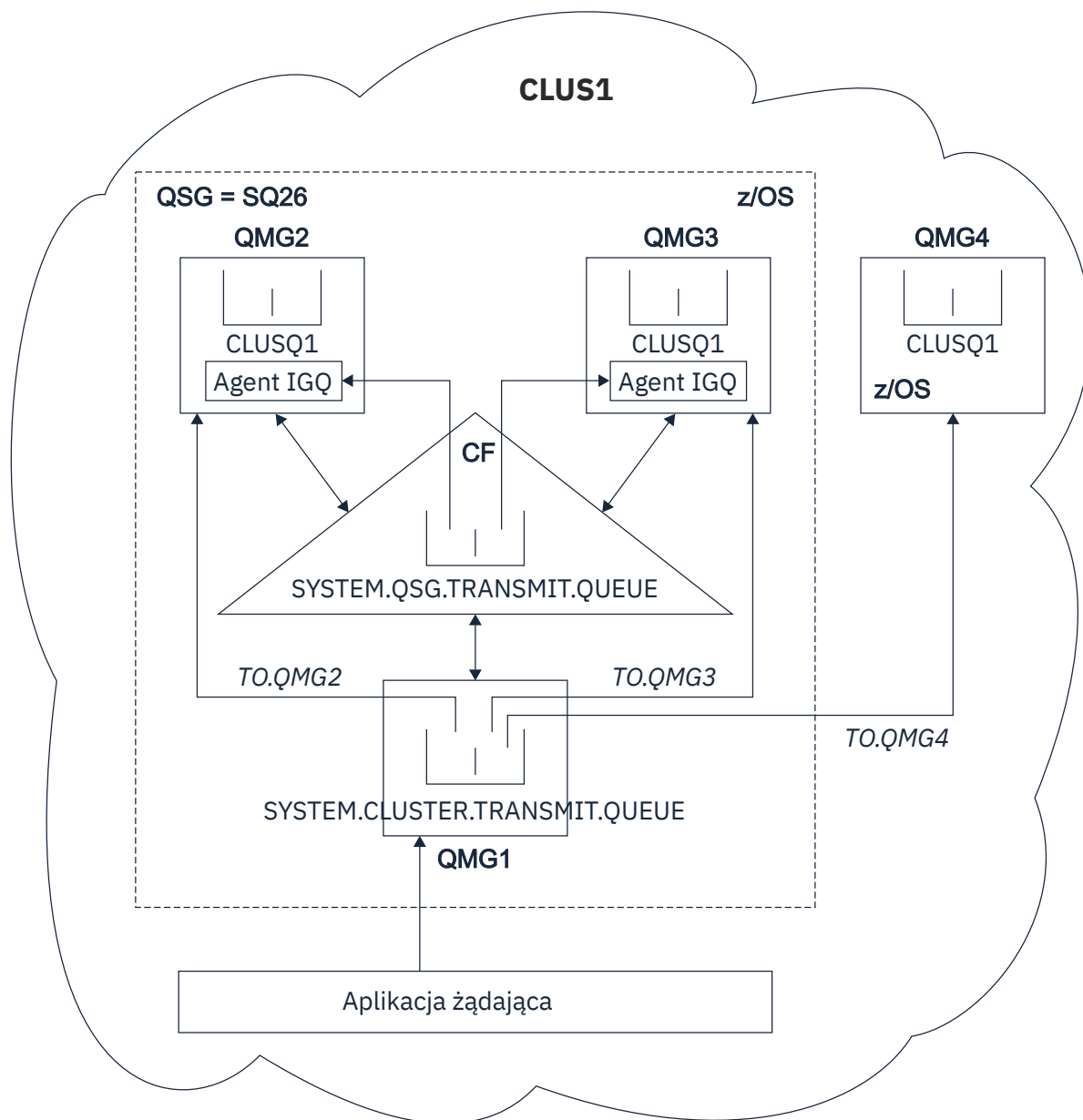
Punkty do uwagi

1. Aplikacja żądająca nie musi mieć informacji o mechanizmie bazowym używanym do dostarczania komunikatów.
2. Dla małych komunikatów można osiągnąć potencjalnie szybszy mechanizm dostarczania komunikatów.
3. Dla dostarczania komunikatów dostępnych jest wiele ścieżek (zwykła trasa kanału i wewnątrzgrupowa trasa kolejkowania).
4. Trasa kolejkowania wewnątrz grupy, potencjalnie szybsza, jest wybierana zamiast zwykłej trasy kanału. W zależności od charakterystyki komunikatu dostarczanie komunikatów może być podzielone na dwie ścieżki. Dlatego komunikaty mogą być dostarczane poza kolejnością (choć jest to możliwe również wtedy, gdy komunikaty są dostarczane tylko przy użyciu normalnej trasy kanału).
5. Po wybraniu trasy i umieszczeniu komunikatów w kolejkach transmisji, do dostarczania komunikatów używana jest tylko wybrana trasa. Wszystkie nieprzetworzone komunikaty w systemie SYSTEM.QSG.TRANSMIT.QUEUE nie jest kierowana do kolejki transmisji XQ1.

Łączenie w klastry z kolejkowaniem wewnątrz grupy (wiele ścieżek dostarczania)

Menedżery kolejek można skonfigurować w taki sposób, aby były w klastrze, a także w grupie współużytkownika kolejek.

Jeśli komunikaty są wysyłane do kolejki klastra, a lokalne i zdalne menedżery kolejek znajdują się w tej samej grupie współużytkowania kolejek, do dostarczania małych komunikatów (za pomocą systemu `SYSTEM.QSG.TRANSMIT.QUEUE`) oraz dostarczanie dużych komunikatów, jeśli kolejowanie wewnątrz grupy obsługuje wielkość komunikatu. Dotyczy to również systemu `SYSTEM.CLUSTER.TRANSMIT.QUEUE` służy do dostarczania komunikatów do dowolnego menedżera kolejek w klastrze, ale poza grupą współużytkowania kolejek. Poniższy diagram ilustruje tę konfigurację (inicjatory kanałów nie są wyświetlane).



Rysunek 70. Przykład łączenia w klastry z kolejowaniem wewnątrz grupy

Diagram przedstawia:

- Cztery z/OS menedżery kolejek QMG1, QMG2, QMG3 i QMG4 skonfigurowane w klastrze CLUS1.
- Menedżery kolejek QMG1, QMG2 i QMG3 skonfigurowane w grupie współużytkowania kolejek SQ26.
- Agenty IGQ działające w menedżerach kolejek QMG2 i QMG3.
- Lokalny system `SYSTEM.CLUSTER.TRANSMIT.QUEUE` zdefiniowany w QMG1.

Uwaga: Dla jasności, SYSTEM.CLUSTER.TRANSMIT.QUEUE w innych menedżerach kolejek nie jest wyświetlana.

- Współużytkowany system SYSTEM.QSG.TRANSMIT.QUEUE zdefiniowana w systemie CF, która jest w strukturze IBM MQ skonfigurowanej z atrybutem CFLEVEL (3) RECOVER (YES).
- Kanały klastra TO.QMG2 (łączenie QMG1 z QMG2), TO.QMG3 (połączenie QMG1 z QMG3) i TO.QMG4 (połączenie QMG1 z QMG4).
- Kolejka klastra CLUSQ1 udostępniana przez menedżery kolejek QMG2, QMG3 i QMG4.

Założmy, że aplikacja żądająca otwiera kolejkę klastra z opcją MQOO_BIND_NOT_FIXED, tak że docelowy menedżer kolejek dla kolejki klastra jest wybierany w czasie umieszczania.

Jeśli wybranym docelowym menedżerem kolejek jest QMG2:

- Wszystkie duże komunikaty umieszczane przez aplikację żądającą są następujące:
 - Umieść w systemie SYSTEM.CLUSTER.TRANSMIT.QUEUE w QMG1, przyczyna: SYSTEM.QSG.TRANSMIT.QUEUE ma strukturę CFLEVEL (3), dlatego obsługuje komunikaty o wielkości do 63 kB.
 - Przesłano do kolejki klastra CLUSQ1 w QMG2 za pomocą kanału klastra TO.QMG2
- Wszystkie małe komunikaty umieszczane przez aplikację żądającą są
 - Umieść we współużytkowanej kolejce transmisji SYSTEM.QSG.TRANSMIT.QUEUE. Ta kolejka ma strukturę skonfigurowaną z atrybutem RECOVER (YES), dlatego jest używana zarówno dla trwałych, jak i nietrwałych małych komunikatów.
 - Wczytane przez agenta IGQ w QMG2
 - Umieść w kolejce klastra CLUSQ1 w QMG2

Jeśli wybranym docelowym menedżerem kolejek jest QMG4:

- Ponieważ menedżer kolejek QMG4 nie jest elementem grupy współużytkowania kolejek SQ26, wszystkie komunikaty umieszczane przez aplikację żądającą są
 - Umieść w systemie SYSTEM.CLUSTER.TRANSMIT.QUEUE w QMG1
 - Przesłano do kolejki klastra CLUSQ1 w QMG4 za pomocą kanału klastra TO.QMG4

Punkty do uwagi

- Aplikacja żądająca nie musi mieć informacji o mechanizmie bazowym używanym do dostarczania komunikatów.
- Potencjalnie szybszy mechanizm dostarczania jest osiąganym w przypadku przesyłania małych nietrwałych komunikatów między menedżerami kolejek w grupie współużytkowania kolejek (nawet jeśli te same menedżery kolejek znajdują się w klastrze).
- Na potrzeby dostarczania komunikatów dostępnych jest wiele ścieżek (zarówno trasa klastra, jak i trasa kolejkowania wewnątrz grupy).
- Potencjalnie szybsza trasa kolejkowania wewnątrz grupy jest wybierana zamiast trasy klastra. W zależności od charakterystyki komunikatu dostarczanie komunikatów może być podzielone na dwie ścieżki. Dlatego komunikaty mogą być dostarczane poza kolejnością. Należy pamiętać, że to dostarczenie jest możliwe bez względu na opcję MQOO_BIND_* określoną przez aplikację. Kolejkowanie wewnątrz grupy dystrybuuje komunikaty w taki sam sposób, jak grupowanie w klastrze, w zależności od tego, czy w przypadku operacji otwarcia określono opcję MQOO_BIND_NOT_FIXED, MQOO_BIND_ON_OPEN, MQOO_BIND_ON_GROUP lub MQOO_BIND_AS_Q_DEF.
- Po wybraniu trasy i umieszczeniu komunikatów w kolejkach transmisji, do dostarczania komunikatów używana jest tylko wybrana trasa. Wszystkie nieprzetworzone komunikaty w systemie SYSTEM.QSG.TRANSMIT.QUEUE nie są kierowane do systemu SYSTEM.CLUSTER.TRANSMIT.QUEUE.

Grupowanie, kolejkowanie wewnątrz grupy i kolejkowanie rozproszone

Istnieje możliwość skonfigurowania menedżera kolejek, który jest elementem klastra, a także grupy współużytkowania kolejek i jest połączony z rozproszonym menedżerem kolejek za pomocą pary kanału nadawcy/odbiorczego.

Ta konfiguracja jest kombinacją rozproszonego kolejkowania z kolejkowaniem wewnątrz grupy i grupowania z kolejkowaniem wewnątrz grupy.

Kolejkowanie wewnątrz grupy jest opisane w sekcji [“Kolejkowanie rozproszone z kolejkowaniem wewnątrz grupy \(wiele ścieżek dostarczania\)”](#) na stronie 226.

Łączenie w klastry z kolejkowaniem wewnątrz grupy jest opisane w sekcji [“Łączenie w klastry z kolejkowaniem wewnątrz grupy \(wiele ścieżek dostarczania\)”](#) na stronie 228.

Komunikaty kolejkowania wewnątrz grupy

W tej sekcji opisano komunikaty umieszczane w systemie SYSTEM.QSG.TRANSMIT.QUEUE.

Struktura komunikatu

Podobnie jak wszystkie inne komunikaty, które są umieszczane w kolejkach transmisji, komunikaty, które są umieszczane w systemie SYSTEM.QSG.TRANSMIT.QUEUE jest poprzedzona nagłówkiem kolejki transmisji (MQXQH).

Trwałość komunikatu

W systemach IBM WebSphere MQ 5.3 i nowszych kolejki współużytkowane obsługują zarówno komunikaty trwałe, jak i nietrwałe.

Jeśli menedżer kolejek kończy działanie, gdy agent IGQ przetwarza komunikaty nietrwałe, lub jeśli agent IGQ kończy działanie nieprawidłowo w trakcie przetwarzania komunikatów, wówczas przetwarzane komunikaty nietrwałe mogą zostać utracone. Aplikacje muszą uzgodnić sposób odtwarzania nietrwałych komunikatów, jeśli ich odtwarzanie jest wymagane.

Jeśli żądanie umieszczenia komunikatu nietrwałego wysłane przez agenta IGQ nieoczekiwanie nie powiedzie się, przetwarzany komunikat zostanie utracony.

Dostarczanie komunikatów

Agent IGQ pobiera i dostarcza wszystkie nietrwałe komunikaty poza zasięgiem punktu synchronizacji oraz wszystkie trwałe komunikaty w zasięgu punktu synchronizacji. W tym przypadku agent IGQ działa jako koordynator punktu synchronizacji. Dlatego agent IGQ przetwarza nietrwałe komunikaty, takie jak szybkie i nietrwałe komunikaty w kanale komunikatów. Patrz [Fast, nonpersistent messages \(szybkie, nietrwałe komunikaty\)](#).

Tworzenie partii wiadomości

Agent IGQ używa stałej wielkości partii 50 komunikatów. Wszystkie komunikaty trwałe pobrane w ramach zadania wsadowego są zatwierdzane w odstępach czasu 50 komunikatów. Agent zatwierdza zadanie wsadowe składające się z trwałych komunikatów, gdy w systemie SYSTEM.QSG.TRANSMIT.QUEUE.

Wielkość komunikatu

Maksymalna wielkość komunikatu, który można umieścić w systemie SYSTEM.QSG.TRANSMIT.QUEUE to maksymalna obsługiwana długość komunikatu dla kolejek współużytkowanych pomniejszona o długość nagłówka kolejki transmisji (MQXQH).

Domyślna trwałość komunikatu i domyślny priorytet komunikatu

W przypadku systemu SYSTEM.QSG.TRANSMIT.QUEUE znajduje się w ścieżce rozstrzygnięcia nazw kolejek ustanowionej w czasie otwarcia, a w przypadku komunikatów, które są umieszczane z domyślną trwałością i domyślnym priorytetem (lub z domyślną trwałością lub domyślnym priorytetem), reguły normalne są stosowane w wyborze kolejki, która ma domyślny priorytet i używane wartości trwałości. (Więcej informacji na temat reguł wyboru kolejki zawiera sekcja [Komunikaty produktu IBM MQ](#)).

Pojęcia pokrewne

[“Niedostarczone/nieprzetworzone komunikaty”](#) na stronie 232

W tej sekcji opisano, co dzieje się z niedostarczonymi i nieprzetworzonymi komunikatami w systemie SYSTEM.QSG.TRANSMIT.QUEUE.

[“Komunikaty raportów-kolejkowanie wewnątrz grupy” na stronie 232](#)

W tej sekcji opisano komunikaty raportu: potwierdzenie przybycia, potwierdzenie dostarczenia, raport o utracie ważności i raport o wyjątkach.

Niedostarczone/nieprzetworzone komunikaty

W tej sekcji opisano, co dzieje się z niedostarczonymi i nieprzetworzonymi komunikatami w systemie SYSTEM.QSG.TRANSMIT.QUEUE.

Jeśli agent IGQ nie może dostarczyć komunikatu do kolejki docelowej, agent IGQ:

- Uwzględnia opcję raportu MQRO_DISCARD_MSG (jeśli pole opcji raportu deskryptora MQMD dla niedostarczanego komunikatu wskazuje, że musi) i usuwa niedostarczony komunikat.
- Podejmuje próbę umieszczenia niedostarczonych komunikatów w kolejce niedostarczonych komunikatów dla docelowego menedżera kolejek, jeśli komunikat nie został jeszcze odrzucony. Agent IGQ poprzedza komunikat nagłówkiem kolejki niedostarczonych komunikatów (MQDLH).

Jeśli kolejka niedostarczonych komunikatów nie jest zdefiniowana lub jeśli niedostarczony komunikat nie może zostać umieszczony w kolejce niedostarczonych komunikatów i jeśli niedostarczony komunikat jest następujący:

- , agent IGQ wycofuje bieżącą partię trwałych komunikatów, które przetwarza, i przechodzi w stan ponawiania. Więcej informacji na ten temat zawiera [“Konkretne właściwości kolejkowania wewnątrz grupy” na stronie 233](#).
- nietrwały, agent IGQ usuwa komunikat i kontynuuje przetwarzanie następnego komunikatu.

Jeśli menedżer kolejek w grupie współużytkowania kolejek zostanie zakończony, zanim powiązany z nim agent IGQ będzie miał czas na przetworzenie wszystkich komunikatów, nieprzetworzone komunikaty pozostaną w systemie SYSTEM.QSG.TRANSMIT.QUEUE do momentu następnego uruchomienia menedżera kolejek. Następnie agent IGQ pobiera i dostarcza komunikaty do kolejek docelowych.

Jeśli działanie narzędzia CF zakończy się niepowodzeniem przed wszystkimi komunikatami w systemie SYSTEM.QSG.TRANSMIT.QUEUE została przetworzona, wszystkie nieprzetworzone nietrwałe komunikaty zostały utracone.

Firma IBM zaleca, aby aplikacje nie umieszczać komunikatów bezpośrednio w kolejkach transmisji. Jeśli aplikacja umieściła komunikaty bezpośrednio w systemie SYSTEM.QSG.TRANSMIT.QUEUE: agent IGQ może nie być w stanie przetworzyć tych komunikatów i pozostają one w systemie SYSTEM.QSG.TRANSMIT.QUEUE. Użytkownicy muszą następnie użyć własnych metod, aby poradzić sobie z nieprzetworzonymi komunikatami.

Komunikaty raportów-kolejkowanie wewnątrz grupy

W tej sekcji opisano komunikaty raportu: potwierdzenie przybycia, potwierdzenie dostarczenia, raport o utracie ważności i raport o wyjątkach.

Potwierdzenie odbioru (COA) /potwierdzenie dostarczenia (COD)

Komunikaty COA i COD są generowane przez menedżer kolejek, gdy używane jest kolejkowanie wewnątrz grupy.

Komunikaty raportu o utracie ważności

Komunikaty raportu o utracie ważności są generowane przez menedżer kolejek.

Komunikaty raportów o wyjątkach

W zależności od opcji MQRO_EXCEPTION_* raportu określonej w polu *Opcje raportu* deskryptora komunikatu dla niedostarczonych komunikatów, agent IGQ generuje wymagany raport wyjątku i umieszcza go w określonej kolejce odpowiedzi. Kolejkowanie wewnątrz grupy może być używane do dostarczania raportu o wyjątkach do docelowej kolejki odpowiedzi.

Trwałość komunikatu raportu jest taka sama jak trwałość niedostarczonych komunikatów. Jeśli agent IGQ nie może rozstrzygnąć nazwy docelowej kolejki odpowiedzi lub jeśli nie może umieścić

komunikatu odpowiedzi w kolejce transmisji (w celu późniejszego przestania do docelowej kolejki odpowiedzi), próbuje umieścić raport o wyjątku w kolejce niedostarczonych komunikatów menedżera kolejek, w którym został wygenerowany komunikat raportu. Jeśli nie jest to możliwe, to jeśli niedostarczony komunikat to:

- Agent IGQ usuwa raport o wyjątkach, wycofuje bieżącą partię komunikatów i przechodzi w stan ponawiania. Więcej informacji na ten temat zawiera [“Konkretne właściwości kolejkowania wewnątrz grupy”](#) na stronie 233.
- nietrwały, agent IGQ usuwa raport o wyjątkach i kontynuuje przetwarzanie następnego komunikatu w systemie SYSTEM.QSG.TRANSMIT.QUEUE.

Bezpieczeństwo kolejkowania wewnątrz grupy

W tej sekcji opisano zasady zabezpieczeń dotyczące kolejkowania wewnątrz grupy.

Atrybuty menedżera kolejek IGQAUT (uprawnienie IGQ) i IGQUSER (ID użytkownika agenta IGQ) mogą być ustawione tak, aby kontrolować poziom sprawdzania zabezpieczeń, który jest wykonywany, gdy agent IGQ otwiera kolejki docelowe.

Upewnienie do kolejkowania wewnątrz grupy (IGQAUT)

Atrybut IGQAUT można ustawić tak, aby wskazywał typ sprawdzeń bezpieczeństwa, które mają być wykonywane, a tym samym określał identyfikatory użytkowników, które mają być używane przez agenta IGQ podczas ustanawiania uprawnień do umieszczania komunikatów w kolejce docelowej.

Atrybut IGQAUT jest analogiczny do atrybutu PUTAUT, który jest dostępny w definicjach kanałów.

Identyfikator użytkownika kolejkowania wewnątrz grupy (IGQUSER)

Atrybut IGQUSER może być użyty do wyznaczenia ID użytkownika, który będzie używany przez agenta IGQ podczas ustanawiania uprawnień do umieszczania komunikatów w kolejce docelowej.

Atrybut IGQUSER jest analogiczny do atrybutu MCAUSER, który jest dostępny w definicjach kanałów.

Konkretne właściwości kolejkowania wewnątrz grupy

W tej sekcji opisano konkretne właściwości kolejkowania wewnątrz grupy, w tym unieważnienie uchwytów obiektów, możliwość samoodtwarzania i ponawiania dla wewnątrzgrupowego agenta kolejkowania oraz wewnątrzgrupowego agenta kolejkowania i serializacji.

Unieważnianie uchwytów obiektów (MQRC_OBJECT_CHANGED)

Jeśli po otwarciu obiektu okaże się, że atrybuty obiektu uległy zmianie, menedżer kolejek unieważnia uchwyt obiektu z wartością MQRC_OBJECT_CHANGED przy jego następnym użyciu.

Kolejkowanie wewnątrz grupy wprowadza następujące reguły unieważniania uchwytu obiektu:

- W przypadku systemu SYSTEM.QSG.TRANSMIT.QUEUE została dołączona do ścieżki rozstrzygania nazw podczas przetwarzania otwarcia, ponieważ kolejkowanie wewnątrz grupy było WŁĄCZONE w czasie otwarcia, ale kolejkowanie wewnątrz grupy było WYŁĄCZONE w czasie umieszczenia, a następnie menedżer kolejek unieważnia uchwyt obiektu i kończy żądanie umieszczenia z opcją MQRC_OBJECT_CHANGED.
- W przypadku systemu SYSTEM.QSG.TRANSMIT.QUEUE nie została uwzględniona w ścieżce rozstrzygania nazw podczas przetwarzania otwarcia, ponieważ kolejkowanie wewnątrz grupy było WYŁĄCZONE w czasie otwarcia, ale kolejkowanie wewnątrz grupy było WŁĄCZONE w czasie umieszczenia, a następnie menedżer kolejek unieważnia uchwyt obiektu i kończy żądanie umieszczenia z opcją MQRC_OBJECT_CHANGED.
- W przypadku systemu SYSTEM.QSG.TRANSMIT.QUEUE została dołączona do ścieżki tłumaczenia nazw podczas przetwarzania otwarcia, ponieważ kolejkowanie wewnątrz grupy było włączone w czasie otwarcia, ale system SYSTEM.QSG.TRANSMIT.QUEUE została zmieniona do czasu umieszczenia, a następnie menedżer kolejek unieważnia uchwyt obiektu i kończy żądanie umieszczenia z wartością MQRC_OBJECT_CHANGED.

Samoistne odtwarzanie wewnątrzgrupowego agenta kolejkowania

Jeśli działanie agenta IGQ zakończy się nieprawidłowo, zostanie wyświetlony komunikat CSQM067E i agent IGQ zostanie ponownie uruchomiony.

Możliwość ponawiania agenta kolejkowania wewnątrz grupy

Jeśli agent IGQ napotka problem z dostępem do systemu SYSTEM.QSG.TRANSMIT.QUEUE (ponieważ nie jest zdefiniowana, na przykład, albo jest zdefiniowana z niepoprawnymi atrybutami, albo jest zablokowana dla operacji pobierania lub z innego powodu), agent IGQ przechodzi w stan ponawiania.

Agent IGQ obserwuje krótkie i długie liczby ponownych prób oraz odstępy czasu. Wartości dla tych liczników i interwałów, których nie można zmienić, są następujące:

Stała	Wartość
Licznik krótkookresowych ponowień	10
Interwał krótkookresowych ponowień	60 sekund = 1 minuta
Licznik długookresowych ponowień	999 999 999
Interwał długookresowych ponowień	1200 sekund = 20 minut

Wewnątrzgrupowy agent kolejkowania i szeregowanie

Próba przekształcenia do postaci szeregowej przez agenta IGQ dostępu do współużytkowanych kolejek w trakcie odtwarzania równorzędnego może się nie powieść.

Jeśli wystąpi awaria menedżera kolejek w grupie współużytkowania kolejek, podczas gdy agent IGQ zajmuje się niezatwierdzonymi komunikatami we współużytkowanej kolejce lub kolejkach, agent IGQ kończy działanie i dla menedżera kolejek, który uległ awarii, wykonywane jest odtwarzanie równorzędne kolejki współużytkowanej. Ponieważ odtwarzanie równorzędne kolejki współużytkowanej jest działaniem asynchronicznym, pozostawia możliwość zrestartowania uszkodzonego menedżera kolejek oraz agenta IGQ dla tego menedżera kolejek przed zakończeniem odtwarzania równorzędnego kolejki współużytkowanej. Co z kolei pozostawia możliwość przetwarzania wszystkich zatwierdzonych komunikatów przed i poza kolejnością, gdy komunikaty są nadal odzyskiwane. Aby upewnić się, że komunikaty nie są przetwarzane poza kolejnością, agent IGQ szereguje dostęp do współużytkowanych kolejek, wywołując funkcję API MQCONN.

Próba przekształcenia do postaci szeregowej przez agenta IGQ dostępu do współużytkowanych kolejek w trakcie odtwarzania równorzędnego może się nie powieść. Zostanie wyświetlony komunikat o błędzie i agent IGQ przejdzie w stan ponawiania. Po zakończeniu odtwarzania równorzędnego menedżera kolejek, na przykład podczas następczej próby, agent IGQ może zostać uruchomiony.

z/OS

Zarządzanie pamięcią masową w systemie z/OS

Produkt IBM MQ for z/OS wymaga trwałych i tymczasowych struktur danych oraz używa zestawów stron i buforów pamięci do przechowywania tych danych. W tych tematach podano więcej szczegółowych informacji na temat sposobu, w jaki produkt IBM MQ używa tych zestawów stron i buforów.

Pojęcia pokrewne

“Zestawy stron dla systemu IBM MQ for z/OS” na stronie 235

W tym temacie opisano, w jaki sposób produkt IBM MQ for z/OS używa zestawów stron do przechowywania komunikatów.

“Klasy pamięci masowej dla IBM MQ for z/OS” na stronie 236

Klasa pamięci masowej to pojęcie IBM MQ for z/OS, które umożliwia menedżerowi kolejek odzworowywanie kolejek na zestawy stron. Za pomocą klas pamięci masowej można kontrolować, które zestawy danych są używane przez które kolejki.

[“Bufory i pule buforów dla systemu IBM MQ for z/OS” na stronie 237](#)

Program IBM MQ for z/OS używa buforów i pul buforów do tymczasowego buforowania danych. W tym temacie opisano sposób organizowania i używania buforów.

Odsyłacze pokrewne

[“Gdzie można znaleźć więcej informacji na temat zarządzania pamięcią masową w systemie IBM MQ for z/OS” na stronie 239](#)

Ta sekcja zawiera dodatkowe informacje na temat zarządzania pamięcią masową w systemie IBM MQ for z/OS.

Zestawy stron dla systemu IBM MQ for z/OS

W tym temacie opisano, w jaki sposób produkt IBM MQ for z/OS używa zestawów stron do przechowywania komunikatów.

Zestaw stron jest liniowym zestawem danych VSAM, który został specjalnie sformatowany do użycia przez system IBM MQ. Zestawy stron są używane do przechowywania większości komunikatów i definicji obiektów.

Wyjątkami są definicje globalne przechowywane we współużytkowanym repozytorium w systemie Db2 oraz komunikaty w kolejkach współużytkowanych. Nie są one przechowywane w zestawach stron menedżera kolejek. Więcej informacji na temat kolejek współużytkowanych zawiera sekcja [“Kolejki współużytkowane i grupy współużytkowania kolejek” na stronie 173](#), a więcej informacji na temat definicji globalnych zawiera sekcja [Definicje prywatne i globalne](#).

Zestawy stron IBM MQ mogą mieć wielkość do 64 GB. Każdy zestaw stron jest identyfikowany przez identyfikator zestawu stron (PSID), będący liczbą całkowitą z zakresu od 00 do 99. Każdy menedżer kolejek musi mieć własne zestawy stron.

Produkt IBM MQ używa zerowego zestawu stron (PSID=00) do przechowywania definicji obiektów i innych ważnych informacji dotyczących menedżera kolejek. W przypadku normalnego działania produktu IBM MQ istotne jest, aby zerowy zestaw stron nie był pełny, dlatego nie należy go używać do zapisywania komunikatów.

Aby zwiększyć wydajność systemu, należy również oddzielić komunikaty o krótkim czasie życia od komunikatów o długim czasie życia, umieszczając je w różnych zestawach stron.

Należy sformatować zestawy stron, a program IBM MQ udostępnia w tym celu program narzędziowy FORMAT; patrz sekcja [Formatowanie zestawów stron \(FORMAT\)](#). Zestawy stron muszą być również zdefiniowane w podsystemie IBM MQ.

Produkt IBM MQ for z/OS można skonfigurować w taki sposób, aby dynamicznie rozwijał zestaw stron po zapelnieniu. Program IBM MQ kontynuuje rozwijanie zestawu stron, jeśli jest to wymagane, aż do chwili, gdy istnieją 123 obszary logiczne, jeśli dostępna jest wystarczająca ilość miejsca na dysku. Przydziały mogą obejmować woluminy, jeśli liniowy zestaw danych jest zdefiniowany w ten sposób, jednak program IBM MQ nie może rozszerzyć zestawów stron powyżej 64 GB.

Nie można używać zestawów stron z jednego menedżera kolejek systemu IBM MQ w innym menedżerze kolejek systemu IBM MQ ani zmieniać nazwy menedżera kolejek. Aby przestać dane z jednego menedżera kolejek do innego, należy usunąć wszystkie obiekty i komunikaty z pierwszego menedżera kolejek i przetaładować je do innego menedżera kolejek.

W menedżerze kolejek z wersją wcześniejszą niż V6 nie można używać zestawów stron większych niż 4 GB. W okresie migracji, gdy istnieje prawdopodobieństwo, że konieczne będzie powrót do poprzedniej wersji kodu:

- Nie zmieniaj zbioru stron 0 na większy niż 4 GB.
- Inne zestawy stron o wielkości większej niż 4 GB pozostaną w trybie bez połączenia podczas restartowania menedżera kolejek w poprzedniej wersji.

Więcej informacji na temat migrowania istniejących zestawów stron, które mogą rozszerzać się poza 4 GB, zawiera sekcja [Definiowanie zestawu stron o wielkości większej niż 4 GB](#).

Administrator może dynamicznie dodawać zestawy stron do działającego menedżera kolejek lub usuwać zestawy stron z działającego menedżera kolejek (z wyjątkiem zerowego zestawu stron). Komenda DEFINE PSID może zostać uruchomiona po zakończeniu restartu menedżera kolejek tylko wtedy, gdy komenda zawiera słowo kluczowe DSN.

Klasy pamięci masowej dla IBM MQ for z/OS

Klasa pamięci masowej to pojęcie IBM MQ for z/OS, które umożliwia menedżerowi kolejek odwzorowywanie kolejek na zestawy stron. Za pomocą klas pamięci masowej można kontrolować, które zestawy danych są używane przez które kolejki.

Wprowadzenie do klas pamięci masowej

Klasa pamięci masowej odwzorowuje jedną lub więcej kolejek na zestaw stron. Oznacza to, że komunikaty dla tej kolejki są przechowywane w tym zestawie stron.

Klasy pamięci masowej umożliwiają sterowanie miejscem przechowywania niewspółużytkowanych danych komunikatów na potrzeby administrowania, zarządzania pamięcią masową zestawu danych i ładowania oraz izolowania aplikacji. Można również użyć klas pamięci masowej do zdefiniowania grupy XCF i nazwy elementu regionu IMS, jeśli używany jest most IMS (patrz sekcja [“IBM MQ i IMS” na stronie 296](#)).

Kolejki współużytkowane nie używają klas pamięci do uzyskania odwzorowania zestawu stron, ponieważ komunikaty w tych kolejkach nie są przechowywane w zestawach stron.

Jak działają klasy pamięci

- Klasę pamięci masowej definiuje się za pomocą komendy DEFINE STGCLASS, określając identyfikator zestawu stron (PSID).
- Podczas definiowania kolejki należy określić klasę pamięci w atrybucie STGCLASS.

W poniższym przykładzie kolejka lokalna QE5 jest odwzorowywana na zestaw stron 21 za pomocą klasy pamięci ARC2.

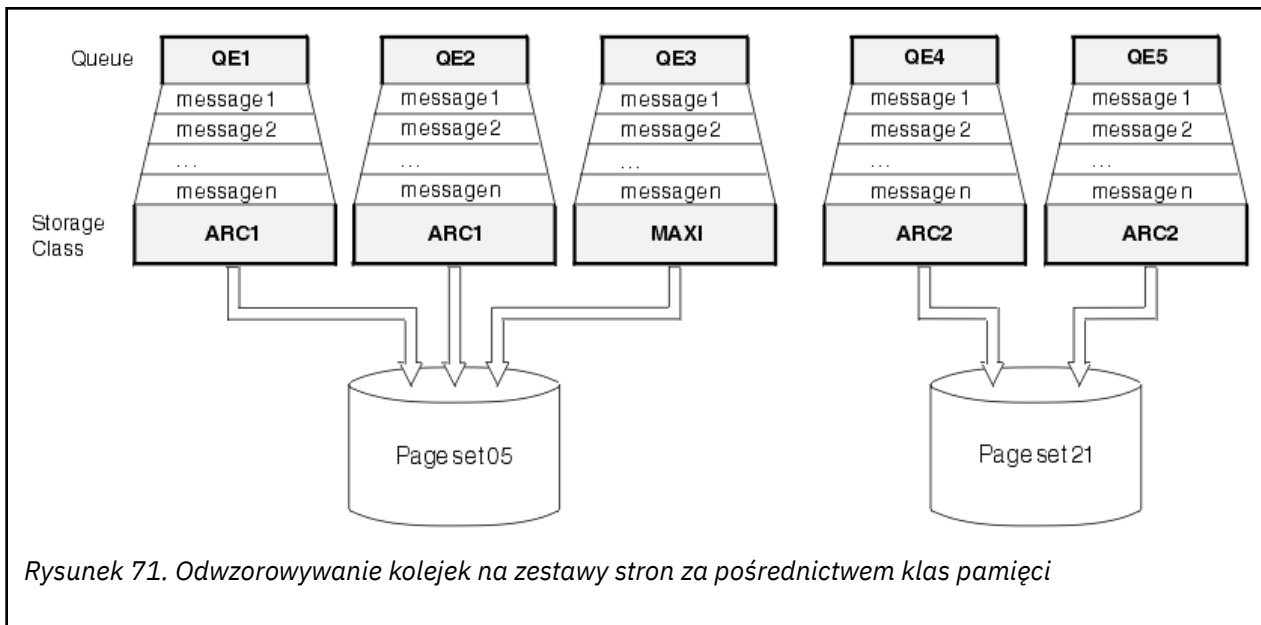
```
DEFINE STGCLASS(ARC2) PSID(21)
DEFINE QLOCAL(QE5) STGCLASS(ARC2)
```

Oznacza to, że komunikaty umieszczane w kolejce QE5 są składowane w zestawie stron 21 (jeśli pozostają w kolejce na tyle długo, aby można je było zapisać w pamięci DASD).

Ta sama klasa pamięci masowej może być używana przez więcej niż jedną kolejkę i można zdefiniować dowolną liczbę klas pamięci masowej. Na przykład można rozszerzyć poprzedni przykład, aby zawierał więcej definicji klas pamięci masowej i kolejek w następujący sposób:

```
DEFINE STGCLASS(ARC1) PSID(05)
DEFINE STGCLASS(ARC2) PSID(21)
DEFINE STGCLASS(MAXI) PSID(05)
DEFINE QLOCAL(QE1) STGCLASS(ARC1) ...
DEFINE QLOCAL(QE2) STGCLASS(ARC1) ...
DEFINE QLOCAL(QE3) STGCLASS(MAXI) ...
DEFINE QLOCAL(QE4) STGCLASS(ARC2) ...
DEFINE QLOCAL(QE5) STGCLASS(ARC2) ...
```

W produkcie Rysunek 71 na stronie 237 obie klasy pamięci masowej ARC1 i MAXI są powiązane z zestawem stron 05. Dlatego kolejki QE1, QE2 i QE3 są odwzorowywane na zestaw stron 05. Podobnie klasa pamięci ARC2 wiąże kolejki QE4 i QE5 z zestawem stron 21.



Jeśli użytkownik zdefiniuje kolejkę bez określania klasy pamięci, program IBM MQ użyje domyślnej klasy pamięci.

Jeśli komunikat jest umieszczany w kolejce, która nazywa nieistniejącą klasę pamięci, aplikacja otrzymuje błąd. Należy zmienić definicję kolejki, aby nadać jej nazwę istniejącej klasy pamięci lub utworzyć klasę pamięci nazwaną przez kolejkę.

Klasę pamięci masowej można zmienić tylko wtedy, gdy:

- Wszystkie kolejki używające tej klasy pamięci masowej są puste i nie mają niezatwierdzonego działania.
- Wszystkie kolejki używające tej klasy pamięci masowej są zamknięte.

z/OS Bufory i pule buforów dla systemu IBM MQ for z/OS

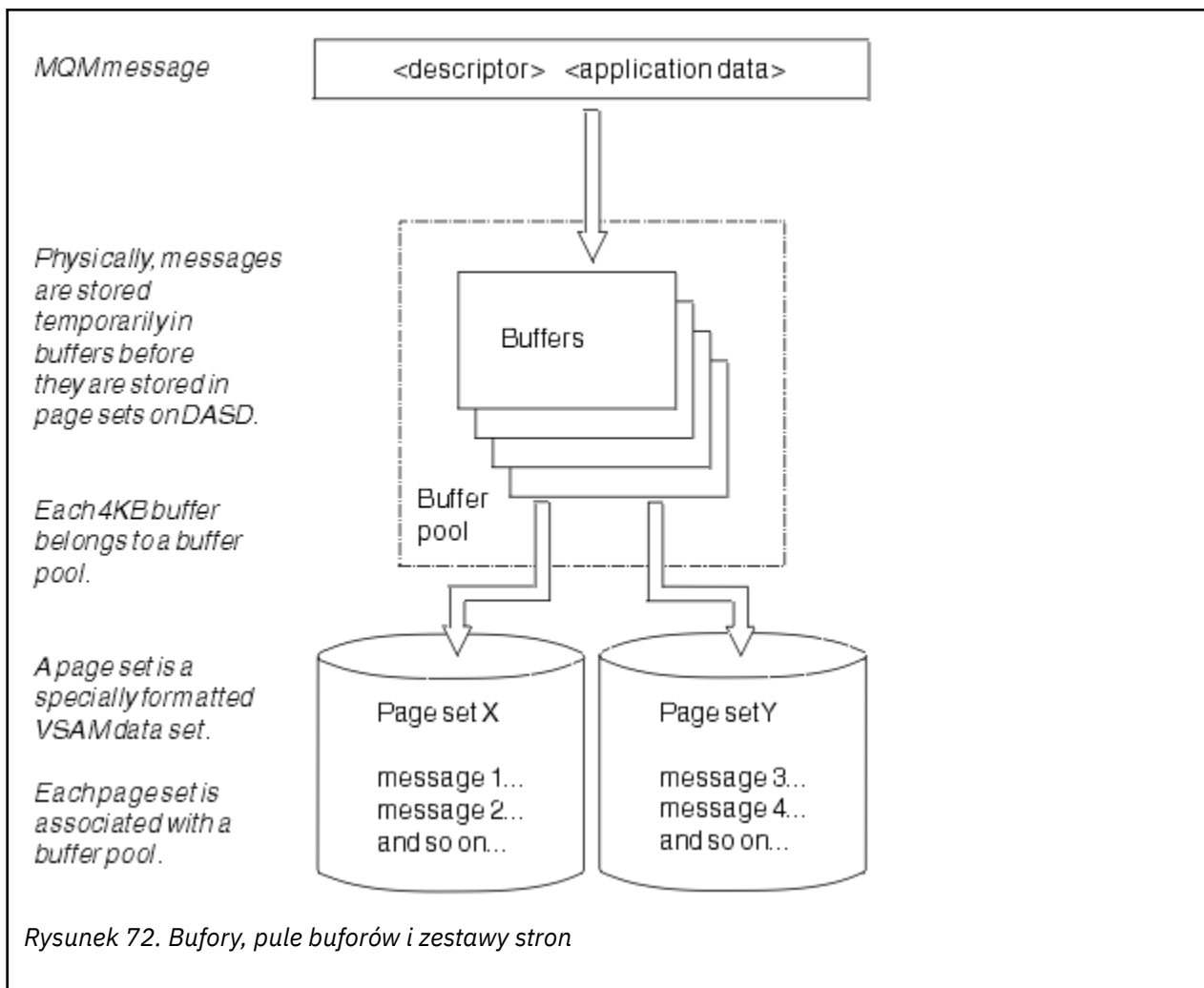
Program IBM MQ for z/OS używa buforów i pul buforów do tymczasowego buforowania danych. W tym temacie opisano sposób organizowania i używania buforów.

W celu zwiększenia efektywności produkt IBM MQ używa formy buforowania, w której komunikaty (i definicje obiektów) są tymczasowo przechowywane w buforach przed zapisaniem ich w zestawach stron na urządzeniu DASD. Komunikaty krótkotrwałe, czyli komunikaty, które są pobierane z kolejki krótko po ich odebraniu, mogą być przechowywane tylko w buforach. Ta aktywność buforowania jest kontrolowana przez menedżera buforów, który jest komponentem produktu IBM MQ.

Bufory są zorganizowane w *pule buforów*. Dla każdego menedżera kolejek można zdefiniować maksymalnie 100 pul buforów (od 0 do 99).

Zaleca się użycie minimalnej liczby pul buforów spójnej z separacją typu obiektu i komunikatu opisaną w sekcji [Rysunek 72 na stronie 238](#) oraz wszystkich wymagań dotyczących odseparowania danych, jakie może mieć aplikacja. Każdy bufor ma długość 4 kB. Pule buforów domyślnie używają 31-bitowej pamięci masowej, w tym trybie maksymalna liczba buforów jest określana przez ilość 31-bitowej pamięci masowej dostępnej w przestrzeni adresowej menedżera kolejek. Dla buforów nie należy używać więcej niż 70%. Alternatywnie przydzielanie pamięci puli buforów można wykonać z 64-bitowej pamięci masowej (należy użyć atrybutu LOCATION komendy **DEFINE BUFFPOOL**). Korzystanie z parametru LOCATION (ABOVE) w celu użycia 64-bitowej pamięci masowej ma dwie zalety. Po pierwsze, jest dużo więcej 64-bitowej pamięci masowej, więc pule buforów mogą być znacznie większe, a po drugie, 31-bitowa pamięć masowa jest udostępniana do użytku przez inne funkcje. Zwykle im więcej buforów, tym efektywniejsze buforowanie i lepsza wydajność systemu IBM MQ.

[Rysunek 72 na stronie 238](#) przedstawia relację między komunikatami, buforami, pulami buforów i zestawami stron. Pula buforów jest powiązana z jednym lub większą liczbą zestawów stron; każdy zestaw stron jest powiązany z pojedynczą pulą buforów.



Za pomocą komendy **ALTER BUFFPOOL** można dynamicznie wydawać komendy służące do modyfikowania wielkości i położenia puli buforów. Zestawy stron można dodawać dynamicznie za pomocą komendy **DEFINE PSID** lub usuwać za pomocą komendy **DELETE PSID**.

Jeśli pula buforów jest zbyt mała, IBM MQ wysyła komunikat **CSQP020E**. Następnie można dynamicznie dodać więcej buforów do puli buforów, której to dotyczy (w tym celu może być konieczne usunięcie buforów z innych pul buforów).

Liczbę buforów w puli określa się za pomocą komendy **DEFINE BUFFPOOL**, a wielkość pul buforów można dynamicznie zmieniać za pomocą komendy **ALTER BUFFPOOL**. Bieżącą liczbę buforów w puli można określić dynamicznie, wyświetlając zestaw stron korzystający z puli buforów za pomocą komendy **DISPLAY USAGE**.

