

7.5

*Administrace produktu IBM WebSphere
MQ*

IBM

Poznámka

Než začnete používat tyto informace a produkt, který podporují, přečtěte si informace, které uvádí [“Poznámky” na stránce 159](#).

Toto vydání se vztahuje k verzi 7, vydání 5 produktu IBM® WebSphere MQ a ke všem následujícím vydáním a modifikacím, dokud nebude v nových vydáních uvedeno jinak.

Když odešlete informace do IBM, udělíte společnosti IBM nevýlučné právo použít nebo distribuovat informace libovolným způsobem, který společnost považuje za odpovídající, bez vzniku jakýchkoliv závazků vůči vám.

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

Obsah

Správa serveru.....	5
Lokální a vzdálená administrace.....	7
Jak používat řídicí příkazy produktu IBM WebSphere MQ.....	8
Automatizace administračních úloh.....	8
Úvod do formátu programových příkazů.....	9
Použití rozhraní MQAI ke zjednodušení používání PCFs.....	19
Úvod do rozhraní IBM WebSphere MQ Administration Interface (MQAI).....	19
Rozhraní IBM WebSphere MQ Administration Interface (MQAI).....	21
Administrace pomocí Průzkumníka IBM WebSphere MQ.....	56
Co lze dělat s Průzkumníkem IBM WebSphere MQ.....	56
Nastavení produktu IBM WebSphere MQ Explorer.....	58
Zabezpečení v systému Windows.....	64
Rozšíření produktu IBM WebSphere MQ Explorer (pouze platformy Windows a Linux x86).....	67
Použití aplikace Taskbar produktu IBM WebSphere MQ (pouze Windows).....	67
Aplikace pro monitorování výstrah produktu IBM WebSphere MQ (pouze Windows).....	67
Správa lokálních objektů produktu IBM WebSphere MQ.....	68
Spuštění a zastavení správce front.....	68
Ruční zastavování správců front.....	70
Provádění lokálních úloh administrace pomocí příkazů MQSC.....	72
Práce se správci front.....	80
Práce s lokálními frontami.....	82
Práce s frontami aliasů.....	87
Práce s modelové fronty.....	88
Práce s administrativními tématy.....	89
Práce s odběry.....	92
Práce se službami.....	95
Správa objektů pro spuštění.....	101
Správa vzdálených objektů produktu IBM WebSphere MQ.....	103
Kanály, klastry a vzdálené fronty.....	103
Vzdálená administrace z lokálního správce front.....	105
Vytvoření lokální definice vzdálené fronty.....	111
Použití definic vzdálených front jako aliasů.....	113
Převod dat.....	113
Správa produktu IBM WebSphere MQ Telemetry.....	115
Konfigurace správce front pro telemetrie v systémech Linux a AIX.....	116
Konfigurace správce front pro telemetrii v systému Windows.....	118
Konfigurovat správce front pro odesílání zpráv klientům MQTT.....	119
Identifikace, autorizace a ověření klienta.....	122
Ověření kanálu telemetrie pomocí zabezpečení SSL.....	128
Ochrana soukromí při publikování pomocí SSL.....	130
Konfigurace SSL.....	131
Konfigurace JAAS.....	136
Koncepte zařízení IBM WebSphere MQ Telemetry pro zařízení.....	138
Administrace výběrového vysílání.....	148
Začínáme s výběrovým vysíláním.....	149
Topologie témat výběrového vysílání IBM WebSphere MQ.....	150
Zmenšení velikosti zpráv výběrového vysílání.....	150
Povolení konverze dat pro systém zpráv výběrového vysílání.....	152
Administrace multicast a monitorování.....	153
Nastavení historie zpráv odběru výběrového vysílání.....	154
Rozšířené úlohy výběrového vysílání.....	154
Administrace serveru HP Integrity NonStop Server.....	157

Ruční spuštění TMF/Gateway z Pathway.....	158
Zastavení TMF/Gateway z Pathway.....	158
Poznámky.....	159
Informace o programovacím rozhraní.....	160
Ochranné známky.....	160

Správa serveru IBM WebSphere MQ

Administrace správců front a přidružených prostředků zahrnuje úlohy, které se často provádějí při aktivaci a správě těchto prostředků. Vyberte si způsob, kterým chcete spravovat správce front a přidružené prostředky.

Objekty produktu IBM WebSphere MQ můžete spravovat lokálně nebo vzdáleně, viz [“Lokální a vzdálená administrace”](#) na stránce 7.

Existuje řada různých metod, které můžete použít k vytváření a administraci správců front a souvisejících prostředků v produktu IBM WebSphere MQ. Tyto metody zahrnují rozhraní příkazového řádku, grafické uživatelské rozhraní a rozhraní API administrace. Prohlédněte si sekce a odkazy v tomto tématu, kde získáte další informace o každém z těchto rozhraní.

K dispozici jsou různé sady příkazů, které můžete použít ke správě produktu IBM WebSphere MQ v závislosti na platformě:

- [“Řídící příkazy produktu IBM WebSphere MQ”](#) na stránce 5
- [“IBM WebSphere MQ Skript skriptů \(MQSC\)”](#) na stránce 5
- [“Programovatelné formáty příkazů \(PCFs\)”](#) na stránce 6

Pro vytváření a správu objektů produktu IBM WebSphere MQ existují také další následující volby:

- [“Příkaz IBM WebSphere MQ Explorer”](#) na stránce 6
- [“Výchozí konfigurační aplikace Windows”](#) na stránce 6
- [“Služba MSCS \(Microsoft Cluster Service\).”](#) na stránce 7

Pomocí příkazů PCF můžete automatizovat některé úlohy administrace a monitorování pro lokální i vzdálené správce front. Tyto příkazy lze také zjednodušit prostřednictvím použití rozhraní MQAI (IBM WebSphere MQ Administration Interface) na některých platformách. Další informace o automatizaci administrativních úloh viz [“Automatizace administrativních úloh”](#) na stránce 8.

Řídící příkazy produktu IBM WebSphere MQ

Řídící příkazy vám umožňují provádět administrativní úlohy na samotných správcích front.

IBM WebSphere MQ for Windows, systémy UNIX and Linux® poskytují *řídící příkazy*, které vydáváte na příkazovém řádku systému.

Řídící příkazy jsou popsány v tématu [Vytvoření a správa správců front](#). Informace o příkazu pro řídící příkazy naleznete v příručce [IBM WebSphere MQ Control commands](#).

IBM WebSphere MQ Skript skriptů (MQSC)

Pomocí příkazů MQSC spravujte objekty správce front včetně samotného správce front, front, definic procesů, seznamů názvů, kanálů, kanálů připojení klienta, služeb listenerů, služeb ověřování a objektů ověřovacích informací.

Příkazy MQSC zadáte do správce front pomocí příkazu `runmqsc`. Můžete to provést interaktivně, vydávat příkazy z klávesnice nebo můžete přesměrovat standardní vstupní zařízení (`stdin`) tak, aby se spustily posloupnosti příkazů z textového souboru ASCII. V obou případech je formát příkazů stejný.

Příkaz `runmqsc` můžete spustit ve třech režimech, v závislosti na parametrech nastavených v příkazu:

- *Režim ověření*, kde jsou příkazy MQSC ověřeny v lokálním správcí front, ale nejsou spuštěny
- *Přímý režim*, kde jsou příkazy MQSC spuštěny v lokálním správcí front
- *Nepřímý režim*, kde jsou příkazy MQSC spuštěny ve vzdáleném správcí front

Atributy objektu uvedené v příkazech MQSC se zobrazí v této sekci velkými písmeny (např. `RQMNAME`), i když nejsou citlivé na velikost písmen. Názvy atributů příkazu MQSC jsou omezeny na osm znaků.

Příkazy MQSC jsou k dispozici na všech platformách. Příkazy MQSC jsou shrnuty v části [Porovnávání sad příkazů](#).

V systémech Windows, UNIX nebo Linux můžete použít MQSC jako jednotlivé příkazy vydané na příkazovém řádku systému. Chcete-li vydat více komplikovanější nebo více příkazů, lze prostředí MQSC sestavit do souboru, který spustíte z příkazového řádku systému Windows, UNIX nebo Linux . Prostředí MQSC lze odeslat do vzdáleného správce front. Podrobné informace najdete v tématu [Odkaz na MQSC](#).

“Příkazy skriptu (MQSC)” na stránce 72 obsahuje popis každého příkazu MQSC a jeho syntaxi.

Další informace o použití příkazů MQSC v lokální administraci viz [“Provádění lokálních úloh administrace pomocí příkazů MQSC”](#) na stránce 72 .

Programovatelné formáty příkazů (PCFs)

Programovatelné formáty příkazů (PCFs) definují příkaz a odpovědi na zprávy, které lze vyměňovat mezi programem a libovolným správcem front (který podporuje PCFs) v síti. Můžete použít příkazy PCF v aplikačním programu správy systému pro administraci objektů produktu IBM WebSphere MQ : objekty ověřovacích informací, kanály, listenery kanálů, seznamy názvů, definice procesů, správce front, fronty, služby a paměťové třídy. Aplikace může pracovat z jediného místa v síti ke komunikaci s informacemi o příkazech a odpovědích s libovolným správcem front, lokálním nebo vzdáleným pomocí lokálního správce front.

Další informace o PCFs viz [“Úvod do formátu programových příkazů”](#) na stránce 9.

Definice PCFs a struktur pro příkazy a odpovědi naleznete v části [Odkaz na formátovatelné formáty příkazů](#).

Příkaz IBM WebSphere MQ Explorer

Pomocí konzoly IBM WebSphere MQ Explorer můžete provádět následující akce:

- Definujte a řídíte různé prostředky včetně správců front, front, definic procesů, seznamů názvů, kanálů, kanálů připojení klienta, modulů listener, služeb a klastrů.
- Spustit nebo zastavit lokálního správce front a jeho přidružené procesy.
- Zobrazte si správce front a jejich přidružené objekty na pracovní stanici nebo z jiných pracovních stanic.
- Zkontrolujte stav správců front, klastrů a kanálů.
- Zkontrolujte, které aplikace, uživatelé nebo kanály mají otevřenou konkrétní frontu ze stavu fronty.

V systémech Windows a Linux můžete produkt IBM WebSphere MQ Explorer spustit pomocí systémové nabídky, spustitelného souboru MQExplorer nebo příkazu **strmqcfig** .

Chcete-li úspěšně spustit IBM WebSphere MQ Explorer na Linux, musíte být schopni zapsat soubor do svého domovského adresáře a domovský adresář musí existovat.

Produkt IBM WebSphere MQ Explorer můžete použít ke správě vzdálených správců front na jiných platformách včetně produktu z/OS, pro podrobnosti a ke stažení balíku SupportPac MS0T, viz <https://www.ibm.com/support/docview.wss?uid=swg24021041>.

Další informace viz [“Administrace pomocí produktu IBM WebSphere MQ Explorer”](#) na stránce 56.

Výchozí konfigurační aplikace Windows

Výchozí konfigurační program systému Windows můžete použít k vytvoření *startéru* (nebo výchozí) sady objektů IBM WebSphere MQ . Souhrn vytvořených výchozích objektů je uveden v tabulce [Tabulka 1: Objekty vytvořené výchozí aplikací pro konfiguraci systému Windows](#).

Služba MSCS (Microsoft Cluster Service).

Služba Microsoft Cluster Service (MSCS) vám umožňuje připojit servery do *klastru*, což poskytuje vyšší dostupnost dat a aplikací a usnadňuje správu systému. MSCS může automaticky zjišťovat a zotavovat se ze selhání serveru nebo aplikací.

Je důležité nesměšujte klastry v prostředí MSCS s klastry IBM WebSphere MQ . Rozlišování je:

IBM WebSphere MQKlastry

jsou skupiny dvou nebo více správců front na jednom nebo více počítačích, poskytují automatické propojení a umožňují sdílení front mezi nimi pro vyrovnávání zátěže a redundanci.

Klastry MSCS

Skupiny počítačů, připojené a konfigurované takovým způsobem, že pokud jeden z nich selže, MSCS provede *přepnutí při selhání*, převede stavová data aplikací ze selhávajícího počítače na jiný počítač v klastru a znovu iniciuje jejich činnost.

Podpora služby Microsoft Cluster Service (MSCS) poskytuje podrobné informace o tom, jak nakonfigurovat systém IBM WebSphere MQ pro systém Windows pro použití MSCS.

Související pojmy

[Technický přehled produktu WebSphere MQ](#)

[“Správa lokálních objektů produktu IBM WebSphere MQ” na stránce 68](#)

Tato sekce vám sděluje, jak spravovat lokální objekty produktu IBM WebSphere MQ pro podporu aplikačních programů, které používají rozhraní MQI (Message Queue Interface). V tomto kontextu lokální administrace znamená vytváření, zobrazování, změny, kopírování a odstraňování objektů IBM WebSphere MQ .

[“Správa vzdálených objektů produktu IBM WebSphere MQ” na stránce 103](#)

[Aspekty při ztrátě kontaktu se správcem prostředků XA](#)

Související úlohy

[Naplánování](#)

[Konfigurace](#)

Související odkazy

[Scénáře transakčního podpory](#)

Lokální a vzdálená administrace

Objekty produktu WebSphere MQ lze spravovat lokálně nebo vzdáleně.

Lokální administrace znamená provádění úloh administrace u všech správců front, které jste definovali na svém lokálním systému. Můžete přistoupit k jiným systémům, například přes emulační program terminálu TCP/IP **telnet**, a provádět administraci zde. V produktu WebSphere MQ je to možné považovat za lokální administraci, protože nejsou zahrnuty žádné kanály, což znamená, že komunikace je spravována operačním systémem.

Produkt WebSphere MQ podporuje administraci z jednoho kontaktního místa pomocí známého jako *vzdálená administrace*. To umožňuje vydávat příkazy z lokálního systému, které jsou zpracovány v jiném systému, a platí také pro program Průzkumník produktu WebSphere MQ . Chcete-li například změnit definici fronty ve vzdáleném správci front, můžete zadat vzdálený příkaz. K tomuto systému se nemusíte přihlašovat, ačkoli byste měli mít definované vhodné kanály. Musí být spuštěn správce front a příkazový server na cílovém systému.

Některé příkazy nelze tímto způsobem vydat, zejména při vytváření nebo spouštění správců front a spouštění příkazových serverů. Chcete-li provést tento typ úlohy, musíte se buď přihlásit ke vzdálenému systému a zadat příkazy z tohoto systému, nebo vytvořit proces, který může vydat příkazy pro vás. Toto omezení platí také pro program Průzkumník produktu WebSphere MQ .

[“Správa vzdálených objektů produktu IBM WebSphere MQ” na stránce 103](#) podrobně popisuje téma vzdálené administrace.

Jak používat řídicí příkazy obslužného programu IBM WebSphere MQ

Tato část popisuje, jak používat řídicí příkazy IBM WebSphere MQ .

Chcete-li vydávat příkazy pro řízení, vaše ID uživatele musí být členem skupiny mqm. Další informace viz [Oprávnění k administraci produktu WebSphere MQ v systémech UNIX, Linux a Windows](#) . Kromě toho si všimněte následujících informací specifických pro prostředí:

WebSphere MQ for Windows

Všechny řídicí příkazy mohou být vydány z příkazového řádku. Názvy příkazů a jejich příznaky nejsou citlivé na velikost písmen: můžete je zadat velkými písmeny, malými písmeny nebo kombinací velkých a malých písmen. Nicméně argumenty pro řízení příkazů (jako jsou názvy front) rozlišují velká a malá písmena.

V popisech syntaxe se spojovník (-) používá jako indikátor příznaku. Namísto pomlčky můžete místo pomlčky použít dopředné lomítko (/).

WebSphere MQ pro systémy UNIX and Linux

Všechny řídicí příkazy produktu WebSphere MQ mohou být vydávány ze shellu. Všechny příkazy jsou citlivé na velikost písmen.

Podmnožinu řídicích příkazů lze zadat pomocí Průzkumníka IBM WebSphere MQ .

Další informace naleznete v tématu [Řídicí příkazy produktu WebSphere MQ](#)

Automatizace administračních úloh

Můžete se rozhodnout, že pro vaši instalaci bude výhodné automatizovat některé úlohy administrace a monitorování. Úlohy administrace pro lokální i vzdálené správce front můžete automatizovat pomocí příkazů PCF (programmable command format). Tato sekce předpokládá, že máte zkušenosti s administrací objektů produktu WebSphere MQ .

příkazy PCF

Příkazy WebSphere MQ programmable command format (PCF) lze použít pro úlohy administrativy programu do administrativního programu. Tímto způsobem lze z programu manipulovat s objekty správce front (fronty, definice procesů, seznamy názvů, kanály, kanály připojení klienta, moduly listener, služby a objekty ověřovacích informací) a dokonce manipulovat se správci front.

Příkazy PCF pokrývají stejný rozsah funkcí poskytovaných příkazy MQSC. Můžete napsat program, který vydá příkazy PCF do libovolného správce front v síti z jednoho uzlu. Tímto způsobem můžete centralizovat a automatizovat administrativní úlohy.

Každý příkaz PCF je datová struktura, která je vložena do části dat aplikace zprávy produktu WebSphere MQ . Každý příkaz je odeslán do cílového správce front s použitím funkce MQI MQPUT stejným způsobem jako jakákoli jiná zpráva. Za předpokladu, že je příkazový server spuštěn ve správci front přijímající zprávu, interpretuje jej jako příkazovou zprávu a spustí příkaz. Chcete-li získat odpovědi, aplikace vydá volání MQGET a data odpovědi jsou vrácena v jiné datové struktuře. Aplikace pak může zpracovat odpověď a podle toho jednat.

Poznámka: Narozdíl od příkazů MQSC, příkazy PCF a jejich odpovědi nejsou v textovém formátu, který můžete číst.

Stručně řečeno, jsou to některé z věcí potřebných k vytvoření zprávy příkazu PCF:

deskriptor zprávy

Jedná se o standardní deskriptor zpráv produktu WebSphere MQ , v němž:

- Typ zprávy (*MsgType*) je MQMT_REQUEST.
- Formát zprávy (*Format*) je MQFMT_ADMIN.

Data aplikace

Obsahuje zprávu PCF včetně záhlaví PCF, v níž:

- Typ zprávy PCF (*Type*) určuje MQCFT_COMMAND.
- Identifikátor příkazu určuje příkaz, například *Change Queue* (MQCMD_CHANGE_Q).

Úplný popis datových struktur PCF a jejich implementace viz [“Úvod do formátu programových příkazů”](#) na stránce 9.

Atributy objektu PCF

Atributy objektu v PCF nejsou omezeny na osm znaků, jak jsou určeny pro příkazy MQSC. Tyto informace jsou zobrazeny v této příručce kurzívou. Například ekvivalent PCF RQMNAME je *RemoteQMGrName*.

Únikové PC

Únikové PCFs jsou příkazy PCF, které obsahují příkazy MQSC v textu zprávy. PCFs můžete použít k odeslání příkazů vzdálenému správci front. Další informace o řídicích PCF viz [Úniková](#).

Úvod do formátu programových příkazů

Programovatelné formáty příkazů (PCFs) definují příkaz a odpovědi na zprávy, které lze vyměňovat mezi programem a libovolným správcem front (který podporuje PCFs) v síti. PCFs zjednodušují administraci správců front a další správu sítě. Mohou být použity k řešení problému složité správy distribuovaných sítí, zvláště když rostou sítě ve velikosti a složitosti sítě.

Programovatelné formáty příkazů popsané v této dokumentaci produktu jsou podporovány:

- IBM WebSphere MQ for AIX
- IBM WebSphere MQ for HP-UX
- IBM WebSphere MQ for Linux
- IBM WebSphere MQ for Solaris
- IBM WebSphere MQ for Windows
- IBM WebSphere MQ for HP Integrity NonStop Server

Řešení problémů PCF

Administrativa distribuovaných sítí může být složitá. Problémy správy se stále zvětšují, neboť se zvyšují velikost a složitost sítí.

Příklady administrace, které jsou specifické pro systém zpráv a front zpráv, zahrnují:

- Správa prostředků.

Například vytvoření a odstranění fronty.

- Monitorování výkonu.

Příklad: Maximální hloubka fronty nebo četnost zpráv.

- Kontrola.

Například ladicí parametry fronty, jako je například maximální hloubka fronty, maximální délka zprávy a povolení a zakázání front.

- Směrování zpráv.

Definice alternativních cest prostřednictvím sítě.

Příkazy PCF produktu WebSphere MQ lze použít ke zjednodušení administrace správce front a jiné administrativní administrace sítě. Příkazy PCF umožňují použít jednu aplikaci k provedení síťové administrace z jednoho správce front v rámci sítě.

Co jsou PCFs?

PCFs definují zprávy příkazu a odpovědi, které lze vyměňovat mezi programem a libovolným správcem front (který podporuje PCFs) v síti. Můžete použít příkazy PCF v aplikačním programu správy systému pro administraci objektů produktu WebSphere MQ : objekty ověřovacích informací, kanály, listenery kanálů, názvy procesů, definice procesů, správce front, fronty, služby a paměťové třídy. Aplikace může pracovat z jediného místa v síti ke komunikaci s informacemi o příkazech a odpovědích s libovolným správcem front, lokálním nebo vzdáleným pomocí lokálního správce front.

Každý správce front má frontu administrace se standardním názvem fronty a vaše aplikace může odesílat zprávy příkazu PCF do této fronty. Každý správce front má také příkazový server pro obsluhu příkazových zpráv z fronty administrace. Zprávy příkazu PCF proto mohou být zpracovány libovolným správcem front v síti a data odpovědi mohou být vrácena do vaší aplikace pomocí uvedené fronty odpovědí. Příkazy PCF a zprávy odpovědí jsou odesílány a přijímány s použitím běžného rozhraní MQI (Message Queue Interface).

Seznam dostupných příkazů PCF včetně jejich parametrů naleznete v části [Definice formátů Programovatelných příkazů](#).

Použití uživatelem programovatelných formátů příkazů

PCF můžete použít v programu správy systému pro vzdálenou administraci WebSphere MQ .

Tento oddíl obsahuje:

- [“Zprávy příkazu PCF” na stránce 10](#)
- [“Odezvy” na stránce 12](#)
- [Pravidla pro pojmenování IBM WebSphere MQ objects](#)
- [“Kontrola oprávnění pro příkazy PCF” na stránce 14](#)

Zprávy příkazu PCF

Zprávy příkazu PCF se skládají ze záhlaví PCF, parametrů identifikovaných v tomto záhlaví a také uživatelem definovaných dat zprávy. Zprávy jsou vydávány pomocí volání rozhraní fronty zpráv.

Každý příkaz a jeho parametry jsou odeslány jako samostatná příkazová zpráva obsahující záhlaví PCF následovaná počtem struktur parametrů; podrobnosti o záhlaví PCF viz záhlaví MQCFH-PCFa v případě příkladu struktury parametrů viz MQCFST-Parametr řetězce PCF-PCF. Hlavička PCF identifikuje příkaz a počet struktur parametrů, které následují ve stejné zprávě. Každá struktura parametrů poskytuje parametr příkazu.

Odpovědi na příkazy, generované příkazovým serverem, mají podobnou strukturu. Existuje hlavička PCF, za kterou následuje řada struktur parametrů. Odpovědi se mohou skládat z více než jedné zprávy, ale příkazy se vždy skládají pouze z jedné zprávy.

Na jiných platformách než z/OSse fronta, do níž jsou odesílány příkazy PCF, vždy nazývá SYSTEM.ADMIN.COMMAND.QUEUE.

Jak zadat zprávy příkazu PCF

Běžné volání rozhraní MQI (Message Queue Interface), MQPUT, MQGET, atd. lze použít k umístění a načtení zpráv příkazu PCF a odpovědí na zprávy z jejich front a z jejich front.

Poznámka:

Ujistěte se, že je příkazový server spuštěný v cílovém správci front, aby mohl příkaz PCF zpracovat na daném správci front.

Seznam dodaných hlavičkového souboru najdete v tématu [WebSphere MQ COPY, header, include a module files](#).

Deskriptor zpráv pro příkaz PCF

Deskriptor zpráv produktu WebSphere MQ je plně zdokumentován v příručce [MQMD-Message descriptor](#).

Zpráva příkazu PCF obsahuje následující pole v deskriptoru zprávy:

Report

Libovolná platná hodnota, jak je požadováno.

MsgType

Toto pole musí být MQMT_REQUEST, aby se označilo, že zpráva vyžaduje odpověď.

Expiry

Libovolná platná hodnota, jak je požadováno.

Feedback

Nastavit na hodnotu MQFB_NONE

Encoding

Pokud odesíláte do systémů Windows, UNIX nebo Linux, nastavte toto pole na kódování použité pro data zprávy; převod se provede, je-li to nutné.

CodedCharSetId

Pokud odesíláte systémy Windows, UNIX nebo Linux nastaví toto pole na identifikátor kódované znakové sady použitý pro data zprávy; konverze se provede, je-li to nutné.

Format

Nastavit na MQFMT_ADMIN.

Priority

Libovolná platná hodnota, jak je požadováno.

Persistence

Libovolná platná hodnota, jak je požadováno.

MsgId

Odesílající aplikace může určit libovolnou hodnotu, nebo může být zadána hodnota MQMI_NONE, která požádá správce front o vygenerování jedinečného identifikátoru zprávy.

CorrelId

Odesílající aplikace může určovat libovolnou hodnotu, nebo lze zadat hodnotu MQCI_NONE, která udává, že neexistuje žádný identifikátor korelace.

ReplyToQ

Název fronty, která má přijmout odezvu.

ReplyToQMGr

Název správce front pro odezvu (nebo prázdný).

Pole kontextu zprávy

Tato pole lze podle potřeby nastavit na libovolné platné hodnoty. Obvykle je volba Vložit zprávu MQPMO_DEFAULT_CONTEXT použita k nastavení polí kontextu zprávy na výchozí hodnoty.

Používáte-li strukturu MQMD version-2, musíte nastavit následující dodatečná pole:

GroupId

Nastavit na hodnotu MQGI_NONE

MsgSeqNumber

Nastavit na 1

Offset

Nastavit na hodnotu 0

MsgFlags

Nastavit na MQMF_NONE

OriginalLength

Nastavit na MQOL_UNDEFINED

Odesílání uživatelských dat

Struktury PCF lze také použít k odeslání uživatelem definovaných dat zprávy. V tomto případě musí být pole deskriptoru zpráv *Format* nastaveno na MQFMT_PCF.

Odesílání a příjem zpráv PCF v určené frontě

Odesílání zpráv PCF do určené fronty

Chcete-li odeslat zprávu do určené fronty, volání funkce mqPutpřevede obsah uvedeného balíku do zprávy PCF a odešle zprávu do určené fronty. Obsah vaku se po zavolání ponechá nezměněný.

Jako vstup pro toto volání musíte dodat:

- Popisovač připojení MQI.
- Popisovač objektu pro frontu, na které má být zpráva umístěna.
- Deskriptor zprávy. Další informace o deskriptoru zpráv viz [MQMD-Message descriptor](#).
- Volby vložení zprávy pomocí struktury MQPMO. Další informace o struktuře MQPMO naleznete v tématu [Volby MQPMO-Put-message](#).
- Manipulátor balíku, který má být převeden na zprávu.

Poznámka: Pokud balík obsahuje zprávu administrace a bylo použito dotazové volání mqAddk vložení hodnot do balíku, hodnota datové položky MQIASY_COMMAND musí být příkaz INQUIRE rozpoznávaný rozhraním MQAI.

Úplný popis volání objektu mqPutBag naleznete v části [mqPutBag](#).

Příjem zpráv PCF z uvedené fronty

Chcete-li přijmout zprávu z uvedené fronty, volání funkce mqGetBag získá zprávu PCF z určené fronty a převede data zprávy na datový balík.

Jako vstup pro toto volání musíte dodat:

- Popisovač připojení MQI.
- Popisovač objektu fronty, ze které má být přečtena zpráva.
- Deskriptor zprávy. V rámci struktury MQMD musí být parametrem *Format* hodnota MQFMT_ADMIN, MQFMT_EVENT nebo MQFMT_PCF.

Poznámka: Je-li zpráva přijata v rámci pracovní jednotky (tj. s volbou MQGMO_SYNCPOINT) a zpráva má nepodporovaný formát, může být jednotka práce vrácena. Zpráva se pak znovu obnoví do fronty a lze ji načíst pomocí volání MQGET namísto volání objektu mqGetBag. Další informace o deskriptoru zpráv naleznete v tématu [MQGMO-Get-message options](#).

- Získejte volby zprávy pomocí struktury MQGMO. Další informace o struktuře MQGMO naleznete v příručce [MQMD-Message Descriptor](#).
- Popisovač balíku, který má obsahovat převedenou zprávu.

Úplný popis volání objektu mqGetBag naleznete v dokumentu [mqGetBag](#).

Odezvy

V odpovědi na každý příkaz generuje příkazový server jednu nebo více zpráv odpovědí. Zpráva s odezvou má podobný formát jako příkazová zpráva.

Hlavička PCF má stejnou hodnotu identifikátoru příkazu jako příkaz, na který se jedná o odezvu (podrobnosti viz [záhlaví MQCFH-PCF](#)). Identifikátor zprávy a korelační identifikátor se nastavují podle voleb sestavy požadavku.

Je-li typ záhlaví PCF zprávy příkazu MQCFT_COMMAND, vygenerují se pouze standardní odpovědi. Tyto příkazy jsou podporovány na všech platformách s výjimkou produktu z/OS. Starší aplikace nepodporují

PCF v systému z/OS; aplikace Průzkumník WebSphere MQ Okna je jedna z takových aplikací (Průzkumník IBM WebSphere MQ verze 6.0 nebo novější však podporuje PCF na systému z/OS).

Je-li typ záhlaví PCF zprávy příkazu MQCFT_COMMAND_XR, bude vygenerována buď rozšířená, nebo standardní odpověď. Tyto příkazy jsou podporovány v systému z/OS a na některých jiných platformách. Příkazy vydané v systému z/OS generují pouze rozšířené odpovědi. Na jiných platformách může být vygenerován typ odezvy.

Pokud jeden příkaz uvádí generický název objektu, vrátí se samostatná odezva ve vlastní zprávě pro každý odpovídající objekt. Pro generování odezvy se s generickým názvem zachází jako s více individuálními příkazy (s výjimkou pole řízení MQCFC_LAST nebo MQCFC_NOT_LAST). Jinak jedna zpráva příkazu vygeneruje jednu zprávu odezvy.

Určité odpovědi PCF mohou vrátit strukturu i v případě, že není vyžádáno. Tato struktura je zobrazena v definici odezvy ([Definice formátů Programovatelných příkazů](#)) jako *vždy vrácena*. Důvodem je, že pro tyto odezvy je nezbytné pojmenovat objekty v odpovědi, aby identifikovaly, na který objekt se data vztahují.

Deskriptor zpráv pro odezvu

Zpráva odpovědi obsahuje následující pole v deskriptoru zprávy:

MsgType

Toto pole je MQMT_REPLY.

MsgId

Toto pole je generováno správcem front.

CorrelId

Toto pole je generováno v souladu s volbami sestavy pro příkazovou zprávu.

Format

Toto pole je MQFMT_ADMIN.

Encoding

Nastavit na hodnotu MQENC_NATIVE.

CodedCharSetId

Nastavit na hodnotu MQCCSI_Q_MGR.

Persistence

Stejně jako ve zprávě příkazu.

Priority

Stejně jako ve zprávě příkazu.

Odezva se vygeneruje s MQPMO_PASS_IDENTITY_CONTEXT.

Standardní odezvy

Zprávy příkazu s typem záhlaví MQCFT_COMMAND, jsou generovány standardní odpovědi. Tyto příkazy jsou podporovány na všech platformách s výjimkou produktu z/OS.

Existují tři typy standardní odpovědi:

- Odezva OK
- Odezva na chybu
- Odezva dat

Odezva OK

Tato odezva se skládá ze zprávy, která začíná se záhlavím formátu příkazu, s polem *CompCode* MQCC_OK nebo MQCC_WARNING.

Pro MQCC_OK je hodnota *Reason* MQRC_NONE.

Pro funkci MQCC_WARNING identifikuje produkt *Reason* charakter varování. V tomto případě může za záhlavím příkazového formátu následovat jedna nebo více varovných struktur parametrů odpovídajících tomuto kódu příčiny.

V obou případech může pro příkaz dotazu následovat další struktury parametrů, jak je popsáno v následujících sekcích.

Odezva na chybu

Pokud má příkaz chybu, odešle se jedna nebo více chybových zpráv o chybě (více než jedna může být odeslána dokonce i pro příkaz, který by normálně měl pouze jednu zprávu odezvy). Tyto chybové zprávy odpovědi mají nastavitodpovídající nastavení MQCFC_LAST nebo MQCFC_NOT_LAST.

Každá taková zpráva začíná hlavičkou formátu odezvy, s hodnotou *CompCode* MQCC_FAILED a polem *Reason*, které identifikuje konkrétní chybu. Každá zpráva obecně popisuje jinou chybu. Kromě toho má každá zpráva buď nula, nebo jednu (nikdy více než jednu) strukturu chybových parametrů, které následují za záhlavím. Tato struktura parametru, pokud existuje jedna, je struktura MQCFIN, s polem *Parameter* obsahujícím jednu z následujících možností:

- MQIACF_PARAMETER_ID

Pole *Value* ve struktuře je identifikátor parametru pro parametr, který byl chybný (například MQCA_Q_NAME).

- MQIACF_ERROR_ID

Tato hodnota se používá s hodnotou *Reason* (v záhlaví příkazového formátu) MQRC_UNEXPECTED_ERROR. Pole *Value* ve struktuře MQCFIN představuje neočekávaný kód příčiny přijatý příkazovým serverem.

- MQIACF_SELEC

Tato hodnota se vyskytne, pokud struktura seznamu (MQCFIL) odeslaná s příkazem obsahuje duplicitní selektor nebo ten, který není platný. Pole *Reason* v záhlaví příkazového formátu identifikuje chybu a pole *Value* ve struktuře MQCFIN představuje hodnotu parametru ve struktuře MQCFIL příkazu, který byl v chybě.

- MQIACF_ERROR_OFFSET

Tato hodnota se vyskytne, když se vyskytne chyba porovnání dat v příkazu kanálu ping. Pole *Value* ve struktuře je posunutí chyby porovnání kanálu Ping kanálu.

- MQIA_CODE_CHAR_SET_ID

Tato hodnota se vyskytne, když identifikátor kódované znakové sady v deskriptoru zprávy přichodí zprávy příkazu PCF neodpovídá identifikátoru cílového správce front. Pole *Value* ve struktuře je identifikátor kódované znakové sady správce front.

Poslední (nebo jediná) chybová zpráva odezvy je souhrnná odpověď, s polem *CompCode* MQCC_FAILED a polem *Reason* MQRCCF_COMMAND_FAILED. Tato zpráva nemá žádnou strukturu parametrů za záhlavím.

Odezva dat

Tato odezva se skládá z odpovědi OK (jak je popsáno výše) na příkaz dotazu. Po odpovědi na OK následují další struktury obsahující požadovaná data, jak je popsáno v tématu [Definice formátů Programovatelných příkazů](#).

Aplikace nesmí záviset na tom, že jsou tyto přídavné struktury parametrů vráceny v libovolném pořadí.

Kontrola oprávnění pro příkazy PCF

Když je zpracován příkaz PCF, *UserIdentifier* z deskriptoru zpráv ve zprávě příkazu se použije pro požadované kontroly oprávnění k objektu WebSphere MQ. Kontrola oprávnění se provádí na každé platformě jinak, jak je popsáno v tomto tématu.

Kontroly se provádějí na systému, na kterém je příkaz zpracováván; proto musí toto ID uživatele existovat na cílovém systému a musí mít požadovaná oprávnění ke zpracování příkazu. Pokud zpráva přišla ze vzdáleného systému, jedním ze způsobů, jak dosáhnout ID existujícího na cílovém systému, je mít odpovídající ID uživatele jak na lokálním tak i na vzdáleném systému.

IBM WebSphere MQ pro systémy Windows, systémy UNIX and Linux



Aby bylo možné zpracovat libovolný příkaz PCF, musí mít ID uživatele oprávnění *dsp* pro objekt správce front v cílovém systému. Kromě toho jsou pro určité příkazy PCF prováděny kontroly oprávnění k objektu WebSphere MQ, jak ukazuje [Tabulka 1](#) na stránce [16](#).

Chcete-li zpracovat kterýkoli z následujících příkazů, musí ID uživatele patřit do skupiny *mqm*.

Poznámka: Pro systém Windows **pouze** může ID uživatele patřit do skupiny *Administrátoři* nebo skupiny *mqm*.

- Změnit kanál
- Kopírovat kanál
- Vytvořit kanál
- Odstranit kanál
- Odeslat signál Ping pro kanál
- Resetovat kanál
- Spustit kanál
- Ukončit kanál
- Spustit inicializátor kanálu
- Spustit modul listener kanálu
- Vyřešit kanál
- Reset klastru
- Aktualizovat klastr
- Pozastavit správce front
- Obnovit správce front

WebSphere MQ for HP Integrity NonStop Server

Aby bylo možné zpracovat libovolný příkaz PCF, musí mít ID uživatele oprávnění *dsp* pro objekt správce front v cílovém systému. Kromě toho jsou provedeny kontroly oprávnění objektu IBM WebSphere MQ pro určité příkazy PCF, jak je zobrazeno v [Tabulka 1](#) na stránce [16](#).