Ze względu na wydajność nie należy umieszczać komunikatów i definicji obiektów w tej samej puli buforów. Należy użyć jednej puli buforów (na przykład zero) wyłącznie dla zerowego zestawu stron, w którym przechowywane są definicje obiektów. Podobnie komunikaty krótkotrwałe i długotrwałe należy przechowywać w różnych pulach buforów, a zatem w różnych zestawach stron i w różnych kolejkach.

Nie można użyć komendy **DEFINE BUFFPOOL** po restarcie w celu utworzenia nowej puli buforów. Zamiast tego, jeśli komenda **DEFINE PSID** używa słowa kluczowego DSN, może jawnie zidentyfikować pulę buforów, która nie jest aktualnie zdefiniowana. Nowa pula buforów zostanie utworzona.

Gdzie można znaleźć więcej informacji na temat zarządzania pamięcią masową w systemie IBM MQ for z/OS

Ta sekcja zawiera dodatkowe informacje na temat zarządzania pamięcią masową w systemie IBM MQ for z/OS.

Więcej informacji na temat tematów w tej sekcji można znaleźć w następujących źródłach:

Temat	Gdzie szukać
Jaka ilość pamięci masowej jest potrzebna	Planowanie wymagań dotyczących pamięci masowej i wydajności w systemie z/OS
Wielkość zestawów stron i pule buforów	Planowanie zestawów stron i pul buforów
Zarządzanie zestawami stron	Zarządzanie zestawami stron
Komendy MQSC	Komendy MQSC

logowanie IBM MQ for z/OS

IBM MQ przechowuje *dzienniki* zmian danych i istotnych zdarzeń w momencie ich wystąpienia. Te dzienniki mogą być używane do odtwarzania danych do poprzedniego stanu, jeśli jest to wymagane.

Zestaw danych programu startowego (BSDS) przechowuje informacje o zestawach danych, które zawierają dzienniki.

Dziennik nie zawiera informacji o statystykach, śledzeniu ani ocenie wydajności. Więcej informacji na temat informacji statystycznych i monitorowania gromadzonych przez IBM MQ zawiera sekcja [Monitorowanie i statystyki](#).

Więcej informacji na temat rejestrowania zawierają następujące tematy:

- [“Pliki dziennika w katalogu IBM MQ for z/OS” na stronie 239](#)
- [“Struktura dziennika” na stronie 244](#)
- [“Sposób zapisywania dzienników IBM MQ for z/OS” na stronie 244](#)
- [“Większy dziennik-względny adres bajtowy” na stronie 247](#)
- [“Zestaw danych programu startowego” na stronie 248](#)

Zadania pokrewne

[Planowanie środowiska rejestrowania](#)

[Ustawianie dzienników za pomocą modułu parametrów systemowych](#)

 [administrowanie z/OS](#)

Odsyłacze pokrewne

 [Komunikaty dla IBM MQ for z/OS](#)

Pliki dziennika w katalogu IBM MQ for z/OS

Pliki dzienników zawierają informacje potrzebne do odtwarzania transakcji. Aktywne pliki dziennika mogą być archiwizowane, dzięki czemu dane dziennika mogą być przechowywane przez długi czas.

Co to jest plik dziennika

IBM MQ rejestruje wszystkie istotne zdarzenia występujące w *aktywnym dzienniku*. Dziennik zawiera informacje potrzebne do odtworzenia:

- Komunikaty trwałe
- Obiekty IBM MQ , takie jak kolejki
- Menedżer kolejek produktu IBM MQ

Aktywny dziennik zawiera zbiór zestawów danych (do 310), które są używane cyklicznie.

Archiwizowanie dzienników można włączyć w taki sposób, aby po zapetnieniu aktywnego dziennika kopia była tworzona w archiwalnym zestawie danych. Archiwizacja umożliwia zachowanie danych dziennika przez dłuższy czas. Jeśli archiwizacja nie jest używana, dzienniki zawijają się, a wcześniejsze dane są nadpisywane. Aby odtworzyć zestaw stron lub dane w strukturze CF, potrzebne są dane dziennika po utworzeniu kopii zapasowej zestawu stron lub struktury. Dziennik archiwalny można utworzyć na dysku lub na taśmie.

Archiwizowanie

Ponieważ aktywny dziennik ma stałą wielkość, IBM MQ okresowo kopiuje zawartość każdego zestawu danych dziennika do *dziennika archiwalnego*, który zwykle jest zestawem danych na urządzeniu pamięci masowej o dostępie bezpośrednim (DASD) lub na taśmie magnetycznej. Jeśli wystąpi awaria podsystemu lub transakcji, program IBM MQ używa aktywnego dziennika i, w razie potrzeby, dziennika archiwalnego do odtwarzania.

Dziennik archiwum może zawierać do 1000 sekwencyjnych zestawów danych. Każdy zestaw danych można wpisać do katalogu za pomocą zintegrowanego narzędzia katalogowego (Integrated Catalog Facility-ICF) systemu z/OS .

Archiwizacja jest podstawowym komponentem odtwarzania IBM MQ . Jeśli jednostka odzyskiwania jest długotrwała, rekordy dziennika w tej jednostce odzyskiwania mogą znajdować się w dzienniku archiwalnym. W takim przypadku odtwarzanie wymaga danych z dziennika archiwalnego. Jeśli jednak archiwizacja jest wyłączona, aktywny dziennik z nowymi rekordami dziennika nadpisuje wcześniejsze rekordy dziennika. Oznacza to, że program IBM MQ może nie być w stanie wycofać jednostki odzyskiwania, a komunikaty mogą zostać utracone. Menedżer kolejek kończy działanie nieprawidłowo.

Dlatego w środowisku produkcyjnym **nigdy nie należy wyłączać archiwizowania**. W przeciwnym razie istnieje ryzyko utraty danych po awarii systemu lub transakcji. Tylko w przypadku pracy w środowisku testowym można rozważyć wyłączenie archiwizacji. W tym celu należy użyć makra CSQ6LOGP , które zostało opisane w sekcji [Korzystanie z komendy CSQ6LOGP](#).

Aby zapobiec problemom z nieplanowanymi długo działającymi jednostkami pracy, program IBM MQ wysyła komunikat ([CSQJ160I](#) lub [CSQJ161I](#)) jeśli podczas przetwarzania przenoszenia aktywnego dziennika zostanie wykryta długotrwała jednostka pracy.

podwójne rejestrowanie

W przypadku rejestrowania podwójnego każdy rekord dziennika jest zapisywany w dwóch różnych aktywnych zestawach danych dziennika, aby zminimalizować prawdopodobieństwo wystąpienia problemów z utratą danych podczas restartu.

Produkt IBM MQ można skonfigurować do działania z *pojedynczym rejestrowaniem* lub *podwójnym rejestrowaniem*. W przypadku pojedynczego rejestrowania rekordy dziennika są zapisywane raz w aktywnym zestawie danych dziennika. Każdy aktywny zestaw danych dziennika jest liniowym zestawem danych (LDS) VSAM pojedynczego przydziału. W przypadku podwójnego rejestrowania każdy rekord dziennika jest zapisywany w dwóch różnych zestawach danych aktywnego dziennika. Podwójne rejestrowanie minimalizuje prawdopodobieństwo wystąpienia problemów z utratą danych podczas restartu.

Przetwasowanie dziennika

Przetwasowanie dziennika powoduje, że rekordy dziennika dla niektórych jednostek pracy są zapisywane dalej w dzienniku. Zmniejsza to ilość danych dziennika, które muszą zostać odczytane podczas restartowania lub wycofywania menedżera kolejek w przypadku długotrwałych lub długotrwałych wątpliwych jednostek pracy.

Jeśli jednostka pracy jest traktowana jako długa, w dalszej części dziennika zapisywana jest reprezentacja każdego rekordu dziennika. Technika ta jest nazywana *shunting*. Po przetworzeniu całej jednostki pracy jednostka pracy jest w stanie *shunted* (przerobiona). Każde działanie wycofania lub restartu związane z odłączoną jednostką pracy może korzystać z rekordów dziennika, zamiast z oryginalnych rekordów dziennika jednostki pracy.

Wykrywanie długotrwałej jednostki pracy jest funkcją procesu punktu kontrolnego. W czasie punktu kontrolnego każda aktywna jednostka pracy jest sprawdzana w celu określenia, czy należy ją usunąć. Jeśli jednostka pracy przeszła przez dwa wcześniejsze punkty kontrolne od czasu jej utworzenia lub od czasu jej ostatniego przetwasowania, jednostka pracy jest odpowiednia do przetwasowania. Oznacza to, że pojedyncza jednostka pracy może być przetwaszana więcej niż jeden raz. Jest to jednostka pracy *wielowartościowa*.

Jednostka pracy jest odłączana co trzy punkty kontrolne. Punkt kontrolny jest jednak wykonywany asynchronicznie względem przetwasznika dziennika (lub zapisu rekordu dziennika, który spowodował przekroczenie wartości LOGLOAD).

W danym momencie występuje tylko jeden punkt kontrolny, dlatego przed zakończeniem punktu kontrolnego może istnieć wiele przetwaszników dziennika.

Oznacza to, że jeśli nie ma wystarczającej liczby aktywnych dzienników lub jeśli są one zbyt małe, to przed zapelnieniem wszystkich dzienników może nie zostać zakończone usuwanie dużej jednostki pracy.

Jeśli nie można zakończyć operacji *shunting*, zostanie wyświetlony komunikat [CSQR027I](#).

Jeśli archiwizacja dziennika jest wyłączona, występuje błąd ABEND 5C6 z przyczyną 00D1032A, jeśli wystąpi próba wyrejestrowania jednostki pracy, dla której wyłączenie nie powiodło się. Aby uniknąć tego problemu, należy użyć parametru OFFLOAD=YES.

Funkcja *shunting* dziennika jest zawsze aktywna i działa niezależnie od tego, czy archiwizacja dziennika jest włączona, czy nie.

Uwaga: Mimo że wszystkie rekordy dziennika dla jednostki pracy są przetwaswane, cała zawartość każdego rekordu nie jest przetwaszana, tylko część, która jest niezbędna do wycofania. Oznacza to, że ilość zapisywanych danych dziennika jest utrzymywana na minimalnym poziomie, a w przypadku wystąpienia błędu zestawu stron nie można użyć takich rekordów. Długotrwała jednostka pracy to jednostka, która działała przez więcej niż trzy punkty kontrolne menedżera kolejek.

Więcej informacji na temat tworzenia dzienników zawiera sekcja [Zarządzanie dziennikami](#).

Kompresja dziennika

Produkt IBM MQ for z/OS można skonfigurować w taki sposób, aby rekordy dziennika były kompresowane i dekompresowane w miarę ich zapisywania i odczytywania z zestawu danych dziennika.

Kompresji dziennika można użyć w celu zmniejszenia ilości danych zapisywanych w dzienniku dla trwałych komunikatów w kolejkach prywatnych. Osiągnięta kompresja zależy od typu danych zawartych w komunikatach. Na przykład kodowanie RLE (Run Length Encoding) działa poprzez upakowywanie powtarzających się instancji bajtów, co może dać dobre wyniki w przypadku danych ustrukturyzowanych lub zorientowanych na rekordy.



Ostrzeżenie: Komunikaty trwałe umieszczone w kolejce współużytkowanej nie podlegają kompresji dziennika.

Pola w sekcji Menedżer rejestrowania rekordów SMF (System Management Facility 115) umożliwiają monitorowanie stopnia kompresji danych. Więcej informacji na temat środowiska SMF zawiera sekcja

Korzystanie z narzędzia do zarządzania systemem oraz sekcja Komunikaty dotyczące rozliczania i statystyk.

Kompresja dziennika zwiększa wykorzystanie procesora w systemie. Kompresji należy używać tylko wtedy, gdy przepustowość menedżera kolejek jest ograniczona przez przepustowość we/wy zapisaną w zestawach danych dziennika lub gdy pamięć dyskowa potrzebna do przechowywania zestawów danych dziennika jest ograniczona. Jeśli używane są kolejki współużytkowane, ograniczenia przepustowości we/wy mogą zostać zwolnione przez dodanie dodatkowych menedżerów kolejek do grupy współużytkowania kolejek i rozłożenie obciążenia na więcej menedżerów kolejek.

W razie potrzeby można włączyć i wyłączyć opcję kompresji dziennika bez konieczności zatrzymywania i restartowania menedżera kolejek. Menedżer kolejek może odczytywać dowolne skompresowane rekordy dziennika bez względu na bieżące ustawienie kompresji dziennika.

Menedżer kolejek obsługuje 3 ustawienia kompresji dziennika.

Brak

Kompresja danych dziennika nie jest używana. Jest to wartość domyślna.

RLE

Kompresja danych dziennika jest wykonywana przy użyciu kodowania RLE (run-length encoding).

ANY

Włącz menedżer kolejek, aby wybrać algorytm kompresji, który zapewnia największy stopień kompresji rekordów dziennika. Ta opcja powoduje kompresję RLE.

Kompresją rekordów dziennika można sterować za pomocą jednej z następujących opcji:

- Komendy SET i DISPLAY LOG w MQSC; patrz [SET LOG](#) i [DISPLAY LOG](#)
- Funkcje Set Log i Inquire Log w interfejsie PCF; patrz [Set log](#) i [Inquire log](#)
- Makro CSQ6LOGP w module parametrów systemowych; patrz sekcja [Korzystanie z komendy CSQ6LOGP](#)

Dodatkowo program narzędziowy Log Print CSQ1LOGP obsługuje rozwijanie skompresowanych rekordów dziennika.

Dane dziennika

Dziennik może zawierać maksymalnie 18 milionów ($1.8 \cdot 10^{19}$) bajtów. Każdy bajt może być adresowany przez przesunięcie od początku dziennika, a to przesunięcie jest nazywane jego *względny adresem bajtowym* (RBA).

Adres RBA jest wskazywany przez 6-bajtowe lub 8-bajtowe pole zawierające całkowity adresowalny zakres 2^{48} bajtów lub 2^{64} bajtów, w zależności od tego, czy używane są 6-bajtowe lub 8-bajtowe dzienniki RBA.

Jeśli jednak program IBM MQ wykryje, że używany zakres jest poza F00000000000 (jeśli używane są 6-bajtowe RBA) lub FFFF800000000000 (jeśli używane są 8-bajtowe dzienniki RBA), zostaną wyświetlone komunikaty [CSQI045](#), [CSQI046](#), [CSQI047](#) i [CSQJ032](#), ostrzeżenie o zresetowaniu dziennika RBA.

Jeśli wartość RBA osiągnie wartość FFF800000000 (jeśli używane są 6-bajtowe dzienniki RBA) lub FFFFFFFC000000000 (jeśli używane są 8-bajtowe dzienniki RBA), menedżer kolejek kończy działanie z kodem przyczyny [00D10257](#).

Po wysłaniu komunikatów ostrzegawczych dotyczących używanego zakresu dziennika należy zaplanować wyłączenie menedżera kolejek, podczas którego menedżer kolejek może zostać przekształcony w taki sposób, aby korzystał z 8-bajtowych dzienników RBA, lub można zresetować dziennik. Procedura resetowania dziennika została opisana w sekcji [Resetowanie dziennika menedżera kolejek](#).

Jeśli menedżer kolejek używa 6-bajtowych dzienników RBAs, należy rozważyć przekształcenie menedżera kolejek w taki sposób, aby korzystał z 8-bajtowych dzienników RBAs, zamiast resetować dziennik menedżera kolejek, zgodnie z procedurą opisaną w sekcji [Implementowanie względnego adresu bajtowego większego dziennika](#).

Dziennik składa się z *rekordów dziennika*, z których każdy jest zestawem danych dziennika traktowanym jako pojedyncza jednostka. Rekord dziennika jest identyfikowany przez adres RBA pierwszego bajtu jego nagłówka lub przez numer kolejny rekordu dziennika (LRSN). RBA lub LRSN jednoznacznie identyfikuje rekord, który rozpoczyna się w określonym punkcie dziennika.

To, czy do identyfikowania punktów dziennika używany jest adres RBA, czy numer LRSN, zależy od tego, czy używane są grupy współużytkowania kolejek. W środowisku współużytkowania kolejki nie można użyć względnego adresu bajtowego do jednoznacznego zidentyfikowania punktu dziennika, ponieważ wiele menedżerów kolejek może aktualizować tę samą kolejkę w tym samym czasie, a każdy z nich ma własny dziennik. Aby rozwiązać ten problem, numer kolejny rekordu dziennika jest określany na podstawie wartości znacznika czasu i nie musi oznaczać fizycznego przesunięcia rekordu dziennika w obrębie dziennika.

Każdy rekord dziennika ma nagłówek, który nadaje mu typ, podkomponent IBM MQ, który go zapisał, oraz, w przypadku rekordów jednostki odzyskiwania, identyfikator jednostki odzyskiwania.

Istnieją cztery typy rekordów dziennika, opisane w następujących nagłówkach:

- [Rekordy dziennika jednostki odtwarzania](#)
- [Rekordy punktów kontrolnych](#)
- [Rekordy sterujące zestawu stron](#)
- [Rekordy kopii zapasowej struktury CF](#)

Rekordy dziennika jednostki odtwarzania

Większość rekordów dziennika opisuje zmiany w kolejkach IBM MQ. Wszystkie takie zmiany są wprowadzane w obrębie jednostek odzyskiwania.

IBM MQ wykorzystuje specjalne techniki rejestrowania obejmujące *cofanie/ponawianie* i *kompensowanie rekordów dziennika* w celu skrócenia czasu restartu i zwiększenia dostępności systemu.

Jednym z efektów jest to, że czas restartu jest ograniczony. Jeśli podczas restartu wystąpi niepowodzenie, tak że menedżer kolejek musi zostać zrestartowany po raz drugi, wszystkie działania odtwarzania, które zostały zakończone do punktu awarii podczas pierwszego restartu, nie muszą być ponownie stosowane podczas drugiego restartu. Oznacza to, że kolejne restarty nie trwają stopniowo dłużej.

Rekordy punktów kontrolnych

Aby skrócić czas restartu, program IBM MQ podczas normalnego działania przyjmuje okresowe punkty kontrolne. Występują one w następujący sposób:

- Po zapisaniu predefiniowanej liczby rekordów dziennika. Ta liczba jest definiowana przez operand częstotliwości punktu kontrolnego o nazwie LOGLOAD dla parametru systemowego macro CSQ6SYSP, opisany w sekcji [Korzystanie z komendy CSQ6SYSP](#).
- Po pomyślnym restarcie.
- Przy normalnym zakończeniu.
- Za każdym razem, gdy IBM MQ przełącza się na następny aktywny zestaw danych dziennika w cyklu.

W momencie wykonywania punktu kontrolnego IBM MQ wydaje komendę DISPLAY CONN (opisaną w sekcji [DISPLAY CONN](#)). wewnątrz, aby lista obecnie wrażliwych połączeń została zapisana w dzienniku konsoli z/OS.

Rekordy sterujące zestawu stron

Rekordy te rejestrują zestawy stron i pule buforów znane menedżerowi kolejek systemu IBM MQ w każdym punkcie kontrolnym oraz rejestrują informacje o zakresach dzienników wymaganych do przeprowadzenia odtwarzania nośników zestawu stron w czasie punktu kontrolnego.

Niektóre dynamiczne zmiany w zestawach stron i pulach buforów są również zapisywane jako rekordy sterujące zestawu stron, dzięki czemu zmiany mogą zostać odtworzone i automatycznie przywrócone podczas następnego restartu menedżera kolejek.

Rekordy kopii zapasowej struktury CF

Te rekordy zawierają dane odczytane ze struktury listy narzędzia CF w odpowiedzi na komendę BACKUP CFSTRUCT. W mało prawdopodobnym przypadku awarii struktury narzędzia CF rekordy te są używane razem z rekordami jednostki odtwarzania przez komendę RECOVER CFSTRUCT do odtworzenia nośnika struktury narzędzia CF do punktu awarii.

Zadania pokrewne

[Implementowanie większego dziennika względnego adresu bajtowego](#)

Struktura dziennika

W tym temacie opisano terminologię używaną do opisywania rekordów dziennika.

Każdy aktywny zestaw danych dziennika musi być liniowym zestawem danych VSAM (LDS). Fizyczną jednostką wyjściową zapisaną w aktywnym zestawie danych dziennika jest odstęp czasu sterowania (CI) o wielkości 4 kB. Każdy CI zawiera jeden rekord VSAM.

Rekordy protokołu fizycznego i logicznego

Jeden CI VSAM jest rekordem *fizycznym*. Informacje rejestrowane w określonym czasie tworzą rekord *logiczny* o długości, która różni się w zależności od ilości miejsca dostępnego w elemencie CI. Dlatego jeden rekord fizyczny może zawierać:

- Kilka rekordów logicznych
- Jeden lub więcej rekordów logicznych i część innego rekordu logicznego
- Tylko część jednego rekordu logicznego

Termin *rekord dziennika* odnosi się do rekordu *logicznego*, niezależnie od tego, ile *fizycznych* rekordów jest potrzebnych do jego zapisania.

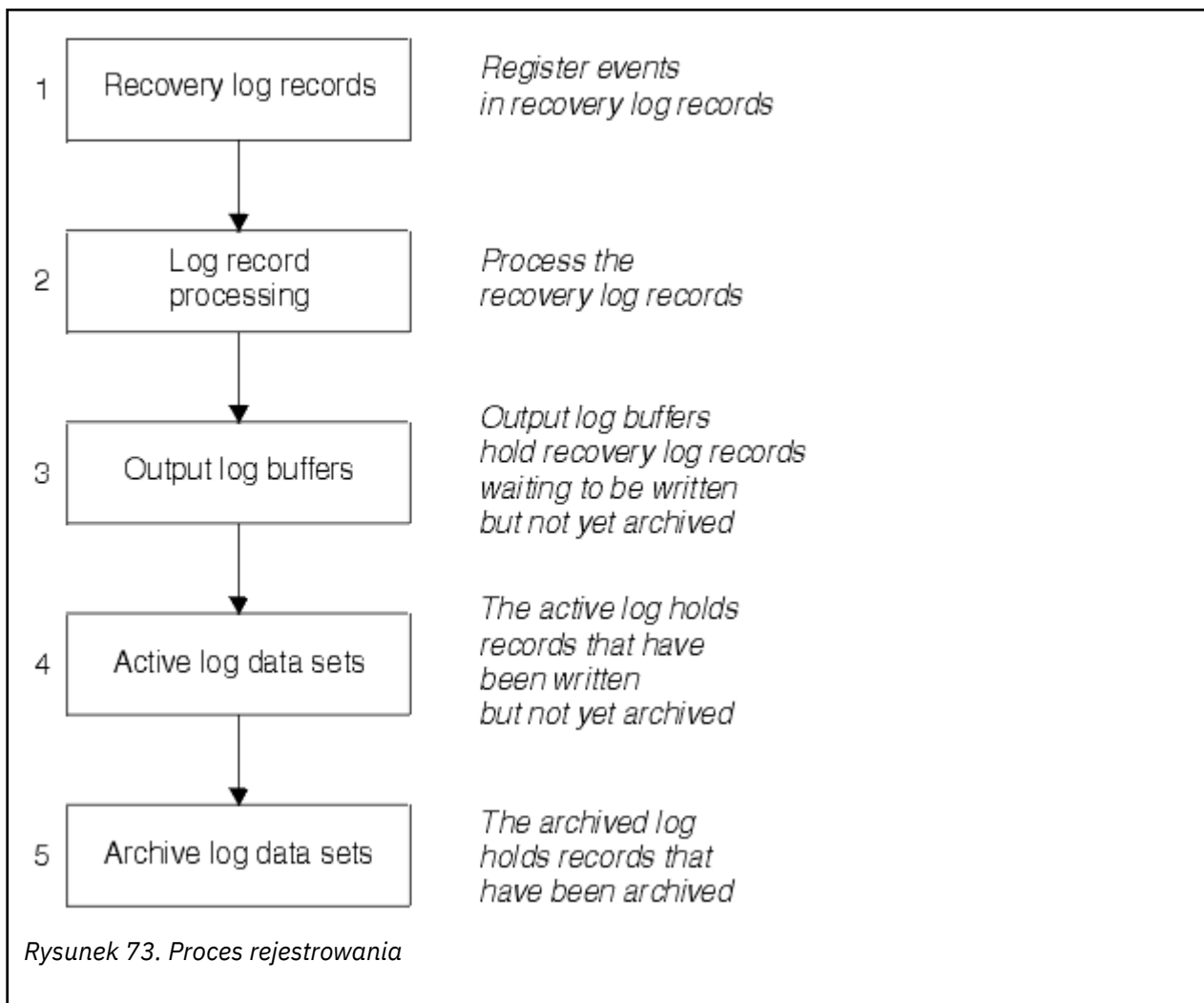
Sposób zapisywania dzienników IBM MQ for z/OS

W tym temacie opisano sposób przetwarzania rekordów pliku dziennika przez produkt IBM MQ.

IBM MQ zapisuje każdy rekord dziennika w zestawie danych DASD nazywanym *aktywnym dziennikiem*. Gdy aktywny dziennik jest pełny, program IBM MQ kopiuje jego zawartość do zestawu danych DASD lub taśm, nazywanego *dziennikiem archiwalnym*. Ten proces jest nazywany *przenoszeniem*.

Rysunek 73 na stronie 245 przedstawia proces rejestrowania. Rekordy dziennika zwykle przechodzą przez następujący cykl:

1. IBM MQ odnotowuje zmiany w danych i istotnych zdarzeniach w rekordach dziennika odtwarzania.
2. Program IBM MQ przetwarza rekordy dziennika odtwarzania i w razie potrzeby dzieli je na segmenty.
3. Rekordy dziennika są umieszczane sekwencyjnie w *wyjściowych buforach dziennika*, które są sformatowane jako VSAM Controls Intervals (CI). Każdy rekord dziennika jest identyfikowany przez względny adres bajtowy z zakresu od 0 do $2^{64} - 1$.
4. Elementy CI są zapisywane w zestawie predefiniowanych zestawów danych aktywnego dziennika DASD, które są używane sekwencyjnie i ponownie wykorzystywane.
5. Jeśli archiwizacja jest aktywna, po zapelnieniu każdego zestawu danych dziennika aktywnego jego treść jest automatycznie przenoszona do nowego zestawu danych dziennika archiwalnego.



Przy zapisywaniu aktywnego dziennika

Bufory dziennika w pamięci masowej są zapisywane w aktywnym zestawie danych dziennika za każdym razem, gdy wystąpi jeden z następujących zdarzeń:

- Bufory dziennika zostaną wypełnione.
- Osiągnięto próg zapisu (określony w makrze CSQ6LOGP).
- Występują pewne istotne zdarzenia, takie jak punkt zatwierdzania lub wywołanie komendy IBM MQ BACKUP CFSTRUCT.

Po zainicjowaniu menedżera kolejek aktywne zestawy danych dziennika o nazwie określonej w zestawie BSDS są dynamicznie przydzielane do wyłącznego użytku przez menedżer kolejek i pozostają przydzielone wyłącznie do programu IBM MQ do czasu zakończenia działania menedżera kolejek.

Dynamiczne dodawanie zestawów danych dziennika

Istnieje możliwość dynamicznego definiowania nowych zestawów danych aktywnego dziennika podczas działania menedżera kolejek. Ta funkcja rozwiązuje problem związany z zawieszaniem menedżera kolejek, gdy archiwizacja nie może przenieść aktywnych dzienników z powodu przejściowych problemów. Więcej informacji na ten temat zawiera opis komendy [DEFINE LOG](#).

Uwaga: Aby ponownie zdefiniować lub usunąć aktywne dzienniki, należy zakończyć i zrestartować menedżer kolejek.

IBM MQ i Storage Management Subsystem

Parametry systemu IBM MQ umożliwiają określenie klas pamięci masowej podsystemu zarządzania pamięcią masową (Storage Management Subsystem- MVS/DFP SMS) podczas dynamicznego przydzielania zestawów danych dziennika archiwalnego systemu IBM MQ . Program IBM MQ inicjuje archiwizację zestawów danych dziennika, ale do przydzielania archiwalnego zestawu danych można użyć programu SMS.

Odsyłacze pokrewne

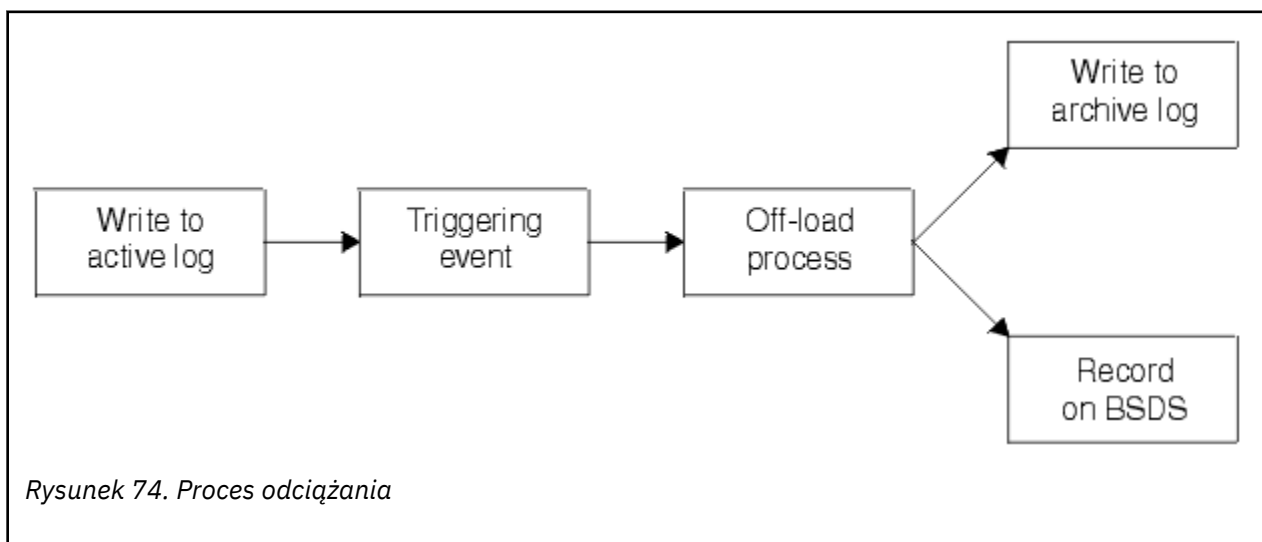
[“Przy zapisywaniu dziennika archiwalnego systemu IBM MQ for z/OS” na stronie 246](#)

Ten temat zawiera informacje o procesie kopiowania aktywnych dzienników do dzienników archiwalnych oraz o tym, kiedy ten proces ma miejsce.

Przy zapisywaniu dziennika archiwalnego systemu IBM MQ for z/OS

Ten temat zawiera informacje o procesie kopiowania aktywnych dzienników do dzienników archiwalnych oraz o tym, kiedy ten proces ma miejsce.

Proces kopiowania aktywnych dzienników do dzienników archiwalnych jest nazywany *przenoszeniem*. Relacja odciążania innych rejestrowanych zdarzeń jest przedstawiona schematycznie w pliku [Rysunek 74](#) na stronie 246.



Wyzwalanie procesu przenoszenia

Proces przenoszenia aktywnego dziennika do dziennika archiwalnego może być wyzwalany przez kilka zdarzeń. Na przykład:

- Wypełnienie aktywnego zestawu danych dziennika.
- Przy użyciu komendy MQSC ARCHIVE LOG.
- Wystąpił błąd podczas zapisywania w aktywnym zestawie danych dziennika.

Zestaw danych jest obcinany przed punktem awarii, a rekord, który nie został zapisany, staje się pierwszym rekordem nowego zestawu danych. Przenoszenie jest wyzwalane dla obciążonego zestawu danych, tak jak w przypadku normalnego pełnego zestawu danych dziennika. Jeśli istnieją dwa aktywne dzienniki, obie kopie są obcinane, dzięki czemu obie kopie pozostają zsynchronizowane.

Komunikat CSQJ110E jest generowany, gdy ostatni dostępny aktywny dziennik jest zapelniony w 5%, a następnie jest zwiększany o 5%, co oznacza procent wielkości dziennika, która jest w użyciu. Jeśli

wszystkie aktywne dzienniki zostaną zapełnione, program IBM MQ zatrzyma przetwarzanie do momentu jego odciążania i wyśle następujący komunikat:

```
CSQJ111A +CSQ1 OUT OF SPACE IN ACTIVE LOG DATA SETS
```

Proces przenoszenia

Gdy wszystkie aktywne dzienniki zostaną zapełnione, program IBM MQ uruchamia proces przenoszenia i zatrzymuje przetwarzanie do momentu zakończenia procesu przenoszenia. Jeśli przetwarzanie przenoszenia nie powiedzie się, gdy aktywne dzienniki będą pełne, program IBM MQ zostanie zakończony awaryjnie.

Gdy aktywny dziennik jest gotowy do odciążania, do operatora konsoli z/OS wysyłane jest żądanie podłączenia taśmy lub przygotowania jednostki DASD. Wartość opcji rejestrowania ARCWTOR (więcej informacji na ten temat zawiera sekcja [Korzystanie z komendy CSQ6ARVP](#)) określa, czy żądanie jest odbierane. Jeśli do odciążania używana jest taśma, należy określić ARCWTOR=YES. Jeśli wartością jest YES, żądanie jest poprzedzone wartością WTOR (numer komunikatu CSQJ008E) nakazującą operatorowi przygotowanie zestawu danych dziennika archiwalnego do przydzielenia.

Operator nie musi natychmiast odpowiadać na ten komunikat. Jednak opóźnienie odpowiedzi opóźnia proces przenoszenia. Nie ma to wpływu na wydajność systemu IBM MQ, chyba że operator opóźnia odpowiedź tak długo, że w systemie IBM MQ zabraknie aktywnych dzienników.

Operator może odpowiedzieć, anulując proces przenoszenia. W takim przypadku, jeśli przydział dotyczy pierwszej kopii zestawów danych z podwójnym archiwum, proces przenoszenia jest tylko opóźniany do momentu zapełnienia następnego aktywnego zestawu danych dziennika. Jeśli przydział dotyczy drugiej kopii, proces archiwizacji przełącza się na tryb pojedynczej kopii, ale tylko dla tego zestawu danych.

Przerwania i błędy podczas przenoszenia

Żądanie zatrzymania menedżera kolejek nie jest uwzględniane do momentu zakończenia przetwarzania przenoszenia. Jeśli operacja IBM MQ zakończy się niepowodzeniem podczas przenoszenia w toku, przenoszenie rozpocznie się ponownie po zrestartowaniu menedżera kolejek.

Komunikaty podczas przetwarzania odciążania

Odciążone komunikaty są wysyłane do konsoli z/OS przez IBM MQ oraz przez proces odciążania. Tych komunikatów można użyć do znalezienia zakresów RBA w różnych zestawach danych dziennika.

Większy dziennik-względny adres bajtowy

Ta funkcja zwiększa dostępność menedżera kolejek, zwiększając czas, po upływie którego należy zresetować dziennik.

Dane odtwarzania są zapisywane w dzienniku, dzięki czemu komunikaty trwałe są dostępne po zrestartowaniu menedżera kolejek. Termin log Relative Byte Address (log RBA) jest używany do odwoływania się do położenia danych jako przesunięcie od początku dziennika.

Przed IBM MQ 8.06-bajtowy dziennik RBA może adresować do 256 terabajtów danych. Przed zapisaniem tej liczby rekordów dziennika należy zresetować dziennik menedżera kolejek, wykonując procedurę opisaną w sekcji [Resetowanie dziennika menedżera kolejek](#).

Resetowanie dzienników menedżerów kolejek nie jest szybkim procesem i może wymagać rozszerzonego wyłączenia z powodu konieczności zresetowania zestawów stron w ramach procesu. W przypadku często używanego menedżera kolejek ta operacja może być zwykle wykonywana raz w roku.

W produkcji IBM MQ 8.0 adres RBA dziennika może mieć długość 8 bajtów, a menedżer kolejek może teraz adresować ponad 64 000 razy więcej danych (16 eksabajtów), zanim będzie konieczne zresetowanie adresu RBA dziennika. Wpływ użycia większego adresu RBA dziennika jest taki, że wielkość zapisywanych danych dziennika zwiększa się o kilka bajtów.

Kiedy ta funkcja jest włączona?

V 9.3.0 Dla menedżerów kolejek utworzonych w wersji IBM MQ 9.3.0 lub nowszej ta funkcja jest już włączona.

Jeśli bieżący adres RBA dziennika zbliża się do końca zakresu adresu RBA dziennika, należy rozważyć przekształcenie menedżera kolejek w taki sposób, aby używał 8-bajтового adresu RBA dziennika zamiast resetowania dziennika menedżera kolejek. Przekształcenie menedżera kolejek w taki sposób, aby korzystał z 8-bajtowych strategii RBA, wymaga krótszego wyłączenia niż resetowanie dziennika i znacznie wydłuża czas, zanim będzie konieczne zresetowanie dziennika.

Komunikat CSQJ034I, wydany podczas inicjowania menedżera kolejek, wskazuje koniec zakresu RBA dziennika dla menedżera kolejek zgodnie z konfiguracją i może zostać użyty do określenia, czy używane są 6-bajtowe, czy 8-bajtowe dzienniki RBA.

W jaki sposób ta funkcja jest włączona?

8-bajtowy adres RBA dziennika jest włączany przez uruchomienie menedżera kolejek w formacie BSDS w wersji 2. Podsumowując, można to osiągnąć poprzez:

1. Upewnianie się, że wszystkie menedżery kolejek w grupie współużytkowania kolejek spełniają wymagania dotyczące włączania 8-bajтового adresu RBA dziennika
2. Czyste zamykanie menedżera kolejek
3. Uruchomienie programu narzędziowego do konwersji BSDS w celu utworzenia kopii BSDS w formacie wersji 2.
4. Restartowanie menedżera kolejek z przekształconym BSDS.

Po przekształceniu menedżera kolejek w celu użycia 8-bajtowych dzienników RBA nie można wrócić do używania 6-bajтового dziennika RBA.

Szczegółowe informacje na temat włączania 8-bajтового dziennika RBA zawiera sekcja [Implementowanie większego dziennika względnego adresu bajtowego](#).

Zadania pokrewne

[Planowanie zwiększenia maksymalnego zakresu adresowalnego dziennika](#)

Odsyłacze pokrewne

[Program narzędziowy do konwersji BSDS \(CSQJUCNV\)](#)

Z/OS Zestaw danych programu startowego

Zestaw danych programu startowego jest wymagany przez program IBM MQ jako mechanizm odwoływania się do zestawów danych dziennika i rekordów dziennika. Informacje te są wymagane podczas normalnego przetwarzania i odtwarzania po restarcie.

Do czego służy zestaw danych programu startowego

Zestaw danych programu startowego (BSDS) jest zestawem danych uporządkowanych według klucza VSAM (KSDS), który przechowuje informacje wymagane przez produkt IBM MQ. Zawiera on następujące elementy:

- Spis wszystkich aktywnych i zarchiwizowanych zestawów danych dziennika znanych programowi IBM MQ. Produkt IBM MQ używa tego magazynu do:
 - Śledzenie aktywnych i zarchiwizowanych zestawów danych dziennika
 - Znajdź rekordy dziennika, aby spełnić żądania odczytu dziennika podczas normalnego przetwarzania
 - Znajdź rekordy dziennika, aby mógł on obsłużyć przetwarzanie restartu

Program IBM MQ zapisuje informacje w spisie zasobów za każdym razem, gdy zestaw danych dziennika archiwalnego jest zdefiniowany lub gdy zestaw danych dziennika aktywnego jest ponownie wykorzystywany. W przypadku aktywnych dzienników w spisie zasobów są wyświetlane informacje

o tym, które są pełne i dostępne do ponownego wykorzystania. Magazyn zawiera względny adres bajtu (RBA) każdej części dziennika przechowywanej w tym zestawie danych.

- *Zawijanie* spisu zasobów wszystkich ostatnich działań IBM MQ . Jest to konieczne, jeśli konieczne jest zrestartowanie menedżera kolejek.

Zestaw danych BSDS jest wymagany, jeśli w menedżerze kolejek wystąpił błąd i konieczne jest jego zrestartowanie. IBM MQ **musi** mieć BSDS. Aby zminimalizować prawdopodobieństwo wystąpienia problemów podczas restartu, można skonfigurować produkt IBM MQ z podwójnymi zestawami BSD, z których każdy rejestruje te same informacje. Użycie podwójnych zestawów BSD jest znane jako działanie w trybie podwójnym. Jeśli to możliwe, należy umieścić kopie na oddzielnych woluminach. Zmniejsza to ryzyko ich utraty w przypadku uszkodzenia lub zniszczenia woluminu. Użyj podwójnych zestawów BSD zamiast podwójnego zapisu na DASD.

Zestaw danych BSDS jest konfigurowany podczas dostosowywania programu IBM MQ i można zarządzać spisem zasobów za pomocą programu narzędziowego do spisywania zasobów dziennika zmian (CSQJU003). Więcej informacji na temat tego narzędzia znajduje się w sekcji Administrowanie programem IBM MQ for z/OS. Jest ona przywoływana przez instrukcję DD w procedurze uruchamiania menedżera kolejek.

Zwykle program IBM MQ przechowuje zduplikowane kopie BSDS. Jeśli wystąpi błąd we/wy, zwalnia błędną kopię i kontynuuje działanie z pojedynczym BSDS. Istnieje możliwość odtworzenia operacji w trybie podwójnym, co opisano w sekcji Administrowanie programem IBM MQ for z/OS.

Aktywne dzienniki są najpierw rejestrowane w serwerze BSDS podczas instalowania produktu IBM MQ . Nie można zastąpić aktywnych dzienników bez zakończenia i zrestartowania menedżera kolejek.

Zestawy danych dziennika archiwalnego są przydzielane dynamicznie. Po przydzieleniu nazwa zestawu danych jest rejestrowana w BSDS. Lista zestawów danych dziennika archiwalnego jest rozwijana po dodaniu archiwów i zawijana po osiągnięciu określonej przez użytkownika liczby pozycji. Maksymalna liczba pozycji wynosi 1000 dla pojedynczego rejestrowania archiwalnego i 2000 dla podwójnego rejestrowania.

Do usunięcia zestawów danych dziennika archiwalnego można użyć systemu zarządzania taśmami (produkt IBM MQ nie ma zautomatyzowanej metody). Dlatego informacje o zestawie danych dziennika archiwalnego mogą znajdować się w zestawie BSDS długo po usunięciu zestawu danych dziennika archiwalnego przez administratora systemu.

I odwrotnie, maksymalna liczba zestawów danych dziennika archiwalnego mogła zostać przekroczona, a dane z BSDS zostały usunięte na długo przed osiągnięciem daty ważności zestawu danych.

Za pomocą następującej komendy MQSC można określić obszar dziennika oraz nazwę aktywnego lub archiwalnego zestawu danych dziennika, w którym znajduje się najwcześniejszy adres RBA dziennika wymagany dla różnych typów odtwarzania nośnika lub menedżera kolejek:

```
DISPLAY USAGE TYPE(DATASET)
```

Jeśli moduł parametrów systemowych określa, że zestawy danych dziennika archiwalnego są katalogowane po przydzieleniu, BSDS wskazuje katalog ICF (Integrated Catalog Facility) dla informacji potrzebnych do późniejszego przydzielenia. W przeciwnym razie pozycje BSDS dla każdego woluminu rejestrują numer seryjny woluminu i informacje o jednostce, które są potrzebne do późniejszego przydzielania.

Wersja BSDS

Format BSDS różni się w zależności od wersji. Zwiększenie wersji BSDS umożliwia korzystanie z nowych funkcji. IBM MQ obsługuje następujące wersje BSDS:

Wersja 1

Obsługiwane przez wszystkie wersje systemu IBM MQ. BSDS w wersji 1 obsługuje 6-bajtowe wartości RBA dziennika.

Wersja 2

Obsługiwane przez IBM MQ 8.0 i nowsze. BSDS w wersji 2 umożliwia użycie 8-bajtowych wartości RBA dziennika i maksymalnie 310 zestawów danych w każdej aktywnej kopii dziennika.

V 9.3.0 Domyślnie włączone dla menedżerów kolejek utworzonych w wersji IBM MQ 9.3.0 i nowszych.

3

Obsługiwane przez IBM MQ 8.0 i nowsze. Zestaw danych BSDS jest automatycznie przekształcany do wersji 3, z wersji 2, po dodaniu więcej niż 31 zestawów danych do jednej z aktywnych kopii dziennika.

Wersję serwera BSDS można określić, uruchamiając program narzędziowy do drukowania map dziennika (CSQJU004). Aby przekształcić zestaw BSDS z wersji 1 do wersji 2, należy uruchomić program narzędziowy do konwersji BSDS (CSQJUCNV).

Więcej informacji na temat 6-bajtowych i 8-bajtowych logów RBA zawiera sekcja [“Większy dziennik-względny adres bajtowy”](#) na stronie 247 .

Archiwalne zestawy danych dziennika i kopie BSDS

Za każdym razem, gdy tworzony jest nowy zestaw danych dziennika archiwalnego, tworzona jest również kopia BSDS. Jeśli dziennik archiwizacji znajduje się na taśmie, BSDS jest pierwszym zestawem danych na pierwszym woluminie wyjściowym. Jeśli dziennik archiwalny znajduje się na DASD, BSDS jest oddzielnym zestawem danych.