Chcete-li zpracovat kterýkoli z následujících příkazů, musí ID uživatele patřit do skupiny *mqm*:

- Změnit kanál
- Kopírovat kanál
- Vytvořit kanál
- Odstranit kanál
- Odeslat signál Ping pro kanál
- Resetovat kanál
- Spustit kanál
- Ukončit kanál
- Spustit inicializátor kanálu
- Spustit modul listener kanálu
- Vyřešit kanál

- Reset klastru
- Aktualizovat klastr
- Pozastavit správce front
- Obnovit správce front

Oprávnění k objektu WebSphere MQ

<i>Tabulka 1. Windows, HP Integrity NonStop Server, UNIX and Linux systems-object authorities</i>		
Příkaz	Oprávnění k objektu WebSphere MQ	Oprávnění třídy (pro typ objektu)
Změnit ověřovací informace	dsp a chg	Není k dispozici.
Změnit kanál	dsp a chg	Není k dispozici.
Změnit modul listener kanálu	dsp a chg	Není k dispozici.
Změnit kanál připojení klienta	dsp a chg	Není k dispozici.
Změnit seznam názvů	dsp a chg	Není k dispozici.
Změnit proces	dsp a chg	Není k dispozici.
Změnit frontu	dsp a chg	Není k dispozici.
Změnit správce front	chg viz <i>poznámka 3 a poznámka 5</i>	Není k dispozici.
Změnit službu	dsp a chg	Není k dispozici.
Vymazat frontu	clr	Není k dispozici.
Kopírovat ověřovací informace	dsp	crt
Kopírovat ověřovací informace (Nahradit) viz <i>poznámka 1</i>	od: dsp do: chg	crt
Kopírovat kanál	dsp	crt
Kopírovat kanál (Nahradit) viz <i>poznámka 1</i>	od: dsp do: chg	crt
Kopírovat modul listener kanálu	dsp	crt
Kopírovat modul listener kanálu (Nahradit) viz <i>poznámka 1</i>	od: dsp do: chg	crt
Kopírovat kanál připojení klienta	dsp	crt
Kopírovat kanál připojení klienta (Nahradit) viz <i>poznámka 1</i>	od: dsp do: chg	crt
Kopírovat seznam názvů	dsp	crt
Zkopírovat seznam názvů (nahradit) viz <i>poznámka 1</i>	od: dsp to: dsp a chg	crt
Kopírovat proces	dsp	crt
Kopírovat proces (nahradit) viz <i>poznámka 1</i>	od: dsp do: chg	crt
Kopírovat frontu	dsp	crt
Kopírovat frontu (nahradit) viz <i>poznámka 1</i>	od: dsp to: dsp a chg	crt

Tabulka 1. Windows, HP Integrity NonStop Server, UNIX and Linux systems-object authorities
(pokračování)

Příkaz	Oprávnění k objektu WebSphere MQ	Oprávnění třídy (pro typ objektu)
Vytvořit ověřovací informace	(systémové výchozí ověřovací informace) dsp	crt
Vytvořit ověřovací informace (Nahradit) viz poznámka 1	(systémové výchozí ověřovací informace) dsp na: chg	crt
Vytvořit kanál	(systémový výchozí kanál) dsp	crt
Vytvořit kanál (Nahradit) viz poznámka 1	(systémový výchozí kanál) dsp to: chg	crt
Vytvořit modul listener kanálu	(systémový výchozí modul listener) dsp	crt
Vytvořit modul listener kanálu (Nahradit) viz poznámka 1	(systémový výchozí modul listener) dsp to: chg	crt
Vytvořit kanál připojení klienta	(systémový výchozí kanál) dsp	crt
Vytvořit kanál připojení klienta (Nahradit) viz poznámka 1	(systémový výchozí kanál) dsp to: chg	crt
Vytvořit seznam názvů	(systémový výchozí seznam názvů) dsp	crt
Vytvořit seznam názvů (Nahradit) viz poznámka 1	(systémový výchozí seznam názvů) dsp to: dsp a chg	crt
Vytvořit proces	(systémový výchozí proces) dsp	crt
Vytvořit proces (Nahradit) viz poznámka 1	(systémový výchozí proces) dsp na: chg	crt
Vytvořit frontu	(systémová výchozí fronta) dsp	crt
Vytvořit frontu (nahradit) viz poznámka 1	(systémová výchozí fronta) dsp na: dsp a chg	crt
Vytvořit službu	(systémová výchozí fronta) dsp	crt
Vytvořit službu (Nahradit) viz poznámka 1	(systémová výchozí fronta) dsp na: chg	crt
Odstranit ověřovací informace	dsp a dlt	Není k dispozici.
Odstranit záznam oprávnění	(objekt správce front) chg viz poznámka 4	viz poznámka 4
Odstranit kanál	dsp a dlt	Není k dispozici.
Odstranit modul listener kanálu	dsp a dlt	Není k dispozici.
Odstranit kanál připojení klienta	dsp a dlt	Není k dispozici.
Odstranit seznam názvů	dsp a dlt	Není k dispozici.
Odstranit proces	dsp a dlt	Není k dispozici.
Odstranit frontu	dsp a dlt	Není k dispozici.
Odstranit službu	dsp a dlt	Není k dispozici.
Zjistit ověřovací informace	dsp	Není k dispozici.
Zjistit záznamy oprávnění	viz poznámka 4	viz poznámka 4

Tabulka 1. Windows, HP Integrity NonStop Server, UNIX and Linux systems-object authorities
(pokračování)

Příkaz	Oprávnění k objektu WebSphere MQ	Oprávnění třídy (pro typ objektu)
Zjistit kanál	dsp	Není k dispozici.
Zjistit modul listener kanálu	dsp	Není k dispozici.
Zjišťovat stav kanálu (pro ChannelType MQCHT_CLSSDR)	inq	Není k dispozici.
Zjistit kanál připojení klienta	dsp	Není k dispozici.
Zjistit seznam názvů	dsp	Není k dispozici.
Zjistit proces	dsp	Není k dispozici.
Zjistit frontu	dsp	Není k dispozici.
Zjistit správce front	<i>viz poznámka 3</i>	Není k dispozici.
Zjistit stav fronty	dsp	Není k dispozici.
Zjistit službu	dsp	Není k dispozici.
Odeslat signál Ping pro kanál	ctrl	Není k dispozici.
Odeslat signál Ping pro správce front	<i>viz poznámka 3</i>	Není k dispozici.
Aktualizovat správce front	(objekt správce front) chg	Není k dispozici.
Aktualizovat zabezpečení (pro SecurityType MQSECTYPE_SSL)	(objekt správce front) chg	Není k dispozici.
Resetovat kanál	ctrlx	Není k dispozici.
Obnovit správce front	(objekt správce front) chg	Není k dispozici.
Obnovit statistiku front	dsp a chg	Není k dispozici.
Vyřešit kanál	ctrlx	Není k dispozici.
Nastavit záznam oprávnění	<i>(objekt správce front) chg viz poznámka 4</i>	<i>viz poznámka 4</i>
Spustit kanál	ctrl	Není k dispozici.
Ukončit kanál	ctrl	Není k dispozici.
Zastavit připojení	(objekt správce front) chg	Není k dispozici.
Spustit listener	ctrl	Není k dispozici.
Ukončit listener	ctrl	Není k dispozici.
Spustit službu	ctrl	Není k dispozici.
Zastavit službu	ctrl	Není k dispozici.
Esc	<i>viz poznámka 2</i>	<i>viz poznámka 2</i>

Notes:

1. Tento příkaz se použije, pokud objekt, který má být nahrazen, existuje, jinak je kontrola oprávnění jako pro Vytvořit, nebo Kopírovat bez náhrady.

2. Požadované oprávnění je určeno pomocí příkazu MQSC definovaného řídicím textem a je ekvivalentem jednoho z předchozích příkazů.
3. Aby bylo možné zpracovat libovolný příkaz PCF, musí mít ID uživatele oprávnění dsp pro objekt správce front v cílovém systému.
4. Tento příkaz PCF je autorizován, pokud nebyl spuštěn příkazový server s parametrem -a. Při výchozím nastavení je příkazový server spuštěn při spuštění správce front a bez parametru -a. Další informace najdete v příručce System Administration Guide.
5. Udělení oprávnění k ID uživatele *chg* pro správce front poskytuje schopnost nastavit záznamy oprávnění pro všechny skupiny a uživatele. Neudělíte tomuto oprávnění běžným uživatelům nebo aplikacím.

Produkt WebSphere MQ také poskytuje některé výstupní body zabezpečení kanálu, takže můžete dodat vlastní uživatelské programy pro kontrolu zabezpečení. Podrobnosti jsou uvedeny v příručce [Zobrazení kanálu](#).

Použití rozhraní MQAI ke zjednodušení používání PCFs

Rozhraní MQAI je administrativní rozhraní produktu WebSphere MQ, které je k dispozici na platformách AIX, HP-UX, IBM i, Linux, Solaris a Windows.

Rozhraní MQAI provádí administrativní úlohy ve správci front prostřednictvím použití *datových balíků*. Datové váčky umožňují manipulaci s vlastnostmi (nebo parametry) objektů způsobem, který je jednodušší než použití PCF.

Rozhraní MQAI lze používat následujícími způsoby:

Zjednodušení použití zpráv PCF

MQAI je jednoduchý způsob, jak spravovat produkt WebSphere MQ; nemusíte psát své vlastní zprávy PCF, čímž se vyhnete problémům spojeným se složitými datovými strukturami.

Chcete-li předávat parametry v programech zapsaných pomocí volání MQI, musí zpráva PCF obsahovat příkaz a podrobnosti o řetězci nebo celočíselných datech. K tomu potřebujete několik příkazů ve vašem programu pro každou strukturu a paměťový prostor musí být alokovan. Tato úloha může být dlouhá a pracná.

Programy napsané pomocí parametrů MQAI pass do příslušného datového balíku a vy potřebujete pro každou strukturu pouze jeden příkaz. Použití datových pytlů MQAI odstraňuje potřebu použití polí a přidělování paměti a poskytuje určitou míru izolace od podrobností PCF.

Snadnější zacházení s chybovými podmínkami

Je obtížné získat návratové kódy zpět z příkazů PCF, ale rozhraní MQAI usnadňuje práci programu pro zpracování chybových stavů.

Po vytvoření a naplnění datového balíku dat můžete odeslat zprávu příkazu administrace na příkazový server správce front pomocí volání mqExecute, které čeká na zprávy odpovědi. Volání mqExecute zpracovává výměnu s příkazovým serverem a vrací odpovědi v *vaku odezvy*.

Další informace o rozhraní MQAI najdete v tématu [“Úvod do administrativního rozhraní produktu IBM WebSphere MQ \(MQAI\)”](#) na stránce 19.

Úvod do administrativního rozhraní produktu IBM WebSphere MQ (MQAI)

Rozhraní IBM WebSphere MQ Administration Interface (MQAI) je programovací rozhraní produktu IBM WebSphere MQ. Provádí administrativní úlohy ve správci front produktu IBM WebSphere MQ pomocí datových balíků pro zpracování vlastností (nebo parametrů) objektů způsobem, který je snazší než použití Programovatelných formátů příkazů (PCFs).

Koncepty a terminologie MQAI

Rozhraní MQAI je programovací rozhraní pro produkt WebSphere MQs použitím jazyka C a také jazyka Visual Basic for Windows. Je k dispozici na jiných platformách než z/OS.

Provádí administrativní úlohy ve správci front produktu WebSphere MQ pomocí datových balíčků. Datové pytle umožňují manipulaci s vlastnostmi (nebo parametry) objektů způsobem, který je jednodušší než použití jiného administrativního rozhraní, Programmable Command Format (PCFs). Rozhraní MQAI nabízí snazší manipulaci s PCFs než pomocí volání MQGET a MQPUT.

Další informace o datových sáčcích naleznete v tématu [“Datové sáčky”](#) na stránce 46. Další informace o PCFs viz [“Úvod do formátu programových příkazů”](#) na stránce 9

Použití přípravku MQAI

Rozhraní MQAI můžete použít k těmto účelům:

- Zjednodušte si použití zpráv PCF. MQAI je jednoduchý způsob, jak spravovat produkt WebSphere MQ; nemusíte psát své vlastní PCF zprávy a vyhnout se tak problémům spojeným se složitými datovými strukturami.
- Snadněji zpracovat chybové podmínky. Je velmi obtížné získat návratové kódy zpět z příkazů skriptu WebSphere MQ (MQSC), ale rozhraní MQAI zjednodušuje program tak, aby zvládl chybový stav.
- Výměna dat mezi aplikacemi. Data aplikace jsou odeslána ve formátu PCF a jsou komprimovaná a rozbalena rozhraním MQAI. Pokud se data zprávy skládají z celých čísel a znakových řetězců, můžete použít rozhraní MQAI k využití výhod vestavěného převodu dat produktu WebSphere MQ pro data PCF. Tím se vyvarujete nutnosti psát uživatelské procedury pro zápis dat. Další informace o použití rozhraní MQAI ke správě produktu WebSphere MQ a k výměně dat mezi aplikacemi naleznete v tématu [“Použití rozhraní MQAI ke zjednodušení používání PCFs”](#) na stránce 19.

Příklady použití rozhraní MQAI

Zobrazený seznam uvádí některé vzorové programy, které demonstrují použití rozhraní MQAI. Ukázky provádí následující úlohy:

1. Vytvořte lokální frontu. [“Ukázkový program v jazyce C pro vytvoření lokální fronty \(amqsaicq.c\)”](#) na stránce 21
2. Zobrazit události na obrazovce pomocí jednoduchého monitoru událostí. [“Ukázkový program v jazyce C pro zobrazení událostí pomocí monitoru událostí \(amqsaiem.c\)”](#) na stránce 25
3. Vytiskněte seznam všech lokálních front a jejich aktuálních hloubek. [“Ukázkový program v jazyce C pro dotazování front a tisk informací \(amqsailq.c\)”](#) na stránce 37
4. Vytisknout seznam všech kanálů a jejich typů. [“Ukázkový program jazyka C pro zjišťování informací o objektech kanálu \(amqsaicl.c\)”](#) na stránce 32

Sestavení aplikace MQAI

Chcete-li sestavit aplikaci pomocí rozhraní MQAI, budete odkazovat na stejné knihovny, jako je tomu u produktu WebSphere MQ. Informace o tom, jak sestavit aplikace produktu WebSphere MQ naleznete v tématu [Sestavení aplikace WebSphere MQ](#).

Rady a tipy pro konfiguraci produktu WebSphere MQ pomocí MQAI

Volání MQAI používá PCF zprávy k odeslání příkazů administrace na příkazový server místo přímého zpracování přímo s příkazovým serverem. Tipy pro konfiguraci produktu WebSphere MQ pomocí rozhraní MQAI lze najít v produktu [“Rady a tipy pro konfiguraci produktu IBM WebSphere MQ”](#) na stránce 41

Rozhraní IBM WebSphere MQ Administration Interface (MQAI)

IBM WebSphere MQ pro systémy Windows, AIX, Linux, HP-UX a Solaris podporují rozhraní IBM WebSphere MQ Administration Interface (MQAI). Rozhraní MQAI je programovací rozhraní produktu IBM WebSphere MQ, které poskytuje alternativu k rozhraní MQI pro odesílání a příjem PCF.

MQAI používá *datové vaky*, které vám umožňují zacházet s vlastnostmi (nebo parametry) objektů snadněji než pomocí PCF přímo prostřednictvím rozhraní MQAI.

Rozhraní MQAI poskytuje snazší programovací přístup ke zprávám PCF předáním parametrů do datového balíku, takže pro každou strukturu je požadován pouze jeden příkaz. Tento přístup odstraňuje potřebu programátora zpracovávat pole a přidělovat prostor pro ukládání dat a poskytuje určitou izolaci od podrobností PCF.

Funkce MQAI spravuje produkt WebSphere MQ odesláním zpráv PCF na příkazový server a čekáním na odpověď.

Rozhraní MQAI je popsáno v druhé části této příručky. Popis rozhraní modelu objektu komponenty na rozhraní MQAI najdete v dokumentaci produktu [Použití jazyka Java](#).

Ukázkový program v jazyce C pro vytvoření lokální fronty (amqsaicq.c)

Ukázkový program v jazyce C `amqsaicq.c` vytvoří lokální frontu pomocí MQAI.

```

/*****
/*
/* Program name: AMQSAICQ.C
/*
/* Description: Sample C program to create a local queue using the
/*               WebSphere MQ Administration Interface (MQAI).
/*
/* Statement: Licensed Materials - Property of IBM
/*
/*             84H2000, 5765-B73
/*             84H2001, 5639-B42
/*             84H2002, 5765-B74
/*             84H2003, 5765-B75
/*             84H2004, 5639-B43
/*
/*             (C) Copyright IBM Corp. 1999, 2024.
/*
*****/
/*
/* Function:
/* AMQSAICQ is a sample C program that creates a local queue and is an
/* example of the use of the mqExecute call.
/*
/* - The name of the queue to be created is a parameter to the program.
/*
/* - A PCF command is built by placing items into an MQAI bag.
/*   These are:-
/*     - The name of the queue
/*     - The type of queue required, which, in this case, is local.
/*
/* - The mqExecute call is executed with the command MQCMD_CREATE_Q.
/*   The call generates the correct PCF structure.
/*   The call receives the reply from the command server and formats into
/*   the response bag.
/*
/* - The completion code from the mqExecute call is checked and if there
/*   is a failure from the command server then the code returned by the
/*   command server is retrieved from the system bag that is
/*   embedded in the response bag to the mqExecute call.
/*
/* Note: The command server must be running.
/*
*****/
/*
/* AMQSAICQ has 2 parameters - the name of the local queue to be created
/*                            - the queue manager name (optional)
*****/

```

```

/*****
/* Includes
/*****
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>

#include <cmqc.h> /* MQI */
#include <cmqcfh.h> /* PCF */
#include <cmqbc.h> /* MQAI */

void CheckCallResult(MQCHAR *, MQLONG , MQLONG );
void CreateLocalQueue(MQHCONN, MQCHAR *);

int main(int argc, char *argv[])
{
    MQHCONN hConn; /* handle to WebSphere MQ connection */
    MQCHAR QMName[MQ_Q_MGR_NAME_LENGTH+1]=""; /* default QMgr name */
    MQLONG connReason; /* MQCONN reason code */
    MQLONG compCode; /* completion code */
    MQLONG reason; /* reason code */

    /*****
    /* First check the required parameters
    /*****
    printf("Sample Program to Create a Local Queue\n");
    if (argc < 2)
    {
        printf("Required parameter missing - local queue name\n");
        exit(99);
    }

    /*****
    /* Connect to the queue manager
    /*****
    if (argc > 2)
        strncpy(QMName, argv[2], (size_t)MQ_Q_MGR_NAME_LENGTH);
        MQCONN(QMName, &hConn, &compCode, &connReason);

    /*****
    /* Report reason and stop if connection failed
    /*****
    if (compCode == MQCC_FAILED)
    {
        CheckCallResult("MQCONN", compCode, connReason);
        exit( (int)connReason);
    }

    /*****
    /* Call the routine to create a local queue, passing the handle to the
    /* queue manager and also passing the name of the queue to be created.
    /*****
    CreateLocalQueue(hConn, argv[1]);

    /*****
    /* Disconnect from the queue manager if not already connected
    /*****
    if (connReason != MQRC_ALREADY_CONNECTED)
    {
        MQDISC(&hConn, &compCode, &reason);
        CheckCallResult("MQDISC", compCode, reason);
    }
    return 0;
}

/*****
/*
/* Function: CreateLocalQueue
/* Description: Create a local queue by sending a PCF command to the command
/* server.
/*
/*****
/*
/* Input Parameters: Handle to the queue manager
/* Name of the queue to be created
/*
/*
/* Output Parameters: None
/*
/*
/* Logic: The mqExecute call is executed with the command MQCMD_CREATE_Q.
/* The call generates the correct PCF structure.
/*****

```

```

/*      The default options to the call are used so that the command is sent*/
/*      to the SYSTEM.ADMIN.COMMAND.QUEUE.                               */
/*      The reply from the command server is placed on a temporary dynamic */
/*      queue.                                                           */
/*      The reply is read from the temporary queue and formatted into the */
/*      response bag.                                                    */
/*      */
/*      The completion code from the mqExecute call is checked and if there */
/*      is a failure from the command server then the code returned by the */
/*      command server is retrieved from the system bag that is          */
/*      embedded in the response bag to the mqExecute call.             */
/*      */
/*      */
/*****
void CreateLocalQueue(MQHCONN hConn, MQCHAR *qName)
{
    MQLONG reason;                /* reason code                */
    MQLONG compCode;              /* completion code           */
    MQHBAG commandBag = MQHB_UNUSABLE_HBAG; /* command bag for mqExecute */
    MQHBAG responseBag = MQHB_UNUSABLE_HBAG; /* response bag for mqExecute */
    MQHBAG resultBag;            /* result bag from mqExecute */
    MQLONG mqExecuteCC;          /* mqExecute completion code */
    MQLONG mqExecuteRC;          /* mqExecute reason code     */

    printf("\nCreating Local Queue %s\n", qName);

    /*****
    /* Create a command Bag for the mqExecute call. Exit the function if the */
    /* create fails.                                                         */
    /*****
    mqCreateBag(MQCBO_ADMIN_BAG, &commandBag, &compCode, &reason);
    CheckCallResult("Create the command bag", compCode, reason);
    if (compCode !=MQCC_OK)
        return;

    /*****
    /* Create a response Bag for the mqExecute call, exit the function if the */
    /* create fails.                                                         */
    /*****
    mqCreateBag(MQCBO_ADMIN_BAG, &responseBag, &compCode, &reason);
    CheckCallResult("Create the response bag", compCode, reason);
    if (compCode !=MQCC_OK)
        return;

    /*****
    /* Put the name of the queue to be created into the command bag. This will */
    /* be used by the mqExecute call.                                         */
    /*****
    mqAddString(commandBag, MQCA_Q_NAME, MQBL_NULL_TERMINATED, qName, &compCode,
                &reason);
    CheckCallResult("Add q name to command bag", compCode, reason);

    /*****
    /* Put queue type of local into the command bag. This will be used by the */
    /* mqExecute call.                                                         */
    /*****
    mqAddInteger(commandBag, MQIA_Q_TYPE, MQQT_LOCAL, &compCode, &reason);
    CheckCallResult("Add q type to command bag", compCode, reason);

    /*****
    /* Send the command to create the required local queue.                   */
    /* The mqExecute call will create the PCF structure required, send it to  */
    /* the command server and receive the reply from the command server into  */
    /* the response bag.                                                       */
    /*****
    mqExecute(hConn,                /* WebSphere MQ connection handle */
              MQCMD_CREATE_Q,        /* Command to be executed          */
              MQHB_NONE,            /* No options bag                  */
              commandBag,           /* Handle to bag containing commands */
              responseBag,          /* Handle to bag to receive the response*/
              MQHO_NONE,           /* Put msg on SYSTEM.ADMIN.COMMAND.QUEUE*/
              MQHO_NONE,           /* Create a dynamic q for the response */
              &compCode,           /* Completion code from the mqExecute */
              &reason);            /* Reason code from mqExecute call */

    if (reason == MQRC_CMD_SERVER_NOT_AVAILABLE)
    {
        printf("Please start the command server: <strmqcsv QMgrName>\n");
        MQDISC(&hConn, &compCode, &reason);
        CheckCallResult("MQDISC", compCode, reason);
        exit(98);
    }
}

```

```

/*****
/* Check the result from mqExecute call and find the error if it failed. */
/*****
if ( compCode == MQCC_OK )
    printf("Local queue %s successfully created\n", qName);
else
{
    printf("Creation of local queue %s failed: Completion Code = %d
           qName, compCode, reason);
    if (reason == MQRCCF_COMMAND_FAILED)
    {
        /*****
        /* Get the system bag handle out of the mqExecute response bag. */
        /* This bag contains the reason from the command server why the */
        /* command failed. */
        /*****
        mqInquireBag(responseBag, MQHA_BAG_HANDLE, 0, &resultBag, &compCode,
                    &reason);
        CheckCallResult("Get the result bag handle", compCode, reason);

        /*****
        /* Get the completion code and reason code, returned by the command */
        /* server, from the embedded error bag. */
        /*****
        mqInquireInteger(resultBag, MQIASY_COMP_CODE, MQIND_NONE, &mqExecuteCC,
                        &compCode, &reason);
        CheckCallResult("Get the completion code from the result bag",
                        compCode, reason);
        mqInquireInteger(resultBag, MQIASY_REASON, MQIND_NONE, &mqExecuteRC,
                        &compCode, &reason);
        CheckCallResult("Get the reason code from the result bag", compCode,
                        reason);
        printf("Error returned by the command server: Completion code = %d :
              Reason = %d\n", mqExecuteCC, mqExecuteRC);
    }
}
/*****
/* Delete the command bag if successfully created. */
/*****
if (commandBag != MQHB_UNUSABLE_HBAG)
{
    mqDeleteBag(&commandBag, &compCode, &reason);
    CheckCallResult("Delete the command bag", compCode, reason);
}

/*****
/* Delete the response bag if successfully created. */
/*****
if (responseBag != MQHB_UNUSABLE_HBAG)
{
    mqDeleteBag(&responseBag, &compCode, &reason);
    CheckCallResult("Delete the response bag", compCode, reason);
}
} /* end of CreateLocalQueue */

/*****
/*
/* Function: CheckCallResult
/*
/*****
/*
/* Input Parameters: Description of call
/*                    Completion code
/*                    Reason code
/*
/* Output Parameters: None
/*
/* Logic: Display the description of the call, the completion code and the
/*        reason code if the completion code is not successful
/*
/*****
void CheckCallResult(char *callText, MQLONG cc, MQLONG rc)
{
    if (cc != MQCC_OK)
        printf("%s failed: Completion Code = %d :
              Reason = %d\n", callText, cc, rc);
}

```


Ukázkový program v jazyce C pro zobrazení událostí pomocí monitoru událostí (amqsaiem.c)

Ukázkový program v jazyce C amqsaiem.c demonstruje základní monitor událostí pomocí MQAI.

```
*****/
/*
/* Program name: AMQSAIEM.C
/*
/* Description: Sample C program to demonstrate a basic event monitor
/* using the WebSphere MQ Admin Interface (MQAI).
/* Licensed Materials - Property of IBM
/*
/* 63H9336
/* (c) Copyright IBM Corp. 1999, 2024. All Rights Reserved.
/*
/* US Government Users Restricted Rights - Use, duplication or
/* disclosure restricted by GSA ADP Schedule Contract with
/* IBM Corp.
/*
*****/
/*
/* Function:
/* AMQSAIEM is a sample C program that demonstrates how to write a simple
/* event monitor using the mqGetBag call and other MQAI calls.
/*
/* The name of the event queue to be monitored is passed as a parameter
/* to the program. This would usually be one of the system event queues:-
/* SYSTEM.ADMIN.QMGR.EVENT Queue Manager events
/* SYSTEM.ADMIN.PERFM.EVENT Performance events
/* SYSTEM.ADMIN.CHANNEL.EVENT Channel events
/* SYSTEM.ADMIN.LOGGER.EVENT Logger events
/*
/* To monitor the queue manager event queue or the performance event queue,
/* the attributes of the queue manager needs to be changed to enable
/* these events. For more information about this, see Part 1 of the
/* Programmable System Management book. The queue manager attributes can
/* be changed using either MQSC commands or the MQAI interface.
/* Channel events are enabled by default.
/*
/* Program logic
/* Connect to the Queue Manager.
/* Open the requested event queue with a wait interval of 30 seconds.
/* Wait for a message, and when it arrives get the message from the queue
/* and format it into an MQAI bag using the mqGetBag call.
/* There are many types of event messages and it is beyond the scope of
/* this sample to program for all event messages. Instead the program
/* prints out the contents of the formatted bag.
/* Loop around to wait for another message until either there is an error
/* or the wait interval of 30 seconds is reached.
/*
*****/
/*
/* AMQSAIEM has 2 parameters - the name of the event queue to be monitored
/* - the queue manager name (optional)
/*
*****

/*****/
/* Includes
/*****/
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>

#include <cmqc.h> /* MQI
#include <cmqcfc.h> /* PCF
#include <cmqbc.h> /* MQAI

/*****/
/* Macros
/*****/
#if MQAT_DEFAULT == MQAT_WINDOWS_NT
#define Int64 "I64"
#elif defined(MQ_64_BIT)
#define Int64 "l"
#else
#define Int64 "ll"
#endif
```

```

#endif

/*****
/* Function prototypes */
/*****
void CheckCallResult(MQCHAR *, MQLONG , MQLONG);
void GetQEvents(MQHCONN, MQCHAR *);
int PrintBag(MQHBAG);
int PrintBagContents(MQHBAG, int);

/*****
/* Function: main */
/*****
int main(int argc, char *argv[])
{
    MQHCONN hConn; /* handle to connection */
    MQCHAR QMName[MQ_Q_MGR_NAME_LENGTH+1]=""; /* default QM name */
    MQLONG reason; /* reason code */
    MQLONG connReason; /* MQCONN reason code */
    MQLONG compCode; /* completion code */

    /*****
    /* First check the required parameters */
    /*****
    printf("Sample Event Monitor (times out after 30 secs)\n");
    if (argc < 2)
    {
        printf("Required parameter missing - event queue to be monitored\n");
        exit(99);
    }

    /*****
    /* Connect to the queue manager */
    /*****
    if (argc > 2)
        strncpy(QMName, argv[2], (size_t)MQ_Q_MGR_NAME_LENGTH);
    MQCONN(QMName, &hConn, &compCode, &connReason);
    /*****
    /* Report the reason and stop if the connection failed */
    /*****
    if (compCode == MQCC_FAILED)
    {
        CheckCallResult("MQCONN", compCode, connReason);
        exit( (int)connReason);
    }

    /*****
    /* Call the routine to open the event queue and format any event messages */
    /* read from the queue. */
    /*****
    GetQEvents(hConn, argv[1]);

    /*****
    /* Disconnect from the queue manager if not already connected */
    /*****
    if (connReason != MQRC_ALREADY_CONNECTED)
    {
        MQDISC(&hConn, &compCode, &reason);
        CheckCallResult("MQDISC", compCode, reason);
    }

    return 0;
}

/*****
/*
/* Function: CheckCallResult */
/*
/*
/*****
/*
/* Input Parameters: Description of call */
/* Completion code */
/* Reason code */
/*
/* Output Parameters: None */
/*
/* Logic: Display the description of the call, the completion code and the */
/* reason code if the completion code is not successful */
/*
/*****
void CheckCallResult(char *callText, MQLONG cc, MQLONG rc)

```

```

{
    if (cc != MQCC_OK)
        printf("%s failed: Completion Code = %d : Reason = %d\n",
            callText, cc, rc);
}

/*****
*/
/* Function: GetQEvents
*/
/*****
*/
/* Input Parameters: Handle to the queue manager
*/
/* Name of the event queue to be monitored
*/
/* Output Parameters: None
*/
/* Logic: Open the event queue.
*/
/* Get a message off the event queue and format the message into
*/
/* a bag.
*/
/* A real event monitor would need to be programmed to deal with
*/
/* each type of event that it receives from the queue. This is
*/
/* outside the scope of this sample, so instead, the contents of
*/
/* the bag are printed.
*/
/* The program waits for 30 seconds for an event message and then
*/
/* terminates if no more messages are available.
*/
/*****
void GetQEvents(MQHCONN hConn, MQCHAR *qName)
{
    MQLONG openReason; /* MQOPEN reason code */
    MQLONG reason; /* reason code */
    MQLONG compCode; /* completion code */
    MQHOBJ eventQueue; /* handle to event queue */

    MQHBAG eventBag = MQHB_UNUSABLE_HBAG; /* event bag to receive event msg */
    MQOD od = {MQOD_DEFAULT}; /* Object Descriptor */
    MQMD md = {MQMD_DEFAULT}; /* Message Descriptor */
    MQGMO gmo = {MQGMO_DEFAULT}; /* get message options */
    MQLONG bQueueOK = 1; /* keep reading msgs while true */

    /*****
    /* Create an Event Bag in which to receive the event.
    /* Exit the function if the create fails.
    /*****
    mqCreateBag(MQCBO_USER_BAG, &eventBag, &compCode, &reason);
    CheckCallResult("Create event bag", compCode, reason);
    if (compCode != MQCC_OK)
        return;

    /*****
    /* Open the event queue chosen by the user
    /*****
    strncpy(od.ObjectName, qName, (size_t)MQ_Q_NAME_LENGTH);
    MQOPEN(hConn, &od, MQOO_INPUT_AS_Q_DEF+MQOO_FAIL_IF QUIESCING, &eventQueue,
        &compCode, &openReason);
    CheckCallResult("Open event queue", compCode, openReason);

    /*****
    /* Set the GMO options to control the action of the get message from the
    /* queue.
    /*****
    gmo.WaitInterval = 30000; /* 30 second wait for message
    gmo.Options = MQGMO_WAIT + MQGMO_FAIL_IF QUIESCING + MQGMO_CONVERT;
    gmo.Version = MQGMO_VERSION_2; /* Avoid need to reset Message ID
    gmo.MatchOptions = MQMO_NONE; /* and Correlation ID after every
    /* mqGetBag

    /*****
    /* If open fails, we cannot access the queue and must stop the monitor.
    /*****
    if (compCode != MQCC_OK)
        bQueueOK = 0;

    /*****
    /* Main loop to get an event message when it arrives
    /*****
    while (bQueueOK)
    {
        printf("\nWaiting for an event\n");

    /*****

```

```

/* Get the message from the event queue and convert it into the event */
/* bag. */
/*****
mqGetBag(hConn, eventQueue, &md, &gmo, eventBag, &compCode, &reason);

/*****
/* If get fails, we cannot access the queue and must stop the monitor. */
/*****
if (compCode != MQCC_OK)
{
    bQueueOK = 0;

    /*****
    /* If get fails because no message available then we have timed out, */
    /* so report this, otherwise report an error. */
    /*****
    if (reason == MQRC_NO_MSG_AVAILABLE)
    {
        printf("No more messages\n");
    }
    else
    {
        CheckCallResult("Get bag", compCode, reason);
    }
}

/*****
/* Event message read - Print the contents of the event bag */
/*****
else
{
    if ( PrintBag(eventBag) )
        printf("\nError found while printing bag contents\n");

} /* end of msg found */
} /* end of main loop */
/*****
/* Close the event queue if successfully opened */
/*****
if (openReason == MQRC_NONE)
{
    MQCLOSE(hConn, &eventQueue, MQCO_NONE, &compCode, &reason);
    CheckCallResult("Close event queue", compCode, reason);
}

/*****
/* Delete the event bag if successfully created. */
/*****
if (eventBag != MQHB_UNUSABLE_HBAG)
{
    mqDeleteBag(&eventBag, &compCode, &reason);
    CheckCallResult("Delete the event bag", compCode, reason);
}

} /* end of GetQEvents */

/*****
/*
/* Function: PrintBag */
/*
/*****
/*
/* Input Parameters: Bag Handle */
/*
/* Output Parameters: None */
/*
/* Returns: Number of errors found */
/*
/* Logic: Calls PrintBagContents to display the contents of the bag. */
/*****

int PrintBag(MQHBAG dataBag)
{
    int errors;

    printf("\n");
    errors = PrintBagContents(dataBag, 0);
    printf("\n");

    return errors;
}

```

```

/*****
/*
/* Function: PrintBagContents
/*
/*****
/*
/* Input Parameters:  Bag Handle
/*                    Indentation level of bag
/*
/*
/* Output Parameters: None
/*
/*
/* Returns:          Number of errors found
/*
/*
/* Logic: Count the number of items in the bag
/*          Obtain selector and item type for each item in the bag.
/*          Obtain the value of the item depending on item type and display the
/*          index of the item, the selector and the value.
/*          If the item is an embedded bag handle then call this function again
/*          to print the contents of the embedded bag increasing the
/*          indentation level.
/*
/*****
int PrintBagContents(MQHBAG dataBag, int indent)
{
    /*****
    /* Definitions
    /*****
    #define LENGTH 500                /* Max length of string to be read*/
    #define INDENT 4                 /* Number of spaces to indent
                                     /* embedded bag display

    /*****
    /* Variables
    /*****
    MQLONG  itemCount;                /* Number of items in the bag
    MQLONG  itemType;                /* Type of the item
    int     i;                        /* Index of item in the bag
    MQCHAR  stringVal[LENGTH+1];     /* Value if item is a string
    MQBYTE  byteStringVal[LENGTH];   /* Value if item is a byte string
    MQLONG  stringLength;            /* Length of string value
    MQLONG  ccsid;                   /* CCSID of string value
    MQINT32 iValue;                 /* Value if item is an integer
    MQINT64 i64Value;               /* Value if item is a 64-bit
                                     /* integer
    MQLONG  selector;               /* Selector of item
    MQHBAG  bagHandle;              /* Value if item is a bag handle
    MQLONG  reason;                 /* reason code
    MQLONG  compCode;               /* completion code
    MQLONG  trimLength;             /* Length of string to be trimmed
    int     errors = 0;              /* Count of errors found
    char    blanks[] = "           /* Blank string used to
                                     /* indent display

    /*****
    /* Count the number of items in the bag
    /*****
    mqCountItems(dataBag, MQSEL_ALL_SELECTORS, &itemCount, &compCode, &reason);

    if (compCode != MQCC_OK)
        errors++;
    else
    {
        printf("
        printf("
        printf("
    }

    /*****
    /* If no errors found, display each item in the bag
    /*****
    if (!errors)
    {
        for (i = 0; i < itemCount; i++)
        {

            /*****
            /* First inquire the type of the item for each item in the bag
            /*****
            mqInquireItemInfo(dataBag,                /* Bag handle
                              MQSEL_ANY_SELECTOR,    /* Item can have any selector*/

```

```

        i,                /* Index position in the bag */
        &selector,        /* Actual value of selector */
                        /* returned by call */
        &itemType,        /* Actual type of item */
                        /* returned by call */
        &compCode,        /* Completion code */
        &reason);        /* Reason Code */

if (compCode != MQCC_OK)
    errors++;

switch(itemType)
{
case MQITEM_INTEGER:
    /******
    /* Item is an integer. Find its value and display its index,
    /* selector and value.
    /******
    mqInquireInteger(dataBag, /* Bag handle */
                    MQSEL_ANY_SELECTOR, /* Allow any selector */
                    i, /* Index position in the bag */
                    &iValue, /* Returned integer value */
                    &compCode, /* Completion code */
                    &reason); /* Reason Code */

    if (compCode != MQCC_OK)
        errors++;
    else
        printf("%.s %-2d %-4d (%d)\n",
            indent, blanks, i, selector, iValue);
    break

case MQITEM_INTEGER64:
    /******
    /* Item is a 64-bit integer. Find its value and display its
    /* index, selector and value.
    /******
    mqInquireInteger64(dataBag, /* Bag handle */
                    MQSEL_ANY_SELECTOR, /* Allow any selector */
                    i, /* Index position in the bag */
                    &i64Value, /* Returned integer value */
                    &compCode, /* Completion code */
                    &reason); /* Reason Code */

    if (compCode != MQCC_OK)
        errors++;
    else
        printf("%.s %-2d %-4d (%"Int64"d)\n",
            indent, blanks, i, selector, i64Value);
    break;

case MQITEM_STRING:
    /******
    /* Item is a string. Obtain the string in a buffer, prepare
    /* the string for displaying and display the index, selector,
    /* string and Character Set ID.
    /******
    mqInquireString(dataBag, /* Bag handle */
                    MQSEL_ANY_SELECTOR, /* Allow any selector */
                    i, /* Index position in the bag */
                    LENGTH, /* Maximum length of buffer */
                    stringVal, /* Buffer to receive string */
                    &stringLength, /* Actual length of string */
                    &ccsid, /* Coded character set id */
                    &compCode, /* Completion code */
                    &reason); /* Reason Code */

    /******
    /* The call can return a warning if the string is too long for
    /* the output buffer and has been truncated, so only check
    /* explicitly for call failure.
    /******
    if (compCode == MQCC_FAILED)
        errors++;
    else
    {
        /******
        /* Remove trailing blanks from the string and terminate with
        /* a null. First check that the string should not have been
        /* longer than the maximum buffer size allowed.
        /******

```

```

        if (stringLength > LENGTH)
            trimLength = LENGTH;
        else
            trimLength = stringLength;
        mqTrim(trimLength, stringVal, stringVal, &compCode, &reason);
        printf("%.s %-2d %-4d 's' %d\n",
            indent, blanks, i, selector, stringVal, ccsid);
    }
    break;

case MQITEM_BYTE_STRING:
    /******
    /* Item is a byte string. Obtain the byte string in a buffer, */
    /* prepare the byte string for displaying and display the */
    /* index, selector and string. */
    /******
    mqInquireByteString(dataBag, /* Bag handle */
        MQSEL_ANY_SELECTOR, /* Allow any selector */
        i, /* Index position in the bag */
        LENGTH, /* Maximum length of buffer */
        byteStringVal, /* Buffer to receive string */
        &stringLength, /* Actual length of string */
        &compCode, /* Completion code */
        &reason); /* Reason Code

    /******
    /* The call can return a warning if the string is too long for */
    /* the output buffer and has been truncated, so only check */
    /* explicitly for call failure. */
    /******
    if (compCode == MQCC_FAILED)
        errors++;
    else
    {
        printf("%.s %-2d %-4d X'",
            indent, blanks, i, selector);

        for (i = 0 ; i < stringLength ; i++)
            printf("

        printf("\n");
    }
    break;

case MQITEM_BAG:
    /******
    /* Item is an embedded bag handle, so call the PrintBagContents*/
    /* function again to display the contents. */
    /******
    mqInquireBag(dataBag, /* Bag handle */
        MQSEL_ANY_SELECTOR, /* Allow any selector */
        i, /* Index position in the bag */
        &bagHandle, /* Returned embedded bag hdl*/
        &compCode, /* Completion code */
        &reason); /* Reason Code

    if (compCode != MQCC_OK)
        errors++;
    else
    {
        printf("%.s %-2d %-4d (%d)\n", indent, blanks, i,
            selector, bagHandle);
        if (selector == MQHA_BAG_HANDLE)
            printf("

        else
            printf("
            PrintBagContents(bagHandle, indent+INDENT);
    }
    break;

default:
    printf("

    }
}
}
return errors;
}

```

Ukázkový program jazyka C pro zjišťování informací o objektech kanálu (amqsaicl.c)