Nazwy zestawów danych dziennika archiwalnego i kopii BSDS są takie same, z tą różnicą, że kwalifikator najniższego poziomu nazwy dziennika archiwalnego rozpoczyna się od litery A, a kopia BSDS rozpoczyna się od litery B, na przykład:

Nazwa dziennika archiwalnego

CSQ.ARCHLOG1.E00186.T2336229. A 0000001

Nazwa kopii BSDS

CSQ.ARCHLOG1.E00186.T2336229. B 0000001

Jeśli podczas kopiowania BSDS wystąpi błąd odczytu, kopia nie zostanie utworzona, zostanie wygenerowany komunikat [CSQJ125E](#) , a przenoszenie do nowego zestawu danych dziennika archiwalnego będzie kontynuowane bez kopii BSDS.

z/OS

Definicja systemu w systemie z/OS

Produkt IBM MQ for z/OS używa wielu domyślnych definicji obiektów i udostępnia przykładowy kod JCL do tworzenia tych domyślnych obiektów. W tym temacie opisano te domyślne obiekty i przykładowy kod JCL.

Ustawianie parametrów systemowych

W produkcie IBM MQ for z/OS moduł parametrów systemowych steruje środowiskami rejestrowania, archiwizowania, śledzenia i połączeń, które są używane przez produkt IBM MQ w jego działaniu. Parametry systemowe są określane przez trzy makra asemblera w następujący sposób:

CSQ6SYSP

Parametry systemowe, w tym ustawianie środowiska połączenia i śledzenia.

CSQ6LOGP

Parametry rejestrowania.

CSQ6ARVP

Parametry archiwum dziennika.

Domyślne moduły parametrów są dostarczane z produktem IBM MQ for z/OS. Jeśli nie zawierają one wartości, które mają być używane, można utworzyć własne moduły parametrów przy użyciu przykładowego dostarczonego z produktem IBM MQ. Przykład: `th1qua1.SCSQPROC (CSQ4ZPRM)`.

Niektóre parametry systemowe można zmieniać podczas działania menedżera kolejek. Zapoznaj się z informacjami na temat komend SET SYSTEM, SET LOG i SET ARCHIVE w sekcji [Komendy MQSC](#).

Więcej informacji na temat definiowania zawierają następujące tematy:

- [“Definiowanie obiektów systemowych dla systemu IBM MQ for z/OS” na stronie 251](#)
- [“Strojenie menedżera kolejek w systemie IBM MQ for z/OS” na stronie 256](#)
- [“Przykładowe definicje dostarczane z produktem IBM MQ for z/OS” na stronie 257](#)

Pojęcia pokrewne

[Dostosuj przykładowe zestawy danych wejściowych inicjowania](#)

[Źródła, z których można wydawać komendy MQSC i PCF w systemie IBM MQ for z/OS](#)

Zadania pokrewne

[administrowaniez/OS](#)

[Konfigurowanie klastrów](#)

[MonitorowanieIBM MQ](#)

Definiowanie obiektów systemowych dla systemu IBM MQ for z/OS

Produkt IBM MQ for z/OS wymaga dodatkowych predefiniowanych obiektów dla aplikacji publikowania/subskrypcji, klastra i sterowania kanałami oraz innych funkcji administrowania systemem.

Obiekty systemowe wymagane przez IBM MQ for z/OS można podzielić na następujące kategorie:

- [Obiekty publikowania/subskrypcji](#)
- [Domyślne obiekty systemowe](#)
- [Obiekty komend systemowych](#)
- [Obiekty administrowania systemem](#)
- [Kolejki kanałów](#)
- [Kolejki klastra](#)
- [Kolejki grup współużytkowania kolejek](#)
- [Klasy pamięci masowej](#)
- [Definiowanie kolejki niedostarczonych komunikatów obiektu systemowego](#)
- [Domyślna kolejka transmisji](#)
- [Kolejki wewnętrzne](#)
- [“Kolejka uwierzytelniania kanału” na stronie 255](#)

Obiekty publikowania/subskrypcji

Istnieje kilka obiektów systemowych, które należy zdefiniować przed użyciem aplikacji publikowania/subskrypcji w produkcie IBM MQ for z/OS. Przykładowe definicje są dostarczane wraz z produktem IBM MQ w celu ułatwienia definiowania tych obiektów. Przykłady te zostały opisane w sekcji [CSQ4INSG](#).

Aby używać publikowania/subskrybowania, należy zdefiniować następujące obiekty:

- Kolejka lokalna o nazwie SYSTEM.RETAINED.PUB.QUEUE, która jest używana do przechowywania kopii każdej zachowanej publikacji w menedżerze kolejek. Każda pełna nazwa tematu może mieć maksymalnie jedną zachowaną publikację przechowywaną w tej kolejce. Jeśli aplikacje będą korzystały z zachowanych publikacji w wielu różnych tematach lub jeśli zachowane komunikaty publikacji są duże, należy starannie zaplanować wymagania dotyczące pamięci dla tej kolejki, w tym przypisać ją do własnego zestawu stron, jeśli wymagania dotyczące pamięci dla tej kolejki są duże. Aby zwiększyć wydajność, należy zdefiniować tę kolejkę z indeksem typu MSGID (jak pokazano w dostarczonej definicji przykładowej kolejki).

- Kolejka lokalna o nazwie SYSTEM.DURABLE.SUBSCRIBER.QUEUE, która jest używana do przechowywania trwałej kopii trwałych subskrypcji w menedżerze kolejek. Aby zwiększyć wydajność, należy zdefiniować tę kolejkę przy użyciu typu indeksu CORRELID (jak pokazano w dostarczonej definicji przykładowej kolejki).
- Kolejka lokalna o nazwie SYSTEM.DURABLE.MODEL.QUEUE, która jest używana jako model dla zarządzanych trwałych subskrypcji.
- Kolejka lokalna o nazwie SYSTEM.INDURABLE.MODEL.QUEUE, która jest używana jako model dla zarządzanych nietrwałych subskrypcji.
- Lista nazw o nazwie SYSTEM.QPUBSUB.QUEUE.NAMELIST, który zawiera listę nazw kolejek monitorowanych przez umieszczony w kolejce interfejs publikowania/subskrypcji.
- Lista nazw o nazwie SYSTEM.QPUBSUB.SUBPOINT.NAMELIST, który zawiera listę obiektów tematu używanych przez umieszczony w kolejce interfejs publikowania/subskrypcji w celu dopasowania obiektów tematu do punktów subskrypcji.
- Temat o nazwie SYSTEM.BASE.TOPIC, który jest używany jako temat podstawowy do rozstrzygania atrybutów.
- Temat o nazwie SYSTEM.BROKER.DEFAULT.STREAM, który jest domyślnym strumieniem używanym przez umieszczony w kolejce interfejs publikowania/subskrypcji.
- Temat o nazwie SYSTEM.BROKER.DEFAULT.SUBPOINT, który jest domyślnym punktem subskrypcji RFH2 używanym przez umieszczony w kolejce interfejs publikowania/subskrypcji.
- Temat o nazwie SYSTEM.BROKER.ADMIN.STREAM, który jest strumieniem administracyjnym używanym przez umieszczony w kolejce interfejs publikowania/subskrypcji.
- Subskrypcja o nazwie SYSTEM.DEFAULT.SUB, który jest domyślnym obiektem subskrypcji używanym do udostępniania wartości domyślnych w komendach DEFINE SUB.

Domyślne obiekty systemowe

Domyślne obiekty systemowe są używane do udostępniania domyślnych atrybutów podczas definiowania obiektu i nie określają nazwy innego obiektu, na podstawie którego ma zostać utworzona definicja.

Nazwy domyślnych definicji obiektów systemowych zaczynają się od znaków "SYSTEM.DEFAULT" lub "SYSTEM.DEF. ". Na przykład domyślna systemowa kolejka lokalna nosi nazwę SYSTEM.DEFAULT.LOCAL.QUEUE.

Te obiekty definiują systemowe wartości domyślne dla atrybutów tych obiektów IBM MQ :

- Kolejki lokalne
- Kolejki modelowe
- Kolejki aliasowe
- Kolejki zdalne
- Procesy
- Listy nazw
- Kanały
- Klasy pamięci masowej
- Informacje uwierzytelniające

Kolejki współużytkowane są specjalnym typem kolejki lokalnej, dlatego podczas definiowania kolejki współużytkowanej definicja jest oparta na systemie SYSTEM.DEFAULT.LOCAL.QUEUE. Należy pamiętać o podaniu wartości dla nazwy struktury narzędzia CF, ponieważ nie jest ona określona w definicji domyślnej. Alternatywnie można zdefiniować własną domyślną definicję kolejki współużytkowanej, która będzie używana jako podstawa dla kolejek współużytkowanych, tak aby wszystkie dziedziczyły wymagane atrybuty. Należy pamiętać, że należy zdefiniować kolejkę współużytkowaną tylko w jednym menedżerze kolejek w grupie współużytkowania kolejek.

Obiekty komend systemowych

Nazwy obiektów komend systemowych zaczynają się od znaków SYSTEM.COMMAND. Należy zdefiniować te obiekty przed użyciem operacji IBM MQ i paneli sterujących do wydawania komend do podsystemu IBM MQ .

Istnieją dwa obiekty komend systemowych:

1. Kolejka wejściowa komend systemowych jest kolejką lokalną, w której komendy są umieszczane przed przetworzeniem przez procesor komend IBM MQ . Musi mieć nazwę SYSTEM.COMMAND.INPUT. Alias o nazwie SYSTEM.ADMIN.COMMAND.QUEUE powinien być również zdefiniowany, w celu zachowania zgodności z IBM MQ for Multiplatforms oraz w celu użycia przez IBM MQ Console i administrative REST API.
2. SYSTEM SYSTEM.COMMAND.REPLY.MODEL to kolejka modelowa, która definiuje kolejkę odpowiedzi komendy systemowej.

Istnieją dwa dodatkowe obiekty do użycia przez IBM MQ Explorer:

- SYSTEM SYSTEM.MQEXPLORER.REPLY.MODEL
- SYSTEM SYSTEM.ADMIN.SVRCONN

SYSTEM SYSTEM.REST.REPLY.QUEUE to kolejka odpowiedzi używana przez IBM MQ administrative REST API.

Komendy są zwykle wysyłane przy użyciu nietrwałych komunikatów, dlatego oba obiekty komend systemowych powinny mieć atrybut DEFPSIST (NO), aby aplikacje korzystające z nich (w tym dostarczone aplikacje, takie jak program narzędziowy i panele sterowania) domyślnie otrzymały komunikaty nietrwałe. W przypadku aplikacji, która używa trwałych komunikatów dla komend, należy ustawić atrybut DEFTYPE (PERMDYN) dla kolejki odpowiedzi, ponieważ komunikaty odpowiedzi dla takich komend są trwałe.

Obiekty administrowania systemem

Nazwy obiektów administracyjnych systemu zaczynają się od znaków SYSTEM.ADMIN.

Istnieją siedem obiektów administrowania systemem:

- SYSTEM SYSTEM.ADMIN.CHANNEL.EVENT
- SYSTEM SYSTEM.ADMIN.COMMAND.EVENT
- SYSTEM SYSTEM.ADMIN.CONFIG.EVENT
- SYSTEM SYSTEM.ADMIN.PERFM.EVENT
- SYSTEM SYSTEM.ADMIN.QMGR.EVENT
- SYSTEM SYSTEM.ADMIN.TRACE.ROUTE.QUEUE queue (kolejka)
- SYSTEM SYSTEM.ADMIN.ACTIVITY.QUEUE queue (kolejka)

Kolejki kanałów

Aby użyć rozproszonego kolejkowania, należy zdefiniować następujące obiekty:

- Kolejka lokalna o nazwie SYSTEM.CHANNEL.SYNCQ, który jest używany do obsługi numerów kolejnych i identyfikatorów logicznych jednostek pracy (LUWID) kanałów. Aby zwiększyć wydajność kanału, należy zdefiniować tę kolejkę z indeksem typu MSGID (jak pokazano w dostarczonej definicji przykładowej kolejki).
- Kolejka lokalna o nazwie SYSTEM.CHANNEL.INITQ, który jest używany dla komend kanału.

Nie można zdefiniować tych kolejek jako kolejek współużytkowanych.

Kolejki klastra

Aby używać klastrów IBM MQ , należy zdefiniować następujące obiekty:

- Kolejka lokalna o nazwie SYSTEM.CLUSTER.COMMAND.QUEUE, która jest używana do przekazywania zmian repozytorium między menedżerami kolejek. Komunikaty zapisane w tej kolejce zawierają aktualizacje danych repozytorium, które mają zostać zastosowane do lokalnej kopii repozytorium lub żądania danych repozytorium.
- Kolejka lokalna o nazwie SYSTEM.CLUSTER.REPOSITORY.QUEUE, która jest używana do przechowywania trwałej kopii repozytorium.
- Kolejka lokalna o nazwie SYSTEM.CLUSTER.TRANSMIT.QUEUE, która jest kolejką transmisji dla wszystkich miejsc docelowych w klastrze. Ze względu na wydajność należy zdefiniować tę kolejkę z indeksem typu CORRELID (jak pokazano w przykładowej definicji kolejki).

Te kolejki zwykle zawierają dużą liczbę komunikatów.

Nie można zdefiniować tych kolejek jako kolejek współużytkowanych.

Kolejki grupy współużytkowania kolejek

Aby używać kanałów współużytkowanych i kolejkowania wewnątrz grupy, należy zdefiniować następujące obiekty:

- Kolejka współużytkowana o nazwie SYSTEM.QSG.CHANNEL.SYNCQ, który jest używany do przechowywania informacji o synchronizacji dla kanałów współużytkowanych.
- Kolejka współużytkowana o nazwie SYSTEM.QSG.TRANSMIT.QUEUE, która jest używana jako kolejka transmisji dla kolejkowania wewnątrz grupy. Jeśli użytkownik pracuje w grupie współużytkowania kolejek, należy zdefiniować tę kolejkę, nawet jeśli nie jest używana kolejkowanie wewnątrz grupy.

Klasy pamięci masowej

Zaleca się zdefiniowanie następujących sześciu klas pamięci masowej. Należy zdefiniować cztery z nich, ponieważ są one wymagane przez produkt IBM MQ. Inne definicje klas pamięci masowej są zalecane, ponieważ są używane w przykładowych definicjach kolejek.

DOMYŚLNE (wymagane)

Ta klasa pamięci masowej jest używana dla wszystkich kolejek komunikatów, które nie mają newralgicznego znaczenia dla wydajności i nie mieszczą się w żadnej innej klasie pamięci masowej. Jest to również dostarczana domyślna klasa pamięci, jeśli nie zostanie podana podczas definiowania kolejki.

NODEFINE (wymagane)

Ta klasa pamięci jest używana, jeśli klasa pamięci określona podczas definiowania kolejki nie jest zdefiniowana.

REMOTE (wymagane)

Ta klasa pamięci masowej jest używana głównie dla kolejek transmisji, czyli kolejek związanych z systemem z krótkotrwałą komunikatami o newralgicznym znaczeniu dla wydajności.

SYSNLGLV,

Ta klasa pamięci masowej jest używana dla długotrwałych komunikatów o newralgicznym znaczeniu dla wydajności.

SYSTEM (wymagane)

Ta klasa pamięci masowej jest używana dla newralgicznych kolejek komunikatów związanych z wydajnością, na przykład dla systemu SYSTEM.CHANNEL.SYNQ i system SYSTEM.CLUSTER.* .

SYSVOLAT

Ta klasa pamięci masowej jest używana dla krótkotrwałego komunikatu o newralgicznym znaczeniu dla wydajności.

W razie potrzeby można zmodyfikować ich atrybuty i dodać inne definicje klas pamięci masowej.

Definiowanie kolejki niedostarczonych komunikatów obiektu systemowego

Jeśli miejsce docelowe komunikatów jest niepoprawne, używana jest kolejka niedostarczonych komunikatów. Program IBM MQ umieszcza takie komunikaty w lokalnej kolejce nazywanej kolejką niedostarczonych komunikatów. Chociaż kolejka niedostarczonych komunikatów nie jest obowiązkowa, należy ją uważać za niezbędną, szczególnie w przypadku korzystania z rozproszonego kolejkowania lub jednego z mostów IBM MQ .

Nie definiuj kolejki niedostarczonych komunikatów jako kolejki współużytkowanej. Umieszczenie w kolejce lokalnej w jednym menedżerze kolejek może zostać umieszczone w kolejce niedostarczonych komunikatów. Jeśli kolejka niedostarczonych komunikatów była kolejką współużytkowaną, program obsługi kolejki niedostarczonych komunikatów w innym systemie mógł przetworzyć komunikat i umieścić go w kolejce o tej samej nazwie, ale ponieważ znajduje się on w innym menedżerze kolejek, może to być błędna kolejka lub mieć inny profil zabezpieczeń. Jeśli kolejka nie istnieje, jej ponowne przetworzenie nie powiedzie się.

Jeśli użytkownik zdecyduje się na zdefiniowanie kolejki niedostarczonych komunikatów, musi również podać nazwę menedżera kolejek. W tym celu należy użyć komendy ALTER QMGR DEADQ (*nazwa_kolejki*). Więcej informacji na ten temat zawiera sekcja [Wyświetlanie i zmienianie atrybutów menedżera kolejek](#).

Domyślna kolejka transmisji

Domyślna kolejka transmisji jest używana, gdy nie jest dostępna żadna inna odpowiednia kolejka transmisji do wysyłania komunikatów do innego menedżera kolejek. Jeśli zostanie zdefiniowana domyślna kolejka transmisji, należy również zdefiniować kanał do obsługi kolejki. Jeśli kanał nie zostanie zdefiniowany, komunikaty umieszczane w domyślnej kolejce transmisji nie są przesyłane do zdalnego menedżera kolejek i pozostają w kolejce.

Jeśli zostanie zdefiniowana domyślna kolejka transmisji, należy również podać nazwę menedżera kolejek. W tym celu należy użyć komendy ALTER QMGR.

Kolejki wewnętrzne

• Kolejka danych oczekujących

- Kolejka zdefiniowana do użytku wewnętrznego, SYSTEM.PENDING.DATA.QUEUE: Obsługuje używanie trwałych subskrypcji w środowisku publikowania/subskrypcji produktu JMS .

• JMS 2.0 kolejka pomostowa opóźnienia dostarczania

- Jeśli używana jest funkcja opóźnienia dostarczania udostępniana przez produkt JMS 2.0 , wówczas wewnętrzna kolejka pomostowa SYSTEM.DDELAY.LOCAL.QUEUE, musi być zdefiniowana. Ta kolejka jest używana przez menedżer kolejek do tymczasowego przechowywania komunikatów wysłanych z niezerowym opóźnieniem dostarczenia do czasu zakończenia opóźnienia dostarczenia i umieszczenia komunikatu w miejscu docelowym. Przykładowa definicja tej kolejki jest przekształcona w komentarz w sekcji CSQ4INSG.
- Podczas definiowania systemu SYSTEM.DDELAY.LOCAL.QUEUE queue, należy ustawić atrybuty STGCLASS, MAXMSGL i MAXDEPTH dla przewidywanej liczby komunikatów, które będą wysyłane z opóźnieniem dostarczania. Dodatkowo podczas definiowania systemu SYSTEM.DDELAY.LOCAL.QUEUE upewnij się, że tylko menedżer kolejek może umieszczać komunikaty w tej kolejce. Należy zachować ostrożność, aby żaden identyfikator użytkownika nie miał uprawnień do umieszczania komunikatów w tej kolejce.

Kolejka uwierzytelniania kanału

Do wewnętrznego wykorzystania uwierzytelniania kanału SYSTEM.CHLAUTH.DATA.QUEUE jest wymagana. Przykładowe definicje są dostarczane wraz z produktem IBM MQ w celu ułatwienia definiowania tych obiektów. Ten przykład jest opisany w sekcji CSQ4INSA, która również definiuje niektóre reguły domyślne.

Strojenie menedżera kolejek w systemie IBM MQ for z/OS

Istnieje kilka prostych kroków, które można wykonać, aby upewnić się, że menedżer kolejek jest dostrojony w celu uniknięcia podstawowych problemów z wydajnością.

Istnieje kilka sposobów zwiększenia wydajności menedżera kolejek, które są kontrolowane przez atrybuty menedżera kolejek ustawione za pomocą komendy ALTER QMGR. Ta sekcja zawiera informacje o tym, w jaki sposób można to zrobić, ustawiając maksymalną liczbę komunikatów dozwolonych w menedżerze kolejek lub wykonując czynności konserwujące w menedżerze kolejek. Więcej informacji na temat wydajności i strojenia zawiera sekcja IBM MQ SupportPac [MP16 - IBM MQ for z/OS -planowanie i strojenie mocy obliczeniowej](#) .

Punkty synchronizacji

Jedną z ról menedżera kolejek jest sterowanie punktem synchronizacji w aplikacji. Aplikacja tworzy jednostkę pracy zawierającą dowolną liczbę wywołań MQPUT lub MQGET zakończonych wywołaniem MQCMIT.

Wraz ze wzrostem liczby wywołań MQPUT lub MQGET w zasięgu jednego MQCMIT, koszt zatwierdzenia znacząco wzrasta. Ogólnie aplikacje powinny być zaprojektowane w taki sposób, aby w pojedynczym punkcie synchronizacji nie były używane wywołania MQPUT/MQGET z dużą liczbą komunikatów.

Liczbę komunikatów w pojedynczym punkcie synchronizacji można ograniczyć administracyjnie za pomocą atrybutu menedżera kolejek MAXUMSGS. Jeśli aplikacja przekracza ten limit, odbiera wywołanie MQRC_SYNCPOINT_LIMIT_REACHED w przypadku wywołania MQPUT, MQPUT1 lub MQGET, które przekracza ten limit. Aplikacja powinna następnie odpowiednio wydać komendę MQCMIT lub MQBACK.

Wartością domyślną parametru MAXUMSGS jest 10000. Tę wartość można obniżyć, aby wymusić niższy limit, co może również pomóc w zabezpieczeniu przed zapętlaniem się aplikacji. Przed redukcją wartości MAXUMSGS należy zapoznać się z istniejącymi aplikacjami, aby upewnić się, że nie przekraczają one limitu, lub zaakceptować kod powrotu MQRC_SYNCPOINT_LIMIT_REACHED.

Przedawnione komunikaty

Komunikaty, które utraciły ważność, są usuwane przez następne odpowiednie wywołanie MQGET. Jeśli jednak takie wywołanie nie wystąpi, komunikaty, które utraciły ważność, nie są usuwane, a w przypadku niektórych kolejek, w szczególności tych, w których komunikaty są pobierane przez parametry MessageId, CorrelId lub GroupId, a kolejka jest indeksowana pod kątem wydajności, może gromadzić się wiele komunikatów, które utraciły ważność. Menedżer kolejek może okresowo skanować wszystkie kolejki w poszukiwaniu komunikatów, które utraciły ważność, a następnie je usuwać. Można wybrać częstotliwość skanowania, jeśli w ogóle. Można to zrobić na dwa sposoby:

Jawne żądanie

Można kontrolować, które kolejki są skanowane i kiedy. Wydać komendę REFRESH QMGR TYPE (utrata ważności), podając kolejkę lub kolejki, które mają być skanowane.

Skanowanie okresowe

Odstęp czasu utraty ważności w obiekcie menedżera kolejek można określić za pomocą atrybutu EXPRYINT. Menedżer kolejek przechowuje informacje o komunikatach, które utraciły ważność w każdej kolejce, i wie, o której godzinie warto szukać komunikatów, które utraciły ważność. Za każdym razem, gdy zostanie osiągnięty odstęp czasu EXPRYINT, menedżer kolejek wyszukuje kolejki kandydujące, które są warte skanowania w poszukiwaniu komunikatów, które utraciły ważność,

i skanuje tylko te kolejki, które uzna za wartościowe. Nie skanuje wszystkich kolejek. Pozwala to uniknąć marnowania czasu procesora na niepotrzebne skanowania.

Kolejki współużytkowane są skanowane tylko przez jednego menedżera kolejek w grupie współużytkowania kolejek. Zwykle skanowanie jest wykonywane przez pierwszego menedżera kolejek, który ma zostać zrestartowany, lub pierwszego, który ma ustawiony parametr EXPRYINT.

Uwaga: Należy ustawić tę samą wartość EXPRYINT dla wszystkich menedżerów kolejek w grupie współużytkowania kolejek.

z/OS Przykładowe definicje dostarczane z produktem IBM MQ for z/OS

Ten temat zawiera informacje uzupełniające dotyczące przykładowego kodu JCL i kodu dostarczonego z produktem IBM MQ for z/OS.

Poniższe przykładowe definicje są dostarczane z produktem IBM MQ w bibliotece thlqual.SCSQPROC . Można ich używać do definiowania obiektów systemowych oraz do dostosowywania własnych obiektów. Niektóre z nich można dołączyć do zestawów danych wejściowych inicjowania (opisanych w sekcji [Komendy inicjowania](#)).

<i>Tabela 22. Przykładowe definicje obiektów systemowych w systemie IBM MQ</i>	
zestaw danych wejściowych inicjowania	Przykładowa nazwa
CSQINP1	CSQ4INP1 CSQ4INPR
CSQINP2	CSQ4INSA CSQ4INYS ¹ CSQ4INSX CSQ4INSG CSQ4INSR CSQ4INSS CSQ4INSJ CSQ4INSM CSQ4INYG CSQ4INYR CSQ4INYC CSQ4INYD CSQ4INSC
CSQINPT (obszar CSQINPT)	CSQ4INST CSQ4INYT
Inne	CSQ4DISP CSQ4INPX CSQ4IVPQ CSQ4IVPG CSQ4MSTR CSQ4MSRR CSQ4QMIN

Uwaga:

1. Kolejność tych przykładowych definicji jest istotna: błąd występuje, jeśli INYS, INSX i INSG są uporządkowane niepoprawnie.

Przykłady komendy CSQINP1

Należy użyć przykładowego zestawu danych CSQINP1 `thlqual.SCSQPROC(CSQ4INP1)`, jeśli dla każdej klasy komunikatu używany jest jeden zestaw stron lub `thlqual.SCSQPROC(CSQ4INPR)`, jeśli dla głównych klas komunikatu używanych jest wiele zestawów stron. Zawiera on definicje pul buforów, zestaw stron dla powiązań puli buforów oraz komendę ALTER SECURITY. Dołącz przykład do konkatenacji CSQINP1 procedury uruchomionego zadania menedżera kolejek.

Przykłady komendy CSQINP2

Przykład obiektu systemowego CSQ4INSG

Przykładowy zestaw danych CSQINP2 `thlqual.SCSQPROC(CSQ4INSG)` zawiera definicje następujących obiektów systemowych do ogólnego użytku:

- Domyślne obiekty systemowe
- Obiekty komend systemowych
- Obiekty administrowania systemem
- Inne obiekty do użytku systemowego

W tym przykładzie należy zdefiniować obiekty, ale należy to zrobić tylko raz przy pierwszym uruchomieniu podsystemu. Najlepszym sposobem na to jest dołączenie definicji do zestawu danych CSQINP2. Są one utrzymywane po zamknięciu i restarcie menedżera kolejek. Nie wolno zmieniać nazw obiektów, ale w razie potrzeby można zmienić ich atrybuty.

Jeśli spełnione są następujące warunki, w systemie SYSTEM.DURABLE.SUBSCRIBER.QUEUE (nawet jeśli subskrypcja publikowania nie jest aktywna):

- Atrybut PSMODE menedżera kolejek jest ustawiony na wartość DISABLED.
- Przykładowy obiekt CSQ4INST instrukcja `DEFINE SUB('SYSTEM.DEFAULT.SUB')` jest obecny.

Aby tego uniknąć, należy usunąć lub przekształcić w komentarz instrukcję `DEFINE SUB('SYSTEM.DEFAULT.SUB')`.

Kolejka pomostowa opóźnienia dostarczania JMS 2.0, SYSTEM.DDELAY.LOCAL.QUEUE musi być zdefiniowana tylko wtedy, gdy używane jest opóźnienie dostarczania JMS 2.0. Domyślnie definicja kolejki jest przekształcona w komentarz, który można w razie potrzeby usunąć z komentarza.

Obiekt systemowy i przykład uwierzytelniania CSQ4INSA

Przykładowy zestaw danych CSQINP2 `thlqual.SCSQPROC(CSQ4INSA)` zawiera definicję kolejki systemu uwierzytelniania kanału. Ta kolejka zawiera rekordy uwierzytelniania kanału. Zawiera również domyślne reguły uwierzytelniania kanału.

Należy zdefiniować obiekty w tym przykładzie, jeśli parametr CHLAUTH ma wartość ENABLED w menedżerze kolejek i mają być uruchamiane kanały lub jeśli ma być ustawiony lub wyświetlany rekord CHLAUTH. Należy je zdefiniować tylko raz przy pierwszym uruchomieniu podsystemu. Najlepszym sposobem na to jest dołączenie definicji do zestawu danych CSQINP2. Są one utrzymywane podczas zamykania i restartowania menedżera kolejek. Nie można zmieniać nazwy kolejki.

Przykład obiektu systemowego CSQ4INSS

Jeśli używane są grupy współużytkowania kolejek, można zdefiniować dodatkowe obiekty systemowe.

Przykładowy zestaw danych `thlqual.SCSQPROC(CSQ4INSS)` zawiera przykładowe komendy do użycia ze strukturami CF oraz zestaw definicji dla obiektów systemowych wymaganych dla kanałów współużytkowanych i kolejkowania wewnątrz grupy.

Nie można użyć tego przykładu w stanie, w jakim się znajduje ("as is"); należy go dostosować przed użyciem. Następnie można dołączyć ten element do konkatenacji CSQINP2 DD procedury

uruchamiania menedżera kolejek lub użyć go jako danych wejściowych dla funkcji COMMAND programu narzędziowego CSQUTIL w celu wydania wymaganych komend.

Definiując grupę lub obiekty współużytkowane, należy je dołączyć do konkatenacji definicji danych CSQINP2 tylko dla jednego menedżera kolejek w grupie współużytkowania kolejek.

Przykład obiektu systemowego CSQ4INSX

Jeśli używane jest rozproszone kolejkowanie i łączenie w klastry, należy zdefiniować dodatkowe obiekty systemowe.

Przykładowy zestaw danych thlqual.SCSQPROC(CSQ4INSX) zawiera wymagane definicje kolejek. Można dołączyć ten element do konkatenacji definicji danych CSQINP2 procedury uruchamiania menedżera kolejek lub użyć go jako danych wejściowych dla funkcji COMMAND w programie narzędziowym CSQUTIL w celu wydania wymaganych komend DEFINE.

Istnieją dwa typy definicji obiektów:

- SYSTEM SYSTEM.CHANNEL.xx, wymagany dla dowolnego rozproszonego kolejkowania
- SYSTEM SYSTEM.CLUSTER.xx, wymagany do grupowania

Przykład obiektu CSQ4INSJ systemowego JMS

Definiuje kolejki używane w domenie publikowania/subskrypcji JMS .

Przykład obiektu systemowego CSQ4INSM

Jeśli używane są zaawansowane zabezpieczenia komunikatów, należy zdefiniować dodatkowe obiekty systemowe. Przykładowy zestaw danych thlqual.SCSQPROC(CSQ4INSM) zawiera wymagane definicje kolejek.

Przykład obiektu CSQ4INSR

Definiuje kolejki używane przez produkt WebSphere Application Server i brokery.

Przykład obiektu CSQ4INYD

Jeśli używane jest kolejkowanie rozproszone i konieczne jest skonfigurowanie własnych kolejek, procesów i kanałów.

Przykładowy zestaw danych thlqual.SCSQPROC(CSQ4INYD) zawiera przykładowe definicje, których można użyć do dostosowania rozproszonych obiektów kolejkowania. Obejmuje ona:

- Zestaw definicji dla wysyłającego zakończenia
- Zestaw definicji dla odbierającego zakończenia
- Zestaw definicji do używania klientów

Nie można użyć tego przykładu w stanie, w jakim się znajduje ("as is")-należy go dostosować przed użyciem. Następnie można dołączyć ten element do konkatenacji CSQINP2 DD procedury uruchamiania menedżera kolejek lub użyć go jako danych wejściowych dla funkcji COMMAND programu narzędziowego CSQUTIL w celu wydania wymaganych komend DEFINE. (Jest to preferowane, ponieważ oznacza, że nie ma konieczności ponownego definiowania tych obiektów przy każdym restarcie menedżera kolejek).

Przykład obiektu CSQ4INYC

Jeśli używane jest łączenie w klastry, definicje równoważne definicjom kanałów i definicjom kolejek zdalnych rozproszonego kolejkowania są tworzone automatycznie w razie potrzeby. Jednak niektóre definicje kanału ręcznego są wymagane-kanał odbiorczy klastra dla klastra i definicja nadajnika klastra dla co najmniej jednego menedżera kolejek repozytorium klastra.

Przykładowy zestaw danych: thlqual.SCSQPROC(CSQ4INYC) zawiera następujące przykładowe definicje, których można użyć do dostosowywania obiektów grupowania:

- Definicje menedżera kolejek

- Definicje kanału odbiorczego
- Definicje kanału nadawczego
- Definicje kolejek klastra
- Definicje list klastrów

Nie można użyć tego przykładu w stanie, w jakim się znajduje ("as is")-należy go dostosować przed użyciem. Następnie można dołączyć ten element do konkatenacji CSQINP2 DD procedury uruchamiania menedżera kolejek lub użyć go jako danych wejściowych dla funkcji COMMAND programu narzędziowego CSQUTIL w celu wydania wymaganych komend DEFINE. Jest to preferowane, ponieważ oznacza, że nie ma konieczności ponownego definiowania tych obiektów przy każdym restarcie programu IBM MQ.

Przykład obiektu CSQ4INYG

Przykładowy zestaw danych: thlqual.SCSQPROC(CSQ4INYG) zawiera następujące przykładowe definicje, których można użyć do dostosowywania własnych obiektów do ogólnego użytku:

- Kolejka niedostarczonych komunikatów
- Domyślna kolejka transmisji
- Obiekty adaptera CICS

Nie można użyć tego przykładu w stanie, w jakim się znajduje ("as is")-należy go dostosować przed użyciem. Następnie można dołączyć ten element do konkatenacji CSQINP2 DD procedury uruchamiania menedżera kolejek lub użyć go jako danych wejściowych dla funkcji COMMAND programu narzędziowego CSQUTIL w celu wydania wymaganych komend DEFINE. Jest to preferowane, ponieważ oznacza, że nie ma konieczności ponownego definiowania tych obiektów przy każdym restarcie programu IBM MQ.

Oprócz przykładowych definicji w tym miejscu można użyć definicji obiektów systemowych jako podstawy własnych definicji zasobów. Na przykład można utworzyć działającą kopię systemu SYSTEM.DEFAULT.LOCAL.QUEUE i nadać jej nazwę MY.DEFAULT.LOCAL.QUEUE. W razie potrzeby można zmienić dowolne parametry w tej kopii. Następnie można wydać komendę DEFINE niezależnie od wybranej metody, pod warunkiem, że użytkownik ma uprawnienia do tworzenia zasobów tego typu.

Domyślna kolejka transmisji

Przed podjęciem decyzji o definiowaniu domyślnej kolejki transmisji należy zapoznać się z opisem w sekcji [Domyślna kolejka transmisji](#).

- Aby zdefiniować domyślną kolejkę transmisji, należy również zdefiniować kanał, który ma ją obsługiwać.
- Jeśli nie chcesz go definiować, pamiętaj, aby usunąć instrukcję DEFEXMITQ z komendy ALTER QMGR w przykładzie.

Obiekty adaptera CICS

Przykład definiuje kolejkę inicjującą o nazwie CICS01.INITQ. Ta kolejka jest używana przez dostarczoną przez IBM MQ transakcję CKTI. Można zmienić nazwę tej kolejki, ale musi ona być zgodna z nazwą określoną w tabeli inicjowania systemu (SIT) CICS lub przestonieniu SYSIN w instrukcji INITPARM.

Przykłady obiektów CSQ4INYS/CSQ4INYR

Definicje klas pamięci masowej do użycia:

- jeden zestaw stron dla każdej klasy komunikatu
- wiele zestawów stron dla głównych klas komunikatów

Na przykład SYSTEM.COMMAND.INPUT używa STGCLASS ('SYSVOLAT') i SYSTEM.CLUSTER.TRANSMIT.QUEUE używa STGCLASS ('REMOTE'). W tabeli CSQ4INYSobie te klasy pamięci masowej używają tego samego zestawu stron. W przypadku komendy CSQ4INYRte klasy

pamięci masowej używają różnych zestawów stron, aby zmniejszyć wpływ na zapewnianie kolejki transmisji.

CSQINPT, próbki

CSQ4INST

Przykładowy zestaw danych: thlqual.SCSQPROC(CSQ4INST) zawiera definicję domyślnej subskrypcji systemu.

W tym przykładzie należy zdefiniować obiekt, ale należy to zrobić tylko raz przy pierwszym uruchomieniu mechanizmu publikowania/subskrybowania. Najlepszym sposobem na to jest dołączenie definicji do zestawu danych CSQINPT. Jest on utrzymywany między zamknięciem i restartem menedżera kolejek. Nie można zmienić nazwy obiektu, ale w razie potrzeby można zmienić ich atrybuty.

CSQ4INYT

Przykładowy zestaw danych: thlqual.SCSQPROC(CSQ4INYT) zawiera zestaw komend, które mogą być uruchamiane po uruchomieniu mechanizmu publikowania/subskrypcji. W tym przykładzie wyświetlane są informacje o temacie i subskrypcji.

Inne

Przykład ekranu CSQ4DISP

Przykładowy zestaw danych: thlqual.SCSQPROC(CSQ4DISP) zawiera zestaw ogólnych komend DISPLAY, które wyświetlają wszystkie zdefiniowane zasoby w menedżerze kolejek. Obejmuje to definicje wszystkich obiektów IBM MQ oraz definicje, takie jak klasy pamięci i śledzenie. Komendy te mogą generować dużą ilość danych wyjściowych. Tego przykładu można użyć w zestawie danych CSQINP2 lub jako danych wejściowych dla funkcji COMMAND programu narzędziowego CSQUTIL.

Przykład CSQ4INPX

Przykładowy zestaw danych: thlqual.SCSQPROC(CSQ4INPX) zawiera zestaw komend, które mogą być wykonywane przy każdym uruchomieniu inicjatora kanału. Przed użyciem tego przykładu należy go dostosować. Następnie można go dołączyć do zestawu danych CSQINPX dla inicjatora kanału.

Przykłady CSQ4IVPQ i CSQ4IVPG

Przykładowe zestawy danych: thlqual.SCSQPROC(CSQ4IVPQ) i thlqual.SCSQPROC(CSQ4IVPG) zawierają zestawy komend DEFINE, które są wymagane do uruchomienia programów weryfikujących instalację (IVP).

Przykłady te można dołączyć do zestawu danych CSQINP2. Po pomyślnym uruchomieniu procesów IVP nie trzeba uruchamiać ich ponownie przy każdym restarcie menedżera kolejek. Dlatego nie trzeba trwale przechowywać tych przykładów w konkatenacji CSQINP2.

Przykłady CSQ4MSTR i CSQ4MSRR

Oto przykładowe procedury uruchomionych zadań dla menedżera kolejek: thlqual.SCSQPROC(CSQ4MSTR) i thlqual.SCSQPROC(CSQ4MSRR).

CSQ4MSRR używa CSQ4INYR w konkatenacji CSQINP2, dzięki czemu ważne kolejki są rozmieszczane w różnych zestawach stron.

Komentarze można usunąć, aby w razie potrzeby można było użyć karty CSQMINI dla nowo utworzonych menedżerów kolejek.

Przykład CSQ4QMIN

Przykładowy zestaw danych QMINI thlqual.SCSQPROC(CSQ4QMIN).

Szczegółowe informacje na temat zestawu danych QMINI i sekcji **TransportSecurity** zawiera sekcja [Zestaw danych QMINI](#).

Odtwarzanie i restartowanie w systemie z/OS

Skorzystaj z odsyłaczy w tym temacie, aby dowiedzieć się więcej o funkcjach produktu IBM MQ for z/OS służących do restartowania i odtwarzania.

Produkt IBM MQ for z/OS oferuje stabilne funkcje restartowania i odtwarzania. Informacje na temat sposobu odzyskiwania menedżera kolejek po jego zatrzymaniu oraz na temat tego, co się dzieje po jego zrestartowaniu, zawierają następujące podtematy:

- [“Sposób wprowadzania zmian w danych w produkcie IBM MQ for z/OS” na stronie 262](#)
- [“Sposób zachowania spójności w produkcie IBM MQ for z/OS” na stronie 264](#)
- [“Co się dzieje podczas kończenia w programie IBM MQ for z/OS” na stronie 266](#)
- [“Co się dzieje podczas restartowania i odtwarzania w produkcie IBM MQ for z/OS” na stronie 268](#)
- [“Sposób rozstrzygnięcia wątpliwych jednostek odzyskiwania” na stronie 270](#)
- [“Odtwarzanie kolejki współużytkowanej” na stronie 272](#)

Pojęcia pokrewne

[z/OS](#) [IBM MQ for z/OS działania odtwarzania](#)

Zadania pokrewne

[Planowanie składowania i odtwarzania](#)

[z/OS](#) [administrowanie z/OS](#)

Odsyłacze pokrewne

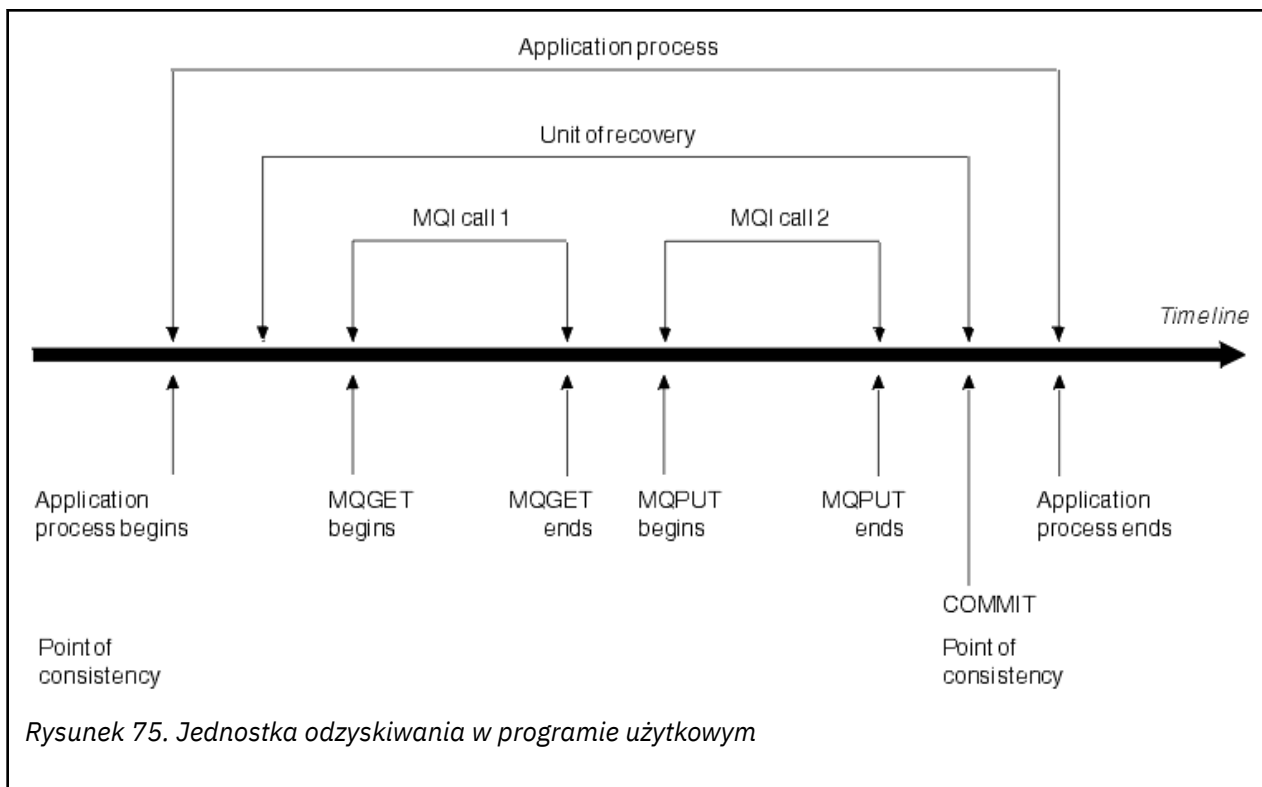
[z/OS](#) [Komunikaty dla IBM MQ for z/OS](#)

[z/OS](#) Sposób wprowadzania zmian w danych w produkcie IBM MQ for z/OS

Program IBM MQ for z/OS musi współdziałać z innymi podsystemami, aby zachować spójność wszystkich danych. Ten temat zawiera informacje o *jednostkach odzyskiwania*, o tym, czym one są i jak są używane w *wycofanych*.

Jednostki odzyskiwania

Jednostka odzyskiwania to przetwarzanie wykonywane przez pojedynczy menedżer kolejek dla aplikacji, które zmienia dane programu IBM MQ z jednego punktu spójności na inny. *Punkt spójności* - nazywany również *punktem synchronizacji* lub *punktem zatwierdzenia* - jest punktem w czasie, w którym wszystkie dane odtwarzalne, do których uzyskuje dostęp aplikacja, są spójne.



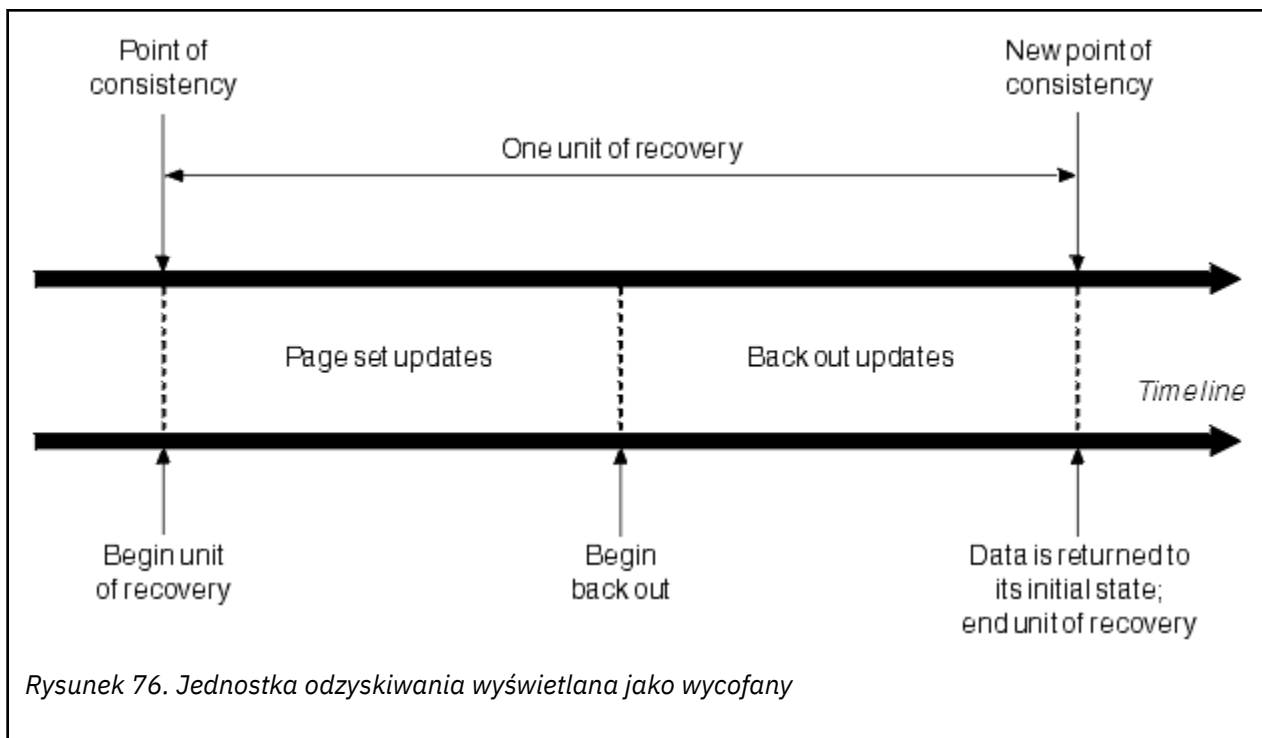
Jednostka odzyskiwania rozpoczyna się od pierwszej zmiany danych po rozpoczęciu programu lub po poprzednim punkcie spójności; kończy się na późniejszym punkcie spójności. Rysunek 75 na stronie 263 przedstawia relację między jednostkami odtwarzania, punktem spójności i programem użytkowym. W tym przykładzie aplikacja wprowadza zmiany w kolejkach za pomocą wywołań MQI 1 i 2. Aplikacja może zawierać więcej niż jedną jednostkę odzyskiwania lub tylko jedną jednostkę. Jednak każda pełna jednostka odzyskiwania kończy się w punkcie zatwierdzenia.

Na przykład transakcja bankowa przekazuje środki z jednego rachunku na inny. Najpierw program odejmuje kwotę od pierwszego konta, konta A. Następnie dodaje kwotę do drugiego konta, B. Po odjęciu kwoty od A, dwa konta są niespójne i IBM MQ nie może zatwierdzić. Stają się one spójne, gdy kwota jest dodawana do rachunku B. Po wykonaniu obu kroków program może ogłosić punkt spójności poprzez zatwierdzenie, co spowoduje, że zmiany będą widoczne dla innych aplikacji.

Normalne zakończenie aplikacji automatycznie powoduje powstanie punktu spójności. Niektóre żądania programów w programach CICS i IMS również powodują powstanie punktu spójności, na przykład EXEC CICS SYNCPOINT.

Wycofywanie pracy

Jeśli w jednostce odzyskiwania wystąpi błąd, program IBM MQ usuwa wszelkie zmiany danych, przywracając ich stan na początku jednostki odzyskiwania, co oznacza, że program IBM MQ wycofuje pracę. Zdarzenia są wyświetlane w sekcji [Rysunek 76](#) na stronie 264.



z/OS Sposób zachowania spójności w produkcie IBM MQ for z/OS

Dane w systemie IBM MQ for z/OS muszą być spójne z zadaniami wsadowymi, CICS, IMS lub TSO. Każda zmiana danych w jednym z nich musi być zgodna ze zmianą w drugim.

Zanim jeden system zatwierdzi zmienione dane, musi wiedzieć, że drugi system może wprowadzić odpowiednią zmianę. Tak więc, systemy muszą się komunikować.

Podczas *zatwierdzania dwufazowego* (na przykład w systemie CICS) jeden podsystem koordynuje proces. Ten podsystem jest nazywany *koordynatorem*, a drugi jest *uczestnikiem*. CICS lub IMS jest zawsze koordynatorem w interakcjach z IBM MQ, a IBM MQ jest zawsze uczestnikiem. W środowisku wsadowym lub TSO produkt IBM MQ może uczestniczyć w protokołach zatwierdzania dwufazowego koordynowanych przez usługi z/OS RRS.

Podczas *zatwierdzania jednofazowego* (na przykład w TSO lub w zadaniu wsadowym), IBM MQ jest zawsze koordynatorem w interakcjach i w pełni steruje procesem zatwierdzania.

W środowisku WebSphere Application Server semantyka obiektu sesji JMS określa, czy używana jest koordynacja zatwierdzania jednofazowego, czy dwufazowego.

Spójność z CICS lub IMS

Połączenie między systemami IBM MQ i CICS lub IMS obsługuje następujące protokoły punktu synchronizacji:

- Zatwierdzanie dwufazowe-dla transakcji, które aktualizują zasoby należące do więcej niż jednego menedżera zasobów.

Jest to standardowy protokół rozproszonego punktu synchronizacji. Wiąże się to z większym rejestrowaniem i przepływami komunikatów niż zatwierdzanie jednofazowe.

- Zatwierdzanie jednofazowe-dla transakcji, które aktualizują zasoby należące do pojedynczego menedżera zasobów (IBM MQ).

Ten protokół jest zoptymalizowany pod kątem rejestrowania i przepływów komunikatów.

- Pominięcie punktu synchronizacji-dla transakcji, które dotyczą produktu IBM MQ , ale nie są wykonywane żadne działania w menedżerze kolejek wymagające punktu synchronizacji (na przykład przeglądanie kolejki).

W każdym przypadku produkt CICS lub IMS działa jako menedżer punktów synchronizacji.

Etapy zatwierdzania dwufazowego używane przez produkt IBM MQ do komunikacji z systemem CICS lub IMS są następujące:

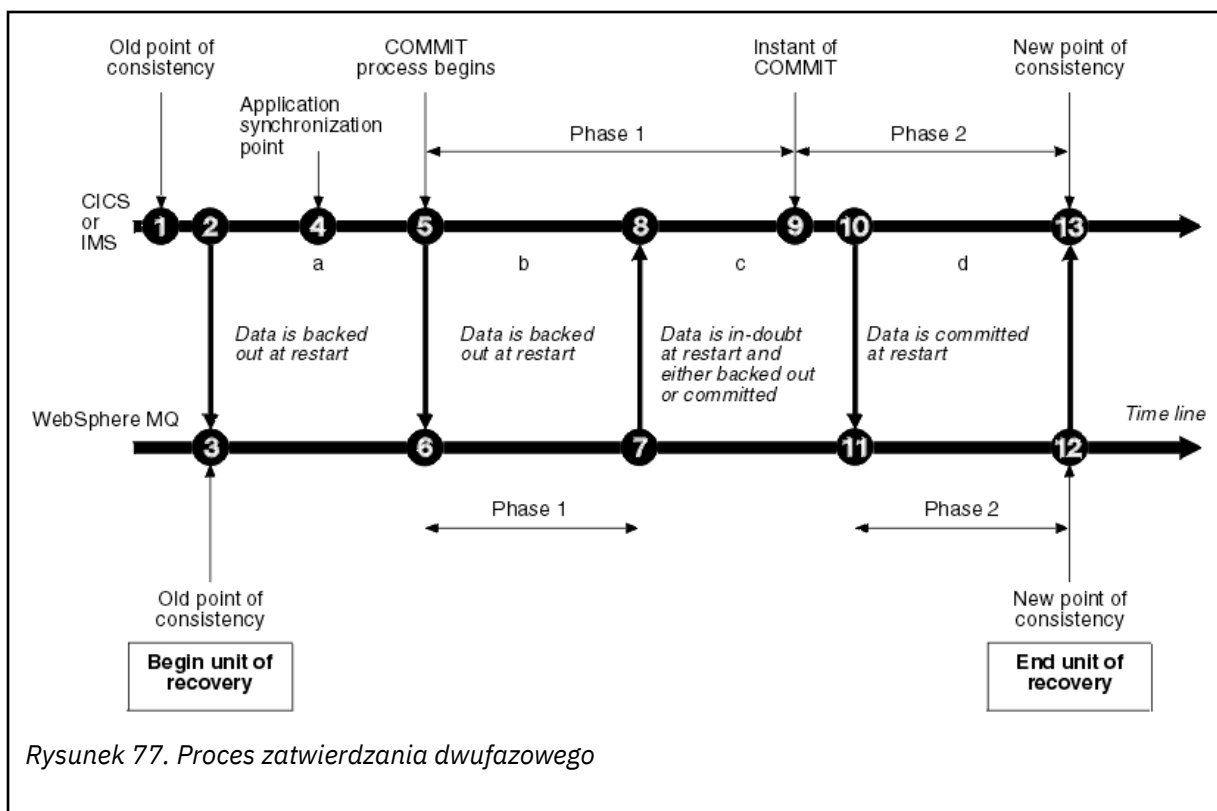
1. W fazie 1 każdy system określa niezależnie, czy zapisał wystarczającą ilość informacji o odtwarzaniu w swoim dzienniku i może zatwierdzić swoją pracę.

Po zakończeniu fazy systemy komunikują się ze sobą. Jeśli się zgodzą, każda z nich rozpoczyna następną fazę.

2. W fazie 2 zmiany są wprowadzane na stałe. Jeśli jeden z systemów zostanie awaryjnie zakończony podczas fazy 2, operacja zostanie zakończona przez proces odtwarzania podczas restartu.

Ilustracja procesu zatwierdzania dwufazowego

Rysunek 77 na stronie 265 przedstawia proces zatwierdzania dwufazowego. Zdarzenia w koordynatorze CICS lub IMS są wyświetlane w górnej linii, zdarzenia w IBM MQ w dolnej linii.



Rysunek 77. Proces zatwierdzania dwufazowego

Liczby w poniższej sekcji są powiązane z liczbami przedstawionymi na rysunku.

1. Dane koordynatora znajdują się w punkcie spójności.
2. Aplikacja w koordynatorze wywołuje funkcję IBM MQ w celu zaktualizowania kolejki przez dodanie komunikatu.
3. Spowoduje to uruchomienie jednostki odzyskiwania w programie IBM MQ.
4. Przetwarzanie jest kontynuowane w koordynatorze do momentu osiągnięcia punktu synchronizacji aplikacji.
5. Następnie koordynator rozpoczyna przetwarzanie zatwierdzania. Programy CICS uruchamiają zatwierdzanie za pomocą komendy SYNCPOINT lub normalnego zakończenia działania aplikacji. Programy IMS mogą uruchomić zatwierdzanie za pomocą wywołania CHKP, wywołania SYNC,

wywołania GET UNIQUE do IOPCB lub normalnego zakończenia aplikacji. Rozpoczyna się faza 1 przetwarzania zatwierdzania.

6. Jako koordynator rozpoczyna przetwarzanie fazy 1, podobnie jak IBM MQ.
7. IBM MQ pomyślnie kończy fazę 1, zapisuje ten fakt w swoim dzienniku i powiadamia koordynatora.
8. Koordynator otrzymuje powiadomienie.
9. Koordynator pomyślnie zakończył przetwarzanie fazy 1. Obecnie oba podsystemy zgadzają się na zatwierdzenie zmian danych, ponieważ oba podsystemy zakończyły fazę 1 i mogą odzyskać sprawność po wystąpieniu błędów. Koordynator rejestruje w swoim dzienniku moment zatwierdzenia-nieodwołalną decyzję dwóch podsystemów o wprowadzeniu zmian.
Koordynator rozpoczyna teraz fazę 2 przetwarzania-rzeczywiste zobowiązanie.
10. Koordynator powiadamia IBM MQ o rozpoczęciu fazy 2.
11. IBM MQ rejestruje początek fazy 2.
12. Faza 2 została pomyślnie zakończona i jest to nowy punkt spójności dla produktu IBM MQ. Następnie program IBM MQ powiadamia koordynatora o zakończeniu przetwarzania fazy 2.
13. Koordynator kończy przetwarzanie fazy 2. Dane kontrolowane przez oba podsystemy są teraz spójne i dostępne dla innych aplikacji.

Zachowanie spójności po nieprawidłowym zakończeniu

Gdy menedżer kolejek jest restartowany po nieprawidłowym zakończeniu, musi określić, czy zatwierdzić, czy wycofać jednostki odzyskiwania, które były aktywne w momencie zakończenia. W przypadku niektórych jednostek odzyskiwania IBM MQ ma wystarczającą ilość informacji, aby podjąć decyzję. Dla innych nie, i musi uzyskać informacje od koordynatora po ponownym nawiązaniu połączenia.

Rysunek 77 na stronie 265 przedstawia cztery okresy w dwóch fazach: a, b, c i d. Status jednostki odzyskiwania zależy od okresu, w którym nastąpiło zakończenie. Status może mieć jedną z następujących wartości:

W trakcie lotu

Menedżer kolejek został zakończony przed zakończeniem fazy 1 (okres a lub b); podczas restartowania program IBM MQ wycofuje aktualizacje.

Wątpliwy

Menedżer kolejek zakończył działanie po zakończeniu fazy 1 i przed rozpoczęciem fazy 2 (okres c); tylko koordynator wie, czy błąd wystąpił przed zatwierdzeniem, czy po nim (punkt 9). Jeśli miało to miejsce wcześniej, program IBM MQ musi wycofać swoje zmiany. Jeśli miało to miejsce później, program IBM MQ musi wprowadzić zmiany i zatwierdzić je. Przy restarcie IBM MQ oczekuje na informacje od koordynatora przed przetworzeniem tej jednostki odzyskiwania.

W trakcie zatwierdzania

Menedżer kolejek zakończył działanie po rozpoczęciu własnej fazy 2 przetwarzania (okres d); wprowadza zatwierdzone zmiany.

Wycofanie

Menedżer kolejek został zakończony po rozpoczęciu wycofywania jednostki odzyskiwania, ale przed zakończeniem procesu (nieprzedstawionego na rysunku) podczas restartowania, program IBM MQ nadal wycofuje zmiany.

Co się dzieje podczas kończenia w programie IBM MQ for z/OS

Menedżer kolejek kończy działanie normalnie w odpowiedzi na komendę STOP QMGR. Jeśli menedżer kolejek zostanie zatrzymany z innego powodu, zakończenie jest nieprawidłowe.

Należy zauważyć, że podczas kończenia działania menedżera kolejek produkt IBM MQ wewnętrznie wydaje komendę.

```
DISPLAY CONN(*) TYPE(CONN) ALL WHERE (APPLTYPE NE SYSTEMAL)
```

Dzięki temu wiadomo, jakie wątki mogą uniemożliwić menedżerowi kolejek zakończenie zamykania systemu.

SYSTEMAL dopasowuje APPLTYPES albo SYSTEM albo CHINIT, dlatego komenda DISPLAY CONN filtrująca typy aplikacji niezgodne z SYSTEMAL powraca do informacji protokołu zadania o wątkach, które mogą uniemożliwiać normalne zamknięcie systemu.

Normalne zakończenie

W przypadku normalnego zakończenia program IBM MQ zatrzymuje wszystkie działania w uporządkowany sposób. Serwer IBM MQ można zatrzymać w trybie wyciszenia, wymuszenia lub restartu. Efekty są podane w sekcji [Tabela 23 na stronie 267](#).

Tabela 23. Zakończenie za pomocą QUIESCE, FORCE i RESTART

Typ wątku	QUIESCE	Wymuszenie	RESTARTOWANIE
Aktywne wątki	Wykonaj do końca	Wycofanie	Wycofanie
Nowe wątki	Można uruchomić	Niedozwolone	Niedozwolone
Nowe połączenia	Niedozwolone	Niedozwolone	Niedozwolone

Aplikacje wsadowe są powiadamiane, jeśli wystąpi zakończenie, gdy aplikacja jest nadal połączona.

W systemie CICSbieżący wątek działa tylko do końca jednostki odzyskiwania. W przypadku serwera CICSzatrzymanie menedżera kolejek w trybie wyciszania powoduje zatrzymanie adaptera CICS , a więc jeśli aktywne zadanie zawiera więcej niż jedną jednostkę odtwarzania, zadanie nie musi zostać wykonane do końca.

Jeśli menedżer kolejek zostanie zatrzymany w trybie wymuszania lub restartowania, nie zostaną przydzielone nowe wątki, a praca z połączonymi wątkami zostanie wycofana. Za pomocą tych trybów można tworzyć wątpliwe jednostki odzyskiwania dla wątków, które znajdują się między fazami przetwarzania zatwierdzania. Są one rozstrzygane, gdy program IBM MQ jest ponownie połączony z sterującym podsystemem CICS, IMSlub RRS.

Po zatrzymaniu menedżera kolejek w dowolnym trybie należy wykonać następujące kroki:

1. Połączenia zostały zakończone.
2. Program IBM MQ przestaje akceptować komendy.
3. IBM MQ zapewnia, że wszystkie zaległe aktualizacje zestawów stron zostaną zakończone.
4. Komenda DISPLAY USAGE jest wydawana wewnętrznie przez IBM MQ , aby restart RBA był rejestrowany w dzienniku konsoli z/OS .
5. Punkt kontrolny zamknięcia systemu jest pobierany i aktualizowany jest BSDS.

Zakończenia, które określają tryb wyciszania, nie mają wpływu na wątpliwe jednostki odzyskiwania. Każda jednostka, która ma wątpliwości, pozostaje wątpliwa.

Nieprawidłowe zakończenie

Nieprawidłowe zakończenie może spowodować, że dane będą w stanie niespójnym, na przykład:

- Jednostka odzyskiwania została przerwana przed osiągnięciem punktu spójności.
- Zatwierdzone dane nie zostały zapisane w zestawach stron.
- Niezatwierdzone dane zostały zapisane w zestawach stron.

- Aplikacja została przerwana między fazą 1 a fazą 2 procesu zatwierdzania, pozostawiając wątpliwą jednostkę odzyskiwania.

Produkt IBM MQ usuwa wszelkie niespójności danych wynikające z nieprawidłowego zakończenia działania podczas restartowania i odtwarzania.

Co się dzieje podczas restartowania i odtwarzania w produkcie IBM MQ for z/OS

Produkt IBM MQ używa dziennika odtwarzania i zestawu danych programu startowego (BSDS) do określenia, co należy odzyskać po restarcie. Usługa BSDS identyfikuje aktywne i archiwalne zestawy danych dziennika oraz położenie najnowszego punktu kontrolnego IBM MQ w dzienniku.

Wprowadzenie do restartowania i odtwarzania

Po zainicjowaniu programu IBM MQ proces restartowania menedżera kolejek odbywa się w następujący sposób:

- Inicjowanie dziennika
- Odbudowa bieżącego statusu
- Odtwarzanie dziennika przekazywania
- Odtwarzanie dziennika wstecznego
- Odbudowywanie indeksu kolejki

Po zakończeniu odtwarzania:

- Zatwierdzone zmiany są odzwierciedlane w danych.
- Działanie wątpliwe jest odzwierciedlane w danych. Jednak dane są zablokowane i nie można ich używać, dopóki program IBM MQ nie rozpozna i nie podejmie decyzji wątpliwej.
- Przerwane zmiany w trakcie przetwarzania i w trakcie przerwania zostały usunięte z kolejek. Komunikaty są spójne i można ich używać.
- Utworzono nowy punkt kontrolny.
- Utworzono nowe indeksy dla kolejek indeksowanych zawierających trwałe komunikaty (patrz sekcja [“Odbudowywanie indeksów kolejki”](#) na stronie 270).

Jeśli używane są podwójne BSD, program IBM MQ sprawdza spójność datowników w BSDS:

- Jeśli obie kopie BSDS są aktualne, produkt IBM MQ sprawdza, czy dwa znaczniki czasu są równe. Jeśli nie, IBM MQ wysyła komunikat CSQJ120E i kończy działanie. Taka sytuacja może wystąpić, gdy dwie kopie BSDS są przechowywane na oddzielnych woluminach DASD, a jeden z woluminów został odtworzony, gdy menedżer kolejek był zatrzymany. Program IBM MQ wykrywa sytuację podczas restartu.
- Jeśli jedna kopia BSDS została zwolona, a rejestrowanie było kontynuowane z pojedynczym BSDS, może wystąpić problem. Jeśli obie kopie BSDS są przechowywane na pojedynczym woluminie i wolumin został odtworzony lub jeśli obie kopie BSDS zostały odtworzone oddzielnie, program IBM MQ może nie wykryć odtwarzania. W takim przypadku rekordy dziennika, które nie zostały zapisane w BSDS, będą nieznanne w systemie.

Aplikacje wsadowe nie są powiadamiane o restarcie *po* zażądaniu połączenia przez aplikację.

Zakres dziennika wymagany do odtwarzania

Podczas restartu zakres danych dziennika, które muszą zostać odczytane, zależy od wielu czynników:

- W momencie nieprawidłowego zakończenia systemu zwykle występuje wiele niekompletnych jednostek pracy. Jak opisano wcześniej, przetwarzanie restartu przywróci system do stanu spójności, co może obejmować wycofanie jednostek pracy w trakcie przetwarzania lub odzyskanie blokad na wątpliwych jednostkach pracy. Odtwarzanie jednostki pracy wymaga, aby wszystkie rekordy dziennika jednostki

pracy dla bieżących, wycofanych i wątpliwych jednostek pracy były dostępne. IBM MQ będzie "przetasaować" stare jednostki pracy, tak aby można było wykonać odzyskiwanie jednostki pracy przy użyciu znacznie mniejszego zakresu danych dziennika.

- W czasie nieprawidłowego zakończenia istnieje zwykle wiele trwałych aktualizacji, które są przechowywane tylko w pamięci podręcznej puli buforów. Nie zostały one jeszcze zapisane na dysku. Zmiany te muszą zostać odczytane z dziennika i ponownie zastosowane do danych przechowywanych w zestawach stron. Strategia RBA odtwarzania zestawu stron w punkcie kontrolnym opisuje najniższy adres RBA dziennika wymagany do zaktualizowania zestawów stron do stanu spójności.
- Jeśli stare zestawy stron zostały wprowadzone do systemu, na przykład w celu odtworzenia po awarii nośnika wprowadzono kopię zapasową zestawu stron, wszystkie zmiany muszą zostać odczytane z dziennika od momentu utworzenia kopii zapasowej. Te zmiany są ponownie stosowane do danych przechowywanych w odtwarzanym zestawie stron. Strategia RBAs odtwarzania zestawu stron wstrzymana na stronie 0 zestawu stron opisuje najniższy adres RBA dziennika wymagany do odtwarzania nośników zestawu stron.
- Jeśli w kolejkach współużytkowanych używane są komunikaty trwałe, do odtworzenia CFSTRUCT, które przechowują komunikaty trwałe, wymagany jest zakres danych dziennika. Najwcześniejsze dane dziennika, które byłyby wymagane do wykonania odtwarzania CFSTRUCT, pochodzą z mniej więcej tego samego okresu, co stara komenda CFSTRUCT BACKUP.

Podczas normalnego działania można użyć komendy DISPLAY USAGE TYPE (DATASET), aby wyświetlić zakres dziennika odtwarzania powiązany z tymi czynnikami (oczywiście nie można udostępnić informacji z powodu ponownego wprowadzenia starych zestawów stron). Aby uniknąć problemów, które mogą wydłużyć restart menedżera kolejek w przypadku nieprawidłowego zakończenia, należy regularnie monitorować wartości wyjściowe komendy DISPLAY USAGE TYPE (DATASET).

Ponadto menedżer kolejek wysyła komunikaty informacyjne dotyczące następujących czynników:

- CSQJ160I i CSQJ161I ostrzegają o długotrwałych jednostkach pracy.
- CSQR026I i CSQR027I zawierają informacje o tym, czy te długotrwałe jednostki pracy zostały pomyślnie przetaczone.
- CSQE040I i CSQE041E ostrzegają, że tworzenie kopii zapasowych struktury jest coraz starsze i w związku z tym operacja RECOVER CFSTRUCT może zająć dużo czasu.

Określanie, która aplikacja ma długo działającą jednostkę pracy

Istnieje możliwość określenia aplikacji z długo działającą jednostką pracy. W tym celu należy użyć komendy DISPLAY CONN.

Komenda DISPLAY CONN zwraca informacje o połączeniu dla wszystkich aplikacji połączonych z menedżerem kolejek wraz z informacjami dodatkowymi, które ułatwiają określenie, które aplikacje aktualnie mają długotrwałą jednostkę pracy. Informacje zwracane przez komendę DISPLAY CONN są podobne do informacji zwracanych przez komendę DISPLAY QSTATUS, ale główną różnicą jest to, że komenda DISPLAY CONN wyświetla informacje o obiektach i informacje transakcyjne dla konkretnego połączenia, a nie szczegóły, które połączenia są powiązane z konkretnym obiektem.

Dla każdej połączonej aplikacji komenda DISPLAY CONN zwraca następujące informacje:

- Podstawowe informacje, w tym identyfikator połączenia i identyfikator PID.
- Informacje transakcyjne dla tego połączenia, w tym data i godzina utworzenia transakcji (tzn. gdy pierwsza operacja MQGET/PUT była wykonywana w punkcie synchronizacji) oraz czas pierwszego zapisu transakcji w dzienniku.
- Rejestruj informacje o czasie wskazujące, która aplikacja nadal ma długo działającą jednostkę pracy.
- Lista wszystkich obiektów, które są aktualnie otwarte w połączeniu. Szczegóły dla każdego obiektu są zwracane jako oddzielny komunikat z identyfikatorem połączenia używanym jako klucz. Ponieważ istnieją różne typy obiektów, takie jak kolejki i menedżery kolejek, informacje wyświetlane z obiektem są specyficzne dla jego typu.

Odbudowywanie indeksów kolejki

Aby zwiększyć szybkość operacji MQGET w kolejce, w której komunikaty nie są pobierane sekwencyjnie, można określić, że program IBM MQ ma obsługiwać indeks identyfikatorów komunikatów lub korelacji albo identyfikator grupy dla wszystkich komunikatów w tej kolejce.

Po zrestartowaniu menedżera kolejek te indeksy są odbudowywane dla każdej kolejki. Dotyczy to tylko komunikatów trwałych; komunikaty nietrwałe są usuwane podczas restartu. Jeśli indeksowane kolejki zawierają dużą liczbę trwałych komunikatów, wydłuża to czas potrzebny do zrestartowania menedżera kolejek.

Za pomocą parametru QINDEXBLD makra CSQ6SYSP można wybrać opcję asynchronicznego odbudowywania indeksów podczas uruchamiania menedżera kolejek. Jeśli zostanie ustawiona wartość QINDEXBLD=NOWAIT, produkt IBM MQ zostanie zrestartowany bez oczekiwania na odbudowanie indeksów.

Sposób rozstrzygnięcia wątpliwych jednostek odzyskiwania

Jeśli program IBM MQ utraci połączenie z innym menedżerem zasobów, zwykle próbuje odtworzyć wszystkie niespójne obiekty podczas restartu.

Jeśli produkt IBM MQ utraci połączenie z produktem CICS, IMS lub RRS, zwykle próbuje odtworzyć wszystkie niespójne obiekty po restarcie. Informacje wymagane do rozstrzygnięcia wątpliwych jednostek odzyskiwania muszą pochodzić z systemu koordynującego. W następnych sekcjach opisano proces rozstrzygnięcia dla różnych środowisk.

- [Jak rozstrzygane są wątpliwe jednostki odzyskiwania CICS](#)
- [Jak rozstrzygane są wątpliwe jednostki odzyskiwania IMS](#)
- [Sposób rozstrzygnięcia wątpliwych jednostek odzyskiwania z usługi RRS](#)
- [Jak rozstrzygane są wątpliwe jednostki odzyskiwania z dyspozycją GROUP jednostek odzyskiwania](#)

Sposób rozstrzygnięcia wątpliwych jednostek odzyskiwania na podstawie CICS

W pewnych okolicznościach produkt CICS nie może uruchomić procesu IBM MQ w celu rozstrzygnięcia wątpliwych jednostek odzyskiwania. W takim przypadku program IBM MQ wysyła jeden z następujących komunikatów:

- CSQC404E
- CSQC405E
- CSQC406E
- CSQC407E

po którym następuje komunikat CSQC408I.

Szczegółowe informacje na temat znaczenia tych komunikatów zawiera podręcznik [Komunikaty systemu IBM MQ for z/OS](#), kody zakończenia i kody przyczyny.

Rozstrzygnięcie jednostek wątpliwych nie ma wpływu na zasoby CICS. Program CICS kontroluje koordynację odtwarzania, a po restarcie automatycznie zatwierdza lub wycofuje każdą jednostkę, w zależności od tego, czy istniał rekord dziennika oznaczający początek zatwierdzenia. Istnienie obiektów wątpliwych nie blokuje zasobów CICS podczas ponownego łączenia z serwerem IBM MQ.

Jedną z funkcji adaptera CICS jest synchronizowanie danych między produktami CICS i IBM MQ. Jeśli menedżer kolejek zostanie zakończony awaryjnie podczas połączenia z programem CICS, program CICS może zatwierdzić lub wycofać pracę, nie informując o tym programu IBM MQ. Po zrestartowaniu menedżera kolejek praca ta jest określona jako *wątpliwa*.

Produkt IBM MQ nie może rozstrzygnąć tych wątpliwych jednostek odtwarzania (czyli zatwierdzić lub wycofać zmian wprowadzonych w zasobach IBM MQ) do momentu zrestartowania lub ponownego połączenia z serwerem CICS.

Proces rozstrzygnięcia wątpliwych jednostek odtwarzania jest inicjowany podczas uruchamiania adaptera CICS . Proces jest uruchamiany, gdy adapter żąda listy wątpliwych jednostek odzyskiwania. Następnie:

- Adapter odbiera listę wątpliwych jednostek odzyskiwania dla tego identyfikatora połączenia z produktu IBM MQi przekazuje je do produktu CICS w celu rozstrzygnięcia.
- Program CICS porównuje wpisy z tej listy z wpisami we własnym dzienniku. CICS określa na podstawie własnej listy, jakie działanie podjęło dla każdej wątpliwej jednostki odzyskiwania.

Dla wszystkich rozstrzygniętych jednostek program IBM MQ w razie potrzeby aktualizuje kolejki i zwalnia odpowiednie blokady. nierozstrzygnięte jednostki mogą pozostać po restarcie. Należy je rozstrzygnąć przy użyciu metod opisanych w sekcji [Administrowanie programem IBM MQ for z/OS](#).

Sposób rozstrzygnięcia wątpliwych jednostek odzyskiwania na podstawie IMS

Rozstrzygnięcie wątpliwych jednostek odzyskiwania w systemie IMS nie ma wpływu na zasoby DL/I. Program IMS steruje koordynacją odtwarzania i po restarcie automatycznie zatwierdza lub wycofuje niekompletne prace w języku DL/I. Decyzja o zatwierdzeniu lub wycofaniu regionów w trybie z połączeniem (bez krótkiej ścieżki) dotyczy obecności lub nieobecności rekordów dziennika IMS o typach X'3730 'i X'3801'. Istnienie wątpliwych jednostek odzyskiwania nie oznacza, że rekordy DL/I są blokowane do momentu nawiązania połączenia przez program IBM MQ .

Podczas restartowania menedżera kolejek program IBM MQ tworzy listę wątpliwych jednostek odzyskiwania. IMS buduje własną listę pozycji odzyskiwania reszt (RRE). Środowiska JRE są rejestrowane w punktach kontrolnych IMS do momentu rozstrzygnięcia wszystkich pozycji.

Podczas ponownego łączenia IMS regionu z IBM MQ IMS wskazuje IBM MQ , czy jednostki pracy oznaczone w programie IBM MQ jako wątpliwe mają być zatwierdzone, czy wycofane.

Gdy jednostki wątpliwe są rozstrzygane:

1. Jeśli program IBM MQ rozpozna, że oznaczył pozycję do zatwierdzenia, a program IMS oznaczył ją do wycofania, zostanie wygenerowany komunikat IBM MQ CSQQ010E. IBM MQ generuje ten komunikat dla wszystkich niespójności tego typu między systemami IBM MQ i IMS.
2. Jeśli baza danych IBM MQ zawiera jednostki wątpliwe, adapter wysyła komunikat CSQQ008I.

Dla wszystkich rozstrzygniętych jednostek program IBM MQ w razie potrzeby aktualizuje kolejki i zwalnia odpowiednie blokady.

Produkt IBM MQ utrzymuje blokady wątpliwej pracy, która nie została rozwiązana. Może to spowodować powstanie zaległości w systemie, jeśli utrzymywane są ważne blokady. Połączenie pozostaje aktywne, dzięki czemu można rozstrzygnąć środowiska JRE IMS . Odtwórz wątpliwe wątki przy użyciu metod opisanych w sekcji [Administrowanie programem IBM MQ for z/OS](#).

Wszystkie prace wątpliwe powinny zostać rozwiązane, chyba że występują problemy z oprogramowaniem lub działaniem, na przykład z zimnym startem systemu IMS . Wątpliwe rozstrzygnięcie przez region sterujący IMS ma miejsce w dwóch okolicznościach:

1. Na początku połączenia z IBM MQ, podczas którego rozstrzygnięcie jest wykonywane synchronicznie.
2. Po awariach programu, podczas których rozstrzygnięcie jest wykonywane asynchronicznie.

Rozstrzygnięcie wątpliwych jednostek odzyskiwania na podstawie usługi RRS

Jedną z funkcji adaptera RRS jest synchronizowanie danych między produktem IBM MQ i innymi menedżerami zasobów uczestniczącymi w usłudze RRS. Jeśli wystąpi awaria, gdy produkt IBM MQ zakończył fazę pierwszej zatwierdzania i oczekuje na decyzję z usługi RRS (koordynator zatwierdzania), jednostka odzyskiwania przechodzi w stan wątpliwy.

Po ponownym ustanowieniu komunikacji między RRS i IBM MQ usługa RRS automatycznie zatwierdza lub wycofuje każdą jednostkę odtwarzania, w zależności od tego, czy istniał rekord dziennika oznaczający początek zatwierdzania. Produkt IBM MQ nie może rozstrzygnąć tych wątpliwych jednostek odtwarzania

(czyli zatwierdzić lub wycofać zmian wprowadzonych w zasobach IBM MQ) do momentu ponownego nawiązania połączenia z usługą RRS.

W pewnych okolicznościach usługa RRS nie może rozstrzygnąć wątpliwych jednostek odtwarzania. W takim przypadku program IBM MQ wysyła jeden z następujących komunikatów do konsoli z/OS :

- CSQ3011I
- CSQ3013I
- CSQ3014I
- CSQ3016I

Szczegółowe informacje na temat znaczenia tych komunikatów zawiera podręcznik Komunikaty systemu IBM MQ for z/OS , kody zakończenia i kody przyczyny .

Dla wszystkich rozstrzygniętych jednostek odzyskiwania program IBM MQ aktualizuje kolejki zgodnie z potrzebami i zwalnia odpowiednie blokady. Nierozstrzygnięte jednostki odzyskiwania mogą pozostać po restarcie. Należy je rozstrzygnąć przy użyciu metody opisanej w sekcji Administrowanie programem IBM MQ for z/OS.

Jak rozstrzygane są wątpliwe jednostki odzyskiwania z dyspozycją jednostki odzyskiwania GROUP

Transakcje wątpliwe, które mają dyspozycję GROUP jednostki odzyskiwania, mogą być rozstrzygane przez koordynatora transakcji przez dowolny menedżer kolejek w grupie współużytkowania kolejek (QSG), w której włączony jest atrybut GROUPUR menedżera kolejek. Za każdym razem, gdy koordynator transakcji ponownie nawiązuje połączenie, zwykle żąda listy wszystkich oczekujących transakcji wątpliwych, a następnie rozstrzyga je przy użyciu informacji z dzienników.

Gdy koordynator transakcji, który nawiązał połączenie z jednostką odzyskiwania GROUP, żąda listy wątpliwych transakcji, zwracana lista obejmuje wszystkie wątpliwe transakcje z jednostką odzyskiwania GROUP, które istnieją w grupie współużytkowania kolejek. Ta lista nie zależy od tego, w którym menedżerze kolejek zostały uruchomione transakcje wątpliwe. Menedżer kolejek przetwarzający takie żądanie kompiluje listę, komunikując się ze wszystkimi innymi aktywnymi menedżerami kolejek w grupie współużytkowania kolejek za pomocą systemu SYSTEM.QSG.UR.RESOLUTION.QUEUE. Następnie menedżer kolejek odczytuje dzienniki wszystkich nieaktywnych menedżerów kolejek z ostatniego punktu kontrolnego, aby zidentyfikować dodatkowe wątpliwe transakcje, które zostałyby zgłoszone, gdyby były aktywne.

Gdy koordynator transakcji żąda rozstrzygnięcia transakcji wątpliwej, menedżer kolejek, z którym jest połączony, określa, czy transakcja została zainicjowana na samym sobie, a jeśli tak, to rozstrzyga ją w taki sam sposób, jak transakcje z jednostką odzyskiwania QMGR. Jeśli transakcja została zainicjowana w innym aktywnym menedżerze kolejek w menedżerze kolejek QSG, żądanie zakończenia rozstrzygnięcia jest kierowane do tego menedżera kolejek za pomocą systemu SYSTEM.QSG.UR.RESOLUTION.QUEUE. W przypadku, gdy transakcja została zainicjowana w nieaktywnym menedżerze kolejek w menedżerze kolejek QSG, wszystkie zadania kolejki współużytkowanej są natychmiast rozstrzygane, a żądanie rozstrzygnięcia pozostałych zadań kolejki prywatnej jest umieszczane w systemie SYSTEM.QSG.UR.RESOLUTION.QUEUE. Nieaktywny menedżer kolejek przetwarza to żądanie po uruchomieniu przed zaakceptowaniem nowej pracy. W tym scenariuszu dzienniki oryginalnego menedżera kolejek nadal wskazują, że jednostka odtwarzania jest wątpliwa do momentu zrestartowania i przetworzenia żądania.

Odtwarzanie kolejki współużytkowanej

Ten temat zawiera informacje dotyczące odtwarzania systemu IBM MQ i odporności różnych komponentów w środowisku grupy współużytkowania kolejek.

- [“Odtwarzanie transakcyjne” na stronie 273](#)
- [“Odtwarzanie równorzędne” na stronie 273](#)

- [“Definicje kolejek współużytkowanych” na stronie 274](#)
- [“Rejestrowanie” na stronie 274](#)
- [“Awaryjne narzędzia CF i struktury” na stronie 274](#)
- [“Scenariusze awarii struktury” na stronie 275](#)
- [“Odporność na awaryjne połączenia narzędzia CF” na stronie 276](#)
- [“Zarządzanie odpornością na awaryjne połączenia narzędzia CF” na stronie 277](#)
- [“Zachowanie operacyjne” na stronie 279](#)

Odtwarzanie transakcyjne

Jeśli aplikacja wywoła wywołanie MQBACK lub zakończy działanie nieprawidłowo (na przykład z powodu wycofania CICS ROLLBACK lub nieprawidłowego zakończenia IMS), informacje na poziomie wątku przechowywane w menedżerze kolejek gwarantują, że jednostka pracy w trakcie przetwarzania zostanie wycofana. Operacje MQPUT i MQGET w punkcie synchronizacji kolejek współużytkowanych są wycofywane w taki sam sposób, jak aktualizacje kolejek niewspółużytkowanych.

Odtwarzanie równorzędne

Jeśli menedżer kolejek ulegnie awarii, zostanie nieprawidłowo odłączony od struktur narzędzia CF, z którymi jest obecnie połączony. Jeśli połączenie między instancją z/OS i narzędziem CF nie powiedzie się (na przykład awaria łącza fizycznego lub wyłączenie zasilania partycji lub narzędzia CF), zostanie ono również wykryte jako nieprawidłowe zakończenie połączenia między menedżerem kolejek i strukturami narzędzia CF, których to dotyczy. Inne menedżery kolejek w tej samej grupie współużytkowania kolejek, które pozostają połączone z tą strukturą, wykryją nieprawidłowe rozłączenie i podejmą próbę zainicjowania *odtworzenia równorzędnego* dla menedżera kolejek, który uległ awarii w tej strukturze. Tylko jeden z tych menedżerów kolejek pomyślnie inicjuje odtwarzanie równorzędne, ale wszystkie pozostałe menedżery kolejek współpracują w odzyskiwaniu jednostek pracy, których właścicielem był menedżer kolejek, który uległ awarii.

Jeśli działanie menedżera kolejek zakończy się niepowodzeniem, gdy żaden z węzłów nie jest połączony ze strukturą, odtwarzanie jest wykonywane, gdy inny menedżer kolejek nawiązuje połączenie z tą strukturą lub gdy menedżer kolejek, którego działanie nie powiodło się, jest restartowany.

Odtwarzanie równorzędne, często nazywane odtwarzaniem na poziomie równorzędnym (Peer Level Recovery-PLR), jest wykonywane na podstawie struktury i pojedynczy menedżer kolejek może uczestniczyć w odtwarzaniu więcej niż jednej struktury w tym samym czasie. Jednak zestaw elementów równorzędnych współpracujących przy odtwarzaniu różnych struktur może się różnić w zależności od tego, które menedżery kolejek były połączone z różnymi strukturami w momencie awarii.

Po zrestartowaniu menedżera kolejek, który uległ awarii, ponownie nawiązuje on połączenie ze strukturami, z którymi był połączony w momencie awarii, i odzyskuje wszystkie pozostałe nierozstrzygnięte jednostki pracy, które nie zostały odtworzone przez odtwarzanie równorzędne.

Odtwarzanie równorzędne jest procesem wielofazowym. Podczas pierwszej fazy odzyskiwane są jednostki pracy, które zostały przesunięte poza fazę w trakcie realizacji. Może to obejmować zatwierdzanie komunikatów dla jednostek pracy, które są w trakcie zatwierdzania, oraz blokowanie komunikatów dla jednostek pracy, które są wątpliwe. Podczas drugiej fazy kolejki, dla których w menedżerze kolejek, w którym wystąpiło niepowodzenie, były aktywne wątki, są sprawdzane, wycofywane są niezatwierdzone komunikaty związane z jednostkami pracy w trakcie przetwarzania, a informacje o aktywnych uchwytach w kolejkach współużytkowanych w menedżerze kolejek, dla którego wystąpiło niepowodzenie, są resetowane. Oznacza to, że program IBM MQ resetuje wszystkie indykatory otwarcia kolejki współużytkowanej dla wyłącznego wejścia menedżera kolejek, który uległ awarii, umożliwiając innym aktywnym menedżerom kolejek otwarcie kolejki dla wejścia.

Definicje kolejek współużytkowanych

Obiekty kolejki reprezentujące atrybuty kolejki współużytkowanej są przechowywane w współużytkowanym repozytorium Db2 używanym przez grupę współużytkowania kolejek. Upewnij się, że istnieją odpowiednie procedury tworzenia i odtwarzania kopii zapasowych tabel Db2 używanych do przechowywania obiektów IBM MQ. Można również użyć programu narzędziowego IBM MQ CSQUTIL do utworzenia komend MQSC służących do odtwarzania w menedżerze kolejek w celu ponownego zdefiniowania obiektów IBM MQ, w tym współużytkowanych definicji kolejek i grup przechowywanych w systemie Db2.

Rejestrowanie

Grupy współużytkowania kolejek mogą obsługiwać komunikaty trwałe, ponieważ komunikaty w kolejkach współużytkowanych mogą być protokołowane w dziennikach menedżera kolejek.