Ukázkový program v jazyce C amqsaicl.c zjišťuje objekty kanálu pomocí rozhraní MQAI.

```
/* **** */
/*
/* Program name: AMQSAICL.C
/*
/* Description: Sample C program to inquire channel objects
/* using the WebSphere MQ Administration Interface (MQAI)
/*
/* <N_OCO_COPYRIGHT>
/* Licensed Materials - Property of IBM
/*
/* 63H9336
/* (c) Copyright IBM Corp. 2008, 2024. All Rights Reserved.
/*
/* US Government Users Restricted Rights - Use, duplication or
/* disclosure restricted by GSA ADP Schedule Contract with
/* IBM Corp.
/* <NOC_COPYRIGHT>
/* **** */
/*
/* Function:
/* AMQSAICL is a sample C program that demonstrates how to inquire
/* attributes of the local queue manager using the MQAI interface. In
/* particular, it inquires all channels and their types.
/*
/* - A PCF command is built from items placed into an MQAI administration
/* bag.
/* These are:-
/* - The generic channel name "*"
/* - The attributes to be inquired. In this sample we just want
/* name and type attributes
/*
/* - The mqExecute MQCMD_INQUIRE_CHANNEL call is executed.
/* The call generates the correct PCF structure.
/* The default options to the call are used so that the command is sent
/* to the SYSTEM.ADMIN.COMMAND.QUEUE.
/* The reply from the command server is placed on a temporary dynamic
/* queue.
/* The reply from the MQCMD_INQUIRE_CHANNEL is read from the
/* temporary queue and formatted into the response bag.
/*
/* - The completion code from the mqExecute call is checked and if there
/* is a failure from the command server, then the code returned by the
/* command server is retrieved from the system bag that has been
/* embedded in the response bag to the mqExecute call.
/*
/* Note: The command server must be running.
/*
/* **** */
/* AMQSAICL has 2 parameter - the queue manager name (optional)
/* - output file (optional) default varies
/* **** */
/* **** */
/* Includes
/* **** */
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>
#if (MQAT_DEFAULT == MQAT_OS400)
#include <recio.h>
#endif

#include <cmqc.h> /* MQI
#include <cmqcfc.h> /* PCF
#include <cmqbc.h> /* MQAI
#include <cmqxc.h> /* MQCD

/* **** */
/* Function prototypes
/* **** */
void CheckCallResult(MQCHAR *, MQLONG , MQLONG);
```



```

/*****
/* DataTypes
/*****
#if (MQAT_DEFAULT == MQAT_OS400)
typedef _RFILE OUTFILEHDL;
#else
typedef FILE OUTFILEHDL;
#endif

/*****
/* Constants
/*****
#if (MQAT_DEFAULT == MQAT_OS400)
const struct
{
    char name[9];
} ChlTypeMap[9] =
{
    " *SDR      ", /* MQCHT_SENDER */
    " *SVR      ", /* MQCHT_SERVER */
    " *RCVR     ", /* MQCHT_RECEIVER */
    " *RQSTR    ", /* MQCHT_REQUESTER */
    " *ALL      ", /* MQCHT_ALL */
    " *CLTCN    ", /* MQCHT_CLNTCONN */
    " *SVRCONN  ", /* MQCHT_SVRCONN */
    " *CLUSRCVR", /* MQCHT_CLUSRCVR */
    " *CLUSSDR  " /* MQCHT_CLUSSDR */
};
#else
const struct
{
    char name[9];
} ChlTypeMap[9] =
{
    "sdr      ", /* MQCHT_SENDER */
    "svr      ", /* MQCHT_SERVER */
    "rcvr     ", /* MQCHT_RECEIVER */
    "rqstr    ", /* MQCHT_REQUESTER */
    "all      ", /* MQCHT_ALL */
    "cltconn  ", /* MQCHT_CLNTCONN */
    "svrcn    ", /* MQCHT_SVRCONN */
    "clusrcvr", /* MQCHT_CLUSRCVR */
    "clusldr  " /* MQCHT_CLUSSDR */
};
#endif

/*****
/* Macros
/*****
#if (MQAT_DEFAULT == MQAT_OS400)
#define OUTFILE "QTEMP/AMQSAICL(AMQSAICL)"
#define OPENOUTFILE(hdl, fname) \
    (hdl) = _Ropen((fname), "wr", rtncode=Y");
#define CLOSEOUTFILE(hdl) \
    _Rclose((hdl));
#define WRITEOUTFILE(hdl, buf, buflen) \
    _Rwrite((hdl), (buf), (buflen));

#elif (MQAT_DEFAULT == MQAT_UNIX)
#define OUTFILE "/tmp/amqsaicl.txt"
#define OPENOUTFILE(hdl, fname) \
    (hdl) = fopen((fname), "w");
#define CLOSEOUTFILE(hdl) \
    fclose((hdl));
#define WRITEOUTFILE(hdl, buf, buflen) \
    fwrite((buf), (buflen), 1, (hdl)); fflush((hdl));

#else
#define OUTFILE "amqsaicl.txt"
#define OPENOUTFILE(fname) \
    fopen((fname), "w");
#define CLOSEOUTFILE(hdl) \
    fclose((hdl));
#define WRITEOUTFILE(hdl, buf, buflen) \
    fwrite((buf), (buflen), 1, (hdl)); fflush((hdl));

#endif

#define ChlType2String(t) ChlTypeMap[(t)-1].name

/*****
/* Function: main
/*****

```

```

/*****
int main(int argc, char *argv[])
{
    /*****/
    /* MQAI variables */
    /*****/
    MQHCONN hConn; /* handle to MQ connection */
    MQCHAR qmName[MQ_Q_MGR_NAME_LENGTH+1]=""; /* default QMgr name */
    MQLONG reason; /* reason code */
    MQLONG connReason; /* MQCONN reason code */
    MQLONG compCode; /* completion code */
    MQHBAG adminBag = MQHB_UNUSABLE_HBAG; /* admin bag for mqExecute */
    MQHBAG responseBag = MQHB_UNUSABLE_HBAG; /* response bag for mqExecute */
    MQHBAG cAttrsBag; /* bag containing chl attributes */
    MQHBAG errorBag; /* bag containing cmd server error */
    MQLONG mqExecuteCC; /* mqExecute completion code */
    MQLONG mqExecuteRC; /* mqExecute reason code */
    MQLONG chlNameLength; /* Actual length of chl name */
    MQLONG chlType; /* Channel type */
    MQLONG i; /* loop counter */
    MQLONG numberOfBags; /* number of bags in response bag */
    MQCHAR chlName[MQ_OBJECT_NAME_LENGTH+1]; /* name of chl extracted from bag */
    MQCHAR OutputBuffer[100]; /* output data buffer */
    OUTFILEHDL *outfp = NULL; /* output file handle */

    /*****/
    /* Connect to the queue manager */
    /*****/
    if (argc > 1)
        strncpy(qmName, argv[1], (size_t)MQ_Q_MGR_NAME_LENGTH);
    MQCONN(qmName, &hConn, &compCode, &connReason);

    /*****/
    /* Report the reason and stop if the connection failed. */
    /*****/
    if (compCode == MQCC_FAILED)
    {
        CheckCallResult("Queue Manager connection", compCode, connReason);
        exit( (int)connReason);
    }

    /*****/
    /* Open the output file */
    /*****/
    if (argc > 2)
    {
        OPENOUTFILE(outfp, argv[2]);
    }
    else
    {
        OPENOUTFILE(outfp, OUTFILE);
    }

    if(outfp == NULL)
    {
        printf("Could not open output file.\n");
        goto MOD_EXIT;
    }

    /*****/
    /* Create an admin bag for the mqExecute call */
    /*****/
    mqCreateBag(MQCBO_ADMIN_BAG, &adminBag, &compCode, &reason);
    CheckCallResult("Create admin bag", compCode, reason);

    /*****/
    /* Create a response bag for the mqExecute call */
    /*****/
    mqCreateBag(MQCBO_ADMIN_BAG, &responseBag, &compCode, &reason);
    CheckCallResult("Create response bag", compCode, reason);

    /*****/
    /* Put the generic channel name into the admin bag */
    /*****/
    mqAddString(adminBag, MQCACH_CHANNEL_NAME, MQBL_NULL_TERMINATED, "*",
        &compCode, &reason);
    CheckCallResult("Add channel name", compCode, reason);

    /*****/
    /* Put the channel type into the admin bag */
    /*****/
    mqAddInteger(adminBag, MQIACH_CHANNEL_TYPE, MQCHT_ALL, &compCode, &reason);
    CheckCallResult("Add channel type", compCode, reason);
}

```

```

/*****
/* Add an inquiry for various attributes */
/*****
mqAddInquiry(adminBag, MQIACH_CHANNEL_TYPE, &compCode;, &reason;);
CheckCallResult("Add inquiry", compCode, reason);

/*****
/* Send the command to find all the channel names and channel types. */
/* The mqExecute call creates the PCF structure required, sends it to */
/* the command server, and receives the reply from the command server into */
/* the response bag. The attributes are contained in system bags that are */
/* embedded in the response bag, one set of attributes per bag. */
/*****
mqExecute(hConn, /* MQ connection handle */
          MQCMD_INQUIRE_CHANNEL, /* Command to be executed */
          MQHB_NONE, /* No options bag */
          adminBag, /* Handle to bag containing commands */
          responseBag, /* Handle to bag to receive the response*/
          MQHO_NONE, /* Put msg on SYSTEM.ADMIN.COMMAND.QUEUE*/
          MQHO_NONE, /* Create a dynamic q for the response */
          &compCode;, /* Completion code from the mqexecute */
          &reason;); /* Reason code from mqexecute call */

/*****
/* Check the command server is started. If not exit. */
/*****
if (reason == MQRC_CMD_SERVER_NOT_AVAILABLE)
{
    printf("Please start the command server: <strmqcsv QMgrName="">\n");
    goto MOD_EXIT;
}

/*****
/* Check the result from mqExecute call. If successful find the channel */
/* types for all the channels. If failed find the error. */
/*****
if ( compCode == MQCC_OK ) /* Successful mqExecute */
{
    /*****
    /* Count the number of system bags embedded in the response bag from the */
    /* mqExecute call. The attributes for each channel are in separate bags. */
    /*****
    mqCountItems(responseBag, MQHA_BAG_HANDLE, &numberOfBags;,
                &compCode;, &reason;);
    CheckCallResult("Count number of bag handles", compCode, reason);

    for ( i=0; i<numberOfbags; i++)
    {
        /*****
        /* Get the next system bag handle out of the mqExecute response bag. */
        /* This bag contains the channel attributes */
        /*****
        mqInquireBag(responseBag, MQHA_BAG_HANDLE, i, &cAttrsBag,
                    &compCode, &reason);
        CheckCallResult("Get the result bag handle", compCode, reason);

        /*****
        /* Get the channel name out of the channel attributes bag */
        /*****
        mqInquireString(cAttrsBag, MQCACH_CHANNEL_NAME, 0, MQ_OBJECT_NAME_LENGTH,
                       chlName, &chlNameLength, NULL, &compCode, &reason);
        CheckCallResult("Get channel name", compCode, reason);

        /*****
        /* Get the channel type out of the channel attributes bag */
        /*****
        mqInquireInteger(cAttrsBag, MQIACH_CHANNEL_TYPE, MQIND_NONE, &chlType,
                        &compCode, &reason);
        CheckCallResult("Get type", compCode, reason);

        /*****
        /* Use mqTrim to prepare the channel name for printing. */
        /* Print the result. */
        /*****
        mqTrim(MQ_CHANNEL_NAME_LENGTH, chlName, chlName, &compCode, &reason);
        sprintf(OutputBuffer, "%-20s%-9s", chlName, ChlType2String(chlType));
        WRITEOUTFILE(outfp,OutputBuffer,29)
    }
}

```

```

else                                     /* Failed mqExecute */
{
    printf("Call to get channel attributes failed: Cc = %ld : Rc = %ld\n",
           compCode, reason);
    /******
    /* If the command fails get the system bag handle out of the mqexecute */
    /* response bag.This bag contains the reason from the command server */
    /* why the command failed. */
    /******
    if (reason == MQRCCF_COMMAND_FAILED)
    {
        mqInquireBag(responseBag, MQHA_BAG_HANDLE, 0, &errorBag,
                     &compCode, &reason);
        CheckCallResult("Get the result bag handle", compCode, reason);

        /******
        /* Get the completion code and reason code, returned by the command */
        /* server, from the embedded error bag. */
        /******
        mqInquireInteger(errorBag, MQIASY_COMP_CODE, MQIND_NONE, &mqExecuteCC,
                         &compCode, &reason );
        CheckCallResult("Get the completion code from the result bag",
                        compCode, reason);
        mqInquireInteger(errorBag, MQIASY_REASON, MQIND_NONE, &mqExecuteRC,
                         &compCode, &reason);
        CheckCallResult("Get the reason code from the result bag",
                        compCode, reason);
        printf("Error returned by the command server: Cc = %ld : Rc = %ld\n",
               mqExecuteCC, mqExecuteRC);
    }
}

MOD_EXIT:
/******
/* Delete the admin bag if successfully created. */
/******
if (adminBag != MQHB_UNUSABLE_HBAG)
{
    mqDeleteBag(&adminBag, &compCode, &reason);
    CheckCallResult("Delete the admin bag", compCode, reason);
}

/******
/* Delete the response bag if successfully created. */
/******
if (responseBag != MQHB_UNUSABLE_HBAG)
{
    mqDeleteBag(&responseBag, &compCode, &reason);
    CheckCallResult("Delete the response bag", compCode, reason);
}

/******
/* Disconnect from the queue manager if not already connected */
/******
if (connReason != MQRC_ALREADY_CONNECTED)
{
    MQDISC(&hConn, &compCode, &reason);
    CheckCallResult("Disconnect from Queue Manager", compCode, reason);
}

/******
/* Close the output file if open */
/******
if(outfp != NULL)
    CLOSEOUTFILE(outfp);

return 0;
}

/******
/*
/* Function: CheckCallResult */
/*
/*
/******
/*
/* Input Parameters: Description of call */
/*                   Completion code */
/*                   Reason code */
/*
/*
/* Output Parameters: None */
/*

```

```

/* Logic: Display the description of the call, the completion code and the */
/* reason code if the completion code is not successful */
/* */
/*****
void CheckCallResult(char *callText, MQLONG cc, MQLONG rc)
{
    if (cc != MQCC_OK)
        printf("%s failed: Completion Code = %ld : Reason = %ld\n", callText,
            cc, rc);
}

```

Ukázkový program v jazyce C pro dotazování front a tisk informací (amqsailq.c)

Ukázkový program v jazyce C amqsailq.c zjišťuje aktuální hloubku lokálních front pomocí rozhraní MQAI.

```

/*****
/*
/* Program name: AMQSAILQ.C */
/* */
/* Description: Sample C program to inquire the current depth of the local */
/* queues using the WebSphere MQ Administration Interface (MQAI) */
/* */
/* Statement: Licensed Materials - Property of IBM */
/* */
/* 84H2000, 5765-B73 */
/* 84H2001, 5639-B42 */
/* 84H2002, 5765-B74 */
/* 84H2003, 5765-B75 */
/* 84H2004, 5639-B43 */
/* */
/* (C) Copyright IBM Corp. 1999, 2024. */
/* */
/*****
/*
/* Function:
/* AMQSAILQ is a sample C program that demonstrates how to inquire
/* attributes of the local queue manager using the MQAI interface. In
/* particular, it inquires the current depths of all the local queues.
/*
/* - A PCF command is built by placing items into an MQAI administration
/* bag.
/* These are:-
/* - The generic queue name "*"
/* - The type of queue required. In this sample we want to
/* inquire local queues.
/* - The attribute to be inquired. In this sample we want the
/* current depths.
/*
/* - The mqExecute call is executed with the command MQCMD_INQUIRE_Q.
/* The call generates the correct PCF structure.
/* The default options to the call are used so that the command is sent
/* to the SYSTEM.ADMIN.COMMAND.QUEUE.
/* The reply from the command server is placed on a temporary dynamic
/* queue.
/* The reply from the MQCMD_INQUIRE_Q command is read from the
/* temporary queue and formatted into the response bag.
/*
/* - The completion code from the mqExecute call is checked and if there
/* is a failure from the command server, then the code returned by
/* command server is retrieved from the system bag that has been
/* embedded in the response bag to the mqExecute call.
/*
/* - If the call is successful, the depth of each local queue is placed
/* in system bags embedded in the response bag of the mqExecute call.
/* The name and depth of each queue is obtained from each of the bags
/* and the result displayed on the screen.
/*
/* Note: The command server must be running.
/*
/*****
/*
/* AMQSAILQ has 1 parameter - the queue manager name (optional)
/*
/*****

```

```

/*****
/* Includes
/*****
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>

#include <cmqc.h>          /* MQI          */
#include <cmqcfc.h>       /* PCF          */
#include <cmqbc.h>        /* MQAI          */

/*****
/* Function prototypes
/*****
void CheckCallResult(MQCHAR *, MQLONG , MQLONG);

/*****
/* Function: main
/*****
int main(int argc, char *argv[])
{
    /*****
    /* MQAI variables
    /*****
    MQHCONN hConn;          /* handle to WebSphere MQ connection */
    MQCHAR qmName[MQ_Q_MGR_NAME_LENGTH+1]=""; /* default QMgr name */
    MQLONG reason;         /* reason code */
    MQLONG connReason;    /* MQCONN reason code */
    MQLONG compCode;      /* completion code */
    MQHBAG adminBag = MQHB_UNUSABLE_HBAG; /* admin bag for mqExecute */
    MQHBAG responseBag = MQHB_UNUSABLE_HBAG; /* response bag for mqExecute */
    MQHBAG qAttrsBag;     /* bag containing q attributes */
    MQHBAG errorBag;      /* bag containing cmd server error */
    MQLONG mqExecuteCC;   /* mqExecute completion code */
    MQLONG mqExecuteRC;   /* mqExecute reason code */
    MQLONG qNameLength;   /* Actual length of q name */
    MQLONG qDepth;       /* depth of queue */
    MQLONG i;            /* loop counter */
    MQLONG numberOfBags; /* number of bags in response bag */
    MQCHAR qName[MQ_Q_NAME_LENGTH+1]; /* name of queue extracted from bag*/

    printf("Display current depths of local queues\n\n");

    /*****
    /* Connect to the queue manager
    /*****
    if (argc > 1)
        strncpy(qmName, argv[1], (size_t)MQ_Q_MGR_NAME_LENGTH);
    MQCONN(qmName, &hConn, &compCode, &connReason);

    /*****
    /* Report the reason and stop if the connection failed.
    /*****
    if (compCode == MQCC_FAILED)
    {
        CheckCallResult("Queue Manager connection", compCode, connReason
    );
        exit( (int)connReason);
    }

    /*****
    /* Create an admin bag for the mqExecute call
    /*****
    mqCreateBag(MQCBO_ADMIN_BAG, &adminBag, &compCode, &reason);
    CheckCallResult("Create admin bag", compCode, reason);

    /*****
    /* Create a response bag for the mqExecute call
    /*****
    mqCreateBag(MQCBO_ADMIN_BAG, &responseBag, &compCode, &reason);
    CheckCallResult("Create response bag", compCode, reason);

    /*****
    /* Put the generic queue name into the admin bag
    /*****
    mqAddString(adminBag, MQCA_Q_NAME, MQBL_NULL_TERMINATED, "*",
                &compCode, &reason);
    CheckCallResult("Add q name", compCode, reason);

    /*****
    /* Put the local queue type into the admin bag
    /*****