Awarie narzędzia CF i struktury

Istnieją dwa typy awarii, które mogą być zgłaszane dla struktury narzędzia CF: awaria struktury i utrata połączenia. Usługi Sysplex dla współużytkowania danych (XES) informują program IBM MQ o awarii struktury CF lub awarii CF ze zdarzeniem awarii struktury. Jeśli zdarzenie XES powoduje utratę połączenia, nie musi to oznaczać, że występuje problem ze strukturą, może to oznaczać, że nie ma dostępnego połączenia do komunikacji ze strukturą. Możliwe, że nie wszystkie menedżery kolejek odbierają zdarzenie utraty połączenia dla struktury. Zależy to od konfiguracji połączeń z systemem CF. Zdarzenie utraty połączenia może również zostać odebrane przez komendy operatora, na przykład VARY PATH OFFLINE lub CONFIG CHP OFFLINE.

Struktury CF, które są używane przez system IBM MQ, można skonfigurować w taki sposób, aby używały dupleksowania zarządzanego przez system. Oznacza to, że w przypadku wystąpienia pojedynczej awarii przetwarzanie przetaczania awaryjnego zarządzanego przez system ukrywa awarię struktury lub utratę połączenia, a menedżer kolejek nie jest informowany o awarii. Jeśli wystąpi awaria obu instancji struktury z dupleksem lub połączenia, menedżer kolejek odbiera odpowiednie zdarzenie i obsługuje je w taki sam sposób, jak zdarzenie niepowodzenia dla struktury simpleks. Szczegółowe informacje na temat obsługi zdarzeń przez menedżer kolejek zawiera sekcja [Scenariusze](#).

W mało prawdopodobnym przypadku awarii systemu CF lub struktury wszystkie nietrwałe komunikaty zapisane w odpowiednich strukturach aplikacji są tracone. Trwałe komunikaty można odzyskać za pomocą komendy RECOVER CFSTRUCT. Jeśli odtwarzalna struktura aplikacji nie powiedzie się, dalsze działanie aplikacji względem tej struktury zostanie uniemożliwione do czasu odtworzenia struktury.

Aby zapewnić możliwość odtworzenia struktury CF w rozsądnym czasie, należy często tworzyć kopie zapasowe za pomocą komendy BACKUP CFSTRUCT. Kopie zapasowe można tworzyć w dowolnych menedżerach kolejek w grupie współużytkowania kolejek lub można dedykować jeden menedżer kolejek do tworzenia wszystkich kopii zapasowych. Zautomatyzuj proces tworzenia kopii zapasowych, aby zapewnić ich regularne wykonywanie.

Każda kopia zapasowa jest zapisywana w aktywnym zestawie danych dziennika menedżera kolejek, który tworzy kopię zapasową. Repozytorium Db2 kolejki współużytkowanej rejestruje nazwę struktury CF, dla której tworzona jest kopia zapasowa, nazwę menedżera kolejek wykonującego kopię zapasową, zakres RBA dla tej kopii zapasowej w dzienniku menedżera kolejek oraz czas kopii zapasowej.

Struktura administracyjna zawiera informacje o niekompletnych jednostkach pracy w kolejkach współużytkowanych w czasie awarii struktury aplikacji, dlatego struktura administracyjna musi być dostępna podczas przetwarzania RECOVER CFSTRUCT. Jeśli struktura administracyjna nie powiodła się, wszystkie menedżery kolejek w grupie współużytkowania kolejek muszą odbudować swoje pozycje struktury administracyjnej przed wydaniem komendy RECOVER CFSTRUCT.

Menedżery kolejek automatycznie i bez przerywania odbudowują swoje pozycje struktury administracyjnej. Jeśli menedżer kolejek nie jest uruchomiony w momencie niepowodzenia, jego

pozycje struktury administracyjnej mogą zostać odbudowane przez innego menedżera kolejek w grupie współużytkowania kolejek, która jest uruchomiona na tym samym lub wyższym poziomie.

Aby odzyskać strukturę aplikacji, należy wprowadzić komendę RECOVER CFSTRUCT do menedżera kolejek, który ma wykonać odtwarzanie. Można odzyskać pojedynczą strukturę CF lub kilka struktur CF jednocześnie. Odtwarzanie można przeprowadzić przy użyciu dowolnego menedżera kolejek w grupie współużytkowania kolejek. Nie musi to być menedżer, który wykonał składowanie, ani menedżer, który był wcześniej połączony z uszkodzoną strukturą.

Komenda RECOVER CFSTRUCT używa kopii zapasowej, która znajduje się w informacjach o repozytorium Db2 (program Db2 musi być dostępny w menedżerze kolejek, w którym przeprowadzane jest odtwarzanie), i odtwarza ją do punktu awarii.

Wykonuje to komenda RECOVER CFSTRUCT, stosując rekordy dziennika z każdego menedżera kolejek w grupie współużytkowania kolejek, który wykonał operację MQPUT lub MQGET między rozpoczęciem tworzenia kopii zapasowej a czasem awarii, do dowolnej kolejki współużytkowanej, która jest odwzorowana na strukturę CF. Wynikowe scalanie dzienników może wymagać odczytania znacznej ilości danych dziennika, ponieważ wszystkie dane dziennika zapisane przez uczestniczące menedżery kolejek od momentu odczytania kopii zapasowej. Stanowczo zaleca się częste (na przykład co godzinę) tworzenie kopii zapasowych, zwłaszcza jeśli w kopii zapasowej znajdują się duże komunikaty.

Scenariusze awarii struktury

Scenariusze

Jeśli dla struktury CF zostanie zgłoszone niepowodzenie, działanie wykonywane przez połączone menedżery kolejek zależy od następujących elementów:

- Typ niepowodzenia zgłoszony przez komponent XES z/OS do IBM MQ.
- Typ struktury (aplikacja lub administracja)
- Poziom menedżera kolejek
- CFLEVEL dla obiektu IBM MQ CFSTRUCT (2, 3, 4 lub 5). Nie jest to poziom CFLEVEL mikro kodu CFCC)
- Atrybut RECAUTO dla obiektu IBM MQ CFSTRUCT na CFLEVEL (5)

Poniższe scenariusze opisują, co dzieje się w przypadku zgłoszenia awarii dla struktury administracyjnej:

- Jeśli dla struktury administracyjnej zostanie odebrane zdarzenie niepowodzenia struktury, struktura zostanie automatycznie ponownie przydzielona i odbudowana bez kończenia działania menedżera kolejek. Nowa instancja struktury jest przydzielana przez XES, gdy menedżer kolejek próbuje nawiązać z nią połączenie.

Gdy menedżer kolejek nawiąże połączenie z nową instancją struktury, zapisuje pozycje dla siebie w strukturze. To przetwarzanie jest wykonywane przez menedżer kolejek i nie jest częścią przetwarzania odbudowywania XES.

Jeśli menedżer kolejek nie był uruchomiony w momencie awarii lub zakończył działanie przed zakończeniem odtwarzania jego części struktury administracyjnej, pozycje struktury administracyjnej mogą zostać odbudowane przez innego menedżera kolejek w grupie współużytkowania kolejek.

Pozycje struktury administracyjnej menedżera kolejek mogą być odbudowywane tylko przez innego menedżera kolejek działającego na tym samym lub wyższym poziomie. Jeśli pozycje struktury administracyjnej menedżera kolejek nie mogą zostać odbudowane przez innego menedżera kolejek w grupie współużytkowania kolejek, zrestartuj menedżer kolejek, aby mógł zakończyć odbudowywanie swojej części struktury.

Niektóre działania są zawieszane do czasu odbudowania pozycji struktury administracyjnej dla wszystkich menedżerów kolejek. Do zawieszonych działań należą:

- Otwieranie i zamykanie kolejek współużytkowanych.

- Zatwierdzanie lub wycofywanie jednostek odzyskiwania.
- Aplikacje przekształcone do postaci szeregowej nawiązujące połączenie z menedżerem kolejek lub rozłączające się z menedżerem kolejek.
- Tworzenie kopii zapasowej lub odtwarzanie struktury aplikacji.

Wszystkie aplikacje przekształcone do postaci szeregowej, które są już połączone z menedżerem kolejek, mogą kontynuować przetwarzanie. Każda aplikacja przekształcona do postaci szeregowej, która próbuje połączyć się z parametrem MQCNO_SERIALIZE_CONN_TAG_QSG lub MQCNO_RESTRICT_CONN_TAG_QSG, otrzyma kod powrotu MQRC_CONN_TAG_NOT_USABLE.

Po odbudowaniu pozycji struktury administracyjnej dla menedżera kolejek zawieszono działania są wznawiane.

Poniższe scenariusze opisują, co dzieje się w przypadku zgłoszenia awarii dla struktury aplikacji:

- Jeśli dla struktury aplikacji zostanie odebrane zdarzenie niepowodzenia struktury, a CFLEVEL ma wartość 1 lub 2, menedżer kolejek kończy działanie. Zrestartuj menedżer kolejek. Pierwszy menedżer kolejek, który próbuje ponownie nawiązać połączenie ze strukturą, powoduje, że serwer XES przydziela nową instancję struktury.
- Jeśli dla struktury aplikacji zostanie odebrane zdarzenie niepowodzenia struktury, a CFLEVEL ma wartość 3, 4 lub 5, menedżery kolejek połączone ze strukturą nadal będą działać. Aplikacje, które nie używają kolejek w uszkodzonej strukturze, mogą kontynuować normalne przetwarzanie.

Jednak aplikacje, które próbują wykonać operacje na kolejkach w strukturze zakończonej niepowodzeniem, otrzymują błąd MQRC_CF_STRUC_FAILED do czasu pomyślnego odbudowania struktury, w którym to momencie aplikacja może ponownie otworzyć kolejki.

Odbudowa struktury jest uruchamiana automatycznie dla struktur aplikacji CFLEVEL (5) zdefiniowanych za pomocą RECAUTO (YES). W przeciwnym razie struktura zostanie odbudowana po wydaniu komendy RECOVER CFSTRUCT.

Odporność na awarie połączeń narzędzia CF

Czym jest odporność na awarie połączeń narzędzia CF?

Odporność na awarie połączeń narzędzia CF oznacza zdolność menedżerów kolejek w grupie współużytkowania kolejek do tolerowania utraty połączenia ze strukturą narzędzia CF bez końca pracy. Ta funkcja próbuje również odbudować strukturę w innym narzędziu CF z lepszą łącznością w celu jak najszybszego odzyskania dostępu do współużytkowanych kolejek.

Co to jest częściowa utrata połączenia?

IBM MQ definiuje częściową utratę połączenia jako sytuację, w której co najmniej jeden system w syspleksie traci połączenie z narzędziem CF, w którym struktura, do której system ma dostęp, jest przydzielona, ale co najmniej jeden system w syspleksie utrzymuje połączenie z tym samym narzędziem CF.

Co to jest całkowita utrata połączenia?

IBM MQ definiuje całkowitą utratę połączenia jako sytuację, w której żaden system w syspleksie nie ma połączenia z narzędziem CF i przydzieloną w nim strukturę.

Po co włączać tę funkcję?

Odporność na awarie połączeń narzędzia CF zwiększa dostępność produktu IBM MQ, co umożliwia pozostawienie niewspółużytkowanych kolejek dostępnych po utracie połączenia menedżera kolejek z co najmniej jedną strukturą narzędzia CF. Ponadto menedżery kolejek, które utracą połączenie ze strukturą narzędzia CF, automatycznie podejmą próbę odbudowania struktury w innym dostępnym narzędziu CF, zwiększając dostępność kolejek współużytkowanych w grupie współużytkowania kolejek.

Uwagi dotyczące włączania tej funkcji

Menedżer kolejek, który toleruje utratę połączenia ze strukturami narzędzia CF bez zakończenia, może przez pewien czas nie być w stanie ponownie nawiązać połączenia ze strukturą narzędzia CF, jeśli nie jest dostępne alternatywne narzędzie CF. Kolejki współużytkowane zdefiniowane w strukturze, która utraciła połączenie, pozostają niedostępne do czasu przywrócenia połączenia ze strukturą. W takiej sytuacji aplikacje łączące się z elementami grupy współużytkowania kolejek w celu wykonania pracy z kolejką współużytkowaną mogą stwierdzić, że kolejki współużytkowane, do których mają dostęp, nie są dostępne. Aby uniknąć takiej sytuacji, zaleca się skonfigurowanie kończenia działania menedżerów kolejek w przypadku utraty połączenia ze strukturą narzędzia CF. To zakończenie wymusza na aplikacjach nawiązanie połączenia z innym elementem grupy współużytkowania kolejek, który ma połączenie ze strukturami narzędzia CF, w których są zdefiniowane kolejki współużytkowane wymagane przez aplikację.

Zarządzanie odpornością na awarie połączeń narzędzia CF

Jak włączyć tę funkcję?

Aby włączyć odporność na połączenia narzędzia CF, należy wykonać następujące kroki:

1. Upewnij się, że zestaw danych pary CFRM został sformatowany do obsługi odbudowywania zarządzanego przez system. Umożliwia to menedżerom kolejek inicjowanie odbudowywania zarządzanego przez system w celu ponownego utworzenia struktury w dostępnym narzędziu CF. Użyj komendy **DISPLAY XCF, COUPLE, TYPE=CFRM**, aby określić format zestawu danych pary CFRM. Aby zapewnić obsługę odbudowywania zarządzanego przez system, należy sformatować zestaw danych pary CFRM, określając:

```
"ITEM NAME(SMREBLD) NUMBER(1) "
```

Więcej informacji na temat formatowania zestawu danych pary CFRM zawiera dokumentacja [z/OS MVS Setting Up a Sysplex](#).

2. Upewnij się, że alternatywne narzędzie CF jest dostępne i znajduje się na liście preferencji CFRM dla wszystkich struktur narzędzia CF IBM MQ. Umożliwia to menedżerom kolejek próbę odbudowania struktur do alternatywnego dostępnego narzędzia CF w celu jak najszybszego przywrócenia dostępu do struktur.

Struktury IBM MQ muszą być zdefiniowane z parametrem ENFORCEORDER (NO) w strategii CFRM, aby system XCF mógł wybrać optymalny system CF w konfiguracji, jeśli produkt IBM MQ ma ponownie przydzielić strukturę.

Więcej informacji na temat list preferencji struktury zawiera dokumentacja [z/OS MVS Setting Up a Sysplex](#).

3. Zmień wszystkie struktury narzędzia CF aplikacji, które muszą tolerować utratę połączenia z CFLEVEL (5). Jest to minimalny poziom, który może tolerować utratę połączenia.
4. Określ wartości wymagane dla atrybutów **QMGR CFCONLOS** i **CFSTRUCT CFCONLOS**, a następnie odpowiednio je zmodyfikuj. Atrybut **QMGR CFCONLOS** określa, czy utrata połączenia ze strukturą administracyjną jest tolerowana, a atrybut **CFSTRUCT CFCONLOS** określa, czy utrata połączenia jest tolerowana przez każdą strukturę narzędzia CF. Jeśli wartości domyślne tych atrybutów zostaną zachowane, menedżer kolejek kończy działanie po utracie połączenia z dowolną strukturą narzędzia CF.
5. Określ wartości wymagane dla atrybutu **CFSTRUCT RECAUTO** dla każdej struktury narzędzia CF aplikacji i odpowiednio je zmodyfikuj. Ten atrybut określa, czy struktury narzędzia CF powinny być automatycznie odzyskiwane przy użyciu zarejestrowanych danych po całkowitej utracie połączenia. Jeśli wartość domyślna tego atrybutu zostanie zachowana, nie jest wykonywane automatyczne odtwarzanie dla struktur aplikacji po całkowitej utracie połączenia.

Scenariusz 1-utrata połączenia ze strukturą administracyjną

Menedżery kolejek mogą tolerować utratę połączenia ze strukturą administracyjną bez przerywania pracy.

Jeśli połączenie ze strukturą administracyjną zostanie utracone przez dowolny menedżer kolejek, który został skonfigurowany do tolerowania utraty połączenia ze strukturą administracyjną, wszystkie elementy grupy współużytkowania kolejek zostaną rozłączone ze strukturą administracyjną. Wszystkie aktywne menedżery kolejek w grupie współużytkowania kolejek podejmą próbę ponownego nawiązania połączenia ze strukturą administracyjną, co spowoduje jej ponowne przydzielenie w narzędziu CF z najlepszą łącznością ze wszystkimi systemami w syspleksie i odbudowanie danych struktury administracyjnej.

Uwaga: Niekoniecznie musi to być narzędzie CF, które ma najlepszą łączność ze wszystkimi systemami, w których znajdują się aktywne menedżery kolejek.

Jeśli menedżer kolejek nie może ponownie nawiązać połączenia ze strukturą administracyjną, na przykład z powodu braku dostępnych narzędzi CF na liście preferencji CFRM dla struktury administracyjnej, niektóre współużytkowane operacje kolejki pozostają niedostępne do czasu, aż menedżer kolejek będzie mógł pomyślnie ponownie nawiązać połączenie ze strukturą administracyjną i odbudować dane struktury administracyjnej. Ponowne połączenie następuje automatycznie, gdy odpowiednie narzędzie CF stanie się dostępne w systemie.

Brak połączenia ze strukturą administracyjną podczas uruchamiania menedżera kolejek z powodu braku połączenia z narzędziem CF lub brak odpowiedniego narzędzia CF dostępnego do przydzielania struktury nie jest tolerowany. Wszystkie aktywne menedżery kolejek w grupie współużytkowania kolejek podejmą próbę ponownego nawiązania połączenia ze strukturą administracyjną, co spowoduje ponowne przydzielenie jej w innym narzędziu CF, jeśli jest dostępna, i odbudowanie danych struktury administracyjnej.

Scenariusz 2-utrata połączenia ze strukturą aplikacji

Utrata połączenia ze strukturami aplikacji w wersji **CFLEVEL (5)** lub nowszej może być tolerowana bez kończenia działania menedżera kolejek. Menedżery kolejek połączone ze strukturami aplikacji w wersji **CFLEVEL (4)** lub niższej albo ze strukturami w wersji **CFLEVEL (5)**, które nie zostały skonfigurowane do tolerowania utraty połączenia, kończą się awaryjnie z kodem przyczyny 00C510AB w przypadku utraty połączenia ze strukturą.

Po utracie połączenia ze strukturą aplikacji, która została skonfigurowana do tolerowania utraty połączenia, wszystkie menedżery kolejek, które utraciły połączenie ze strukturą, zostaną rozłączone. Kolejne zachowanie menedżera kolejek zależy od tego, czy utrata połączenia jest częściowa, czy łączna.

Częściowa utrata połączenia ze strukturą aplikacji

Jeśli utrata połączenia zostanie określona jako częściowa, menedżery kolejek, które utraciły połączenie ze strukturą, podejmą próbę zainicjowania odbudowy zarządzanej przez system w celu przeniesienia struktury do innego narzędzia CF z poprawnymi połączeniami. Jeśli odbudowanie powiedzie się, zarówno trwałe, jak i nietrwałe komunikaty w strukturze zostaną skopiowane do innego narzędzia CF, a dostęp do kolejek w strukturze zostanie odtworzony. Menedżery kolejek, które nie utraciły połączenia, pozostają połączone ze strukturą, jednak operacje, które uzyskują dostęp do struktury, są opóźniane podczas procesu odbudowywania zarządzanego przez system.

Jeśli struktury aplikacji nie można odbudować do innego narzędzia CF z poprawnymi połączeniami lub niektóre menedżery kolejek nadal nie mają połączenia ze strukturą po jej odbudowaniu w innym narzędziu CF, kolejki zdefiniowane w strukturze pozostają niedostępne w menedżerach kolejek, które nie mają połączenia ze strukturą, dopóki nie zostanie przywrócone połączenie z narzędziem CF. Menedżery kolejek automatycznie ponownie nawiązują połączenie ze strukturą, gdy stanie się ona dostępna, a dostęp do współużytkowanych kolejek zdefiniowanych w strukturze zostanie odtworzony.

Łączna utrata połączenia ze strukturą aplikacji

Jeśli wszystkie systemy MVS w syspleksie utracą połączenie z narzędziem CF, w którym jest przydzielona struktura aplikacji, program z/OS zwalnia strukturę z narzędzia CF przy każdej próbie ponownego nawiązania połączenia ze strukturą. Menedżer kolejek może podjąć próbę ponownego nawiązania połączenia ze strukturą z kilku powodów, takich jak próba otwarcia kolejki współużytkowanej przez aplikację lub powiadomienie z systemu o tym, że nowe zasoby narzędzia CF

mogą być dostępne. Dlatego jest prawdopodobne, że wszystkie nietrwałe komunikaty w strukturze, której to dotyczy, zostaną utracone po całkowitej utracie połączenia ze strukturą aplikacji.

Odtwarzalne struktury aplikacji są automatycznie odzyskiwane po całkowitej utracie połączenia, jeśli zostały zdefiniowane w produkcie **RECAUTO (YES)**. Odtwarzanie rozpoczyna się prawie natychmiast, jeśli dostępne jest alternatywne narzędzie CF, w którym można przydzielić strukturę lub gdy takie narzędzie CF stanie się dostępne. Jeśli struktura nie została zdefiniowana za pomocą parametru **RECAUTO (YES)**, odtwarzanie można uruchomić za pomocą komendy **RECOVER CFSTRUCT**. Spowoduje to odzyskanie wszystkich trwałych komunikatów w strukturze, ale wszystkie nietrwałe komunikaty zostaną utracone. Ponieważ ten proces obejmuje odczytywanie dziennika menedżera kolejek, jego wykonanie może zająć trochę czasu, dlatego zaleca się regularne tworzenie kopii zapasowych struktury w celu skrócenia czasu do momentu odtworzenia dostępu do współużytkowanych kolejek w strukturze.

Menedżery kolejek podejmują próbę ponownego nawiązania połączenia z nienaprawialnymi strukturami aplikacji, gdy tylko aplikacja próbuje otworzyć współużytkowaną kolejkę zdefiniowaną w strukturze lub gdy z systemu zostanie odebrane powiadomienie o udostępnieniu nowych zasobów narzędzia CF. Jeśli dostępne jest odpowiednie narzędzie CF, w którym można przydzielić strukturę, przydzielana jest nowa struktura i przywracany jest dostęp do współużytkowanych kolejek zdefiniowanych w strukturze. Ponieważ komunikaty trwałe nie mogą być umieszczane w kolejkach zdefiniowanych w strukturach nienaprawialnych, wszystkie komunikaty w kolejkach współużytkowanych są tracone.

Zachowanie operacyjne

Jeśli menedżer kolejek produktu IBM WebSphere MQ 7.1 lub nowszego skonfigurowany pod kątem tolerowania utraty połączenia z określoną strukturą narzędzia CF utraci połączenie, elementy grupy współużytkowania kolejek podejmą próbę automatycznego odtworzenia po awarii i ponownego nawiązania połączenia ze strukturą. To działanie może obejmować zmianę przydziału struktury w innym narzędziu CF z lepszą łącznością, jeśli jest ono dostępne. Jednak interwencja operatora może być nadal wymagana w celu odzyskania sprawności po utracie połączenia.

Zwykle wymagane działanie operatora ma na celu:

1. Usunąć przyczynę niepowodzenia, która spowodowała utratę połączenia.
2. Upewnić się, że narzędzie CF, w którym można przydzielić struktury IBM MQ, jest dostępne we wszystkich systemach w sysplexie.

Wszystkie struktury, które zostały automatycznie ponownie przydzielone w innym narzędziu CF po utracie połączenia, można przenieść do narzędzia CF z optymalnym połączeniem do wszystkich menedżerów kolejek w grupie współużytkowania kolejek. Jeśli jest to wymagane, można to zrobić, inicjując komendę odbudowywania zarządzaną przez system **SETXCF START, REBUILD** zgodnie z opisem w sekcji [Skorowidz komend systemowych z/OS Serwer MVS](#).

W przypadku częściowej utraty połączenia ze strukturą aplikacji menedżery kolejek, które utraciły połączenie ze strukturą, podejmą próbę zainicjowania odbudowy zarządzanej przez system. Ten proces przydziela strukturę w innym narzędziu CF tylko wtedy, gdy to narzędzie CF ma połączenie ze wszystkimi aktywnymi menedżerami kolejek połączonymi obecnie ze strukturą. Dlatego możliwe jest, że gdy większość menedżerów kolejek w grupie współużytkowania kolejek utraciła połączenie ze strukturą aplikacji, nie są one w stanie odbudować struktury do innego narzędzia CF z powodu menedżerów kolejek, które są nadal połączone z pierwotną strukturą. W takiej sytuacji menedżery kolejek, które są nadal połączone z oryginalną strukturą, można zamknąć, aby umożliwić odbudowanie struktury, lub można wydać komendę **RESET CFSTRUCT ACTION (FAIL)**, aby struktura zakończyła się niepowodzeniem. Odtwarzanie można zainicjować dla odpowiednich struktur, wydając komendę **RECOVER CFSTRUCT**.

Uwaga: W przypadku niepowodzenia i odtwarzania struktury wszystkie nietrwałe komunikaty w strukturze są tracone.

Ten temat zawiera informacje o znaczeniu zabezpieczeń dla systemu IBM MQ oraz o konsekwencjach braku odpowiednich ustawień zabezpieczeń w systemie.

Dlaczego należy chronić zasoby IBM MQ

Program IBM MQ obsługuje przesyłanie informacji, które są potencjalnie cenne. Zastosowanie zabezpieczeń zapewnia, że zasoby, których właścicielem jest IBM MQ i którymi zarządza, są chronione przed dostępem bez uprawnień. Taki dostęp może prowadzić do utraty lub ujawnienia informacji.

Należy upewnić się, że żaden z następujących zasobów nie jest dostępny ani zmieniany przez nieautoryzowanego użytkownika lub proces:

- Połączenia z IBM MQ
- Obiekty IBM MQ, takie jak kolejki, procesy i listy nazw
- IBM MQ łączy transmisji, czyli IBM MQ kanały
- IBM MQ komendy sterujące systemem
- Komunikaty produktu IBM MQ
- Informacje o kontekście powiązane z komunikatami

Aby zapewnić niezbędne zabezpieczenia, produkt IBM MQ używa narzędzia autoryzacji systemu z/OS (SAF) do kierowania żądań autoryzacji do zewnętrznego menedżera zabezpieczeń (ESM), na przykład serwera zabezpieczeń (wcześniej znanego jako RACF). Produkt IBM MQ nie przeprowadza własnej weryfikacji zabezpieczeń. W przypadku używania rozproszonego kolejkowania lub klientów może być konieczne zastosowanie dodatkowych zabezpieczeń, w przypadku których produkt IBM MQ udostępnia rekordy uwierzytelniania kanału, wyjścia kanału, atrybut kanału MCAUSER i protokół TLS.

Decyzja o zezwoleniu na dostęp do obiektu jest podejmowana przez menedżera ESM, a program IBM MQ następuje po tej decyzji. Jeśli menedżer ESM nie może podjąć decyzji, program IBM MQ uniemożliwia dostęp do obiektu.

Co się stanie, jeśli zasoby IBM MQ nie będą chronione

W przypadku braku działań dotyczących bezpieczeństwa najbardziej prawdopodobnym efektem jest to, że wszyscy użytkownicy mogą uzyskiwać dostęp i zmieniać *każdy* zasób. Dotyczy to nie tylko użytkowników lokalnych, ale także tych w systemach zdalnych używających rozproszonego kolejkowania lub klientów, w przypadku których zabezpieczenia logowania mogą być mniej rygorystyczne niż w przypadku systemu z/OS.

Aby włączyć sprawdzanie zabezpieczeń, należy wykonać następujące czynności:

- Zainstaluj i aktywuj moduł ESM (na przykład Security Server).
- Zdefiniuj klasę MQADMIN, jeśli używany jest menedżer ESM inny niż serwer zabezpieczeń.
- Aktywuj klasę MQADMIN.

Należy rozważyć, czy użycie nazw zasobów zawierających wielkie i małe litery będzie korzystne dla przedsiębiorstwa. Jeśli w profilach ESM używane są nazwy zasobów zawierające wielkie i małe litery, należy zdefiniować i aktywować klasę MXADMIN.

z/OS szyfrowanie zestawu danych

Szyfrowanie zestawu danych (DSE) umożliwia szyfrowanie zestawów danych z/OS, dzięki czemu zawarte w nich dane mogą być wyświetlane lub modyfikowane tylko przez identyfikatory użytkowników, którym nadano konkretne uprawnienie. Zapewnia to szyfrowanie przechowywanych danych w systemie plików i zapobiega niezamierzonemu ujawnieniu poufnych informacji użytkownikom, którzy mają uzasadnione potrzeby biznesowe i uprawnienia do samodzielnego zarządzania zestawami danych.

W wersjach wcześniejszych niż IBM MQ for z/OS 9.1.4 produkt IBM MQ for z/OS nie obsługiwał funkcji DSE z aktywnymi dziennikami, zestawami stron i współużytkowanymi zestawami danych komunikatów (SMDS), które udostępniają podstawowe mechanizmy trwałości dla komunikatów IBM MQ .

Zamiast tego produkt *Advanced Message Security* udostępnia kompleksowe rozwiązanie do szyfrowania na potrzeby przesyłania komunikatów w produkcie IBM MQ , które obejmuje całą sieć produktu IBM MQ , szyfrowanie przesyłanych danych, w spoczynku, a nawet wewnątrz procesów środowiska wykonawczego IBM MQ .

Inne VSAM i sekwencyjne zestawy danych używane w podsystemie IBM MQ mogą być szyfrowane za pomocą DSE. Na przykład:

- zestaw danych programu startowego
- Sekwencyjne pliki zawierające komendy konfiguracji systemu (MQSC) odczytywane podczas uruchamiania przy użyciu nazw DDNAME CSQINPX
- Dzienniki archiwalne IBM MQ , często używane do długoterminowego archiwizowania danych dziennika IBM MQ na potrzeby kontroli.

Można szyfrować za pomocą DSE, przydzielając klasę danych zdefiniowaną za pomocą etykiety klucza zestawu danych. Więcej informacji na ten temat zawiera sekcja [Planowanie pamięci archiwalnej dziennika](#).

Począwszy od wersji IBM MQ for z/OS 9.1.4, IBM MQ for z/OS obsługuje użycie funkcji DSE z aktywnymi dziennikami i zestawami stron oprócz obsługi zapewnianej we wcześniejszych wersjach.

IBM MQ for z/OS nie obsługuje DSE dla współużytkowanych zestawów danych komunikatów (SMDS).

Patrz sekcja [poufność danych przechowywanych w systemie IBM MQ for z/OS z szyfrowaniem zestawu danych](#) . :NONE.

Pojęcia pokrewne

[Pojęcia związane z bezpieczeństwem](#)

[Rekordy uwierzytelniania kanału](#)

[Uprawnienia do pracy z obiektami IBM MQ w systemie z/OS](#)

[Szyfrujące protokoły bezpieczeństwa: TLS](#)

Zadania pokrewne

[Konfigurowanie zabezpieczeń w systemie z/OS](#)

[Porównywanie zabezpieczeń na poziomie łącza i zabezpieczeń na poziomie aplikacji](#)

Odsyłacze pokrewne

[Komunikaty dla IBM MQ for z/OS](#)

Elementy sterujące i opcje zabezpieczeń w produkcie IBM MQ for z/OS

Można określić, czy zabezpieczenia są włączone dla całego podsystemu IBM MQ oraz czy mają być wykonywane sprawdzenia zabezpieczeń na poziomie menedżera kolejek lub grupy współużytkowania kolejek. Można również kontrolować liczbę identyfikatorów użytkowników sprawdzanych pod kątem ochrony zasobów interfejsu API.

Bezpieczeństwo podsystemu

Zabezpieczenia podsystemu to element sterujący, który określa, czy dla całego menedżera kolejek jest wykonywane sprawdzanie zabezpieczeń. Jeśli nie jest wymagane sprawdzanie zabezpieczeń (na przykład w systemie testowym) lub jeśli poziom zabezpieczeń wszystkich zasobów, które mogą nawiązywać połączenie z produktem IBM MQ (w tym klientów i kanałów) jest zadowalający, można wyłączyć sprawdzanie zabezpieczeń dla menedżera kolejek lub grupy współużytkowania kolejek, aby nie było wykonywane dalsze sprawdzanie zabezpieczeń.

Jest to jedyne sprawdzenie, które może całkowicie wyłączyć ochronę i określić, czy są wykonywane inne sprawdzenia bezpieczeństwa. Oznacza to, że jeśli sprawdzanie dla menedżera kolejek lub grupy współużytkowania kolejek zostanie wyłączone, nie będzie wykonywane żadne inne sprawdzanie IBM MQ. Jeśli to pole pozostanie włączone, produkt IBM MQ sprawdzi wymagania bezpieczeństwa dla innych zasobów produktu IBM MQ.

Można również włączyć lub wyłączyć zabezpieczenia dla określonych zestawów zasobów, takich jak komendy.

Sprawdzanie na poziomie menedżera kolejek lub grupy współużytkowania kolejek

Zabezpieczenia można zaimplementować na poziomie menedżera kolejek lub na poziomie grupy współużytkowania kolejki. Jeśli zabezpieczenia są implementowane na poziomie grupy współużytkowania kolejki, wszystkie menedżery kolejek w grupie współużytkują te same profile. Oznacza to, że istnieje mniej profili do zdefiniowania i obsługi, co ułatwia zarządzanie bezpieczeństwem. Ułatwia również dodanie nowego menedżera kolejek do grupy współużytkowania kolejek, ponieważ dziedziczy on istniejące profile zabezpieczeń.

Możliwe jest również zaimplementowanie kombinacji obu tych elementów, jeśli instalacja tego wymaga, na przykład podczas migracji lub jeśli w grupie współużytkowania kolejek istnieje jeden menedżer kolejek, który wymaga innych poziomów zabezpieczeń niż inne menedżery kolejek w grupie.

Zabezpieczenia na poziomie grupy współużytkowania kolejki

Sprawdzanie zabezpieczeń na poziomie grupy współużytkowania kolejek jest wykonywane dla całej grupy współużytkowania kolejek. Pozwala to uprościć administrowanie zabezpieczeniami, ponieważ wymaga zdefiniowania mniejszej liczby profili zabezpieczeń. Autoryzacja ID użytkownika do korzystania z określonego zasobu jest obsługiwana na poziomie grupy współużytkowania kolejek i jest niezależna od tego, który menedżer kolejek jest używany przez ten ID użytkownika do uzyskiwania dostępu do zasobu.

Na przykład można powiedzieć, że aplikacja serwera działa z identyfikatorem użytkownika SERVER i chce uzyskać dostęp do kolejki o nazwie SERVER.REQUEST, a użytkownik chce uruchomić instancję SERVER na każdym obrazie z/OS w sysplexie. Zamiast zezwalać serwerowi na otwieranie serwera SERVER.REQUEST dla każdego menedżera kolejek osobno (zabezpieczenia na poziomie menedżera kolejek), dostęp można zezwolić tylko na poziomie grupy współużytkowania kolejek.

Profile zabezpieczeń na poziomie grupy współużytkowania kolejek można używać do ochrony wszystkich typów zasobów, zarówno lokalnych, jak i współużytkowanych.

zabezpieczenia na poziomie menedżera kolejek

Profile zabezpieczeń na poziomie menedżera kolejek mogą być używane do ochrony wszystkich typów zasobów, zarówno lokalnych, jak i współużytkowanych.

Połączenie obu poziomów

Można użyć kombinacji zarówno menedżera kolejek, jak i zabezpieczeń na poziomie grupy współużytkowania kolejek.

Ustawienia zabezpieczeń na poziomie grupy współużytkowania kolejki można przestonić dla konkretnego menedżera kolejek, który jest członkiem tej grupy. Oznacza to, że można wykonać inne sprawdzanie poziomu zabezpieczeń dla pojedynczego menedżera kolejek niż w przypadku innych menedżerów kolejek w grupie.

Więcej informacji na ten temat zawiera sekcja [Profile do sterowania grupą współużytkowania kolejek lub zabezpieczeniami na poziomie menedżera kolejek](#).

Kontrolowanie liczby sprawdzanych identyfikatorów użytkowników

RESLEVEL to profil serwera bezpieczeństwa, który steruje liczbą identyfikatorów użytkowników sprawdzanych pod kątem bezpieczeństwa zasobów IBM MQ. Zwykle, gdy użytkownik próbuje uzyskać

dostęp do zasobu IBM MQ , serwer Security Server sprawdza odpowiedni identyfikator lub identyfikatory użytkownika, aby sprawdzić, czy dostęp do tego zasobu jest dozwolony. Definiując profil RESLEVEL, można określić, czy ma być sprawdzany zero, jeden, czy, jeśli ma to zastosowanie, dwa identyfikatory użytkowników.

Te elementy sterujące są wykonywane na podstawie połączenia i są wykonywane przez cały czas trwania połączenia.

Dla każdego menedżera kolejek istnieje tylko jeden profil RESLEVEL. Sterowanie jest implementowane przez dostęp, który ma użytkownik o danym identyfikatorze do tego profilu.

Klasy IBM MQ RACF z mieszaną wielkością liter lub wielkimi literami

Obecnie można używać obsługi profili RACF z mieszanymi wielkimi literami, co pozwala na używanie nazw zasobów z mieszanymi wielkimi literami i definiowanie profili produktu IBM MQ RACF w celu ich ochrony.

Można wybrać jedną z następujących opcji:

- Kontynuuj używanie tylko wielkich liter w klasach IBM MQ RACF , tak jak w poprzednich wersjach, lub
- Użyj nowych klas IBM MQ RACF z mieszaną wielkością liter.

Bez użycia profili RACF z mieszaną wielkością liter nadal można używać nazw zasobów z mieszaną wielkością liter w produkcie IBM MQ for z/OS ; Jednak te nazwy zasobów mogą być chronione tylko przez ogólne profile RACF w klasach IBM MQ pisanych wielkimi literami. W przypadku korzystania z obsługi profili produktu IBM MQ RACF o różnej wielkości liter można zapewnić bardziej szczegółowy poziom ochrony, definiując profile produktu IBM MQ RACF w klasach IBM MQ o różnej wielkości liter.

z/OS Zasoby, które można chronić w programie IBM MQ for z/OS

Podczas uruchamiania menedżera kolejek lub na żądanie komendy operatora program IBM MQ for z/OS określa, które zasoby mają być chronione.

Użytkownik może sterować tym, które sprawdzenia zabezpieczeń są wykonywane dla poszczególnych menedżerów kolejek. Na przykład można zaimplementować pewną liczbę sprawdzeń zabezpieczeń w produkcyjnym menedżerze kolejek, ale nie można ich zaimplementować w testowym menedżerze kolejek.

Bezpieczeństwo połączenia

Sprawdzanie zabezpieczeń połączenia jest wykonywane, gdy aplikacja próbuje nawiązać połączenie z menedżerem kolejek. W tym celu należy wysłać żądanie MQCONN lub MQCONNX albo wtedy, gdy inicjator kanału lub adapter CICS lub IMS wysyła żądanie połączenia.

Jeśli używane są zabezpieczenia na poziomie menedżera kolejek, można wyłączyć sprawdzanie zabezpieczeń połączenia dla konkretnego menedżera kolejek. Jednak w takim przypadku dowolny użytkownik może nawiązać połączenie z tym menedżerem kolejek.

W przypadku adaptera CICS do sprawdzenia zabezpieczeń połączenia używany jest tylko identyfikator użytkownika przestrzeni adresowej CICS , a nie pojedynczy identyfikator użytkownika terminalu CICS . W przypadku adaptera IMS , gdy element sterujący IMS lub regiony zależne łączą się z serwerem IBM MQ, sprawdzany jest identyfikator użytkownika przestrzeni adresowej IMS . Dla inicjatora kanału sprawdzany jest identyfikator użytkownika używany przez przestrzeń adresową inicjatora kanału.

Sprawdzanie zabezpieczeń połączenia można włączyć lub wyłączyć na poziomie menedżera kolejek lub grupy współużytkownika kolejek.

Bezpieczeństwo komend

Sprawdzanie bezpieczeństwa komend jest wykonywane, gdy użytkownik wywoła komendę MQSC z dowolnego źródła opisanego w sekcji [Źródła](#), z których można wywołać komendy MQSC i PCF w systemie IBM MQ for z/OS. Można wykonać oddzielne sprawdzenie zasobu określonego przez komendę zgodnie z opisem w sekcji ["Ochrona zasobów komend"](#) na stronie 284.

Jeśli sprawdzanie komend zostanie wyłączone, wystawcy komend nie będą sprawdzani, aby sprawdzić, czy mają uprawnienia do wydania komendy.

Jeśli komendy MQSC są wprowadzane z konsoli, konsola musi mieć atrybut uprawnień konsoli SYS z/OS. Komendy wydawane z zestawów danych CSQINP1 lub CSQINP2 albo wewnętrznie przez menedżer kolejek są wyłączone ze sprawdzania zabezpieczeń, podczas gdy komendy dla CSQINPX używają ID użytkownika przestrzeni adresowej inicjatora kanału. Należy kontrolować, kto może aktualizować te zestawy danych za pomocą normalnej ochrony zestawu danych.

Sprawdzanie zabezpieczeń komend można włączyć lub wyłączyć na poziomie menedżera kolejek lub grupy współużytkowania kolejek.

Ochrona zasobów komend

Niektóre komendy MQSC, na przykład definiujące kolejkę lokalną, wymagają manipulowania zasobami IBM MQ. Gdy ochrona zasobów komend jest aktywna, za każdym razem, gdy wydawana jest komenda obejmująca zasób, program IBM MQ sprawdza, czy użytkownik może zmienić definicję tego zasobu.

Ochrona zasobów komend może być pomocna w wymuszaniu standardów nazewnictwa. Na przykład administrator listy płac może usuwać i definiować tylko kolejki o nazwach rozpoczynających się od łańcucha "PAYROLL". Jeśli ochrona zasobów komendy jest nieaktywna, nie są wykonywane żadne operacje sprawdzania zabezpieczeń dla zasobu, który jest przetwarzany przez komendę. Nie należy mylić ochrony zasobów komend z ochroną komend; obie są niezależne.

Wyłączenie sprawdzania ochrony zasobów przez komendy nie ma wpływu na sprawdzanie zasobów, które jest wykonywane specjalnie dla innych typów przetwarzania, które nie są związane z komendami.

Sprawdzanie bezpieczeństwa zasobów komend można włączyć lub wyłączyć na poziomie menedżera kolejek lub grupy współużytkowania kolejek.

Zagadnienia dotyczące bezpieczeństwa kanału

Zabezpieczenia kanału

Jeśli używane są kanały, dostępne opcje ochrony zależą od tego, który protokół komunikacyjny będzie używany. Jeśli używany jest protokół TCP, z protokołem komunikacyjnym nie są dostarczane żadne opcje zabezpieczające, ale można użyć protokołu TLS. Jeśli używany jest protokół APPC, można przestać informacje o identyfikatorze użytkownika z wysyłającego agenta MCA przez sieć do docelowego agenta MCA w celu weryfikacji.

W przypadku obu protokołów można określić identyfikatory użytkowników, które mają być sprawdzane ze względów bezpieczeństwa, oraz określić ich liczbę. Opcje dostępne dla użytkownika zależą od używanego protokołu, od tego, co określono podczas definiowania kanału oraz od ustawień RESLEVEL dla inicjatora kanału.

Więcej informacji na temat dostępnych typów zabezpieczeń kanału zawiera sekcja [Rekordy uwierzytelniania kanału](#) i sekcja [Przegląd wyjścia zabezpieczeń](#).

Odsyłacze pokrewne

["Bezpieczeństwo zasobów interfejsu API w produkcie IBM MQ for z/OS"](#) na stronie 285

Zasoby są sprawdzane, gdy aplikacja otwiera obiekt za pomocą wywołania MQOPEN lub MQPUT1. Dostęp wymagany do otwarcia obiektu zależy od tego, jakie opcje otwarcia są określone podczas otwierania kolejki.

Zasoby są sprawdzane, gdy aplikacja otwiera obiekt za pomocą wywołania MQOPEN lub MQPUT1 . Dostęp wymagany do otwarcia obiektu zależy od tego, jakie opcje otwarcia są określone podczas otwierania kolejki.

Zabezpieczenia zasobów interfejsu API są podzielone na następujące sprawdzenia:

- [kolejka](#)
- [Proces](#)
- [Lista nazw](#)
- [Alternatywny użytkownik](#)
- [Kontekst](#)

Podczas otwierania obiektu menedżera kolejek lub podczas uzyskiwania dostępu do obiektów klasy pamięci nie są wykonywane żadne sprawdzenia zabezpieczeń.

Kolejka

Sprawdzanie zabezpieczeń kolejki steruje tym, kto może otwierać kolejkę i jakie opcje może otwierać tę kolejkę. Na przykład użytkownik może mieć możliwość otwarcia kolejki o nazwie PAYROLL.INCREASE.SALARY w celu przeglądania komunikatów w kolejce (za pomocą opcji MQOO_BROWSE), ale nie w celu usuwania komunikatów z kolejki (za pomocą jednej z opcji MQOO_INPUT_*). Jeśli sprawdzanie kolejek zostanie wyłączone, każdy użytkownik może utworzyć dowolną kolejkę z dowolną poprawną opcją otwarcia (czyli dowolną poprawną opcją MQOO_* w wywołaniu MQOPEN lub MQPUT1).

Sprawdzanie zabezpieczeń kolejki można włączyć lub wyłączyć na poziomie menedżera kolejek lub grupy współużytkowania kolejek.

Proces

Sprawdzanie bezpieczeństwa procesu jest wykonywane, gdy użytkownik otwiera obiekt definicji procesu. Jeśli sprawdzanie procesów zostanie wyłączone, każdy użytkownik może utworzyć dowolny proces.

Sprawdzanie zabezpieczeń procesu można włączyć lub wyłączyć na poziomie menedżera kolejek lub grupy współużytkowania kolejek.

Lista nazw

Sprawdzanie zabezpieczeń listy nazw jest wykonywane, gdy użytkownik otwiera listę nazw. Jeśli sprawdzanie list nazw zostanie wyłączone, każdy użytkownik może utworzyć dowolną listę nazw.

Sprawdzanie zabezpieczeń listy nazw można włączyć lub wyłączyć na poziomie menedżera kolejek lub grupy współużytkowania kolejek.

Alternatywny użytkownik

Alternatywne zabezpieczenia użytkownika określają, czy jeden ID użytkownika może używać uprawnień innego ID użytkownika do otwierania obiektu IBM MQ .

Na przykład:

- Program serwera uruchomiony dla ID użytkownika PAYSERV pobiera komunikat żądania z kolejki, która została umieszczona w kolejce przez ID użytkownika USER1.
- Gdy program serwera otrzyma komunikat żądania, przetwarza żądanie i umieszcza odpowiedź z powrotem w kolejce odpowiedzi określonej w komunikacie żądania.
- Zamiast używać własnego identyfikatora użytkownika (PAYSERV) do autoryzowania otwierania kolejki odpowiedzi, serwer może określić inny identyfikator użytkownika, w tym przypadku

USER1. W tym przykładzie alternatywne zabezpieczenia użytkownika określają, czy ID użytkownika PAYSERV może określać ID użytkownika USER1 jako alternatywny ID użytkownika podczas otwierania kolejki odpowiedzi.

Alternatywny identyfikator użytkownika jest określony w polu *AlternateUserId* deskryptora obiektu (MQOD).

Można użyć alternatywnych identyfikatorów użytkowników dla dowolnego obiektu IBM MQ, na przykład procesów lub list nazw. Nie ma to wpływu na identyfikator użytkownika używany przez inne menedżery zasobów, na przykład dla zabezpieczeń CICS lub dla zabezpieczeń zestawu danych z/OS.

Jeśli alternatywne zabezpieczenia użytkownika nie są aktywne, każdy użytkownik może użyć innego identyfikatora jako alternatywnego identyfikatora użytkownika.

Alternatywne sprawdzanie zabezpieczeń użytkownika można włączyć lub wyłączyć na poziomie menedżera kolejek lub grupy współużytkowania kolejek.

Kontekst

Kontekst to informacje, które mają zastosowanie do konkretnego komunikatu i są zawarte w deskrytorze komunikatu (MQMD) będącym częścią komunikatu. Informacje o kontekście znajdują się w dwóch sekcjach:

Sekcja Tożsamość

Użytkownik aplikacji, która jako pierwsza umieściła komunikat w kolejce. Składa się z następujących pól:

- *UserIdentifier*
- *AccountingToken*
- *ApplIdentityData*

Sekcja pochodzenia

Aplikacja, która umieściła komunikat w kolejce, w której jest obecnie przechowywany. Składa się z następujących pól:

- *PutApplType*
- *PutApplName*
- *PutDate*
- *PutTime*
- *ApplOriginData*

Aplikacje mogą określać dane kontekstu podczas wykonywania wywołania MQPUT lub MQPUT1. Aplikacja może generować dane, dane mogą być przekazywane z innego komunikatu lub menedżer kolejek może domyślnie generować dane. Na przykład programy serwera mogą użyć danych kontekstowych do sprawdzenia tożsamości requestera, to znaczy, czy ten komunikat pochodzi z poprawnej aplikacji? Zwykle pole *UserIdentifier* jest używane do określenia identyfikatora użytkownika alternatywnego.

Zabezpieczenia kontekstu służą do określania, czy użytkownik może określić dowolną z opcji kontekstu w dowolnym wywołaniu MQOPEN lub MQPUT. Więcej informacji na temat opcji kontekstu zawiera sekcja [Opcje MQOPEN dotyczące kontekstu komunikatu](#). Opisy pól deskryptora komunikatu dotyczących kontekstu znajdują się w sekcji [MQMD-Message descriptorMQMD -deskryptor komunikatu](#).

Jeśli sprawdzanie zabezpieczeń kontekstu zostanie wyłączone, każdy użytkownik może użyć dowolnej opcji kontekstu, na którą zezwala ochrona kolejki.

Sprawdzanie zabezpieczeń kontekstu można włączyć lub wyłączyć na poziomie kolejki, menedżera kolejek lub grupy współużytkowania kolejki.

IBM MQ for z/OS oferuje wiele funkcji wysokiej dostępności. W tej sekcji opisano niektóre zagadnienia związane z dostępnością.

Kilka funkcji programu IBM MQ może zwiększyć dostępność systemu w przypadku awarii menedżera kolejek lub inicjatora kanału. Więcej informacji na temat tych funkcji zawierają następujące sekcje:

- [Uwagi dotyczące systemu Sysplex](#)
- [Kolejki współużytkowane](#)
- [Kanały współużytkowane](#)
- [Dostępność sieci w systemie IBM MQ](#)
- [Korzystanie z menedżera z/OS Automatic Restart Manager \(ARM\)](#)
- [Korzystanie z narzędzia XRF \(z/OS Extended Recovery Facility\)](#)
- [Używanie atrybutu z/OS GROUPUR do odtwarzania w grupie współużytkowania kolejek](#)
- [Gdzie można znaleźć więcej informacji na temat dostępności](#)

Uwagi dotyczące systemu Sysplex

W systemie *sysplex* wiele obrazów systemów operacyjnych z/OS współdziela w jednym obrazie systemu i komunikuje się za pomocą narzędzia CF. Produkt IBM MQ może używać narzędzi środowiska *sysplex* w celu zwiększenia dostępności.

Usunięcie powinowactwa między menedżerem kolejek a konkretnym obrazem produktu z/OS umożliwia zrestartowanie menedżera kolejek na innym obrazie produktu z/OS w przypadku awarii obrazu. Mechanizm restartowania może być ręczny, używać ARM lub używać automatyzacji systemu, jeśli spełnione są następujące warunki:

- Wszystkie zestawy stron, dzienniki, zestawy danych programu startowego, biblioteki kodu i zestawy danych konfiguracji menedżera kolejek są definiowane na woluminach współużytkowanych.
- Definicja podsystemu ma zasięg *sysplex* i unikalną nazwę w obrębie *sysplex*.
- Poziom *wczesnego kodu* instalowanego na każdym obrazie systemu z/OS w czasie IPL jest taki sam.
- Wirtualne adresy IP TCP (VIPA) są dostępne na każdym stosie TCP w *sysplexie*, a programy nasłuchujące TCP i połączenia przychodzące systemowi IBM MQ zostały skonfigurowane tak, aby używały wirtualnych adresów IP zamiast domyślnych nazw hostów.

Więcej informacji na temat używania protokołu TCP w *sysplexie* zawiera publikacja *TCP/IP in a sysplex*, SG24-5235, IBM Redbooks .

Dodatkowo można skonfigurować wiele menedżerów kolejek działających na różnych obrazach systemu operacyjnego w *sysplexie*, tak aby działały jako grupa współużytkowania kolejek, która może korzystać ze współużytkowanych kolejek i kanałów w celu zwiększenia dostępności i równoważenia obciążenia.

Kolejki współużytkowane

W środowisku grupy współużytkowania kolejek aplikacja może nawiązać połączenie z dowolnym menedżerem kolejek w grupie współużytkowania kolejek. Ponieważ wszystkie menedżery kolejek w grupie współużytkowania kolejek mogą uzyskać dostęp do tego samego zestawu kolejek współużytkowanych, aplikacja nie zależy od dostępności konkretnego menedżera kolejek; każdy menedżer kolejek w grupie współużytkowania kolejek może obsługiwać dowolną kolejkę. Zapewnia to większą dostępność, jeśli menedżer kolejek zostanie zatrzymany, ponieważ wszystkie inne menedżery kolejek w grupie współużytkowania kolejek mogą kontynuować przetwarzanie kolejki. Informacje na temat wysokiej dostępności kolejek współużytkowanych zawiera sekcja [“Zalety korzystania z kolejek współużytkowanych”](#) na stronie 194.

W celu dalszego zwiększenia dostępności komunikatów w grupie współużytkowania kolejek program IBM MQ wykrywa, czy inny menedżer kolejek w grupie rozłącza się nieprawidłowo od narzędzia CF i w miarę możliwości wykonuje jednostki pracy dla tego menedżera kolejek, które nadal oczekują na wykonanie. Jest to nazywane *odtworzeniem równorzędnym* i jest opisane w sekcji [“Odtwarzanie równorządne”](#) na stronie 273.

Funkcja odtwarzania równorzędnego nie może odzyskać jednostek pracy, które były niepewne w momencie awarii. Za pomocą menedżera automatycznego restartowania (ARM) można zrestartować wszystkie systemy uczestniczące w awarii (na przykład CICS, Db2i IBM MQ) oraz upewnić się, że wszystkie systemy zostały zrestartowane na tym samym nowym procesorze. Oznacza to, że mogą one resynchronizować i szybko odzyskiwać wątpliwe jednostki pracy. Zostało to opisane w sekcji [“Korzystanie z menedżera automatycznego restartowania \(Automatic Restart Manager-ARM\) systemu z/OS”](#) na stronie 288.

Kanały współużytkowane

W środowisku grupy współużytkowania kolejek produkt IBM MQ udostępnia funkcje, które zapewniają wysoką dostępność w sieci. Inicjator kanału umożliwia korzystanie z produktów sieciowych, które równoważą żądania sieciowe z zestawu zakwalifikowanych serwerów i ukrywają awarie serwerów przed siecią (na przykład zasoby ogólne VTAM). Produkt IBM MQ używa portu ogólnego dla żądań przychodzących, aby żądania przyłączenia mogły być kierowane do dowolnego dostępnego inicjatora kanału w grupie współużytkowania kolejek. Zostało to opisane w sekcji [“Kanały współużytkowane”](#) na stronie 216.

Współużytkowane kanały wychodzące pobierają komunikaty wysyłane przez nie ze współużytkowanej kolejki transmisji. Informacje o statusie kanału współużytkowanego są przechowywane w jednym miejscu dla całego poziomu grupy współużytkowania kolejki. Oznacza to, że kanał może zostać automatycznie zrestartowany w innym inicjatorze kanału w grupie współużytkowania kolejek w przypadku awarii inicjatora kanału, menedżera kolejek lub podsystemu komunikacyjnego. Jest to nazywane *odtworzeniem kanału równorzędnego* i jest opisane w sekcji [Współużytkowane kanały wychodzące](#).

Dostępność sieci IBM MQ

Komunikaty produktu IBM MQ są przenoszone z menedżera kolejek do menedżera kolejek w sieci produktu IBM MQ za pomocą kanałów. Istnieje możliwość zmiany konfiguracji na wielu poziomach w celu zwiększenia dostępności sieciowej menedżera kolejek oraz zdolności kanału produktu IBM MQ do wykrywania problemów z siecią i do ponownego nawiązywania połączenia.

Opcja *Keepalive* TCP jest dostępna dla kanałów TCP/IP. Powoduje ona, że TCP okresowo wysyła pakiety między sesjami w celu wykrycia awarii sieci. Atrybut kanału KAINTE określa częstotliwość tych pakietów dla kanału.

AdoptMCA umożliwia zakończenie kanału zablokowanego w przetwarzaniu odbierania w wyniku wyłączenia sieci i zastąpienie go nowym żądaniem połączenia. *AdoptMCA* można sterować za pomocą właściwości menedżera kolejek *adoptMCA* z programem narzędziowym MQSC lub za pomocą właściwości *AdoptNewMCAType* z interfejsem formatów komend programowalnych.

Parametr *ReceiveTimeout* zapobiega trwałemu blokowaniu kanału w wywołaniu odbierania sieciowego. Parametry inicjatora kanału RCVTIME i RCVTMIN określają parametry limitu czasu odbierania dla kanałów jako funkcję interwału pulsu. Więcej informacji na ten temat zawiera sekcja [Parametr menedżera kolejek](#) .

Korzystanie z menedżera automatycznego restartowania (Automatic Restart Manager-ARM) systemu z/OS

Parametr IBM MQ for z/OS może być używany w połączeniu z parametrem menedżera automatycznego restartowania (ARM) systemu z/OS . Jeśli menedżer kolejek lub inicjator kanału uległ awarii, menedżer ARM restartuje go na tym samym obrazie z/OS . Jeśli wystąpi awaria systemu z/OS , cała grupa

powiązanych podsystemów i aplikacji również ulegnie awarii. Menedżer ARM może automatycznie zrestartować wszystkie systemy, które uległy awarii, w predefiniowanej kolejności, na innym obrazie z/OS w obrębie syspleksu. Jest to nazywane restartem międzysystemowym.

ARM umożliwia szybkie odtwarzanie wątpliwych transakcji w środowisku kolejki współużytkowanej. Zapewnia również wyższą dostępność, jeśli nie są używane grupy współużytkowania kolejek.

Za pomocą ARM można zrestartować menedżer kolejek na innym obrazie z/OS w syspleksie w przypadku awarii z/OS .

Aby włączyć automatyczne restartowanie, należy wykonać następujące czynności:

1. Skonfiguruj zestaw danych sprzężenia ARM.
2. Zdefiniuj działania automatycznego restartowania, które program z/OS ma wykonać w *strategii ARM*.
3. Uruchom strategię ARM.

Jeśli menedżery kolejek w różnych obrazach programu z/OS mają być restartowane automatycznie, każdy menedżer kolejek w każdym obrazie systemu z/OS , w którym ten menedżer kolejek może być restartowany, musi być zdefiniowany z unikalną 4-znakową nazwą podsystemu dla całego syspleksu.

Użycie ARM z IBM MQ zostało opisane w sekcji [Używanie ARM w sieci IBM MQ](#).

Korzystanie z narzędzia z/OS Extended Recovery Facility (XRF)

Programu IBM MQ można używać w środowisku XRF (Extended Recovery Facility). Wszystkie zestawy danych należące do IBM MQ(kod wykonywalny, zestawy BSD, dzienniki i zestawy stron) muszą znajdować się na DASD współużytkowanym przez aktywne i alternatywne procesory XRF.

Jeśli do odtwarzania używane jest narzędzie XRF, należy zatrzymać menedżer kolejek w aktywnym procesorze i uruchomić go w procesorze alternatywnym. W systemie CICSmożna to zrobić za pomocą tabeli listy komend (CLT) udostępnionej przez CICS lub operator systemu może to zrobić ręcznie. W przypadku systemu IMSjest to operacja ręczna, którą należy wykonać po zakończeniu przełączania procesora przez system koordynujący IMS .

Przed przełączeniem menedżera kolejek na procesor alternatywny należy zakończyć lub zakończyć działanie programów narzędziowych IBM MQ . Należy dokładnie rozważyć wpływ tej potencjalnej przerwy podczas planowania planów odtwarzania XRF.

Należy zachować ostrożność, aby zapobiec uruchomieniu menedżera kolejek w procesorze alternatywnym przed zakończeniem działania menedżera kolejek w aktywnym procesorze. Przedwczesne uruchomienie może spowodować poważne problemy z integralnością danych, katalogu i dziennika. Użycie globalnej serializacji zasobów (GRS) pomaga uniknąć problemów z integralnością, uniemożliwiając jednocześnie użycie produktu IBM MQ w obu systemach. Należy dołączyć BSDS jako zasób chroniony oraz włączyć aktywne i alternatywne procesory XRF do pierścienia GRS.

Używanie atrybutu z/OS GROUPUR do odtwarzania w grupie współużytkowania kolejek

Grupy współużytkowania kolejek (QSG) umożliwiają korzystanie z dodatkowych narzędzi transakcyjnych opisanych w tym temacie. Atrybut GROUPUR umożliwia aplikacjom klienckim XA wykonywanie na dowolnym elemencie grupy QSG dowolnego wątpliwego odtwarzania transakcji, które może być wymagane.

Jeśli aplikacja kliencka XA łączy się z grupą współużytkowania kolejek (QSG) za pośrednictwem systemu Sysplex, nie może zagwarantować, z którym konkretnym menedżerem kolejek się łączy. Użycie atrybutu GROUPUR przez menedżery kolejek w grupie współużytkowania kolejek może umożliwić dowolne wątpliwe odtwarzanie transakcji, które może być konieczne w dowolnym elemencie grupy QSG. Nawet jeśli menedżer kolejek, z którym aplikacja była początkowo połączona, nie jest dostępny, może zostać przeprowadzone odtwarzanie transakcji.

Ta funkcja zwalnia aplikację kliencką XA z zależności od konkretnych elementów grupy QSG, a tym samym zwiększa dostępność menedżera kolejek. Grupa współużytkowania kolejek jest wyświetlana dla aplikacji transakcyjnej jako pojedynczy obiekt udostępniający wszystkie funkcje produktu IBM MQ i bez pojedynczego punktu awarii menedżera kolejek.

Ta funkcja nie jest widoczna dla aplikacji transakcyjnej.

Gdzie można znaleźć więcej informacji o dostępności

Więcej informacji na ten temat można znaleźć w następujących źródłach:

<i>Tabela 24. Gdzie można znaleźć więcej informacji o dostępności</i>	
Temat	Gdzie szukać
Grupy współużytkowania kolejek	“Kolejki współużytkowane i grupy współużytkowania kolejek” na stronie 173
Parametry systemu	Konfigurowanie parametrów systemowych
Korzystanie z menedżera automatycznego restartowania Programy narzędziowe	Używanie ARM w sieci IBM MQ
Komendy MQSC	Komendy MQSC

z/OS

Monitorowanie i statystyki w systemie IBM MQ for z/OS

Produkt IBM MQ for z/OS zawiera zestaw narzędzi do monitorowania menedżera kolejek i zbierania statystyk.

IBM MQ udostępnia narzędzia do monitorowania systemu i gromadzenia statystyk. Więcej informacji na temat tych narzędzi można znaleźć w następujących sekcjach:

- [“Monitorowanie bezpośrednie” na stronie 290](#)
- [“IBM MQ ślad” na stronie 290](#)
- [“Zdarzenia” na stronie 291](#)

Monitorowanie bezpośrednie

IBM MQ zawiera następujące komendy służące do monitorowania statusu obiektów IBM MQ :

- Komenda DISPLAY CHSTATUS wyświetla status określonego kanału.
- Komenda DISPLAY QSTATUS wyświetla status określonej kolejki.
- DISPLAY CONN wyświetla status określonego połączenia.

Więcej informacji na temat tych komend zawiera sekcja [Komendy MQSC](#).

IBM MQ ślad

Program IBM MQ udostępnia narzędzie śledzenia, którego można użyć do gromadzenia następujących informacji podczas działania menedżera kolejek:

Statystyki wydajności

Śledzenie statystyk zbiera następujące informacje, aby pomóc w monitorowaniu wydajności i dostrojeniu systemu:

- Liczba różnych żądań MQI (statystyki menedżera komunikatów)
- Liczba różnych żądań obiektów (statystyki menedżera danych)
- Informacje o wykorzystaniu Db2 (statystyki menedżera Db2)
- Informacje o wykorzystaniu narzędzia CF (statystyki menedżera narzędzia CF)
- Informacje o wykorzystaniu SMDS (statystyki współużytkowanego zestawu danych komunikatów)
- Informacje o wykorzystaniu puli buforów (statystyki menedżera buforów)
- Informacje o rejestrowaniu (statystyki menedżera rejestrowania)
- Informacje o wykorzystaniu pamięci masowej (statystyki menedżera pamięci masowej)
- Informacje o żądaniach blokad (statystyki menedżera blokad)

Dane księgowe

- Śledzenie rozliczania zbiera informacje na temat czasu procesora przeznaczonego na przetwarzanie wywołań MQI oraz na temat liczby żądań MQPUT i MQGET wykonanych przez danego użytkownika.
- Program IBM MQ może również gromadzić informacje o każdym zadaniu za pomocą programu IBM MQ. Te dane są gromadzone jako rekord rozliczeniowy na poziomie wątku. Dla każdego wątku program IBM MQ gromadzi również informacje o każdej kolejce używanej przez ten wątek.

Dane wygenerowane przez śledzenie są wysyłane do narzędzia SMF (System Management Facility) lub narzędzia GTF (Generalized Trace Facility).

Zdarzenia

Zdarzenia IBM MQ udostępniają informacje o błędach, ostrzeżeniach i innych istotnych wystąpieniach w menedżerze kolejek. Umieszczając te zdarzenia we własnej aplikacji do zarządzania systemem, można monitorować działania wielu menedżerów kolejek dla wielu aplikacji IBM MQ. W szczególności można monitorować wszystkie menedżery kolejek w systemie z poziomu pojedynczego menedżera kolejek.

Zdarzenia mogą być zgłaszane przez mechanizm raportowania napisany przez użytkownika do aplikacji administracyjnej, która obsługuje prezentację zdarzeń operatorowi. Zdarzenia umożliwiają również aplikacjom działającym jako agenty dla innych sieci administracyjnych, na przykład NetView, monitorowanie raportów i tworzenie odpowiednich alertów.

Zadania pokrewne

[Używanie usługi śledzenia produktu IBM MQ](#)

[Korzystanie ze zdarzeń IBM MQ](#)

z/OS

Dyspozycja jednostki odzyskiwania w systemie z/OS

W przypadku połączenia z menedżerem kolejek w grupie współużytkowania kolejek (QSG) niektóre aplikacje transakcyjne mogą używać dyspozycji jednostki odzyskiwania GROUP, a nie QMGR, przez określenie nazwy QSG podczas nawiązywania połączenia zamiast nazwy menedżera kolejek. Dzięki temu odtwarzanie transakcji będzie bardziej elastyczne i stabilne, jeśli zostanie usunięte wymaganie ponownego nawiązania połączenia z tym samym menedżerem kolejek w grupie QSG.

Transakcje uruchomione przez aplikacje połączone za pomocą nazwy grupy współużytkowania kolejek również mają dyspozycję jednostki odtwarzania GROUP.

Gdy aplikacja transakcyjna łączy się z dyspozycją jednostki odtwarzania GROUP, jest ona logicznie połączona z grupą współużytkowania kolejek i nie ma powinowactwa do żadnego konkretnego menedżera kolejek. Wszystkie uruchomione transakcje zatwierdzania dwufazowego, które zakończyły phase-1 procesu zatwierdzania, to znaczy są wątpliwe, mogą zostać zapytane i rozstrzygnięte po nawiązaniu połączenia z dowolnym menedżerem kolejek w obrębie grupy QSG. W scenariuszu odtwarzania oznacza to, że koordynator transakcji nie musi ponownie nawiązywać połączenia z tym samym menedżerem kolejek, który może być w tym momencie niedostępny.

Aplikacje, które łączą się z jednostką odzyskiwania QMGR, mają bezpośrednie powinowactwo z menedżerem kolejek, z którym są połączone. W scenariuszu odtwarzania koordynator transakcji musi ponownie nawiązać połączenie z tym samym menedżerem kolejek, aby rozstrzygnąć wszelkie wątpliwe transakcje, niezależnie od tego, czy menedżer kolejek należy do grupy współużytkowania kolejek.

Gdy aplikacje określają nazwę grupy współużytkowania kolejek, a tym samym łączą się z menedżerem kolejek w grupie QSG z dyspozycją jednostki odzyskiwania GROUP, grupa współużytkowania kolejek jest logicznie oddzielnym menedżerem zasobów. Oznacza to, że transakcje wątpliwe są widoczne dla aplikacji tylko wtedy, gdy łączy się ona ponownie z tą samą jednostką rozporządzenia odzyskiwaniem. Transakcje wątpliwe z dyspozycją jednostki odzyskiwania QMGR nie są widoczne dla aplikacji, które połączyli się z dyspozycją jednostki odzyskiwania GROUP i odwrotnie.

Pojęcia pokrewne

[“Włączanie jednostek odtwarzania GROUP” na stronie 292](#)

Grupa współużytkowania kolejek może skonfigurować i włączyć obsługę jednostek odtwarzania GROUP.

[“Obsługa aplikacji” na stronie 293](#)

Ta strona służy do określania, które aplikacje mogą łączyć się z jednostką odzyskiwania GROUP.

Włączanie jednostek odtwarzania GROUP

Grupa współużytkowania kolejek może skonfigurować i włączyć obsługę jednostek odtwarzania GROUP.

Aby użyć jednostek odtwarzania GROUP w menedżerze kolejek w obrębie grupy QSG, należy włączyć atrybut GROUPUR menedżera kolejek. Więcej informacji na temat tego pojęcia zawiera sekcja [“Dyspozycja jednostki odzyskiwania w systemie z/OS” na stronie 291](#) przed zapoznaniem się z resztą tego tematu.

Gdy atrybut GROUPUR menedżera kolejek jest włączony, menedżer kolejek akceptuje nowe połączenia z dyspozycją odtwarzania jednostki GROUP. Jeśli ten atrybut zostanie wyłączony, nowe połączenia z tym rozporządzeniem nie będą akceptowane, chociaż nie będzie to wpływało na już połączone aplikacje, dopóki nie zostaną rozłączone.

Gdy aplikacja łączy się z dyspozycją jednostki odzyskiwania GROUP i pyta o wątpliwe transakcje lub próbuje rozstrzygnąć transakcję, która została uruchomiona w innym miejscu grupy współużytkowania kolejek (QSG), menedżer kolejek, z którym jest teraz połączony, musi mieć możliwość komunikacji z innymi elementami grupy współużytkowania kolejek, aby mógł przetworzyć żądanie. W tym celu używana jest kolejka współużytkowana o nazwie SYSTEM.QSG.UR.RESOLUTION.QUEUE. Ta kolejka musi znajdować się w odtwarzalnej strukturze aplikacji o nazwie CSQSYSAPPL. Struktura musi być odtwarzalna, ponieważ komunikaty trwałe są przechowywane w tej kolejce podczas przetwarzania żądań rozstrzygnięcia.

Przed włączeniem jednostek odtwarzania GROUP należy upewnić się, że struktura narzędzia CF i kolejka współużytkowana są zdefiniowane. Można użyć definicji z przykładu CSQ4INSS. Gdy kolejka jest zdefiniowana lub wykryta podczas uruchamiania, każdy menedżer kolejek w grupie współużytkowania kolejek otwiera kolejkę, aby mogła odbierać przychodzące żądania. Aby usunąć lub przenieść kolejkę, ponieważ została ona zdefiniowana niepoprawnie, można zażądać, aby menedżery kolejek zamkły w niej otwarte uchwyty, aktualizując obiekt kolejki w celu zablokowania żądań MQGET. Po wprowadzeniu niezbędnych poprawek zezwolenie aplikacjom na pobieranie komunikatów z kolejki po raz kolejny nakazuje każdemu menedżerowi kolejek ponowne otwarcie tej kolejki. Użyj komendy DISPLAY QSTATUS, aby określić, które uchwyty są otwarte w kolejce.

Po zakończeniu tej konfiguracji można włączyć jednostki odtwarzania GROUP dla każdego menedżera kolejek, z którym aplikacje transakcyjne mają się łączyć przy użyciu dyspozycji jednostki odtwarzania GROUP. Nie muszą to być wszystkie menedżery kolejek w grupie współużytkowania kolejek, ale jeśli ta funkcja ma zostać włączona tylko w podzbiorze grupy współużytkowania kolejek, należy upewnić się, że aplikacje podejmą tylko próbę nawiązania połączenia z menedżerami kolejek, w których włączono tę funkcję. Więcej informacji na ten temat zawiera [“Obsługa aplikacji” na stronie 293](#).

Podczas próby włączenia atrybutu GROUPUR menedżera kolejek wykonywana jest pewna liczba operacji sprawdzania konfiguracji. Menedżer kolejek sprawdza, czy:

- Należy do grupy współużytkowania kolejek.
- Kolejka współużytkowana o nazwie SYSTEM.QSG.UR.RESOLUTION.QUEUE została zdefiniowana zgodnie z definicją w CSQ4INSS.
- SYSTEM SYSTEM.QSG.UR.RESOLUTION.QUEUE znajduje się w odtwarzalnej strukturze CF o nazwie CSQSYSAPPL.

Jeśli którekolwiek z powyższych sprawdzeń nie powiedzie się, atrybut GROUPUR pozostanie wyłączony i zostanie zwrócony kod komunikatu.

Te sprawdzenia konfiguracji są również wykonywane podczas uruchamiania menedżera kolejek, jeśli atrybut menedżera kolejek jest włączony. Jeśli którakolwiek ze sprawdzeń nie powiedzie się podczas uruchamiania jednostki odtwarzania GROUP, zostanie wyłączona, a menedżer kolejek wyśle komunikat z informacją o tym, które sprawdzenie nie powiodło się. Po wykonaniu niezbędnych czynności naprawczych należy ponownie włączyć atrybut menedżera kolejek.

Obsługa aplikacji

Ta strona służy do określania, które aplikacje mogą łączyć się z jednostką odzyskiwania GROUP.

Obsługa dyspozycji jednostki odtwarzania GROUP jest ograniczona do pewnych typów aplikacji transakcyjnych, dla których IBM MQ for z/OS jest menedżerem zasobów, ale nie jest koordynatorem transakcji. Obecnie obsługiwane są następujące aplikacje transakcyjne:

- IBM MQ rozszerzone transakcyjne aplikacje klienckie
- Aplikacje IBM MQ classes for JMS działające na serwerze aplikacji, takim jak WebSphere Application Server.
- Aplikacje CICS działające w systemie CICS Transaction Server 4.2 lub nowszym, gdy definicja zasobu CICS MQCONN jest skonfigurowana z opcją RESYNCMEMBER (GROUPRESYNC).

Pojęcia pokrewne

[“IBM MQ rozszerzone transakcyjne aplikacje klienckie” na stronie 293](#)

Ta strona służy do określenia, w jaki sposób aplikacje rozszerzonego klienta transakcyjnego IBM MQ mogą korzystać z dyspozycji jednostki odtwarzania GROUP.

[“Aplikacje CICS” na stronie 294](#)

Ta strona służy do określania, w jaki sposób produkt CICS może korzystać z dyspozycji GROUP jednostek odzyskiwania.

IBM MQ rozszerzone transakcyjne aplikacje klienckie

Ta strona służy do określenia, w jaki sposób aplikacje rozszerzonego klienta transakcyjnego IBM MQ mogą korzystać z dyspozycji jednostki odtwarzania GROUP.

Przykładem rozszerzonej aplikacji klienckiej transakcyjnej IBM MQ jest aplikacja, która używa języka JMS i działa w produkcie WebSphere Application Server, łącząc się z serwerem IBM MQ za pośrednictwem protokołu TCP/IP, a nie powiązań lokalnych. Te aplikacje klienckie łączą się z serwerem IBM MQ for z/OS za pośrednictwem połączeń sieciowych, takich jak TCP/IP. W przypadku tych aplikacji jest to wartość określona dla parametru QMNAME łańcucha xa_info przekazanego w wywołaniu xa_open, która określa, czy ma być używana jednostka QMGR, czy GROUP. Więcej informacji na temat xa_open zawiera sekcja [Format łańcucha xa_open](#) i sekcja [Dodatkowe przetwarzanie błędów dla xa_open](#). W przypadku aplikacji JMS jest to wykonywane przez określenie nazwy grupy współużytkowania kolejek (QSG) w ConnectionFactory zamiast nazwy konkretnego menedżera kolejek.

Aby aplikacje klienckie XA mogły korzystać z dyspozycji jednostki odtwarzania GROUP, należy skonfigurować protokół TCP/IP w taki sposób, aby aplikacje klienckie mogły być kierowane do menedżerów kolejek w grupie współużytkowania kolejek z włączonym atrybutem GROUPUR, a nie do konkretnego menedżera kolejek. Jedną z technologii dynamicznych wirtualnych adresów IP, której można użyć w tym celu, jest dystrybutor z/OS SysPlex . Więcej informacji na ten temat zawierają [Serwer komunikacji z/OS i z/OS Podstawowe umiejętności: dynamiczne adresowanie wirtualne](#) . Aby włączyć jednostki odtwarzania GROUP dla podzbioru menedżerów kolejek w grupie współużytkowania kolejek, należy upewnić się, że aplikacje klienckie nie mogą być kierowane do tych, w których nie są włączone.

Aplikacje klienckie nie muszą łączyć się z grupą współużytkownika kolejek przy użyciu kanałów współużytkowanych.

Aplikacje CICS

Ta strona służy do określania, w jaki sposób produkt CICS może korzystać z dyspozycji GROUP jednostek odzyskiwania.

System CICS 4.2 lub nowszej udostępnia opcję resynchronizacji grupy, RESYNCMEMBER (GROUPRESYNC) w definicji zasobu MQCONN. CICS skonfigurowany z tą opcją może nawiązać połączenie z dowolnym odpowiednim menedżerem kolejek w grupie współużytkowania kolejek, która działa w tej samej partycji LPAR co region CICS. Aby obsłużyć opcję CICS GROUPRESYNC, menedżer kolejek musi być uruchomiony w produkcie MQ V7.1 lub nowszej i musi być włączony dla obsługi GROUPUR.

Transakcje działające w regionie CICS połączonym z produktem MQ za pomocą komendy GROUPRESYNC tworzą jednostki pracy z dyspozycją odtwarzania jednostki GROUP.

Można użyć komendy RESYNCMEMBER (GROUPRESYNC), aby umożliwić szybsze odtwarzanie po awarii menedżera kolejek, ponieważ umożliwia regionowi CICS natychmiastowe nawiązanie połączenia z alternatywnym zakwalifikowanym menedżerem kolejek działającym na tej samej partycji LPAR, rozstrzygając w razie potrzeby wszystkie transakcje wątpliwe bez oczekiwania na restart menedżera kolejek.

RESYNCMEMBER (GROUPRESYNC) włącza również bardziej elastyczne opcje restartu dla systemu CICS. Region CICS z połączeniem MQ skonfigurowany do używania kolejek współużytkowanych GROUPRESYNC i MQ może zostać zrestartowany w dowolnej partycji LPAR, w której istnieje menedżer kolejek działający jako element tej samej grupy współużytkowania kolejek.

IBM MQ i inne produkty z/OS

W tym temacie opisano, w jaki sposób program IBM MQ może współpracować z innymi produktami z/OS.

Pojęcia pokrewne

[“IBM MQ i CICS” na stronie 294](#)

Wszystkie wersje produktu CICS obsługiwane przez produkt IBM MQ 9.0.0i nowsze używają dostarczonej przez firmę CICS wersji adaptera i mostu.

[“IBM MQ for z/OS i WebSphere Application Server” na stronie 301](#)

Ten temat zawiera informacje dotyczące korzystania z języka IBM MQ for z/OS przez WebSphere Application Server.

Odsyłacze pokrewne

[“IBM MQ i IMS” na stronie 296](#)

W tym temacie opisano sposób współpracy produktu IBM MQ z produktem IMS. Adapter IMS umożliwia połączenie menedżera kolejek z produktem IMS i umożliwia aplikacjom produktu IMS korzystanie z interfejsu MQI.

[“IBM MQ i adaptory z/OS Batch, TSO i RRS” na stronie 300](#)

W tym temacie opisano, w jaki sposób produkt IBM MQ współpracuje z adapterami z/OS Batch, TSO i RRS.

IBM MQ i CICS

Wszystkie wersje produktu CICS obsługiwane przez produkt IBM MQ 9.0.0i nowsze używają dostarczonej przez firmę CICS wersji adaptera i mostu.

Więcej informacji na temat konfigurowania adaptera IBM MQ CICS i komponentów produktu IBM MQ CICS bridge zawiera sekcja [Konfigurowanie połączeń z produktem IBM MQ](#) w dokumentacji produktu CICS.

Zadania pokrewne

[Używanie produktu IBM MQ z produktem CICS](#)

CICS przyłączenie grupy

Przyłączenie grupy CICS umożliwia regionowi systemu CICS nawiązanie połączenia z dowolnym aktywnym elementem grupy współużytkowania kolejek systemu IBM MQ w tej samej partycji LPAR zamiast określania pojedynczego menedżera kolejek. Program CICS nadal łączy się z pojedynczym menedżerem kolejek w danym momencie.

Do obsługi przyłączania grup CICS wymagane są co najmniej dwa menedżery kolejek w partycji LPAR. Użycie przyłączenia do grupy zapewnia wyższą dostępność, ponieważ nie ma potrzeby, aby konkretny menedżer kolejek był aktywny. Program CICS łączy się z dowolnym menedżerem kolejek w grupie współużytkowania kolejek na partycji LPAR.

Więcej informacji na ten temat zawiera dokumentacja produktu CICS dotycząca zasobu MQCONN.

Program CICS próbuje nawiązać połączenie z produktem MQNAME przekazanym jako menedżer kolejek:

- Jeśli menedżer kolejek istnieje i jest aktywny, połączenie będzie działać.
- Jeśli nawiązanie połączenia nie powiedzie się, program CICS wysyła zapytanie o status menedżerów kolejek w grupie, aby sprawdzić, które menedżery kolejek są aktywne w tej samej partycji LPAR.
- Jeśli aktywnych jest wiele menedżerów kolejek, program CICS sprawdza status RESYNCMEMBER (YES) i UOW, aby określić, czy produkt CICS musi nawiązać połączenie z konkretnym elementem lub czy powinien nawiązać połączenie z tym elementem, czy nie jest aktywny.
- Jeśli nie ma potrzeby nawiązywania połączenia z konkretnym elementem, program CICS wybiera menedżera kolejek (przy użyciu algorytmu randomizacji).
- Program CICS próbuje nawiązać połączenie z wybranym menedżerem kolejek.
- Jeśli próba nie powiedzie się, w zależności od kodu powrotu CICS wybierze następny element, a następnie ponownie przejdzie przez pętlę wyboru.
- Jeśli żaden menedżer kolejek nie jest aktywny, program CICS wysyła wiele połączeń do listy menedżerów kolejek i oczekuje na liście ECBLIST, aż pierwszy menedżer kolejek stanie się dostępny.

Pojęcia pokrewne

“Jednostki odzyskiwania grupy (GROUPUR) dla systemu CICS” na stronie 295

Produkt IBM MQ GROUPUR for CICS umożliwia odtwarzanie równorzędne w przypadku wątpliwych jednostek pracy w grupie współużytkowania kolejek (QSG). Jeden menedżer kolejek systemu IBM MQ może rozstrzygać wątpliwe jednostki pracy w imieniu innego menedżera kolejek w grupie współużytkowania kolejek. Oznacza to, że jeśli program CICS łączy się ponownie przez przyłączenie grupy do innego menedżera kolejek w QSG, może rozstrzygać wątpliwe transakcje z poprzedniego połączenia IBM MQ .

Informacje pokrewne

Obsługa grup współużytkowania kolejek systemu IBM MQ

Jednostki odzyskiwania grupy (GROUPUR) dla systemu CICS

Produkt IBM MQ GROUPUR for CICS umożliwia odtwarzanie równorzędne w przypadku wątpliwych jednostek pracy w grupie współużytkowania kolejek (QSG). Jeden menedżer kolejek systemu IBM MQ może rozstrzygać wątpliwe jednostki pracy w imieniu innego menedżera kolejek w grupie współużytkowania kolejek. Oznacza to, że jeśli program CICS łączy się ponownie przez przyłączenie grupy do innego menedżera kolejek w QSG, może rozstrzygać wątpliwe transakcje z poprzedniego połączenia IBM MQ .

Jeśli region programu CICS działa z menedżerem kolejek, a menedżer kolejek kończy pracę nieprawidłowo, wszystkie transakcje wątpliwe są odzyskiwane. Eliminuje to konieczność oczekiwania przez region CICS na zrestartowanie menedżera kolejek, z którym pracował, a następnie rozstrzygania wszelkich wątpliwych jednostek pracy. Oznacza to, że potrzebne są co najmniej dwa menedżery kolejek na partycji LPAR, aby program CICS mógł nawiązać połączenie z innym menedżerem kolejek w przypadku nieprawidłowego zakończenia pierwszego menedżera kolejek.

Nowe ustawienie RESYNCMEMBER (GROUPRESYNC) w definicji MQCONN produktu CICS :

- Używa funkcji przyłączania grupy IBM MQ i odtwarzania równorzędnego.
- Wymaga menedżera kolejek z włączonym atrybutem GROUPUR.
- Nadal obsługuje istniejące ustawienia CICS MQCONN RESYNCMEMBER (YES i NO):
 - Używa istniejącej funkcji przyłączania grupy CICS i nie wykonuje odtwarzania równorzędnego.
 - Zmiana ustawień RESYNCMEMBER jest uwzględniana przy następnym połączeniu programu CICS z serwerem IBM MQ.

Pojęcia pokrewne

[“Włączanie jednostek odtwarzania GROUP” na stronie 292](#)

Grupa współużytkowania kolejek może skonfigurować i włączyć obsługę jednostek odtwarzania GROUP.

IBM MQ i IMS

W tym temacie opisano sposób współpracy produktu IBM MQ z produktem IMS. Adapter IMS umożliwia połączenie menedżera kolejek z produktem IMS i umożliwia aplikacjom produktu IMS korzystanie z interfejsu MQI.

Opcjonalny dodatkowy most IBM MQ - IMS umożliwia aplikacjom uruchamianie aplikacji produktu IMS, które nie używają interfejsu MQI. Oznacza to, że można używać wcześniejszych aplikacji z produktem IBM MQ bez konieczności ich przepisywania.

Więcej informacji na temat tych komponentów zawierają następujące podtematy:

Pojęcia pokrewne

[Aplikacje mostu IMS i IMS w systemie IBM MQ for z/OS](#)

Zadania pokrewne

[Konfigurowanie adaptera IMS](#)

[Konfigurowanie mostu IMS](#)

[Obsługa adaptera IMS](#)

Odsyłacze pokrewne

[MQIIH-nagłówek informacji IMS](#)

Adapter IMS

Adapter IMS jest interfejsem między aplikacjami IMS i podsystemem IBM MQ.

Adaptory IBM MQ umożliwiają różnym środowiskom aplikacji wysyłanie i odbieranie komunikatów za pośrednictwem sieci kolejkowania komunikatów. Adapter IMS jest interfejsem między aplikacjami IMS i podsystemem IBM MQ. Umożliwia ona aplikacjom IMS korzystanie z interfejsu MQI.

Adapter IMS odbiera i interpretuje żądania dostępu do systemu IBM MQ za pomocą narzędzia [External Subsystem Attach Facility \(ESAF\)](#) udostępnionego przez IMS. Zwykle program IMS łączy się automatycznie z programem IBM MQ bez interwencji operatora.

Adapter IMS zapewnia dostęp do zasobów IBM MQ dla programów działających w następujących trybach lub stanach:

- Tryb zadania (TCB)
- Stan problemu
- Tryb inny niż międzypamięciowy
- Tryb rejestru bez dostępu

Adapter udostępnia wątek połączenia z bloku kontrolnego zadania aplikacji (TCB) do bazy danych IBM MQ.

Adapter obsługuje protokół zatwierdzania dwufazowego w przypadku zmian w zasobach należących do produktu IBM MQ, w przypadku których IMS pełni rolę koordynatora punktu synchronizacji. Konwersacje, w których IMS nie jest koordynatorem punktu synchronizacji, na przykład konwersacje chronione przez APPC (SYNCLVL = SYNCP), nie są obsługiwane przez adapter IMS.

Adapter udostępnia również transakcję monitora wyzwalacza (CSQQTRMN). Zostało to opisane w sekcji [“Monitor wyzwalacza IMS” na stronie 297](#).

Funkcji IBM MQ można używać razem z narzędziem IMS Extended Recovery Facility (XRF) w celu wspomagania odtwarzania po wystąpieniu błędu systemu IMS .

Uwaga: Produkt IMS 15.2 Extended Recovery Facility (XRF) nie jest już obsługiwany. Więcej informacji na ten temat zawiera dokumentacja produktu [IMS](#) .

Korzystanie z adaptera

Aplikacje i adapter IMS działają w tej samej przestrzeni adresowej. Menedżer kolejek jest oddzielony, we własnej przestrzeni adresowej.

Należy połączyć wszystkie programy, które wywołują co najmniej jedno wywołanie MQI, z odpowiednim modułem interfejsu języka IMS i, jeśli nie używają dynamicznych wywołań MQI, z programem pośredniczącym API dostarczonym przez IBM MQ, CSQQSTUB. Gdy aplikacja wysyła wywołanie MQI, kod pośredniczący przekazuje sterowanie do adaptera za pośrednictwem zewnętrznego interfejsu podsystemu IMS , który zarządza przetwarzaniem żądania przez menedżer kolejek komunikatów.

Administrowanie systemem i jego działanie przy użyciu programu IMS

Autoryzowany operator terminalu IMS może wydawać komendy IMS w celu kontrolowania i monitorowania połączenia z systemem IBM MQ. Jednak operator terminalu IMS nie ma kontroli nad przestrzenią adresową IBM MQ . Na przykład operator nie może wyłączyć IBM MQ z przestrzeni adresowej IMS .