```

```

/*****
mqAddInteger(adminBag, MQIA_Q_TYPE, MQQT_LOCAL, &compCode, &reason);
CheckCallResult("Add q type", compCode, reason);

/*****
/* Add an inquiry for current queue depths */
/*****
mqAddInquiry(adminBag, MQIA_CURRENT_Q_DEPTH, &compCode, &reason);
CheckCallResult("Add inquiry", compCode, reason);

/*****
/* Send the command to find all the local queue names and queue depths. */
/* The mqExecute call creates the PCF structure required, sends it to */
/* the command server, and receives the reply from the command server into */
/* the response bag. The attributes are contained in system bags that are */
/* embedded in the response bag, one set of attributes per bag. */
/*****
mqExecute(hConn, /* WebSphere MQ connection handle */
          MQCMD_INQUIRE_Q, /* Command to be executed */
          MQHB_NONE, /* No options bag */
          adminBag, /* Handle to bag containing commands */
          responseBag, /* Handle to bag to receive the response */
          MQHO_NONE, /* Put msg on SYSTEM.ADMIN.COMMAND.QUEUE */
          MQHO_NONE, /* Create a dynamic q for the response */
          &compCode, /* Completion code from the mqExecute */
          &reason); /* Reason code from mqExecute call */

/*****
/* Check the command server is started. If not exit. */
/*****
if (reason == MQRC_CMD_SERVER_NOT_AVAILABLE)
{
    printf("Please start the command server: <strmqcsv QMgrName>\n");
    MQDISC(&hConn, &compCode, &reason);
    CheckCallResult("Disconnect from Queue Manager", compCode, reason);
    exit(98);
}

/*****
/* Check the result from mqExecute call. If successful find the current */
/* depths of all the local queues. If failed find the error. */
/*****
if ( compCode == MQCC_OK ) /* Successful mqExecute */
{
    /*****
    /* Count the number of system bags embedded in the response bag from the */
    /* mqExecute call. The attributes for each queue are in a separate bag. */
    /*****
    mqCountItems(responseBag, MQHA_BAG_HANDLE, &numberOfBags, &compCode,
                 &reason);
    CheckCallResult("Count number of bag handles", compCode, reason);

    for ( i=0; i<numberOfBags; i++)
    {
        /*****
        /* Get the next system bag handle out of the mqExecute response bag. */
        /* This bag contains the queue attributes */
        /*****
        mqInquireBag(responseBag, MQHA_BAG_HANDLE, i, &qAttrsBag, &compCode,
                   &reason);
        CheckCallResult("Get the result bag handle", compCode, reason);

        /*****
        /* Get the queue name out of the queue attributes bag */
        /*****
        mqInquireString(qAttrsBag, MQCA_Q_NAME, 0, MQ_Q_NAME_LENGTH, qName,
                      &qNameLength, NULL, &compCode, &reason);
        CheckCallResult("Get queue name", compCode, reason);

        /*****
        /* Get the depth out of the queue attributes bag */
        /*****
        mqInquireInteger(qAttrsBag, MQIA_CURRENT_Q_DEPTH, MQIND_NONE, &qDepth,
                       &compCode, &reason);
        CheckCallResult("Get depth", compCode, reason);

        /*****
        /* Use mqTrim to prepare the queue name for printing. */
        /* Print the result. */
        /*****
        mqTrim(MQ_Q_NAME_LENGTH, qName, qName, &compCode, &reason)

```

```

    printf("%4d %-48s\n", qDepth, qName);
}
}
else /* Failed mqExecute */
{
    printf("Call to get queue attributes failed: Completion Code = %d :
        Reason = %d\n", compCode, reason);

    /******
    /* If the command fails get the system bag handle out of the mqExecute */
    /* response bag. This bag contains the reason from the command server */
    /* why the command failed. */
    /******
    if (reason == MQRCCF_COMMAND_FAILED)
    {
        mqInquireBag(responseBag, MQHA_BAG_HANDLE, 0, &errorBag, &compCode,
            &reason);
        CheckCallResult("Get the result bag handle", compCode, reason);

        /******
        /* Get the completion code and reason code, returned by the command */
        /* server, from the embedded error bag. */
        /******
        mqInquireInteger(errorBag, MQIASY_COMP_CODE, MQIND_NONE, &mqExecuteCC,
            &compCode, &reason);
        CheckCallResult("Get the completion code from the result bag",
            compCode, reason);
        mqInquireInteger(errorBag, MQIASY_REASON, MQIND_NONE, &mqExecuteRC,
            &compCode, &reason);
        CheckCallResult("Get the reason code from the result bag",
            compCode, reason);
        printf("Error returned by the command server: Completion Code = %d :
            Reason = %d\n", mqExecuteCC, mqExecuteRC);
    }
}

/******
/* Delete the admin bag if successfully created. */
/******
if (adminBag != MQHB_UNUSABLE_HBAG)
{
    mqDeleteBag(&adminBag, &compCode, &reason);
    CheckCallResult("Delete the admin bag", compCode, reason);
}

/******
/* Delete the response bag if successfully created. */
/******
if (responseBag != MQHB_UNUSABLE_HBAG)
{
    mqDeleteBag(&responseBag, &compCode, &reason);
    CheckCallResult("Delete the response bag", compCode, reason);
}

/******
/* Disconnect from the queue manager if not already connected */
/******
if (connReason != MQRC_ALREADY_CONNECTED)
{
    MQDISC(&hConn, &compCode, &reason);
    CheckCallResult("Disconnect from queue manager", compCode, reason);
}
return 0;
}

*****
*
* Function: CheckCallResult
*
*****
*
* Input Parameters: Description of call
*                   Completion code
*                   Reason code
*
* Output Parameters: None
*
* Logic: Display the description of the call, the completion code and the
*        reason code if the completion code is not successful
*
*****

```



```
void CheckCallResult(char *callText, MQLONG cc, MQLONG rc)
{
    if (cc != MQCC_OK)
        printf("%s failed: Completion Code = %d : Reason = %d\n",
            callText, cc, rc);
}
```

Rady a tipy pro konfiguraci produktu IBM WebSphere MQ

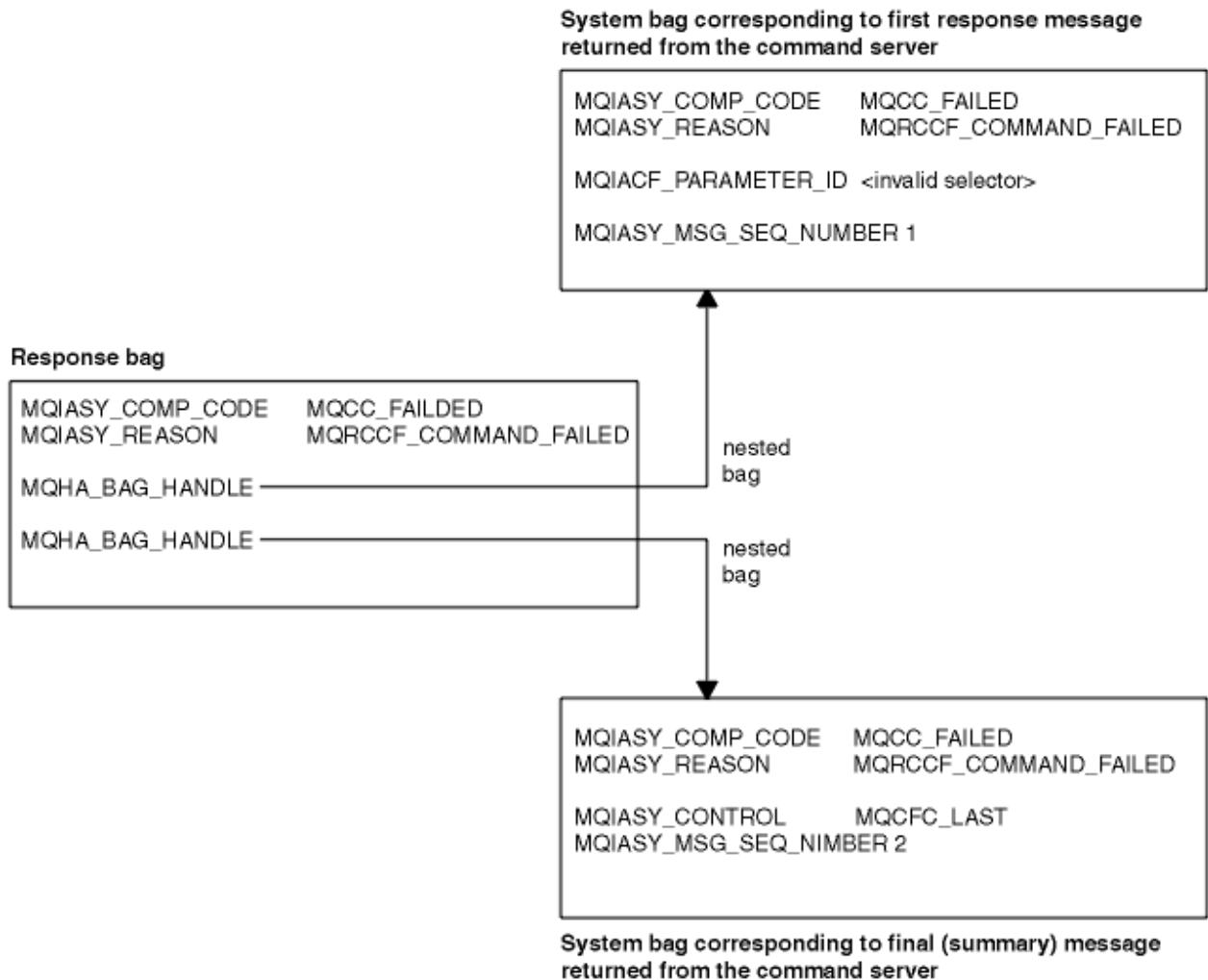
Programování pokynů a tipů při používání rozhraní MQAI.

Volání MQAI používá PCF zprávy k odeslání příkazů administrace na příkazový server místo přímého zpracování přímo s příkazovým serverem. Zde je několik tipů pro konfiguraci produktu WebSphere MQ pomocí rozhraní MQAI:

- Znakové řetězce v produktu WebSphere MQ jsou vyplňovány mezerami na pevnou délku. Řetězce zakončené znakem C lze normálně zadávat jako vstupní parametry do programovacích rozhraní produktu WebSphere MQ .
- Chcete-li vymazat hodnotu řetězcového atributu, nastavte ji na jednu prázdnou hodnotu a ne prázdný řetězec.
- Zvažte v předstihu atributy, které chcete změnit, a dotazujte se pouze na tyto atributy.
- Určité atributy nelze změnit, například název fronty nebo typ kanálu. Ujistěte se, že jste se pokusili změnit pouze ty atributy, které lze upravit. Prohlédněte si seznam požadovaných a volitelných parametrů pro specifický objekt změny PCF. Viz [Definice formátů Programovatelných příkazů](#).
- Pokud volání MQAI selže, vrátí se do vaku odezvy část podrobností o selhání. Další podrobnosti lze poté nalézt ve vnořeném balíku, ke kterému lze přistupovat pomocí selektoru MQHA_BAG_HANDLE. Například, pokud volání mqExecute selže s kódem příčiny MQRCCF_COMMAND_FAILED, vrátí se tyto informace v balíku odezvy. Možný důvod pro tento kód příčiny je, že zadaný selektor nebyl platný pro typ zprávy příkazu a tento detail informací se nachází ve vnořeném vaku, ke kterému lze přistupovat pomocí popisovače balíku.

Další informace o příkazu MQExecute viz [“Odeslání příkazů administrace na příkazový server pomocí volání mqExecute”](#) na stránce 55 .

Tento scénář ukazuje následující diagram:



Rozšířená témata MQAI

Informace o indexování, konverzi dat a používání deskriptoru zpráv

- Indexování

Indexy se používají při nahrazování nebo odebrání existujících datových položek z balíku za účelem uchování pořadí vložení. Úplné podrobnosti o indexování lze nalézt v [“Indexace v rozhraní MQAI”](#) na stránce 42.

- Převod dat

Řetězce obsažené v datovém balíku MQAI se mohou nacházet v různých kódovaných znakových sadách a tyto řetězce lze převést pomocí celočíselného volání mqSet. Úplné podrobnosti o převodu dat lze nalézt v [“Převod dat v MQAI”](#) na stránce 43.

- Použití deskriptoru zpráv

Rozhraní MQAI generuje deskriptor zprávy, který je nastaven na počáteční hodnotu při vytvoření datového balíku. Úplné podrobnosti o použití deskriptoru zpráv lze najít v [“Použití deskriptoru zpráv v MQAI”](#) na stránce 45.

Indexace v rozhraní MQAI

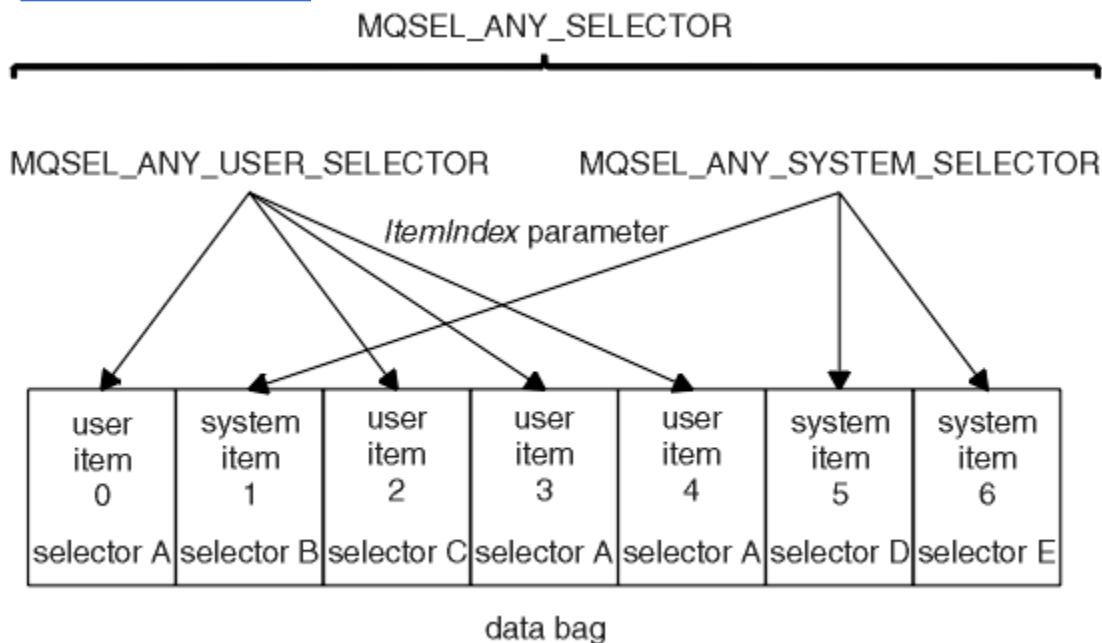
Indexy se používají při nahrazování nebo odebrání existujících datových položek z balíku. Existují tři typy indexování, které umožňují snadné načtení datových položek.

Každý selektor a hodnota v rámci datové položky v balíku mají tři přidružená čísla indexů:

- Index vztahující se k jiným položkám, které mají stejný selektor.

- Index vztahující se ke kategorii selektoru (uživatel nebo systém), ke kterému položka patří.
- Index vztahující se ke všem datovým položkám v tašce (uživatel a systém).

To umožňuje indexování podle selektorů uživatele, selektorů systému, nebo obojí, jak je zobrazeno v [Obrázek 1 na stránce 43](#).



Obrázek 1. Indexování

Na obrázku [Obrázek 1 na stránce 43](#) může uživatel položka 3 (selektor A) odkazovat na následující dvojice indexů:

Selector	ItemIndex
selektor A	1
SELEKTOR MQSEL_ANY_USER_SELECTOR	2
SELEKTOR MQSEL_ANY_SELECTOR	3

Index je založen na nuly jako pole v C; pokud existuje 'n' výskytů, index se pohybuje od nuly přes 'n-1' bez mezer.

Indexy se používají při nahrazování nebo odebírání existujících datových položek z balíku. Použijete-li tímto způsobem, bude zachováno pořadí vložení, ale mohou být ovlivněny indexy ostatních datových položek. Příklady této funkce viz [Změna informací v rámci balíku](#) a [Odstranění datových položek](#).

Tři typy indexování umožňují snadné načítání datových položek. Pokud například existují tři instance určitého selektoru v balíku, volání položek mqCount může spočítat počet instancí tohoto selektoru a volání mqInquire* mohou určovat pouze tyto hodnoty, a to jak selektor, tak index. To je užitečné pro atributy, které mohou mít seznam hodnot, jako jsou některé z výhodů na kanálech.

Převod dat v MQAI

Řetězce obsažené v datovém balíku MQAI se mohou nacházet v různých kódovaných znakových sadách. Tyto řetězce lze převést pomocí celočíselného volání mqSet.

Podobně jako zprávy PCF, řetězce obsažené v datovém balíku MQAI mohou být v různých kódovaných znakových sadách. Obvykle jsou všechny řetězce ve zprávě PCF ve stejné kódované znakové sadě; tj. stejné nastavení jako správce front.

Každá řetězcová položka v datovém balíku obsahuje dvě hodnoty; samotný řetězec a CCSID. Řetězec, který se přidá do balíku, se získá z parametru *Buffer* řetězce mqAddString nebo mqSetString.

Identifikátor CCSID se získá ze systémové položky obsahující selektor MQIASY_CODED_CHAR_SET_ID. To se označuje jako *CCSID balíku* a lze jej změnit pomocí celočíselného volání mqSet.

Když se dotazujete na hodnotu řetězce obsaženého v datovém balíku, CCSID je výstupní parametr z volání.

Tabulka 2 na stránce 44 zobrazuje pravidla používaná při převádění datových pytlů do zpráv a naopak:

<i>Tabulka 2. Zpracování CCSID</i>			
Volání MQAI	CCSID	Vstup pro volání	Výstup pro volání
mqBagToBuffer	CCSID na balík (1)	Ignorováno	Nezměněno
mqBagToBuffer	Řetězcové CCSID v balíku	Použita	Nezměněno
mqBagToBuffer	Řetězcové CCSID ve vyrovnávací paměti	Nelze použít	Zkopírováno z řetězcových CCSID v balíku
mqBufferToBag	CCSID na balík (1)	Ignorováno	Nezměněno
mqBufferToBag	Řetězcové CCSID ve vyrovnávací paměti	Použita	Nezměněno
mqBufferToBag	Řetězcové CCSID v balíku	Nelze použít	Zkopírováno z řetězcových CCSID ve vyrovnávací paměti
mqPutBag	MQMD CCSID	Použita	Nezměněno (2)
mqPutBag	CCSID na balík (1)	Ignorováno	Nezměněno
mqPutBag	Řetězcové CCSID v balíku	Použita	Nezměněno
mqPutBag	Řetězcové CCSID v odeslané zprávě	Nelze použít	Zkopírováno z řetězcových CCSID v balíku
mqGetBag	MQMD CCSID	Používá se pro převod dat zprávy	Nastavit na CCSID vrácených dat (3)
mqGetBag	CCSID na balík (1)	Ignorováno	Nezměněno
mqGetBag	Řetězcové CCSID ve zprávě	Použita	Nezměněno
mqGetBag	Řetězcové CCSID v balíku	Nelze použít	Zkopírováno z řetězcových CCSID ve zprávě
mqExecute	CCSID pro balík požadavků	Používá se pro deskriptor MQMD zprávy požadavku (4)	Nezměněno
mqExecute	CCSID pro vak odpovědi	Používá se pro konverzi dat zprávy odpovědi (4)	Nastavit na CCSID vrácených dat (3)
mqExecute	Řetězcové CCSID v balíku požadavků	Používá se pro zprávu požadavku	Nezměněno
mqExecute	Řetězcové CCSID v tašce odpovědi	Nelze použít	Zkopírováno z řetězcových CCSID ve zprávě odpovědi

Tabulka 2. Zpracování CCSID (pokračování)

Volání MQAI	CCSID	Vstup pro volání	Výstup pro volání
Notes:			
1. CCSID balíku je systémová položka se selektorem MQIASY_CODED_CHAR_SET_ID.			
2. Hodnota MQCCSI_Q_MGR se změnila na aktuální identifikátor CCSID správce front.			
3. Je-li požadována konverze dat, vrátí se CCSID vrácených dat stejná jako výstupní hodnota. Není-li konverze dat požadována, vrátí se CCSID vrácených dat stejná jako hodnota zprávy. Všimněte si, že pokud se požaduje konverze dat, žádná zpráva se nevrátí, ale selže.			
4. Je-li CCSID MQCCSI_DEFAULT, použije se CCSID správce front.			

Použití deskriptoru zpráv v MQAI

Deskriptor zprávy vygenerovaný rozhraním MQAI je nastaven na počáteční hodnotu při vytvoření datového balíku.

Typ příkazu PCF se získá z systémové položky se selektorem MQIASY_TYPE. Při vytváření datového balíku se počáteční hodnota této položky nastavuje v závislosti na typu balíku, který vytvoříte:

Tabulka 3. Typ příkazu PCF

Druh sáčku	Počáteční hodnota položky MQIASY_TYPE
MQCBO_ADMIN_BAG	PŘÍKAZ MQCFT_COMMAND
MQCBO_COMMAND_BAG	PŘÍKAZ MQCFT_COMMAND
MQCBO_*	UŽIVATEL MQCFT_USER

Když adaptér MQAI vygeneruje deskriptor zprávy, hodnoty použité v parametrech *Format* a *MsgType* závisí na hodnotě systémové položky se selektorem MQIASY_TYPE, jak je zobrazeno v [Tabulka 3](#) na stránce 45.

Tabulka 4. Parametry formátu a MsgType deskriptoru MQMD

Typ příkazu PCF	Formát	MsgType
PŘÍKAZ MQCFT_COMMAND	MQFMT_ADMIN	POŽADAVEK MQMT_REQUEST
SESTAVA MQCFT_REPORT	MQFMT_ADMIN	SESTAVA MQMT_REPORT
ODEZVA MQCFT_RESPONSE	MQFMT_ADMIN	MQMT_REPLY
MQCFT_TRACE_ROUTE	MQFMT_ADMIN	MQM_DATAGRAM
UDÁLOST MQCFT_EVENT	UDÁLOST MQFMT_EVENT	MQM_DATAGRAM
MQCFT_*	MQFMT_PCF	MQM_DATAGRAM

Tabulka 4 na stránce 45 ukazuje, že pokud vytvoříte administrativní balík nebo balík příkazů, *Format* deskriptoru zprávy je MQFMT_ADMIN a *MsgType* je MQMT_REQUEST. To je vhodné pro zprávu požadavku PCF odeslanou na příkazový server, když se odezva očekává zpět.

Ostatní parametry v deskriptoru zpráv mají hodnoty zobrazené v [Tabulka 5](#) na stránce 45.

Tabulka 5. Hodnoty deskriptoru zpráv

Parametr	Hodnota
<i>StrucId</i>	ID_STRUKTURY MQM_STRUCT
<i>Version</i>	MQMD_VERSION_1

Tabulka 5. Hodnoty deskriptoru zpráv (pokračování)

Parametr	Hodnota
<i>Report</i>	MQRO_NONE
<i>MsgType</i>	Viz Tabulka 4 na stránce 45.
<i>Expiry</i>	30 sekund (poznámka “1” na stránce 46)
<i>Feedback</i>	MQFB_NONE
<i>Encoding</i>	MQENC_NATIVE
<i>CodedCharSetId</i>	závisí na CCSID balíku (poznámka “2” na stránce 46)
<i>Format</i>	Viz Tabulka 4 na stránce 45.
<i>Priority</i>	MQPRI_PRIORITY_AS_Q_DEF
<i>Persistence</i>	MQPER_NOT_PERSISTENT
<i>MsgId</i>	MQMI_NONE
<i>CorrelId</i>	MQCI_NONE
<i>BackoutCount</i>	0
<i>ReplyToQ</i>	viz poznámka “3” na stránce 46
<i>ReplyToQMgr</i>	prázdná
<p>Notes:</p> <ol style="list-style-type: none"> 1. Tato hodnota může být přepsána při volání mqExecute pomocí parametru OptionsBag . Další informace o tomto tématu naleznete v tématu mqExecute. 2. Viz “Převod dat v MQAI” na stránce 43. 3. Název uživatelem zadané fronty odpovědi nebo dočasné dynamické dynamické fronty generované rozhraním MQAI pro zprávy typu MQMT_REQUEST. Jinak prázdné místo. 	

Datové sáčky

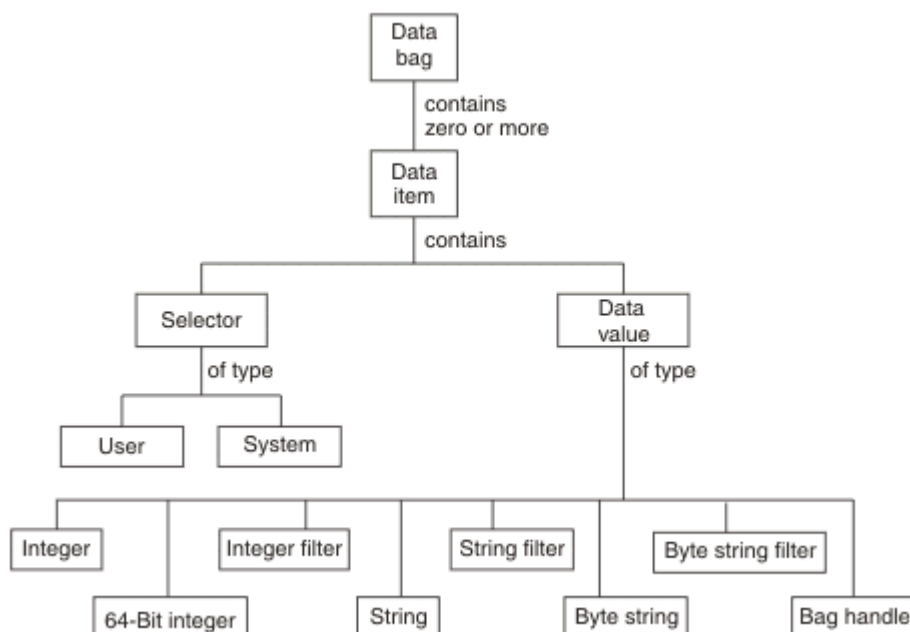
Datový balík je prostředkem pro manipulaci s vlastnostmi nebo parametry objektů za použití rozhraní MQAI.

Datové bagy

- Datový balík obsahuje nula nebo více *datových položek*. Tyto datové položky jsou řazeny v rámci balíku, jak jsou umístěny do tašky. Tomu se říká *pořadí vložení*. Každá datová položka obsahuje *selektor* , který identifikuje datovou položku a *hodnotu* této datové položky, která může být buď celým číslem, 64bitovým celým číslem, celočíselným filtrem, řetězcem, řetězcovým filtrem, bajtovým řetězcem, filtrem bajtového řetězce nebo manipulátorem s jiným vakem. Datové položky jsou popsány v podrobnostech v produktu “Datová položka” na stránce 49

Existují dva typy selektorů: *selektory uživatelů* a *selektory systému*. Ty jsou popsány v části *Selektory MQAI*. Selektory jsou obvykle jedinečné, ale je možné mít více hodnot pro stejný selektor. V tomto případě *index* identifikuje konkrétní výskyt selektoru, který je nezbytný. Indexy jsou popsány v tématu “Indexace v rozhraní MQAI” na stránce 42.

Hierarchie těchto konceptů je zobrazena na [Obrázku 1](#).



Obrázek 2. Hierarchie koncepcí MQAI

Hierarchie byla vysvětlena v předchozím odstavci.

Typy datových pytlů

V závislosti na úloze, kterou chcete provést, si můžete vybrat typ datového balíku, který chcete vytvořit:

uživatelská množina

Jednoduchá taška používaná pro uživatelská data.

administrační množina

Sada pro data používaná k administraci objektů produktu WebSphere MQ pomocí odeslání administračních zpráv na příkazový server. Administrativní vak automaticky implikuje určité volby, jak je popsáno v [“Vytváření a odstraňování datových balíčků”](#) na stránce 48.

množina příkazů

Pro příkazy administrace objektů produktu WebSphere MQ byl vytvořen také balík. Avšak na rozdíl od balíku pro administraci, příkazový balík neimplikuje automaticky určité volby, ačkoli tyto volby jsou k dispozici. Další informace o volbách viz [“Vytváření a odstraňování datových balíčků”](#) na stránce 48.

skupinový balík

Sáček používaný k zadržení sady seskupených datových položek. Skupinové balíky nelze použít pro administraci objektů produktu WebSphere MQ .

Kromě toho je **systemová množina** vytvořena rozhraním MQAI, když se z příkazového serveru vrátí zpráva odpovědi a umístí se do výstupního balíku uživatele. Systemový balík nemůže být upraven uživatelem.

Použití datových pytlů Odlišné způsoby použití datových pytlů jsou uvedeny v tomto tématu:

Použití datových bag

Různé způsoby použití datových pytlů jsou uvedeny v následujícím seznamu:

- Datové pytle [“Vytváření a odstraňování datových balíčků”](#) na stránce 48 můžete vytvářet a odstraňovat.
- Data mezi aplikacemi můžete odesílat pomocí datových balíčků [“Vkládání a přijímání datových pytlů”](#) na stránce 48.
- Datové položky můžete přidat do datových pytlů [“Přidání datových položek do balíčků”](#) na stránce 49.

- Do datového balíku [“Přidání dotazového příkazu do balíku”](#) na stránce 50 můžete přidat dotazovací příkaz.
- Můžete se dotazovat v datových pytích [“Inkviování v rámci datových sáčků”](#) na stránce 51.
- Datové položky můžete spočítat v rámci datového balíku [“Počítání datových položek”](#) na stránce 53.
- Informace můžete změnit v rámci datového balíku [“Změna informací v rámci balíku”](#) na stránce 51.
- Datovou datovou sadu [“Vymazání kabelky pomocí volání mqClearBag”](#) na stránce 52 můžete vymazat.
- Můžete oříznout datový balík [“Oříznutí balíku pomocí volání mqTruncateBag”](#) na stránce 53.
- Je možné převádět vaky a vyrovnávací paměti [“Převod balíků a vyrovnávacích pamětí”](#) na stránce 53.

Vytváření a odstraňování datových balíků

Vytvoření datových sáčků

Chcete-li použít rozhraní MQAI, vytvořte nejprve datový balík pomocí volání funkce mqCreateBag. Jako vstup pro toto volání dodáváte jednu nebo více voleb pro řízení vytvoření balíku.

Parametr *Options* volání MQCreateBag umožňuje zvolit, zda se má vytvořit uživatelská kabelka, balík příkazů, balík skupiny nebo balík pro administraci.

Chcete-li vytvořit tašku pro uživatele, balík příkazů nebo balík skupiny, můžete vybrat jednu nebo více dalších voleb:

- Použijte formulář seznamu, když existují dva nebo více sousedících výskytů stejného selektoru v balíku.
- Změňte pořadí datových položek tak, jak jsou přidány do zprávy PCF, abyste se ujistili, že se parametry nacházejí ve správném pořadí. Další informace o datových položkách viz [“Datová položka”](#) na stránce 49.
- Zkontrolujte hodnoty selektorů uživatele pro položky, které jste přidali do balíku.

Administrativní tašky automaticky implikují tyto možnosti.

Datový balík je identifikován svým popisovačem. Popisovač vaku je vrácen z balíku mqCreatea musí být dodán na všech ostatních voláních, které používají datový balík.

Úplný popis volání objektu mqCreateBag naleznete v dokumentu [mqCreateBag](#).

Odstranění datových sáčků

Jakýkoli datový balík vytvořený uživatelem musí být také odstraněn pomocí volání funkce mqDeleteBag. Je-li např. v uživatelském kódu vytvořen balík, musí být také odstraněn v kódu uživatele.

Systémové váčky se vytvářejí a odstraňují automaticky pomocí rozhraní MQAI. Další informace o tomto tématu viz [“Odeslání příkazů administrace na příkazový server pomocí volání mqExecute”](#) na stránce 55. Uživatelský kód nemůže odstranit systémovou vak.

Úplný popis volání objektu mqDeleteBag naleznete v dokumentu [mqDeleteBag](#).

Vkládání a přijímání datových pytlů

Data lze také posílat mezi aplikacemi pomocí vkládání a získávání datových pytlů pomocí volání mqPutBag a mqGetBag. To umožňuje aplikaci MQAI raději zacházet s vyrovnávací pamětí než s aplikací. Volání mqPutBag převádí obsah uvedeného balíku do zprávy PCF a odesílá zprávu do zadané fronty a volání mqGetBag odebere zprávu z určené fronty a převede je zpět na datový balík. Volání mqPutBag je tedy ekvivalentem volání mqBagToBuffer následovaného operací MQPUT a objekt Bag mqGetje ekvivalentem volání MQGET následovaného parametrem mqBufferToBag.

Další informace o odesílání a přijímání zpráv PCF ve specifické frontě naleznete v tématu [“Odesílání a příjem zpráv PCF v určené frontě”](#) na stránce 12 .

Poznámka: Pokud se rozhodnete použít volání balíku mqGet, musí být podrobnosti o příkazu PCF v rámci zprávy správné; nejsou-li uvedeny, jsou vráceny příslušné chybové výsledky a zpráva PCF se nevrátí.

Datová položka

Datové položky se používají k naplnění datových balíků při jejich vytvoření. Tyto datové položky mohou být uživatelské nebo systémové položky.

Tyto uživatelské položky obsahují uživatelská data, jako například atributy objektů, které jsou spravovány. Systémové položky by měly být použity pro větší kontrolu nad generovanými zprávami: například generování záhlaví zpráv. Další informace o systémových položkách najdete v tématu [“Systémové položky” na stránce 49.](#)

Typy datových položek

Když jste vytvořili datový balík, můžete jej naplnit celým nebo znakovým řetězcovým položkám. Můžete se dotázat na všechny tři typy položek.

Datová položka může být buď celé číslo, nebo položky ve znakovém řetězci. Zde jsou uvedeny typy datové položky dostupné v rámci rozhraní MQAI:

- Celé číslo
- 64bitové celé číslo
- Filtr celých čísel
- Znakový řetězec
- Filtr řetězce
- Bajtový řetězec
- Filtr bajtového řetězce
- Držák sady

Použití datových položek

Jsou to tyto způsoby použití datových položek:

- [“Počítání datových položek” na stránce 53.](#)
- [“Odstranění datových položek” na stránce 54.](#)
- [“Přidání datových položek do balíků” na stránce 49.](#)
- [“Filtrování a dotazování datových položek” na stránce 50.](#)

Systémové položky

Systémové položky lze použít pro:

- Generování záhlaví PCF. Systémové položky mohou řídit identifikátor příkazu PCF, řídicí volby, pořadové číslo zprávy a typ příkazu.
- Převod dat. Systémové položky obsluhují identifikátor znakové sady pro položky znakového řetězce v balíku.

Podobně jako všechny datové položky se systémové položky skládají z selektoru a hodnoty. Informace o těchto selektorech a o tom, co jsou určeny, naleznete v tématu [Selektory MQAI.](#)

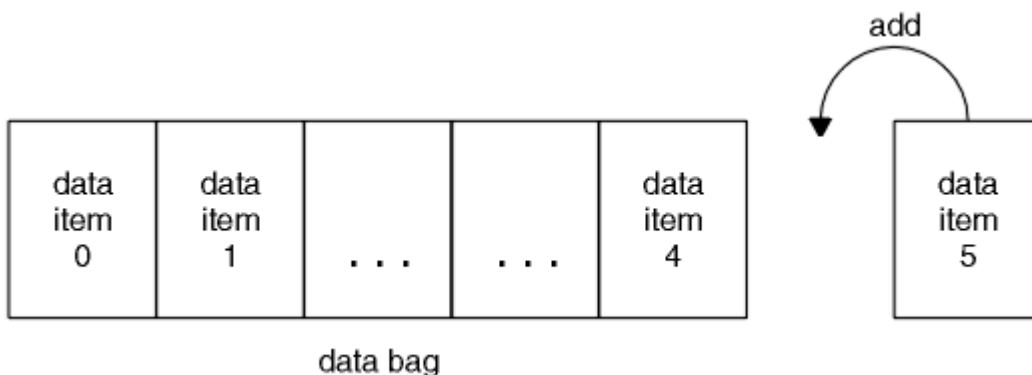
Systémové položky jsou jedinečné. Jedna nebo více systémových položek může být identifikováno selektorem systému. Existuje pouze jeden výskyt každého selektoru systému.

Většinu systémových položek lze upravit (viz [“Změna informací v rámci balíku” na stránce 51](#)), ale uživatel nemůže změnit volby vytvoření balíku. Systémové položky nelze odstranit. (Viz [“Odstranění datových položek” na stránce 54.](#))

Přidání datových položek do balíků

Když je datový balík vytvořen, můžete ho naplnit datovými položkami. Tyto datové položky mohou být uživatelské nebo systémové položky. Další informace o datových položkách viz [“Datová položka”](#) na stránce 49.

Rozhraní MQAI vám umožňuje přidávat celočíselné položky, 64bitové celočíselné položky, celočíselné položky filtru, položky znakového řetězce, filtr řetězce, položky řetězce bajtů a položky filtru bajtového řetězce do pytlů a to se zobrazí v produktu **Obrázek 3** na stránce 50. Položky jsou identifikovány selektorem. Jeden selektor obvykle identifikuje pouze jednu položku, ale to není vždy případ. Je-li datová položka se zadaným selektorem již v balíku přítomna, přidá se další instance tohoto selektoru na konec balíku.



Obrázek 3. Přidání datových položek

Přidejte datové položky do balíku pomocí volání `mqAdd*`:

- Chcete-li přidat celočíselné položky, použijte celočíselné volání `mqAdd`, jak je popsáno v části [mqAddInteger](#).
- Chcete-li přidat 64bitové celočíselné položky, použijte volání `mqAddInteger64`, jak je popsáno v souboru [mqAddInteger64](#).
- Chcete-li přidat celočíselné položky filtru, použijte volání `mqAddIntegerFilter`, jak je popsáno v [mqAddIntegerFilter](#).
- Chcete-li přidat položky znak-řetězec, použijte volání řetězce `mqAdd`, jak je popsáno v části [mqAddString](#).
- Chcete-li přidat položky filtru řetězce, použijte volání `mqAddStringFilter`, jak je popsáno v souboru [mqAddStringFilter](#).
- Chcete-li přidat bajtové řetězcové položky, použijte volání `mqAddByteString` podle popisu v části [mqAddByteString](#).
- Chcete-li přidat položky filtru bajtového řetězce, použijte volání filtru `mqAddByteStringFilter`, jak je popsáno v [mqAddByteStringFilter](#).

Další informace o přidávání datových položek do balíku viz [“Systémové položky”](#) na stránce 49.

Přidání dotazového příkazu do balíku

Dotazovací volání `mqAddse` používá k přidání dotazového příkazu do balíku. Volání je konkrétně pro účely administrace, takže je lze použít pouze s balíky administrace. Umožňuje určit selektory atributů, na které se chcete dotázat z produktu WebSphere MQ.

Úplný popis dotazu `mqAddInquiry` naleznete v tématu [mqAddInquiry](#).

Filtrování a dotazování datových položek

Při použití rozhraní MQAI k zjišťování informací o attributech objektů produktu WebSphere MQ můžete řídit data, která se vrátí do programu, a to dvěma způsoby.

- Můžete **filtrovat** data, která jsou vrácena pomocí polí `mqAddInteger` a `mqAddString`. Tento přístup umožňuje zadat dvojici *Selector* a *ItemValue*, například:

```
mqAddInteger(inputbag, MQIA_Q_TYPE, MQQT_LOCAL)
```

Tento příklad určuje, že typ fronty (*Selector*) musí být lokální (*ItemValue*) a tato specifikace musí odpovídat atributům objektu (v tomto případě fronty), o které se chcete dotazovat.

Ostatní atributy, které lze filtrovat, odpovídají příkazům PCF Inquire *, které lze nalézt v produktu “Úvod do formátu programových příkazů” na stránce 9. Chcete-li se například dotázat na atributy kanálu, podívejte se na příkaz Inquire Channel v této dokumentaci k produktu. Parametry "Required parameters" a "Optional parameters" příkazu Inquire Channel identifikují selektory, které můžete použít k filtrování.

- Můžete **dotaz** konkrétní atributy objektu pomocí volání dotazu `mqAdd`. Tento parametr určuje selektor, o který se zajímáte. Pokud neuvedete selektor, vrátí se všechny atributy objektu.

Zde je příklad filtrování a dotazování na atributy fronty:

```
/* Request information about all queues */
mqAddString(adminbag, MQCA_Q_NAME, "*")

/* Filter attributes so that local queues only are returned */
mqAddInteger(adminbag, MQIA_Q_TYPE, MQQT_LOCAL)

/* Query the names and current depths of the local queues */
mqAddInquiry(adminbag, MQCA_Q_NAME)
mqAddInquiry(adminbag, MQIA_CURRENT_Q_DEPTH)

/* Send inquiry to the command server and wait for reply */
mqExecute(MQCMD_INQUIRE_Q, ...)
```

Další příklady filtrování a dotazování datových položek viz “Příklady použití rozhraní MQAI” na stránce 20.

Inkviování v rámci datových sáčků

Můžete se dotázat na:

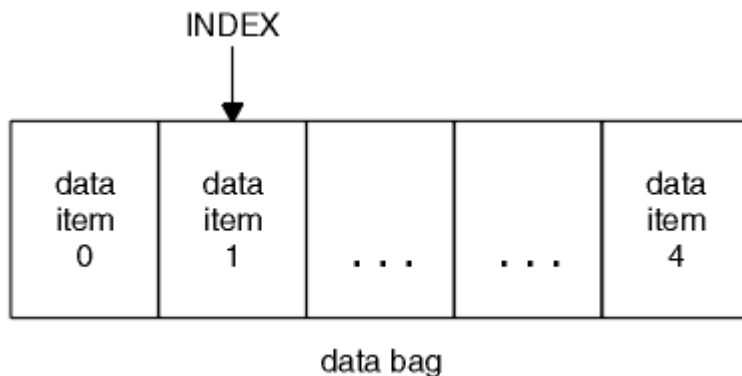
- Hodnota celočíselné položky pomocí celočíselného volání `mqInquire`. Viz [mqInquireInteger](#).
- Hodnota 64-bitové celočíselné položky pomocí volání `mqInquireInteger64`. Viz [mqInquireInteger64](#).
- Hodnota celočíselné položky filtru pomocí volání `mqInquireIntegerFilter`. Viz [mqInquireIntegerFilter](#).
- Hodnota znakové znakové položky pomocí volání řetězce `mqInquire`. Viz řetězec [mqInquireString](#).
- Hodnota položky filtru řetězce pomocí volání `mqInquireStringFilter`. Viz [mqInquireStringFilter](#).
- Hodnota bajtové řetězcové položky pomocí volání `mqInquireByteString`. Viz [mqInquireByteString](#).
- Hodnota položky filtru bajtového řetězce pomocí volání filtru `mqInquireByteString`. Viz filtr [mqInquireByteString](#).
- Hodnota popisovače balíku pomocí volání `mqInquireBag`. Viz [mqInquireBag](#).

Můžete se také dotázat na typ (celé číslo, 64bitové, celočíselný filtr, znakový řetězec, filtr řetězce, řetězec bajtů, filtr řetězců bajtů nebo popisovač balíku) pro konkrétní položku pomocí volání `mqInquireItemInfo`. Viz [mqInquireItemInfo](#).

Změna informací v rámci balíku

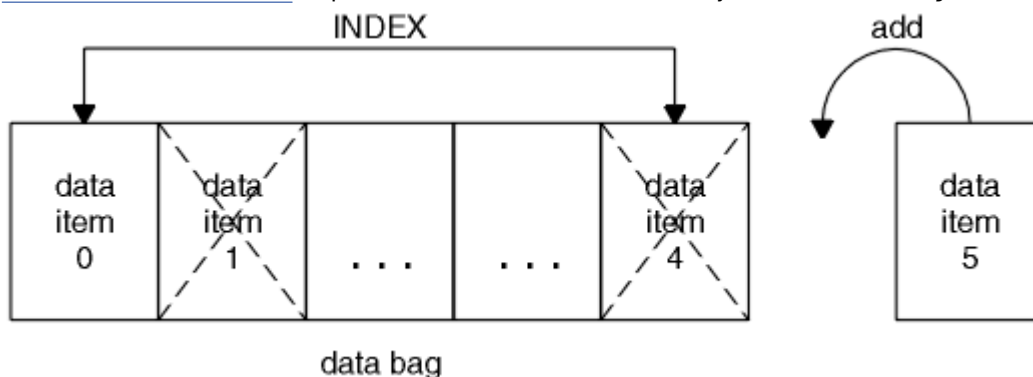
Rozhraní MQAI umožňuje měnit informace v rámci balíku pomocí volání `mqSet*`. Můžete provést následující akce:

1. Upravit datové položky v rámci balíku. Index umožňuje nahrazení jednotlivé instance parametru tím, že identifikuje výskyt položky, která má být upravena (viz [Obrázek 4](#) na stránce 52).



Obrázek 4. Úprava jedné datové položky

2. Odstraňte všechny existující výskyty uvedeného selektoru a přidejte nový výskyt na konec balíku. (Viz Obrázek 5 na stránce 52.) Speciální hodnota indexu umožňuje nahradit **všechny** instance parametru.



Obrázek 5. Úprava všech datových položek

Poznámka: Index zachová pořadí vložení v rámci balíku, ale může ovlivnit indexy ostatních datových položek.

Celočíselný hovor `mqSet` umožňuje upravit celočíselné položky v rámci balíku. Volání `mqSetInteger64` vám umožňuje upravit 64-bitové celočíselné položky. Volání `mqSetIntegerFilter` vám umožňuje upravit celočíselné položky filtru. Volání `mqSetString` vám umožňuje upravit položky ve znakovém řetězci. Volání `mqSetStringFilter` vám umožňuje upravit položky filtru řetězce. Volání `mqSetByteString` umožňuje upravit položky řetězce bajtů. Volání filtru `mqSetByteString` umožňuje upravit položky filtru bajtových řetězců. Alternativně můžete použít tato volání k odstranění všech existujících výskytů uvedeného selektoru a přidat nový výskyt na konci balíku. Datová položka může být uživatelská položka nebo systémová položka.

Úplný popis těchto volání naleznete v následujících tématech:

- [mqSetCelé číslo](#)
- [mqSetInteger64](#)
- [mqSetIntegerFilter](#)
- [mqSetŘetězec](#)
- [mqSetStringFilter](#)
- [mqSetByteString](#)
- [mqSetByteStringFiltr](#)

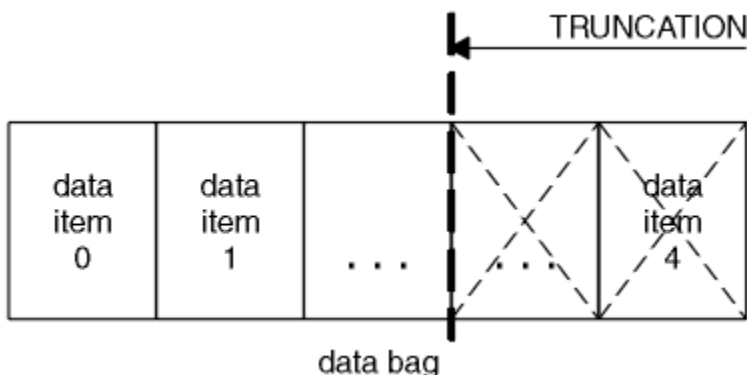
Vymazání kabelky pomocí volání `mqClearBag`

Volání `mqClearBag` odebere všechny uživatelské položky z uživatelské tašky a resetuje systémové položky na jejich počáteční hodnoty. Systémové balíky obsažené v balíku se také odstraní.

Úplný popis volání objektu `mqClearBag` naleznete v dokumentu [mqClearBag](#).

Oříznutí balíku pomocí volání `mqTruncateBag`

Volání `mqTruncateBag` snižuje počet uživatelských položek v uživatelské kabelce odstraněním položek z konce balíku, počínaje nejnověji přidanou položkou. Lze ji například použít při použití stejných informací záhlaví ke generování více než jedné zprávy.

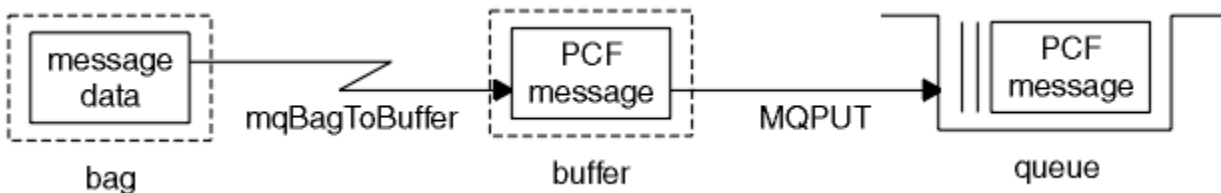


Obrázek 6. Oříznutí balíku

Úplný popis volání balíku `mqTruncateBag` naleznete v části [mqTruncateBag](#).

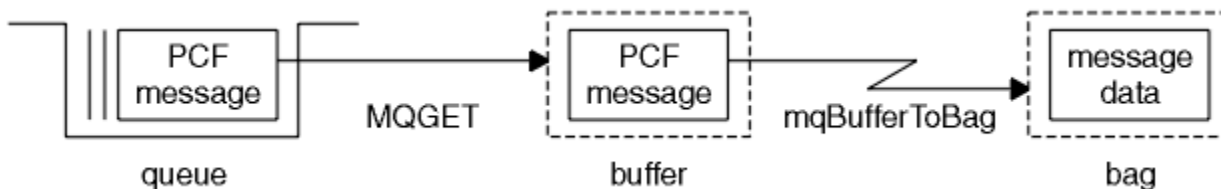
Převod balíků a vyrovnávacích pamětí

Chcete-li odeslat data mezi aplikacemi, za prvé, data zprávy se umístí do tašky. Poté jsou data v balíku převedena na zprávu PCF pomocí volání `mqBagToBuffer`. Zpráva PCF se odešle do požadované fronty pomocí volání `MQPUT`. To je zobrazeno na obrázku [Obrázek 7](#) na stránce 53. Úplný popis volání `mqBagToBuffer` pro `mqBag` naleznete v souboru `mqBagToBuffer`.



Obrázek 7. Převod kabelek do zpráv PCF

Chcete-li přijímat data, je zpráva přijata do vyrovnávací paměti pomocí volání `MQGET`. Data ve vyrovnávací paměti se pak převedou do tašky pomocí volání `mqBufferToBag` a poskytuje vyrovnávací paměť tak, že obsahuje platnou zprávu PCF. Tato hodnota je zobrazena na obrázku [Obrázek 8](#) na stránce 53. Úplný popis volání `mqBufferToBag` naleznete v části `mqBufferToBag`.



Obrázek 8. Převod zpráv PCF do formátu balíku

Počítání datových položek

Volání položek `mqCount` počítá počet položek uživatele, systémových položek nebo obojího, které jsou uloženy v datovém balíku, a vrátí toto číslo. Například, `mqCountItems(Bag, 7, ...)` vrátí počet položek v balíku se selektorem 7. Může počítat položky podle jednotlivých selektorů, selektorů uživatele, podle systémových selektorů nebo podle všech selektorů.

Poznámka: Toto volání počítá počet datových položek, nikoli počet jedinečných selektorů v balíku. Selektor se může vyskytnout vícekrát, takže v balíku může být méně jedinečných selektorů než datových položek.

Úplný popis volání položek mqCountnaleznete v části [mqCount](#) položek.

Odstranění datových položek

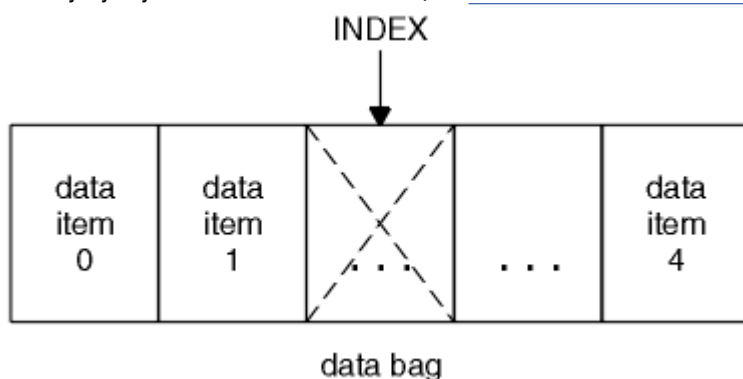
Položky z pytlů můžete odstranit mnoha způsoby. Můžete provést následující akce:

- Odebrat jednu nebo více uživatelských položek z balíku. Podrobnější informace viz téma [“Odstranění datových položek z balíku pomocí volání položky mqDelete”](#) na stránce 54.
- Vymažte **všechny** uživatelské položky z balíku, tj. *vymazat* kabelku. Podrobné informace viz [“Vymazání kabelky pomocí volání mqClearBag”](#) na stránce 52.
- Odstraňte uživatelské položky z konce balíku, tj. *oříznout* balík. Podrobnější informace viz téma [“Oříznutí balíku pomocí volání mqTruncateBag”](#) na stránce 53.

Odstranění datových položek z balíku pomocí volání položky mqDelete

Volání položky mqDeleteodebere jednu nebo více uživatelských položek z balíku. Index se používá k odstranění:

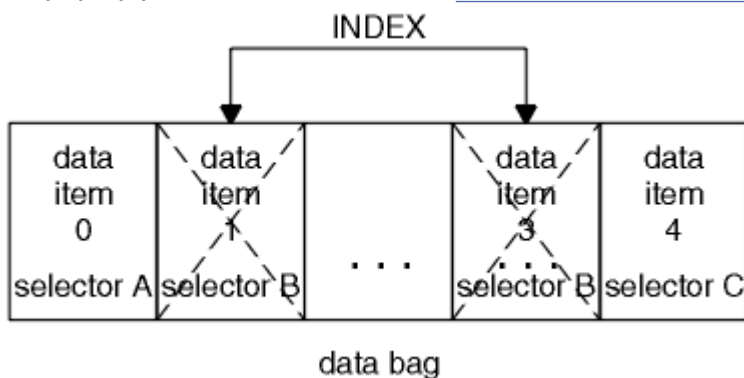
1. Jediný výskyt zadaného selektoru. (Viz [Obrázek 9](#) na stránce 54.)



Obrázek 9. Odstranění jedné datové položky

, nebo

2. Všechny výskyty určeného selektoru. (Viz [Obrázek 10](#) na stránce 54.)



Obrázek 10. Odstranění všech datových položek

Poznámka: Index zachová pořadí vložení v rámci balíku, ale může ovlivnit indexy ostatních datových položek. Například volání položky mqDelete zachová hodnoty indexu datových položek, které následují za odstraněnou položkou, protože indexy jsou reorganizovány tak, aby vyplnily mezeru, která zůstává z odstraněné položky.

Úplný popis volání položky mqDelete naleznete v tématu [mqDeleteItem](#).

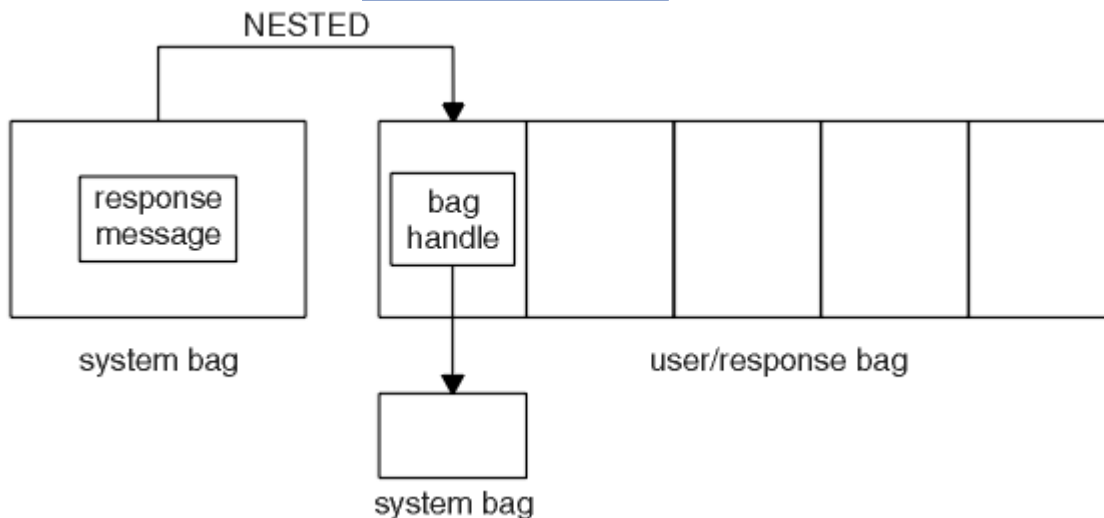
Odeslání příkazů administrace na příkazový server pomocí volání mqExecute

Po vytvoření a naplnění datového balíku lze pomocí volání mqExecute odeslat na příkazový server správce front zprávu administrativního příkazu. To obslouží výměnu s příkazovým serverem a vrací odpovědi v balíku.

Po vytvoření a naplnění datového balíku dat můžete odeslat zprávu příkazu administrace na příkazový server správce front. Nejjednodušším způsobem, jak to provést, je volání pomocí volání mqExecute . Volání mqExecute odešle zprávu příkazu administrace jako přechodnou zprávu a čeká na všechny odpovědi. Odpovědi jsou vráceny v balíku odpovědí. Tyto informace mohou obsahovat informace o atributech týkajících se několika objektů produktu WebSphere MQ nebo například řady zpráv odpovědí na chybu příkazu PCF, například. Proto může balík odezvy obsahovat pouze návratový kód, nebo může obsahovat *vnořené pytle*.

Zprávy odpovědi jsou umístěny do systémových pytlů vytvořených systémem. Například pro dotazy týkající se názvů objektů se vytvoří systémový balík, který bude obsahovat názvy objektů, a do uživatelského balíku se vloží taška. Manipulátory s těmito sáčky se pak vloží do pytle s odezvou a do vnořené balíku lze přistupovat pomocí selektoru MQHA_BAG_HANDLE. Systémový balík zůstane v úložišti, pokud nebude odstraněn, dokud nebude balík odpovědí odstraněn.

Koncept *vnoření* je zobrazen v [Obrázek 11 na stránce 55](#).



Obrázek 11. Vnoření

Jako vstup do volání mqExecute je třeba zadat:

- Popisovač připojení MQI.
- Příkaz, který má být proveden. Mělo by se jednat o jednu z hodnot MQCMD_ *.

Poznámka: Pokud není tato hodnota rozpoznána rozhraním MQAI, je hodnota stále akceptována. Pokud však bylo k vložení hodnot do balíku použity dotazové volání mqAdd, tento parametr musí být příkaz INQUIRE rozpoznán rozhraním MQAI. To znamená, že parametr by měl být ve tvaru MQCMD_INQUIRE_ *.

- Volitelně lze použít popisovač balíku obsahující volby, které řídí zpracování volání. Také zde můžete uvést maximální dobu v milisekundách, po kterou má rozhraní MQAI čekat na každou zprávu odpovědi.
- Popisovač administračního balíku, který obsahuje podrobnosti o příkazu správy, který má být vydán.
- Popisovač pro soubor odpovědí, který přijímá zprávy odpovědi.

Následující volby jsou volitelné:

- Popisovač objektu fronty, do které má být umístěn administrační příkaz.

Není-li zadán žádný popisovač objektu, je příkaz administrace umístěn na SYSTEM.ADMIN.COMMAND.QUEUE náležící k aktuálně připojenému správci front. Toto nastavení je výchozí.

- Manipulátor objektu fronty, do níž mají být umístěny zprávy odpovědí.

Můžete se rozhodnout umístit zprávy odpovědi do dynamické fronty, která je vytvořena automaticky rozhraním MQAI. Vytvořená fronta existuje pouze po dobu trvání volání a je odstraněna rozhraním MQAI při ukončení volání mqExecute .

Příklady použití volání mqExecute viz [Příklad kódu](#) .

Administrace pomocí produktu IBM WebSphere MQ Explorer

Produkt IBM WebSphere MQ Explorer umožňuje provádět lokální nebo vzdálenou administraci vaší sítě z počítače s operačním systémem Windowsnebo Linux (pouze platformyx86 a x86-64) .

IBM WebSphere MQ for Windowsa IBM WebSphere MQ for Linux (platformyx86 a x86-64) poskytují administrativní rozhraní s názvem IBM WebSphere MQ Explorer , aby bylo možné provádět administrativní úlohy jako alternativu k použití řídicích příkazů nebo příkazů MQSC. [Porovnání sad příkazů](#) zobrazuje, co můžete provádět pomocí produktu IBM WebSphere MQ Explorer.

Produkt IBM WebSphere MQ Explorer umožňuje provádět lokální nebo vzdálenou administraci vaší sítě z počítače s operačním