Ograniczenia

Następujące wywołania API IBM MQ nie są obsługiwane w aplikacji używającej adaptera IMS :

- Baza MQCB
- MQCB_FUNKCJA
- Komenda MQCTL

Monitor wyzwalacza IMS

Monitor wyzwalacza IMS (**CSQQTRMN**) jest dostarczoną przez IBM MQ aplikacją IMS , która uruchamia transakcję IMS po wystąpieniu zdarzenia IBM MQ , na przykład po umieszczeniu komunikatu w konkretnej kolejce.

Jak to działa?

Gdy komunikat jest umieszczany w kolejce komunikatów aplikacji, wyzwalacz jest generowany, jeśli spełnione są warunki wyzwalacza. Następnie menedżer kolejek zapisuje komunikat (zawierający dane zdefiniowane przez użytkownika), zwany *komunikatem wyzwalacza*, do kolejki inicjującej, która została określona dla tej kolejki komunikatów. W środowisku IMS można uruchomić instancję komendy CSQQTRMN w celu monitorowania kolejki inicjującej i pobrania z niej komunikatów wyzwalacza po ich odebraniu. Zwykle komenda CSQQTRMN planuje inną transakcję IMS za pomocą komendy INSERT (ISRT) do kolejki komunikatów IMS . Uruchomiona aplikacja IMS odczytuje komunikat z kolejki komunikatów aplikacji, a następnie przetwarza go. CSQQTRMN musi działać jako BMP bez komunikatu.

Każda kopia CSQQTRMN obsługuje pojedynczą kolejkę inicjującą. Po uruchomieniu monitor wyzwalacza działa do momentu zakończenia działania IBM MQ lub IMS .

Makro APPLCTN dla CSQQTRMN musi mieć wartość SCHDTYP=PARALLEL.

Ponieważ monitor wyzwalacza jest BMP zorientowany wsadowo, transakcje IMS , które są uruchamiane przez monitor wyzwalacza, zawierają następujące elementy:

- Puste pola w polu LTERM IOPCB
- Nazwa PSB monitora wyzwalacza BMP w polu ID użytkownika IOPCB

Jeśli docelowa transakcja IMS jest chroniona przez serwer Security Server (wcześniej znany jako RACF), może być konieczne zdefiniowanie CSQQTRMN jako ID użytkownika na serwerze Security Server.

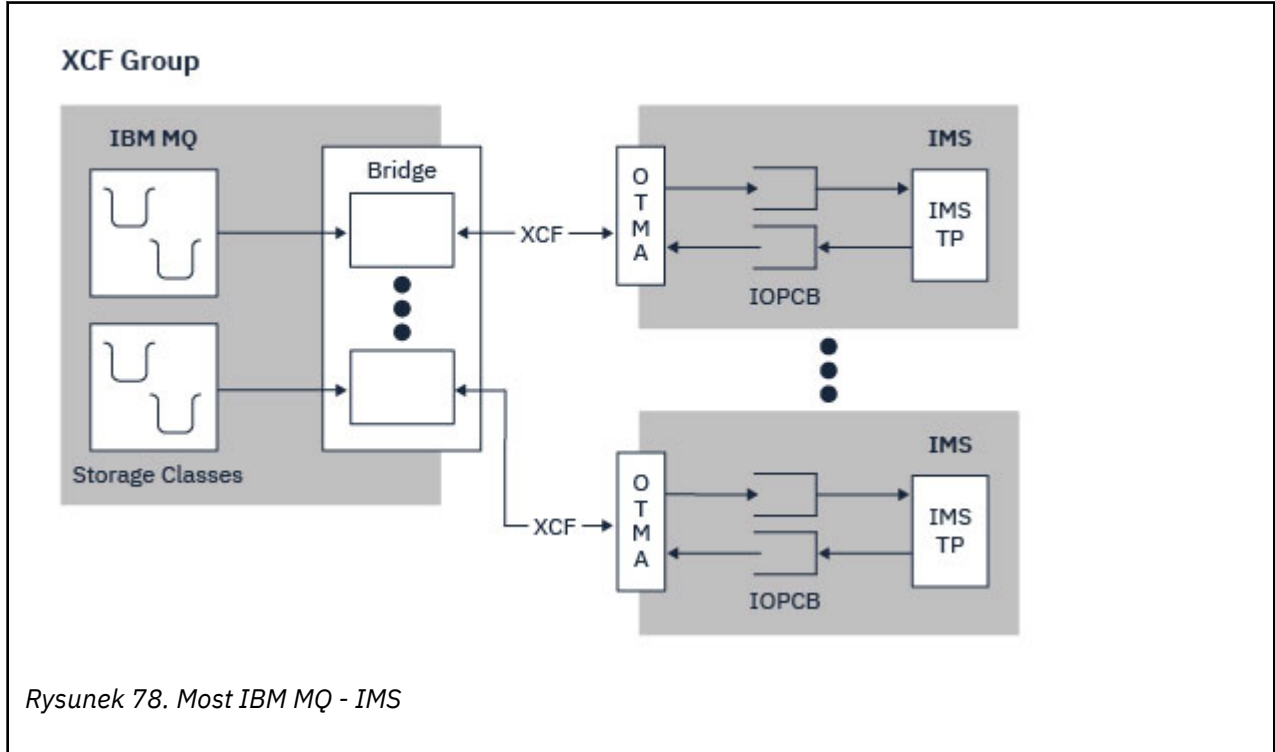
z/OS Most IBM MQ - IMS

Most IBM MQ - IMS jest komponentem produktu IBM MQ for z/OS , który umożliwia bezpośredni dostęp z aplikacji IBM MQ do aplikacji w systemie IMS .

Most IBM MQ - IMS włącza *obsługę niejawnych MQI*. Oznacza to, że można ponownie zaprojektować wcześniejsze aplikacje, które były kontrolowane przez terminale podłączone do terminalu 3270, tak aby były kontrolowane przez komunikaty IBM MQ , bez konieczności ponownego zapisywania, rekompilowania lub ponownego konsolidowania tych aplikacji. Most jest klientem IMS *Open Transaction Manager Access* (OTMA).

W aplikacjach mostu nie ma wywołań IBM MQ w aplikacji IMS . Aplikacja pobiera dane wejściowe za pomocą komendy GET UNIQUE (GU) do IOPCB i wysyła dane wyjściowe za pomocą komendy ISRT do IOPCB. Aplikacje IBM MQ używają nagłówka IMS (struktury MQIIH) w danych komunikatu, aby upewnić się, że aplikacje mogą być wykonywane w taki sam sposób, jak w przypadku, gdy są sterowane przez nieprogramowalne terminale. Jeśli używana jest aplikacja IMS , która przetwarza komunikaty z wieloma segmentami, należy pamiętać, że wszystkie segmenty powinny być zawarte w jednym komunikacie IBM MQ .

Most IMS został zilustrowany w sekcji Rysunek 78 na stronie 298.



Menedżer kolejek może nawiązać połączenie z jednym lub wieloma systemami IMS , a więcej niż jeden menedżer kolejek może nawiązać połączenie z jednym systemem IMS . Jedynym ograniczeniem jest to, że wszystkie muszą należeć do tej samej grupy XCF i muszą znajdować się w tym samym syspleksie.

Informacje na temat konfigurowania mostu IMS i dodawania dodatkowego połączenia IMS do tego samego menedżera kolejek zawiera sekcja [Konfigurowanie mostu IMS](#) .

Co to jest OTMA?

Narzędzie IMS OTMA jest opartym na transakcjach protokołem bezpołączeniowym typu klient/serwer, który działa w systemie IMS. Działa on jako interfejs dla serwerów komunikacyjnych opartych na hoście, uzyskujących dostęp do aplikacji IMS TM za pośrednictwem narzędzia [z/OS Cross Systems Coupling Facility \(XCF\)](#).

OTMA umożliwia klientom łączenie się z IMS w celu zapewnienia wysokiej wydajności dla interakcji między klientami i IMS dla dużej sieci lub dużej liczby sesji. OTMA jest zaimplementowany w środowisku sysplex w systemie z/OS . Dlatego domena OTMA jest ograniczona do domeny XCF.

Monitorowanie zasobów OTMA

Obsługa komunikatów protokołu OTMA x '3C', dostępna w produkcie IMS v10 lub nowszej, jest zawarta w moście IBM MQ - IMS w produkcie IBM MQ for z/OS. Te komunikaty są wysyłane do klientów OTMA przez program IMS w celu raportowania statusu poprawności.

Jeśli partner IMS nie może przetworzyć ilości wysyłanych żądań transakcji, powiadomi IBM MQ o wystąpieniu ostrzeżenia powodziowego. W odpowiedzi IBM MQ spowolni szybkość wysyłania żądań przez most.

Jeśli program IMS nadal nie może przetworzyć żądań transakcji i wystąpi pełny warunek powodziowy, wszystkie programy TPIPE skierowane do partnera IMS są zawieszane. Po powiadomieniu od partnera IMS , że warunek powodziowy lub powodziowy został zwolniony IBM MQ , wznawiane są wszystkie zawieszane programy TPE, jeśli jest to konieczne, i stopniowo zwiększane są szybkość wysyłania żądań transakcji, aż do osiągnięcia maksymalnej szybkości. Komunikaty konsoli są wysyłane przez IBM MQ w odpowiedzi na zmianę statusu partnerów IMS .

Jeśli używane są partnerzy IMS v10 , należy upewnić się, że zastosowano poprawkę PTF UK45082 .

Wprowadzanie transakcji IMS z IBM MQ

Aby wprowadzić transakcję IMS korzystającą z mostu, aplikacje umieszczają komunikaty w kolejce IBM MQ jak zwykle. Komunikaty zawierają dane transakcji IMS ; mogą mieć nagłówek IMS (struktura MQIIH) lub pozwolić mostowi IBM MQ - IMS na przyjęcie założeń dotyczących danych w komunikacie.

Następnie program IBM MQ umieszcza komunikat w kolejce IMS (najpierw jest on umieszczany w kolejce IBM MQ , aby umożliwić użycie punktów synchronizacji w celu zapewnienia integralności danych). Klasa pamięci masowej kolejki IBM MQ określa, czy kolejka jest *kolejką OTMA* (czyli kolejką używaną do przesyłania komunikatów do mostu IBM MQ - IMS) i konkretnym partnerem IMS , do którego są wysyłane dane komunikatu.

Zdalne menedżery kolejek mogą również uruchamiać transakcje IMS , zapisując je w tych kolejkach OTMA w systemie IBM MQ for z/OS.

Dane zwracane z systemu IMS są zapisywane bezpośrednio w kolejce odpowiedzi IBM MQ określonej w strukturze deskryptora komunikatu (MQMD). Może to być kolejka transmisji do menedżera kolejek określonego w polu **ReplyToQMGr** deskryptora MQMD.

Pojęcia pokrewne

[Aplikacje mostu IMS i IMS w systemie IBM MQ for z/OS](#)

Zadania pokrewne

[Dostosowywanie mostu IMS](#)

Odsyłacze pokrewne

[“IBM MQ i IMS” na stronie 296](#)

W tym temacie opisano sposób współpracy produktu IBM MQ z produktem IMS. Adapter IMS umożliwia połączenie menedżera kolejek z produktem IMS i umożliwia aplikacjom produktu IMS korzystanie z interfejsu MQI.

IBM MQ i adaptory z/OS Batch, TSO i RRS

W tym temacie opisano, w jaki sposób produkt IBM MQ współpracuje z adapterami z/OS Batch, TSO i RRS.

Wprowadzenie do adapterów wsadowych

Adaptory Batch/TSO są interfejsem między aplikacjami IBM MQ i z/OS działającymi w systemie JES, TSO lub z/OS UNIX System Services. Te adaptory umożliwiają aplikacjom z/OS korzystanie z interfejsu MQI.

Adaptory zapewniają dostęp do zasobów IBM MQ dla programów działających w następujących trybach lub stanach:

- Tryb zadania (TCB)
- Stan problemu lub nadzorczy
- Tryb inny niż międzypamięciowy
- Tryb rejestru bez dostępu

Połączenia między aplikacjami i programem IBM MQ są nawiązywane na poziomie zadania. Adaptory udostępniają wątek połączenia z bloku kontrolnego zadania aplikacji (TCB) do bazy danych IBM MQ.

Adapter Batch/TSO obsługuje protokół zatwierdzania jednofazowego dla zmian wprowadzonych w zasobach należących do IBM MQ. Nie obsługuje on protokołów zatwierdzania wielofazowego. Adapter RRS umożliwia aplikacjom IBM MQ uczestniczenie w protokołach zatwierdzania dwufazowego z innymi produktami obsługującymi RRS, koordynowanymi przez usługi RRS (z/OS Resource Recovery Services).

Adaptory używają usługi z/OS STIMERM do planowania zdarzenia asynchronicznego co sekundę. To zdarzenie uruchamia blok żądania przerwania (IRB), który nie wymaga oczekiwania przez zadanie aplikacji wsadowej. Ten IRB sprawdza, czy został opublikowany IBM MQ wypowiedzenie EBC. Jeśli EBC został wypowiedziany, IRB publikuje wszystkie ECB aplikacji, które oczekują na zdarzenie w IBM MQ (na przykład sygnał lub oczekiwanie).

Adapter Batch/TSO

Adapter IBM MQ Batch/TSO zapewnia obsługę IBM MQ dla aplikacji z/OS Batch i TSO. Wszystkie aplikacje, które działają w środowisku z/OS Batch lub TSO, muszą mieć edytowany razem z nimi kod pośredniczący interfejsu API CSQBSTUB. Kod pośredniczący zapewnia aplikacji dostęp do wszystkich wywołań MQI. W przypadku aplikacji należy użyć zatwierdzania jednofazowego i wycofywania, wywołując wywołania MQI **MQCMIT** i **MQBACK**.

Adapter RRS

Usługi odtwarzania zasobów (Resource Recovery Services-RRS) to podkomponent produktu z/OS, który udostępnia usługę ogólnosystemową do koordynowania zatwierdzania dwufazowego w produktach z/OS. Adapter IBM MQ Batch/TSO RRS (adapter RRS) udostępnia obsługę IBM MQ dla aplikacji wsadowych i TSO produktu z/OS, które chcą korzystać z tych usług. Adapter RRS umożliwia produktowi IBM MQ pełne uczestnictwo w koordynacji usług RRS. Aplikacje mogą uczestniczyć w przetwarzaniu zatwierdzania dwufazowego z innymi produktami obsługującymi usługi RRS (na przykład Db2).

Adapter RRS udostępnia dwa kody pośredniczące. Należy połączyć aplikacje, które mają używać usługi RRS z jednym z tych kodów pośredniczących.

CSQBRSTB

Ten kod pośredniczący umożliwia używanie zatwierdzania dwufazowego i wycofywania dla aplikacji przy użyciu wywoływalnych usług odtwarzania zasobów RRS zamiast wywołań MQI **MQCMIT** i **MQBACK**.

Należy również dowiązać moduł ATRSCSS z biblioteki SYS1.CSSLIB z aplikacją. Jeśli używane są wywołania MQI **MQCMIT** i **MQBACK**, zostanie wyświetlony kod powrotu MQRC_ENVIRONMENT_ERROR.

CSQBRRSI

Ten kod pośredniczący umożliwia użycie wywołań MQI **MQCMIT** i **MQBACK**; IBM MQ w rzeczywistości implementuje te wywołania jako wywołania **SRRCMIT** i **SRRBACK** RRS.

Informacje na temat budowania aplikacji korzystających z adaptera RRS zawiera sekcja [Adapter wsadowy RRS](#).

Gdzie można znaleźć więcej informacji na temat adapterów z/OS Batch, TSO i RRS

Więcej informacji na temat tematów w tej sekcji można znaleźć w następujących źródłach:

Temat	Gdzie szukać
Konfigurowanie adapterów zadań wsadowych	Czynność 19: Konfigurowanie adapterów Batch, TSO i RRS
Usługi odtwarzania zasobów wywoływalnych RRS	<i>MVS Programming: Callable Services for High Level Languages (Programowanie MVS: usługi wywoływane dla języków wysokiego poziomu)</i>

z/OS

IBM MQ for z/OS i WebSphere Application Server

Ten temat zawiera informacje dotyczące korzystania z języka IBM MQ for z/OS przez WebSphere Application Server.

Aplikacje napisane w języku Java, które działają w systemie WebSphere Application Server, mogą używać specyfikacji Java Message Service (JMS) do przesyłania komunikatów. Przesyłanie komunikatów w trybie punkt z punktem w tym środowisku może być udostępniane przez menedżer kolejek produktu IBM MQ for z/OS.

Zaletą używania menedżera kolejek produktu IBM MQ for z/OS do udostępniania przesyłania komunikatów jest to, że łączące się aplikacje produktu JMS mogą w pełni uczestniczyć w funkcjonalności sieci produktu IBM MQ. Na przykład mogą one używać mostu IMS lub wymieniać komunikaty z menedżerami kolejek działającymi na innych platformach.

Połączenie między programem WebSphere Application Server i menedżerem kolejek

Więcej informacji na ten temat zawiera sekcja [Używanie produktów IBM MQ i WebSphere Application Server razem](#).

Korzystanie z funkcji IBM MQ w aplikacjach JMS

Domyślnie komunikaty produktu JMS przechowywane w kolejkach systemu IBM MQ używają nagłówka MQRFH2 do przechowywania niektórych informacji z nagłówka komunikatu JMS. Wiele wcześniejszych aplikacji IBM MQ nie może przetwarzać komunikatów z tymi nagłówkami i wymaga własnych nagłówków charakterystycznych, na przykład MQCIH dla CICS Bridge lub MQWIH dla IBM MQ Workflow. Więcej informacji na ten temat zawiera sekcja [Odwzorowywanie komunikatów produktu JMS na komunikaty produktu IBM MQ](#).

Managed File Transfer

Program Managed File Transfer przesyła pliki między systemami w sposób zarządzany i kontrolowany, bez względu na wielkość plików lub używane systemy operacyjne.

Za pomocą programu Managed File Transfer można zbudować dostosowane, skalowalne i zautomatyzowane rozwiązanie, które umożliwi zarządzanie przesyłaniem plików, ufanie mu i zabezpieczanie go. Managed File Transfer eliminuje kosztowne nadmiarowości, obniża koszty utrzymania i maksymalizuje dotychczasowe inwestycje IT.

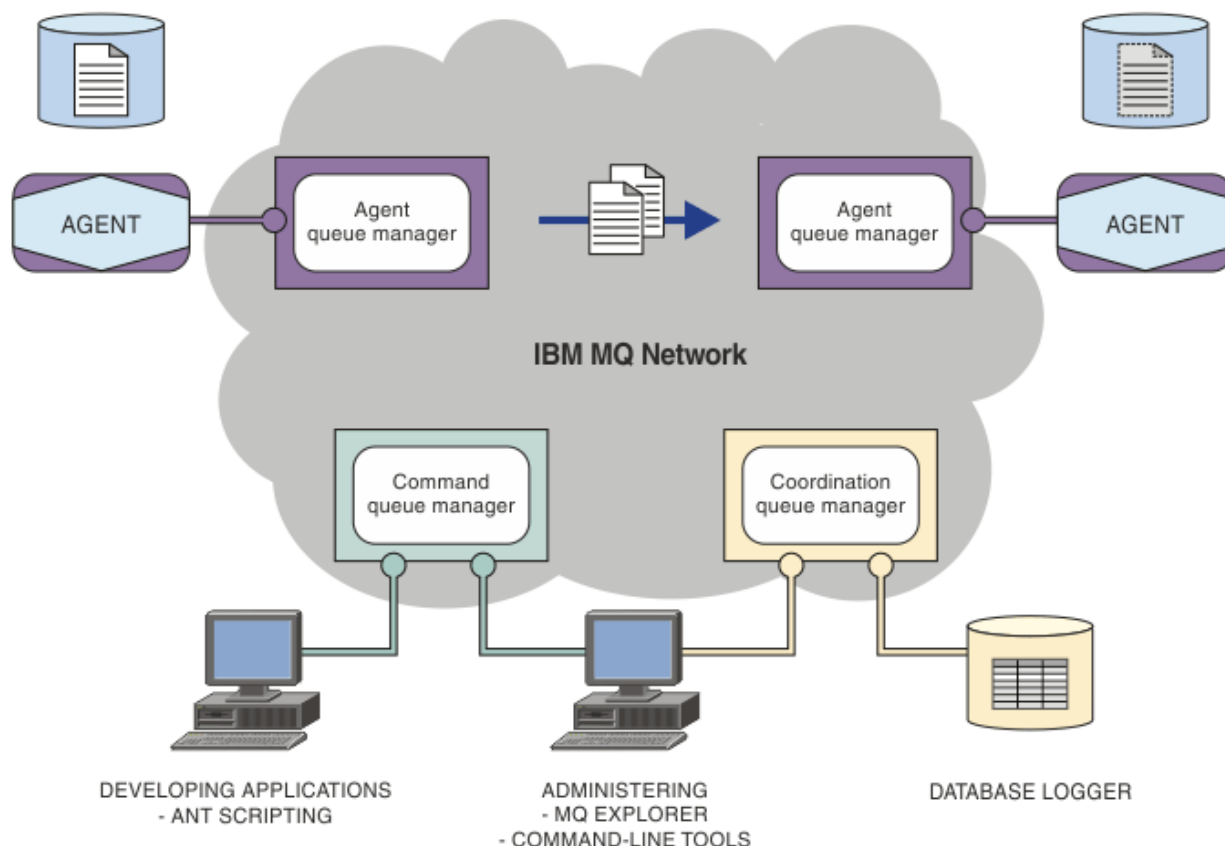



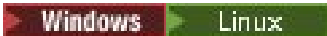


Diagram przedstawia prostą topologię produktu Managed File Transfer. Istnieją dwa agenty, z których każdy łączy się z własnym menedżerem kolejek agenta w sieci IBM MQ. Plik jest przesyłany z agenta po jednej stronie diagramu, przez sieć IBM MQ, do agenta po drugiej stronie diagramu. Ponadto w sieci produktu IBM MQ znajdują się menedżer kolejek koordynacji i menedżer kolejek komend. Aplikacje i narzędzia łączą się z tymi menedżerami kolejek w celu konfigurowania, administrowania, obsługi i rejestrowania działań Managed File Transfer w sieci IBM MQ.

Produkt Managed File Transfer można zainstalować jako cztery różne opcje, w zależności od systemu operacyjnego i konfiguracji ogólnej. Dostępne są następujące opcje: Managed File Transfer Agent, Managed File Transfer Logger, Managed File Transfer Service lub Managed File Transfer Tools. Więcej informacji na ten temat zawiera sekcja [Opcje produktu Managed File Transfer](#).

Za pomocą programu Managed File Transfer można wykonywać następujące zadania:

- Utwórz zarządzane przesyłanie plików
 -   Tworzenie nowych operacji przesyłania plików z systemu IBM MQ Explorer na platformach Linux lub Windows.
 - Utwórz nowe przesyłanie plików z wiersza komend na wszystkich obsługiwanych platformach.
 - Zintegruj funkcję przesyłania plików z narzędziem Apache Ant.

- Napisz aplikacje sterujące Managed File Transfer , umieszczając komunikaty w kolejkach komend agenta.
- Zaplanuj przesyłanie plików w późniejszym czasie. Można również wyzwoić zaplanowane operacje przesyłania plików na podstawie zakresu zdarzeń systemu plików, na przykład tworzenia nowego pliku.
- Stale monitoruj zasób, na przykład katalog, a gdy zawartość tego zasobu spełnia predefiniowany warunek, uruchom zadanie. Tym zadaniem może być przesyłanie plików, skrypt Ant lub zadanie JCL.
- Przesyłanie plików do i z kolejek systemu IBM MQ .
- Przesyłanie plików do i z serwerów FTP, FTPS lub SFTP.
- Przesyłanie plików do i z węzłów Connect:Direct .
- Prześlij zarówno pliki tekstowe, jak i binarne. Pliki tekstowe są automatycznie konwertowane między stronami kodowymi i konwencjami końca wiersza w systemie źródłowym i docelowym.
- Przesyłanie może być zabezpieczone przy użyciu standardów branżowych dla połączeń opartych na protokole SSL (Secure Socket Layer).
- Wyświetlanie trwających transferów i rejestrowanie informacji o wszystkich transferach w sieci
 -  Wyświetlanie statusu trwających operacji przesyłania z systemu IBM MQ Explorer na platformach Linux lub Windows .
 -  Sprawdź status zakończonych operacji przesyłania, korzystając z programu IBM MQ Explorer na platformach Linux lub Windows .
 - Użyj funkcji programu rejestrującego bazy danych Managed File Transfer , aby zapisać komunikaty dziennika w bazie danych Db2 lub Oracle .

Produkt Managed File Transfer jest oparty na produkcie IBM MQ, który zapewnia gwarantowane jednorazowe dostarczanie komunikatów między aplikacjami. Można korzystać z różnych funkcji produktu IBM MQ. Na przykład można użyć kompresji kanału, aby skompresować dane wysyłane między agentami za pomocą kanałów IBM MQ i użyć kanałów SSL do zabezpieczenia danych wysyłanych między agentami. Pliki są przesyłane niezawodnie i mogą tolerować awarię infrastruktury, w której odbywa się przesyłanie plików. W przypadku wyłączenia sieci przesyłanie plików jest restartowane od miejsca, w którym zostało przerwane po przywróceniu połączenia.

Konsolidując przesyłanie plików z istniejącą siecią IBM MQ , można uniknąć wydatków na zasoby wymagane do utrzymania dwóch oddzielnych infrastruktur. Jeśli użytkownik nie jest jeszcze klientem IBM MQ , utworzenie IBM MQ sieci do obsługi Managed File Transfer powoduje utworzenie rdzenia sieci na potrzeby przyszłej implementacji architektury SOA. Jeśli jesteś już klientem IBM MQ , Managed File Transfer może skorzystać z istniejącej infrastruktury IBM MQ , w tym IBM MQ Internet Pass-Thru i IBM Integration Bus.

Aby zwiększyć odporność konfiguracji produktu Managed File Transfer , można skorzystać z rozwiązań wysokiej dostępności systemu IBM MQ . Jeśli agenty używają menedżerów kolejek replikowanych danych (RDQM), należy je skonfigurować do korzystania z funkcji zmiennego adresu IP. Oznacza to, że agenty używają tego samego adresu IP do komunikacji z dowolnym z trzech instancji RDQM, które są obecnie uruchomione i automatycznie ponownie łączą się po przełączeniu awaryjnym (patrz [Wysoka dostępność RDQM](#) i [Tworzenie i usuwanie zmiennego adresu IP](#)). Jeśli używane jest rozwiązanie menedżera kolejek z wieloma instancjami, aplikacje używają innego adresu IP do komunikacji z każdą instancją, który jest obsługiwany przez ponowne połączenie klienta po przełączeniu awaryjnym (patrz sekcja [Menedżery kolejek z wieloma instancjami](#) oraz sekcja [Ponowne połączenie kanału i klienta](#)).

Produkt Managed File Transfer można zintegrować z wieloma innymi produktami IBM :

IBM Integration Bus

Przetwarzanie plików, które zostały przesłane przez program Managed File Transfer w ramach przepływu IBM Integration Bus . Więcej informacji na ten temat zawiera sekcja [Praca z produktem MFT w serwisie IBM Integration Bus](#).

IBM Sterling Connect:Direct

Przesyłanie plików do i z istniejącej sieci Connect:Direct za pomocą mostu Managed File Transfer Connect:Direct . Więcej informacji na ten temat zawiera sekcja [Most Connect:Direct](#).

IBM Tivoli Composite Application Manager

Program IBM Tivoli Composite Application Manager udostępnia agenta, którego można użyć do monitorowania informacji publikowanych w menedżerze kolejek koordynacji.

Pojęcia pokrewne

[Opcje produktu Managed File Transfer](#)

[“MFT przegląd topologii” na stronie 304](#)

Przegląd sposobu, w jaki agenty Managed File Transfer są połączone z menedżerem kolejek koordynacji w sieci produktu IBM MQ .

[“W jaki sposób produkt MFT działa z produktem IBM MQ?” na stronie 304](#)

Produkt Managed File Transfer współdziela na wiele sposobów z produktem IBM MQ.

W jaki sposób produkt MFT działa z produktem IBM MQ?

Produkt Managed File Transfer współdziela na wiele sposobów z produktem IBM MQ.

- Managed File Transfer przesyła pliki między procesami agenta, dzieląc każdy plik na jeden lub więcej komunikatów i przysyłając je przez sieć IBM MQ .
- Procesy agenta przenoszą dane plików za pomocą nietrwałych komunikatów w celu zminimalizowania wpływu na dzienniki IBM MQ . Dzięki komunikacji między sobą procesy agenta regulują przepływ komunikatów zawierających dane plików. Zapobiega to tworzeniu się komunikatów zawierających dane pliku w kolejkach transmisji systemu IBM MQ i zapewnia, że jeśli nie zostaną dostarczone jakiegokolwiek nietrwałe komunikaty, dane pliku zostaną wysłane ponownie.
- Agenty Managed File Transfer używają wielu kolejek systemu IBM MQ . Więcej informacji na ten temat zawiera sekcja [Kolejki systemowe MFT i temat systemowy](#).
- Chociaż niektóre z tych kolejek są przeznaczone wyłącznie do użytku wewnętrznego, agent może akceptować żądania w postaci specjalnie sformatowanych komunikatów komend wysyłanych do konkretnej kolejki, z której agent odczytuje dane. Zarówno komendy wiersza komend, jak i wtyczka IBM MQ Explorer wysyłają do agenta komunikaty IBM MQ w celu poinformowania agenta o konieczności wykonania pożądanego działania. W ten sposób można pisać aplikacje IBM MQ , które współdzielą z agentem. Więcej informacji na ten temat zawiera sekcja [Sterowanie MFT przez umieszczanie komunikatów w kolejce komend agenta](#).
- Agenty dla systemu Managed File Transfer wysyłają informacje o swoim stanie oraz postępie i wyniku operacji przesyłania do menedżera kolejek produktu MQ , który został wyznaczony jako menedżer kolejek koordynacji. Te informacje są publikowane przez menedżer kolejek koordynacji i mogą być subskrybowane przez aplikacje, które chcą monitorować postęp przesyłania lub przechowywać rekordy operacji przesyłania, które wystąpiły. Zarówno komendy wiersza komend, jak i wtyczka IBM MQ Explorer mogą korzystać z opublikowanych informacji. Można pisać aplikacje IBM MQ , które korzystają z tych informacji. Więcej informacji na temat tematu, w którym informacje są publikowane, zawiera sekcja [SYSTEM.FTE Temat produktu FTE](#).
- Kluczowe komponenty produktu Managed File Transfer korzystają z możliwości menedżerów kolejek systemu IBM MQ w zakresie zapisywania i przekazywania komunikatów. Oznacza to, że w przypadku wyłączenia, niezmienione części infrastruktury mogą kontynuować przesyłanie plików. Dotyczy to menedżera kolejek koordynacji, w którym kombinacja subskrypcji przechowywania i przekazywania oraz subskrypcji trwałych umożliwia menedżerowi kolejek koordynacji tolerowanie niedostępności bez utraty kluczowych informacji o transferach plików, które miały miejsce.

MFT przegląd topologii

Przegląd sposobu, w jaki agenty Managed File Transfer są połączone z menedżerem kolejek koordynacji w sieci produktu IBM MQ .

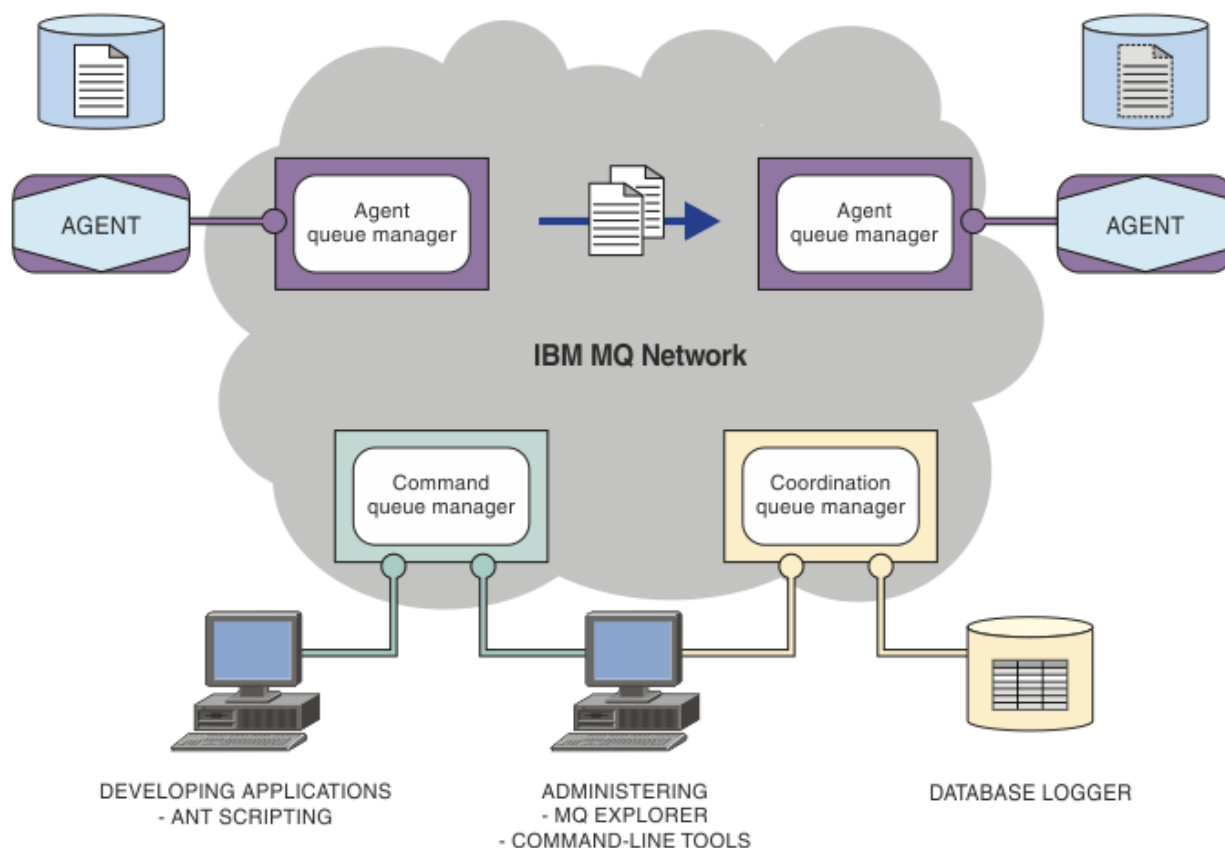
Agenty Managed File Transfer wysyłają i odbierają przesyłane pliki. Każdy agent ma własny zestaw kolejek w powiązaniu z nim menedżerze kolejek, a agent jest przyłączony do swojego menedżera kolejek w trybie powiązań lub w trybie klienta. Agent może również używać menedżera kolejek koordynacji jako swojego menedżera kolejek.

Menedżer kolejek koordynacji rozgłasza informacje o kontroli i przesyłaniu plików. Menedżer kolejek koordynacji reprezentuje pojedynczy punkt gromadzenia informacji o agencie, statusie przesyłania i kontroli przesyłania. Menedżer kolejek koordynacji nie musi być dostępny, aby możliwe było przesyłanie. Jeśli menedżer kolejek koordynacji tymczasowo staje się niedostępny, przesyłanie jest kontynuowane w normalny sposób. Komunikaty kontroli i statusu są przechowywane w menedżerach kolejek agenta do momentu udostępnienia menedżera kolejek koordynacji, a następnie mogą być przetwarzane w normalny sposób.

Agenty rejestrują się w menedżerze kolejek koordynacji i publikują w nim swoje szczegóły. Te informacje o agencie są używane przez wtyczkę Managed File Transfer do włączania uruchamiania operacji przesyłania z serwera IBM MQ Explorer. Informacje o agencie zgromadzone w menedżerze kolejek koordynacji są również używane przez komendy do wyświetlania informacji o agencie i jego statusie.

Status przesyłania i informacje kontroli przesyłania są publikowane w menedżerze kolejek koordynacji. Status przesyłania i informacje kontroli przesyłania są używane przez wtyczkę Managed File Transfer do monitorowania postępu operacji przesyłania z serwera IBM MQ Explorer. Informacje kontroli przesyłania przechowywane w menedżerze kolejek koordynacji mogą być zachowywane w celu zapewnienia możliwości kontroli.

Menedżer kolejek komend jest używany do nawiązywania połączenia z siecią produktu IBM MQ i jest menedżerem kolejek, z którym nawiązywane jest połączenie podczas wywoływania komend produktu Managed File Transfer .



Pojęcia pokrewne

[“Managed File Transfer” na stronie 302](#)

Program Managed File Transfer przesyła pliki między systemami w sposób zarządzany i kontrolowany, bez względu na wielkość plików lub używane systemy operacyjne.

“W jaki sposób produkt MFT działa z produktem IBM MQ?” na stronie 304
Produkt Managed File Transfer współdziela na wiele sposobów z produktem IBM MQ.

[Managed File Transfer scenariusz](#)

Przegląd produktu MFTREST API

REST API obsługuje niektóre komendy Managed File Transfer, w tym wyświetlanie listy operacji przesyłania oraz szczegółowe informacje o agentach przesyłania plików.

W produkcie IBM MQ 9.1.0 REST API dostępne są opcje umożliwiające wyświetlenie listy wszystkich bieżących operacji przesyłania w systemie Managed File Transfer oraz tworzenie zapytań o status agentów Managed File Transfer. Więcej informacji na ten temat zawiera sekcja [Pierwsze kroki z produktem REST API MFT](#).

IBM MQ Internet Pass-Thru

IBM MQ Internet Pass-Thru (MQIPT) to opcjonalny komponent produktu IBM MQ, który może być używany do implementowania rozwiązań przesyłania komunikatów między ośrodkami zdalnymi w Internecie.

Począwszy od wersji IBM MQ 9.2.0, MQIPT jest opcjonalnym komponentem produktu IBM MQ. Aby uzyskać pliki instalacyjne produktu MQIPT dla systemu IBM MQ 9.3.x, należy przejść do serwisu [IBM Fix Central dla systemu IBM MQ](#). W wersjach wcześniejszych niż IBM MQ 9.2.0 produkt MQIPT był dostępny jako pakiet wsparcia.

Nie trzeba uruchamiać programu IBM MQ 9.3, aby używać programu MQIPT w systemie IBM MQ 9.3. Do nawiązania połączenia z dowolną obsługiwaną wersją produktu IBM MQ można użyć programu MQIPT i nie trzeba instalować żadnych innych komponentów IBM MQ w tej samej wersji co program MQIPT.

Jeśli zakupiono uprawnienie IBM MQ, można zainstalować dowolną liczbę kopii produktu MQIPT. Instalacje MQIPT nie są wliczane do zakupionego uprawnienia IBM MQ. Więcej informacji na temat licencjonowania produktu IBM MQ zawiera sekcja [Informacje licencyjne produktu IBM MQ](#).

Uwaga: Ta dokumentacja odnosi się do MQIPT w IBM MQ 9.3. Dokumentację pakietu wsparcia MQIPT (wersja 2.1) w serwisie IBM Documentation można znaleźć w serwisie [MQIPT \(SupportPac MS81\)](#) w dokumentacji IBM MQ 9.0.

Uwaga: Jeśli używany jest produkt MQIPT 2.1 lub wcześniejszy, zaleca się przeprowadzenie aktualizacji do produktu MQIPT for IBM MQ 9.3, ponieważ data zakończenia wsparcia dla pakietu wsparcia MQIPT to 30th września 2020.

Produkt IBM MQ Internet Pass-Thru działa jako usługa autonomiczna, która może odbierać i przekazywać przepływy komunikatów IBM MQ między dwoma menedżerami kolejek produktu IBM MQ lub między klientem produktu IBM MQ a menedżerem kolejek produktu IBM MQ.

MQIPT łączy to połączenie, gdy klient i serwer nie znajdują się w tej samej sieci fizycznej.

Jedna lub więcej instancji programu MQIPT może zostać umieszczonych w ścieżce komunikacji między dwoma menedżerami kolejek systemu IBM MQ lub między klientem IBM MQ a menedżerem kolejek systemu IBM MQ. Instancje MQIPT zezwalają dwóm systemom IBM MQ na wymianę komunikatów bez konieczności bezpośredniego połączenia TCP/IP między tymi dwoma systemami. Jest to przydatne, gdy konfiguracja firewalla uniemożliwia bezpośrednie połączenie TCP/IP między dwoma systemami.

Program MQIPT nasłuchuje połączeń przychodzących na co najmniej jednym porcie TCP/IP, który może zawierać normalne komunikaty IBM MQ, IBM MQ komunikaty tunelowane w HTTP lub komunikaty szyfrowane przy użyciu protokołu TLS (Transport Layer Security) lub SSL (Secure Sockets Layer). Produkt MQIPT może obsługiwać wiele współbieżnych połączeń.

Kanał IBM MQ , który wysyła początkowe żądanie połączenia TCP/IP, jest określany jako *program wywołujący*, kanał, z którym próbuje się połączyć jako *program odpowiadający*, oraz menedżer kolejek, z którym próbuje się połączyć jako *docelowy menedżer kolejek*.

MQIPT przechowuje dane w pamięci podczas przekazywania ich ze źródła do miejsca docelowego. Na dysku nie są zapisywane żadne dane (z wyjątkiem pamięci stronicowanej na dysk przez system operacyjny). Jedyne miejsce, w którym program MQIPT uzyskuje jawny dostęp do dysku, to odczyt jego pliku konfiguracyjnego oraz zapis rekordów dziennika i śledzenia połączenia.

Pełny zakres typów kanałów produktu IBM MQ można utworzyć za pomocą jednej lub wielu instancji produktu MQIPT. Obecność parametru MQIPT w ścieżce komunikacji nie ma wpływu na parametry funkcjonalne podłączonych komponentów IBM MQ , ale może mieć pewien wpływ na wydajność przesyłania komunikatów.

Parametr MQIPT może być używany w połączeniu z parametrami IBM MQ i IBM Integration Bus, zgodnie z opisem w sekcji [“Możliwe konfiguracje produktu MQIPT”](#) na stronie 310.

Aby zainstalować produkt MQIPT, należy zapoznać się z sekcją [Instalowanie produktu MQIPT](#).

Zadania pokrewne

[Konfigurowanie produktu IBM MQ Internet Pass-Thru](#)

[Administrowanie i konfigurowanie produktu IBM MQ Internet Pass-Thru](#)

Odsyłacze pokrewne

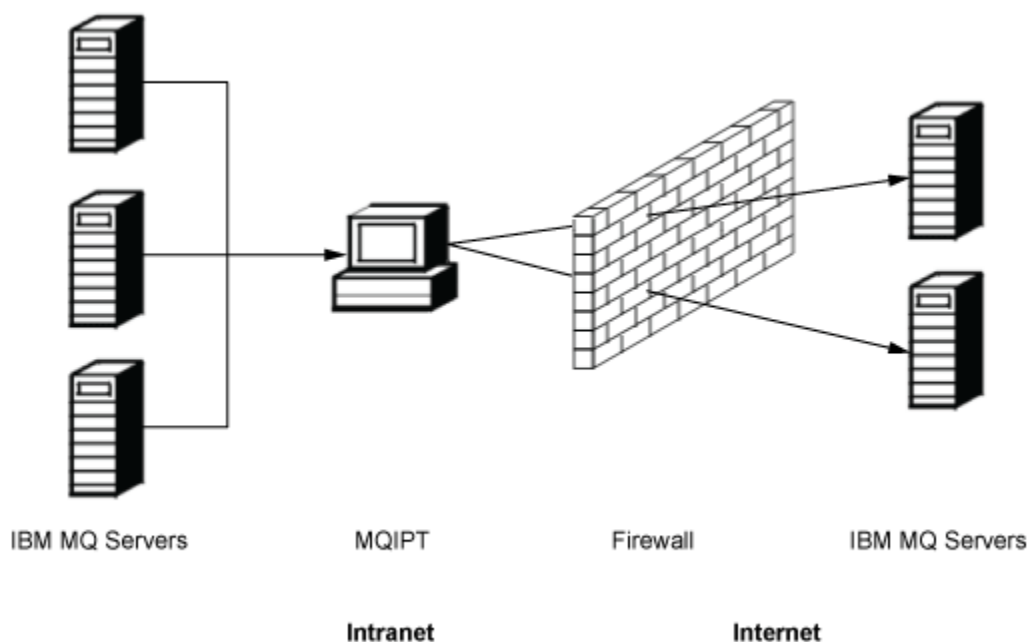
[Informacje dodatkowe o konfiguracji produktu IBM MQ Internet Pass-Thru](#)

Użycie MQIPT

Istnieje wiele potencjalnych zastosowań produktu IBM MQ Internet Pass-Thru (MQIPT).

Produkt MQIPT może być używany jako koncentrator kanałów.

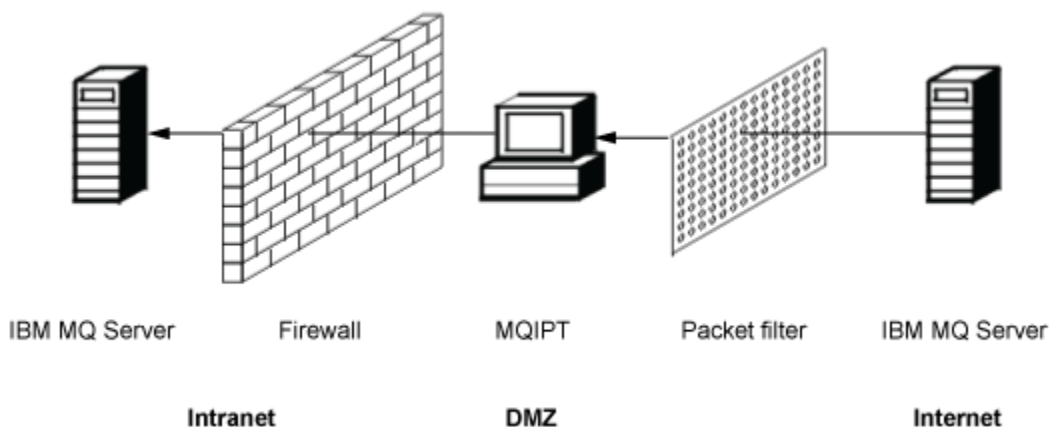
Dzięki użyciu metody MQIPT kanały do lub z wielu oddzielnych hostów mogą być wyświetlane w firewallu tak, jakby wszystkie były do lub od hosta MQIPT . Ułatwia to definiowanie reguł filtrowania firewala i zarządzanie nimi.



Rysunek 79. Przykład MQIPT jako koncentratora kanałów

MQIPT można umieścić w strefie DMZ, aby zapewnić pojedynczy punkt dostępu

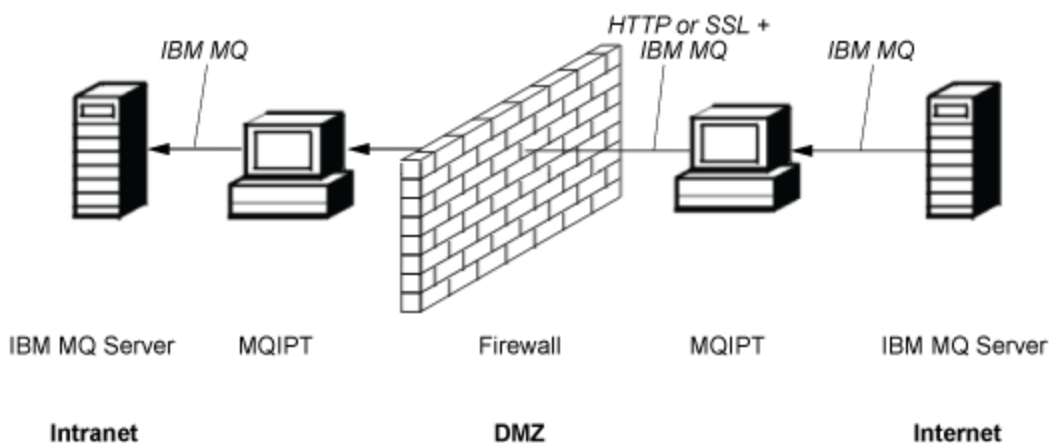
Jeśli program MQIPT jest umieszczony w firewallu strefy DMZ (konfiguracji firewalla na potrzeby zabezpieczania sieci lokalnych), na komputerze ze znanym i zaufanym adresem IP, można użyć programu MQIPT do nasłuchiwania przychodzących połączeń kanału IBM MQ, które mogą być następnie przekazywane do zaufanego intranetu. Wewnętrzna zapora firewall musi zezwalać temu zaufanemu komputerowi na nawiązywanie połączeń przychodzących. W tej konfiguracji produkt MQIPT uniemożliwia zewnętrznym żądaniom dostępu odbieranie rzeczywistych adresów IP komputerów w zaufanym intranecie. W ten sposób produkt MQIPT udostępnia pojedynczy punkt dostępu. Jeśli jest to wymagane, program MQIPT można skonfigurować w taki sposób, aby akceptował połączenia TLS i przekazywał dane do miejsca docelowego przy użyciu oddzielnego połączenia TLS, co spowoduje zakończenie sesji TLS w strefie DMZ.



Rysunek 80. Przykład MQIPT w firewallu strefy DMZ

Produkt MQIPT może komunikować się za pomocą tunelowania HTTP .

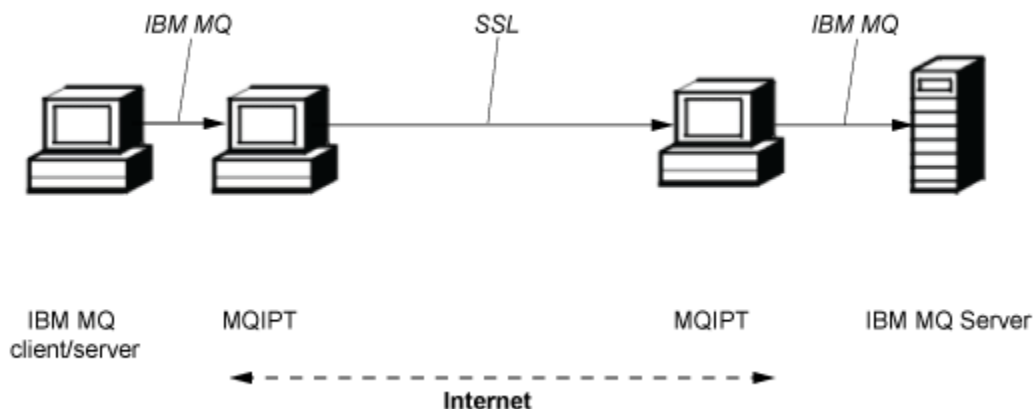
Jeśli dwie instancje MQIPT są wdrożone w sieci, mogą komunikować się za pomocą protokołu HTTP. Funkcja tunelowania HTTP umożliwia przesyłanie żądań przez firewalle przy użyciu istniejących proxy HTTP . Pierwszy element MQIPT wstawia protokół IBM MQ do protokołu HTTP , a drugi wyodrębni protokół IBM MQ z opakowania HTTP i przekazuje go do docelowego menedżera kolejek.



Rysunek 81. Przykład tunelowania MQIPT i HTTP

MQIPT może szyfrować komunikaty

Jeśli parametr MQIPT jest skonfigurowany tak, jak w poprzednim przykładzie, żądania mogą być szyfrowane przed przestaniem przez firewalle. Pierwszy program MQIPT szyfruje dane, a drugi deszyfruje je przy użyciu protokołu SSL/TLS przed wystąpieniem ich do docelowego menedżera kolejek.



Rysunek 82. Przykład zastosowania protokołów MQIPT i SSL/TLS

Jak działa MQIPT

W najprostszej konfiguracji MQIPT działa jako serwer przekazujący protokołu IBM MQ. Nasłuchuje on na porcie TCP/IP i akceptuje żądania połączeń z kanałów IBM MQ.

Jeśli zostanie odebrane poprawnie sformatowane żądanie, program MQIPT nawiąże dalsze połączenie TCP/IP między sobą a docelowym menedżerem kolejek produktu IBM MQ. Następnie przekazuje wszystkie odebrane pakiety protokołu z połączenia przychodzącego do docelowego menedżera kolejek i zwraca pakiety protokołu z docelowego menedżera kolejek z powrotem do oryginalnego połączenia przychodzącego.

Nie jest używana żadna zmiana protokołu IBM MQ (klient/serwer lub menedżer kolejek w menedżerze kolejek), ponieważ żaden z tych elementów nie jest bezpośrednio świadomy obecności pośrednika. Nowe wersje kodu klienta lub serwera IBM MQ nie są wymagane.

Aby używać produktu MQIPT, kanał wywołujący musi być skonfigurowany w taki sposób, aby używał nazwy hosta i portu MQIPT, a nie nazwy hosta i portu docelowego menedżera kolejek. Jest ona definiowana za pomocą właściwości **CONNNAME** kanału IBM MQ. Program MQIPT odczytuje dane przychodzące i po prostu przekazuje je do docelowego menedżera kolejek. Inne pola konfiguracyjne, takie jak ID użytkownika i hasło w kanale klient/serwer, są podobnie przekazywane do docelowego menedżera kolejek.

Wiele menedżerów kolejek

Program MQIPT może być używany w celu umożliwienia dostępu do więcej niż jednego docelowego menedżera kolejek. Aby to zadziałało, musi istnieć mechanizm umożliwiający określenie MQIPT, z którym menedżerem kolejek ma zostać nawiązane połączenie, a więc program MQIPT używa numeru przychodzącego portu TCP/IP do określenia, z którym menedżerem kolejek ma zostać nawiązane połączenie.

Dlatego można skonfigurować program MQIPT do nasłuchiwania na wielu portach TCP/IP. Każdy port nasłuchiwania jest odwzorowywany na docelowy menedżer kolejek za pomocą programu MQIPT trasa. Można zdefiniować do 100 takich tras, które wiążą nasłuchujący port TCP/IP z nazwą hosta i portem docelowego menedżera kolejek. Oznacza to, że nazwa hosta (adres IP) docelowego menedżera kolejek nigdy nie jest widoczna dla kanału źródłowego. Każda trasa może obsługiwać wiele połączeń między portem nasłuchiwania i miejscem docelowym, przy czym każde połączenie działa niezależnie.

MQIPT plik konfiguracyjny

Program MQIPT używa pliku konfiguracyjnego o nazwie `mcipt.conf`. Ten plik zawiera definicje wszystkich tras i powiązanych z nimi właściwości. Więcej informacji na temat produktu `mcipt.conf` zawiera sekcja [Administrowanie i konfigurowanie produktu IBM MQ Internet Pass-Thru](#).

Po uruchomieniu program MQIPT uruchamia każdą trasę wymienioną w pliku konfiguracyjnym. Komunikaty są zapisywane w konsoli systemowej, pokazując status każdej trasy. Jeśli dla trasy zostanie wyświetlony komunikat MQCPI078, trasa jest gotowa do akceptowania żądań połączenia.

Możliwe konfiguracje produktu MQIPT

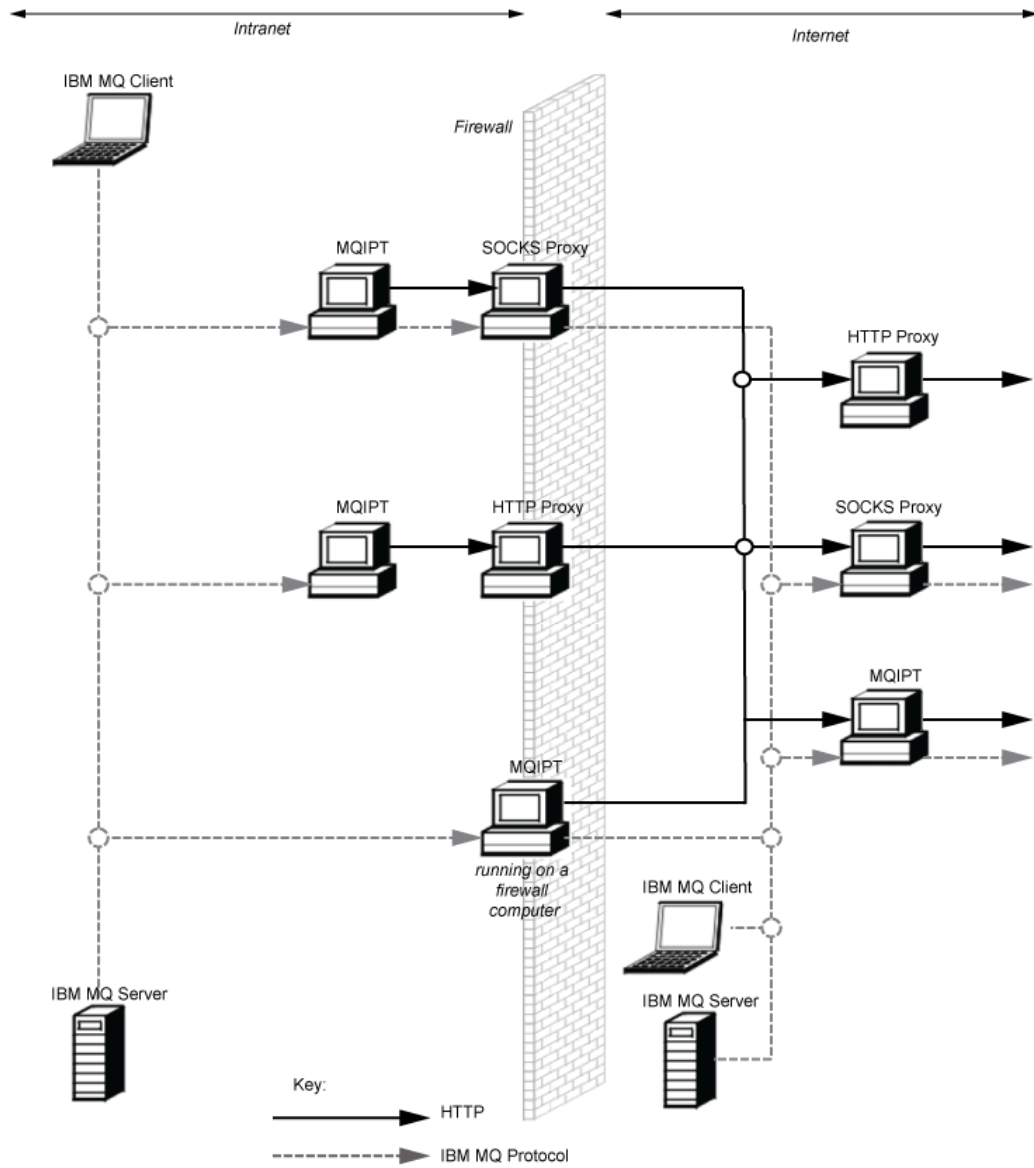
Parametr MQIPT może być używany w połączeniu z parametrami IBM MQ i IBM Integration Bus.

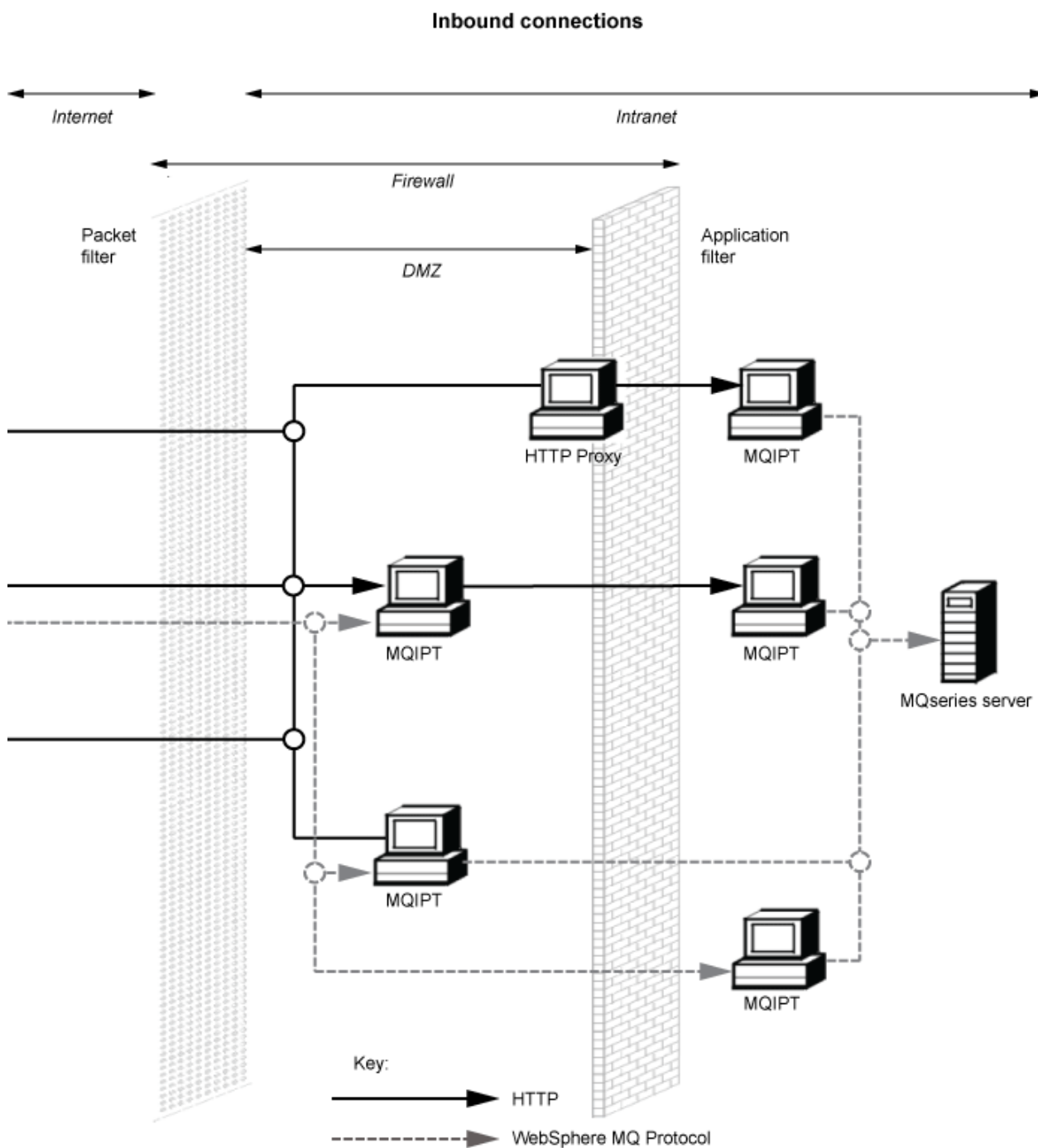
Poniższy wieloczęściowy rysunek przedstawia wiele możliwych konfiguracji produktu MQIPT w topologii IBM MQ. Ilustruje on różne sposoby wysyłania komunikatów przez program MQIPT. Przedstawia on klienty i serwery w intranecie, wewnątrz firewalla i w Internecie poza firewallem, przekazując komunikaty do serwera proxy MQIPT, serwera proxy HTTP lub serwera proxy SOCKS, który je przekazuje.

Komunikaty są odbierane przez serwer proxy MQIPT lub serwer proxy HTTP w strefie DMZ przed przekazaniem komunikatu przez przychodzący firewall do serwera.

Należy zauważyć, że serwery proxy HTTP, proxy SOCKS i MQIPT znajdujące się po stronie intranetowej firewalla reprezentują możliwość połączenia wielu komputerów w sieci Internet. Na przykład komputer z systemem MQIPT może komunikować się za pośrednictwem jednego lub większej liczby komputerów proxy SOCKS lub HTTP lub kolejnych komputerów MQIPT przed osiągnięciem celu.

Outbound connections





Zgodne konfiguracje

Kompatybilne scenariusze połączeń, w których klient lub menedżer kolejek produktu IBM MQ komunikuje się z programem MQIPT. Ta sama lub druga trasa MQIPT jest używana do komunikacji z docelowym menedżerem kolejek.

Kompatybilne konfiguracje z pojedynczą trasą MQIPT

Do komunikacji z produktem IBM MQ można użyć pojedynczej trasy MQIPT .

Kolumny w tabeli [Tabela 26](#) na stronie 313 zawierają następujące informacje:

1. Protokół używany między IBM MQ i trasą MQIPT . Połączenie może zostać utworzone przez klienta lub menedżera kolejek systemu IBM MQ i może korzystać z protokołu FAP (IBM MQ Formats and Protocols) lub SSL/TLS.
2. Tryb, w którym działa trasa MQIPT . Format komunikacji w Internecie między systemami MQIPT i IBM MQ jest określany przez konfigurację trasy MQIPT . Należy zauważyć, że tam, gdzie w tabeli jest mowa o protokole SSL, można również użyć protokołu TLS.
3. Protokół używany między trasą MQIPT i docelowym menedżerem kolejek.

1. IBM MQ Protokół źródła	2. Tryb MQIPT trasy	3. IBM MQ Protokół docelowy
FAP	FAP-proxy (domyślny)	FAP
	FAP-serwer i SSL-klient	Protokół SSL/TLS
Protokół SSL/TLS	SSL-proxy	Protokół SSL/TLS
	SSL-serwer i FAP-klient	FAP
	SSL-serwer i SSL-klient	Protokół SSL/TLS

Kompatybilne konfiguracje z więcej niż jedną trasą MQIPT

Do komunikacji z produktem IBM MQ można użyć więcej niż jednej trasy w jednej lub większej liczbie instancji serwera MQIPT.

Kolumny w tabeli [Tabela 27](#) na stronie 314 zawierają następujące informacje:

1. Protokół używany między IBM MQ a pierwszą trasą MQIPT . Połączenie może zostać utworzone przez klienta lub menedżera kolejek systemu IBM MQ i może korzystać z protokołu FAP (IBM MQ Formats and Protocols) lub SSL/TLS.
2. Tryb, w którym działa pierwsza trasa MQIPT . Format komunikacji w Internecie między systemami MQIPT i IBM MQ jest określany przez konfigurację trasy MQIPT . Należy zauważyć, że tam, gdzie w tabeli jest mowa o protokole SSL, można również użyć protokołu TLS.
3. Tryb, w którym działa druga trasa MQIPT .
4. Protokół używany między drugą trasą MQIPT i docelowym menedżerem kolejek.

Tabela 27. Poprawne konfiguracje z wieloma instancjami MQIPT

1. IBM MQ Protokół źródła	2. Tryb pierwszej MQIPT trasy	3. Tryb drugiej MQIPT trasy	4. IBM MQ Protokół docelowy
FAP (domyślny)	FAP-proxy (domyślny)	FAP-proxy (domyślny)	FAP
	FAP-serwer i SSL-klient	SSL-proxy	Protokół SSL/TLS
		SSL-serwer i FAP-klient	FAP
		SSL-serwer i SSL-klient	Protokół SSL/TLS
	HTTP-client	HTTP-serwer i klient SSL	Protokół SSL/TLS
	HTTPS-client	HTTPS-server i SSL-client	Protokół SSL/TLS
	HTTP-client	HTTP-serwer	FAP
HTTPS-client	HTTPS-serwer	FAP	
Protokół SSL/TLS	SSL-proxy	SSL-proxy	Protokół SSL/TLS
		SSL-serwer i FAP-klient	FAP
		SSL-serwer i SSL-klient	Protokół SSL/TLS
	HTTP-client	HTTP-serwer	FAP
	HTTPS-client	HTTPS-serwer	Protokół SSL/TLS
	HTTP-client	HTTP-serwer i klient SSL	FAP
	HTTPS-client	HTTPS-server i SSL-client	Protokół SSL/TLS

Obsługiwane konfiguracje kanałów

Obsługiwane są wszystkie typy kanałów IBM MQ, ale konfiguracja jest ograniczona do połączeń TCP/IP. Dla klienta lub menedżera kolejek systemu IBM MQ program MQIPT jest wyświetlany jako docelowy menedżer kolejek. Jeśli konfiguracja kanału wymaga hosta docelowego i numeru portu, podana jest nazwa hosta MQIPT i numer portu nastuchiwania.

Kanały klient/serwer

Program MQIPT nastuchuje przychodzących żądań połączeń klienckich, a następnie przekazuje je za pomocą tunelowania HTTP, protokołu SSL/TLS lub jako standardowe pakiety protokołu IBM MQ. Jeśli produkt MQIPT używa tunelowania HTTP lub protokołu SSL/TLS, przekazuje je w połączeniu z drugim serwerem MQIPT. Jeśli nie jest używane tunelowanie HTTP, przekazuje je w połączeniu do tego, co widzi jako docelowy menedżer kolejek (choć może to być kolejny MQIPT). Gdy docelowy menedżer kolejek zaakceptował połączenie klienta, pakiety są przekazywane między klientem a serwerem.

Kanały nadawcze/odbiorcze klastra

Jeśli produkt MQIPT odbierze żądanie przychodzące z kanału nadawczego klastra, przyjmuje, że menedżer kolejek ma włączoną obsługę mechanizmu SOCKS, a prawdziwy adres docelowy zostanie uzyskany podczas procesu uzgadniania mechanizmu SOCKS. Przekazuje żądanie do następnego menedżera kolejek MQIPT lub docelowego menedżera kolejek dokładnie w taki sam sposób, jak w przypadku kanałów połączenia klienta. Obejmuje to również automatycznie zdefiniowane kanały nadawcze klastra.

Nadawca/odbiorca

Jeśli produkt MQIPT odbiera żądanie przychodzące z kanału nadawczego, przekazuje je do następnego menedżera kolejek MQIPT lub docelowego menedżera kolejek dokładnie w taki sam sposób, jak w przypadku kanałów połączenia klienckiego. Docelowy menedżer kolejek sprawdza poprawność przychodzącego żądania i w razie potrzeby uruchamia kanał odbiorczy. Cała komunikacja między kanałem nadawczym i odbiorczym (w tym przepływy zabezpieczeń) jest przekazywana.

Requester/serwer

Ta kombinacja jest obsługiwana w taki sam sposób, jak w poprzednich konfiguracjach. Sprawdzenie poprawności żądania połączenia jest wykonywane przez kanał serwera w docelowym menedżerze kolejek.

Requester/nadawca

Konfiguracja wywołania zwrotnego może być używana, jeśli dwa menedżery kolejek nie mogą nawiązywać bezpośrednich połączeń ze sobą, ale oba mogą nawiązywać połączenia z programem MQIPT i akceptować połączenia z niego.

Serwer/requester i serwer/odbiornik

Są one obsługiwane przez produkt MQIPT w taki sam sposób, w jaki obsługuje on konfigurację Sender/Receiver.

Warunki zakończenia i awarii kanału

Gdy program MQIPT wykryje zamknięcie kanału IBM MQ (normalne lub nieprawidłowe), propaguje zamknięcie kanału. Jeśli trasa zostanie zamknięta za pomocą programu MQIPT, wszystkie kanały przechodzące przez tę trasę zostaną zamknięte.

MQIPT udostępnia opcjonalne narzędzie limitu czasu bezczynności. Jeśli program MQIPT wykryje, że kanał był bezczynny przez czas przekraczający limit czasu, wykona natychmiastowe zamknięcie dwóch przedmiotowych połączeń.

Systemy IBM MQ na końcu kanału obserwują te nieprawidłowe warunki zamknięcia albo jako awarie sieci, albo jako zakończenie kanału przez partnera. Kanał jest następnie w stanie restartować i odtwarzać (jeśli awaria wystąpi podczas okresu wrażliwego protokołu), tak jakby nie była używana opcja MQIPT.

Bezpieczeństwo komunikatów

Zarządzanie kolejkami rozproszonymi IBM MQ zapewnia poprawne dostarczanie komunikatów. Tak jest nadal w przypadku, gdy między dwoma końcami kanału występuje łańcuch MQIPT. Produkt MQIPT nie zapisuje żadnych danych komunikatu ani nie uczestniczy w procedurze punktu synchronizacji, która zapewnia poprawne dostarczanie komunikatów.

W przypadku korzystania z szybkich, nietrwałych komunikatów IBM MQ, jeśli trasa MQIPT nie powiedzie się lub zostanie zrestartowana, gdy komunikat IBM MQ jest w trakcie przesyłania, komunikat może zostać utracony. Przed zrestartowaniem trasy należy upewnić się, że wszystkie kanały IBM MQ używające trasy MQIPT są nieaktywne.

Więcej informacji na temat bezpieczeństwa komunikatów w produkcie IBM MQ zawiera sekcja [Bezpieczeństwo komunikatów](#).

Menedżery kolejek z wieloma instancjami i wysoka dostępność

Produkt MQIPT może być używany z menedżerami kolejek z wieloma instancjami w środowiskach wysokiej dostępności.

MQIPT nie ma trwałego stanu i dlatego nie ma korzyści z przepiętowania MQIPT do innego systemu. Zamiast tego należy uruchomić wiele instancji programu MQIPT z identycznymi plikami konfiguracyjnymi `mqipt.conf` w różnych systemach. Monitoruj dostępność każdej instancji programu MQIPT i w razie potrzeby zrestartuj ją (w tym samym systemie). Zapewnia to zestaw identycznych instancji MQIPT, które mogą być używane do kierowania połączeń. Następnie należy upewnić się, że program IBM MQ może kierować połączenia do programu MQIPT oraz że program MQIPT może przekazywać te połączenia do docelowego menedżera kolejek.

Kanały wychodzące IBM MQ mogą być kierowane do dostępnej instancji MQIPT na różne sposoby, na przykład:

- Należy użyć systemu równoważenia obciążenia lub routera wysokiej dostępności, takiego jak IBM Network Dispatcher w produkcie WebSphere Edge Components.

- Podaj wiele nazw połączeń w definicji kanału IBM MQ , używając listy rozdzielanej przecinkami. Następnie program IBM MQ próbuje połączyć się z każdym adresem MQIPT , dopóki nie znajdzie dostępnej instancji MQIPT .

Należy również skierować połączenia z programu MQIPT do docelowego menedżera kolejek. Jeśli konfiguracja wysokiej dostępności zapewnia, że adres IP jest przetaczany awaryjnie z docelowym menedżerem kolejek, nie jest wymagana żadna specjalna konfiguracja produktu MQIPT : należy określić docelowy adres IP we właściwości trasy **Destination** i zezwolić operacji przetaczania awaryjnego na przeniesienie adresu IP z menedżerem kolejek.

Jeśli jednak adres IP menedżera kolejek ulegnie zmianie po przetaczeniu awaryjnym, należy skonfigurować program MQIPT tak, aby przekazywał połączenie do poprawnego miejsca docelowego. Można to zrobić na kilka sposobów:

- Napisz wyjście routingu, które sprawdza, który adres IP i numer portu są dostępne, a następnie nadpisz miejsce docelowe trasy dla każdego połączenia. Niektóre przykładowe wyjścia routingu są dostarczane z produktem MQIPT i można je w tym celu zaadaptować.
- Użyj systemu równoważenia obciążenia o wysokiej dostępności, aby przekierować połączenie.
- Zdefiniuj wiele tras MQIPT , po jednej dla każdego adresu IP i portu, na którym może działać menedżer kolejek. Następnie skieruj połączenia IBM MQ do różnych tras MQIPT , na przykład wyświetlając wszystkie adresy IP i numery portów tras w postaci listy rozdzielanej przecinkami w nazwie połączenia kanału wychodzącego.

Ważne jest również, aby dostroić wszystkie komponenty na całej ścieżce sieciowej:

1. Próby nawiązania połączenia z niedostępnymi systemami muszą szybko zakończyć się niepowodzeniem, aby próby ponownego nawiązania połączenia mogły zostać przeniesione do pierwszego dostępnego miejsca docelowego.

W przypadku tras MQIPT SSL dostrój właściwość trasy **SSLClientConnectTimeout** , aby zapewnić niepowodzenie połączenia dla niedostępnych miejsc docelowych. Szczegółowe informacje na temat parametrów strojenia bazy danych IBM MQ można znaleźć w dokumentacji systemu IBM MQ . Szczegółowe informacje na temat strojenia TCP/IP dla systemu operacyjnego można znaleźć w dokumentacji systemu operacyjnego. We wszystkich przypadkach nieudane próby nawiązania połączenia powinny szybko zwrócić awarię sieci (na przykład pakiet resetowania TCP) lub bez zbędnej zwłoki.

2. Aktywne połączenia z systemem, który uległ awarii, muszą być natychmiast zerwane, aby można było nawiązywać nowe połączenia.

Należy również wziąć pod uwagę wpływ przetaczania awaryjnego w czasie, gdy połączenia aktywnie korzystają z produktu MQIPT. Prawdopodobnie połączenia sieciowe zostaną zerwane podczas przetaczania awaryjnego. W przypadku aplikacji klienckich można użyć funkcji automatycznego ponownego nawiązywania połączeń z klientem IBM MQ w celu ponownego nawiązania zerwanych połączeń. W przypadku kanałów komunikatów można określić odstęp czasu między krótkimi ponownymi próbami, aby kanał natychmiast nawiązywał połączenie. Więcej informacji na temat automatycznego ponownego nawiązywania połączenia przez klienta i ponownego konfigurowania kanału komunikatów zawiera dokumentacja IBM MQ .

9.3.5 IBM MQ Console i REST API

IBM MQ Console i REST API umożliwiają administrowanie programem IBM MQ i wykonywanie operacji przesyłania komunikatów przy użyciu protokołu HTTP.

- Program IBM MQ Console umożliwia wykonywanie podstawowych czynności administracyjnych z poziomu przeglądarki WWW. Więcej informacji na ten temat zawiera sekcja [Administrowanie przy użyciu produktu IBM MQ Console](#).
- Program administrative REST API umożliwia administrowanie obiektami produktu IBM MQ , takimi jak menedżery kolejek i kolejki, oraz agentami Managed File Transfer i transferami. Więcej informacji na ten temat zawiera sekcja [Administrowanie przy użyciu produktu REST API](#).

- Za pomocą programu messaging REST API można wykonywać proste przesyłanie komunikatów w trybie punkt z punktem i publikowanie. Więcej informacji na ten temat zawiera sekcja [Przesyłanie komunikatów przy użyciu programu REST API](#).

Opcje instalacji

Produkt IBM MQ Console i produkt REST API działają na serwerze WebSphere Liberty o nazwie mqweb. Z poziomu produktu IBM MQ 9.3.5 można zainstalować serwer mqweb jako komponent opcjonalny w instalacji produktu IBM MQ lub jako autonomiczną instalację produktu IBM MQ Web Server .

Autonomiczna instalacja systemu IBM MQ Web Server

Z poziomu produktu IBM MQ 9.3.5 serwer mqweb może działać w instalacji autonomicznej produktu IBM MQ Web Server. Autonomiczna instalacja produktu IBM MQ Web Server umożliwia zainstalowanie i uruchomienie serwera mqweb w systemach innych niż instalacje produktu IBM MQ . Zainstalowanie autonomicznego produktu IBM MQ Web Server zapewnia większą elastyczność w wyborze systemów i liczby systemów, na których mają być uruchamiane serwery mqweb. W razie potrzeby kilka instancji serwera mqweb można uruchomić na różnych komputerach, aby zapewnić wymaganą skalowalność i dostępność.

Jeśli zakupiono uprawnienie IBM MQ , można zainstalować tyle kopii, ile jest wymagane dla autonomicznego IBM MQ Web Server. Instalacje IBM MQ Web Server nie są wliczane do zakupionego uprawnienia IBM MQ . Więcej informacji na temat licencjonowania produktu IBM MQ zawiera sekcja [Informacje licencyjne produktu IBM MQ](#).

W przypadku autonomicznej instalacji IBM MQ Web Server obowiązują następujące ograniczenia:

- Program IBM MQ Console może być używany tylko do administrowania zdalnymi menedżerami kolejek.
- Parametr messaging REST API może być używany tylko ze zdalnymi menedżerami kolejek.
- Plik administrative REST API nie jest dostępny.

Autonomiczny IBM MQ Web Server jest obsługiwany tylko na platformach Linux .

Więcej informacji na temat instalowania autonomicznego serwera IBM MQ Web Server zawiera sekcja [Instalowanie autonomicznego serwera IBM MQ Web Server](#).






Opcjonalny komponent instalacji produktu IBM MQ

Komponenty IBM MQ Console i REST API można zainstalować w ramach instalacji systemu IBM MQ .

Wszystkie składniki IBM MQ Console i REST API są dostępne, gdy serwer mqweb działa w instalacji produktu IBM MQ .

- Program IBM MQ Console może być używany do administrowania lokalnymi i zdalnymi menedżerami kolejek.
- Program messaging REST API może być używany z lokalnymi i zdalnymi menedżerami kolejek.
- Program administrative REST API może być używany do administrowania lokalnymi i zdalnymi menedżerami kolejek.

Aby korzystać z komponentów IBM MQ Console i REST API , należy zainstalować następujący komponent w ramach instalacji produktu IBM MQ :

-  W systemie AIX zainstaluj zestaw plików mqm.web.rte .
-  W systemie IBM i zainstaluj komponent WEB.
-  W systemie Linux zainstaluj komponent MQSeriesWeb .
-  W systemie Windows zainstaluj składnik Web Administration .
-  W systemie z/OS zainstaluj składnik IBM MQ for z/OS UNIX System Services Web Components .

Uwagi

Niniejsza publikacja została opracowana z myślą o produktach i usługach oferowanych w Stanach Zjednoczonych.

IBM może nie oferować w innych krajach produktów, usług lub opcji omawianych w tej publikacji. Informacje o produktach i usługach dostępnych w danym kraju można uzyskać od lokalnego przedstawiciela IBM. Odwołanie do produktu, programu lub usługi IBM nie oznacza, że można użyć wyłącznie tego produktu, programu lub usługi IBM. Zamiast nich można zastosować ich odpowiednik funkcjonalny pod warunkiem, że nie narusza to praw własności intelektualnej firmy IBM. Jednakże cała odpowiedzialność za ocenę przydatności i sprawdzenie działania produktu, programu lub usługi pochodzących od producenta innego niż IBM spoczywa na użytkowniku.

IBM może posiadać patenty lub złożone wnioski patentowe na towary i usługi, o których mowa w niniejszej publikacji. Przedstawienie niniejszej publikacji nie daje żadnych uprawnień licencyjnych do tychże patentów. Pisemne zapytania w sprawie licencji można przesyłać na adres:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Zapytania w sprawie licencji dotyczących informacji kodowanych przy użyciu dwubajtowych zestawów znaków (DBCS) należy kierować do lokalnych działów IBM Intellectual Property Department lub zgłaszać na piśmie pod adresem:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

Poniższy akapit nie obowiązuje w Wielkiej Brytanii, a także w innych krajach, w których jego treść pozostaje w sprzeczności z przepisami prawa miejscowego: INTERNATIONAL BUSINESS MACHINES CORPORATION DOSTARCZA TĘ PUBLIKACJĘ W STANIE, W JAKIM SIĘ ZNAJDUJE ("AS IS"), BEZ JAKICHKOLWIEK GWARANCJI (RĘKOJMIĘ RÓWNIEŻ WYŁĄCZA SIĘ), WYRAŻNYCH LUB DOMNIEMANYCH, A W SZCZEGÓLNOŚCI DOMNIEMANYCH GWARANCJI PRZYDATNOŚCI HANDLOWEJ, PRZYDATNOŚCI DO OKREŚLONEGO CELU ORAZ GWARANCJI, ŻE PUBLIKACJA TA NIE NARUSZA PRAW OSÓB TRZECICH. Ustawodawstwa niektórych krajów nie dopuszczają zastrzeżeń dotyczących gwarancji wyraźnych lub domniemanych w odniesieniu do pewnych transakcji; w takiej sytuacji powyższe zdanie nie ma zastosowania.

Informacje zawarte w niniejszej publikacji mogą zawierać nieścisłości techniczne lub błędy typograficzne. Informacje te są okresowo aktualizowane, a zmiany te zostaną uwzględnione w kolejnych wydaniach tej publikacji. IBM zastrzega sobie prawo do wprowadzania ulepszeń i/lub zmian w produktach i/lub programach opisanych w tej publikacji w dowolnym czasie, bez wcześniejszego powiadomienia.

Wszelkie wzmianki w tej publikacji na temat stron internetowych innych podmiotów zostały wprowadzone wyłącznie dla wygody użytkowników i w żadnym wypadku nie stanowią zachęty do ich odwiedzania. Materiały dostępne na tych stronach nie są częścią materiałów opracowanych dla tego produktu IBM, a użytkownik korzysta z nich na własną odpowiedzialność.

IBM ma prawo do używania i rozpowszechniania informacji przystanych przez użytkownika w dowolny sposób, jaki uzna za właściwy, bez żadnych zobowiązań wobec ich autora.

Licencjodawcy tego programu, którzy chcieliby uzyskać informacje na temat programu w celu: (i) wdrożenia wymiany informacji między niezależnie utworzonymi programami i innymi programami (łącznie

z tym opisywanym) oraz (ii) wspólnego wykorzystywania wymienianych informacji, powinni skontaktować się z:

IBM Corporation
Koordynator współdziałania oprogramowania, dział 49XA
3605 Autostrada 52 N
Rochester, MN 55901
U.S.A.

Informacje takie mogą być udostępnione, o ile spełnione zostaną odpowiednie warunki, w tym, w niektórych przypadkach, zostanie uiszczona stosowna opłata.

Licencjonowany program opisany w niniejszej publikacji oraz wszystkie inne licencjonowane materiały dostępne dla tego programu są dostarczane przez IBM na warunkach określonych w Umowie IBM z Klientem, Międzynarodowej Umowie Licencyjnej IBM na Program lub w innych podobnych umowach zawartych między IBM i użytkownikami.

Wszelkie dane dotyczące wydajności zostały zebrane w kontrolowanym środowisku. W związku z tym rezultaty uzyskane w innych środowiskach operacyjnych mogą się znacząco różnić. Niektóre pomiary mogły być dokonywane na systemach będących w fazie rozwoju i nie ma gwarancji, że pomiary wykonane na ogólnie dostępnych systemach dadzą takie same wyniki. Niektóre z pomiarów mogły być estymowane przez ekstrapolację. Rzeczywiste wyniki mogą być inne. Użytkownicy powinni we własnym zakresie sprawdzić odpowiednie dane dla ich środowiska.

Informacje dotyczące produktów innych niż produkty IBM pochodzą od dostawców tych produktów, z opublikowanych przez nich zapowiedzi lub innych powszechnie dostępnych źródeł. Firma IBM nie testowała tych produktów i nie może potwierdzić dokładności pomiarów wydajności, kompatybilności ani żadnych innych danych związanych z tymi produktami. Pytania dotyczące możliwości produktów innych podmiotów należy kierować do dostawców tych produktów.

Wszelkie stwierdzenia dotyczące przyszłych kierunków rozwoju i zamierzeń IBM mogą zostać zmienione lub wycofane bez powiadomienia.

Publikacja ta zawiera przykładowe dane i raporty używane w codziennych operacjach działalności gospodarczej. W celu kompleksowego ich zilustrowania, podane przykłady zawierają nazwiska osób prywatnych, nazwy przedsiębiorstw oraz nazwy produktów. Wszystkie te nazwy/nazwiska są fikcyjne i jakiegokolwiek podobieństwo do istniejących nazw/nazwisk i adresów jest całkowicie przypadkowe.

LICENCJA W ZAKRESIE PRAW AUTORSKICH:

Niniejsza publikacja zawiera przykładowe aplikacje w kodzie źródłowym, ilustrujące techniki programowania w różnych systemach operacyjnych. Użytkownik może kopiować, modyfikować i dystrybuować te programy przykładowe w dowolnej formie bez uiszczania opłat na rzecz IBM, w celu projektowania, używania, sprzedaży lub dystrybucji aplikacji zgodnych z aplikacyjnym interfejsem programistycznym dla tego systemu operacyjnego, dla którego napisane zostały programy przykładowe. Programy przykładowe nie zostały gruntownie przetestowane. IBM nie może zatem gwarantować ani sugerować niezawodności, użyteczności i funkcjonalności tych programów.

W przypadku przeglądania niniejszych informacji w formie elektronicznej, zdjęcia i kolorowe ilustracje mogą nie być wyświetlane.

Informacje dotyczące interfejsu programistycznego

Informacje o interfejsie programistycznym, jeśli są dostępne, mają na celu pomóc w tworzeniu aplikacji do użycia z tym programem.

Podręcznik ten zawiera informacje na temat interfejsów programistycznych, które umożliwiają klientom pisanie programów w celu uzyskania dostępu do usług produktu WebSphere MQ.

Informacje te mogą również zawierać informacje na temat diagnostyki, modyfikacji i strojenia. Tego typu informacje są udostępniane jako pomoc przy debugowaniu aplikacji.

Ważne: Informacji o diagnostyce, modyfikacji i strojeniu nie należy używać jako interfejsu programistycznego, ponieważ mogą one ulec zmianie.

Znaki towarowe

IBM, logo IBM, ibm.com są znakami towarowymi IBM Corporation zarejestrowanymi w wielu systemach prawnych na całym świecie. Aktualna lista znaków towarowych IBM dostępna jest w serwisie WWW IBM, w sekcji "Copyright and trademark information" (Informacje o prawach autorskich i znakach towarowych), pod adresem www.ibm.com/legal/copytrade.shtml. Nazwy innych produktów lub usług mogą być znakami towarowymi IBM lub innych podmiotów.

Microsoft oraz Windows są znakami towarowymi firmy Microsoft Corporation w Stanach Zjednoczonych i/lub innych krajach.

UNIX jest zastrzeżonym znakiem towarowym The Open Group w Stanach Zjednoczonych i/lub w innych krajach.

Linux jest zastrzeżonym znakiem towarowym Linusa Torvaldsa w Stanach Zjednoczonych i/lub w innych krajach.

Ten produkt zawiera oprogramowanie opracowane przez Eclipse Project (<https://www.eclipse.org/>).

Java oraz wszystkie znaki towarowe i logo dotyczące języka Java są znakami towarowymi lub zastrzeżonymi znakami towarowymi Oracle i/lub przedsiębiorstw afiliowanych Oracle.



Numer pozycji:

(1P) P/N: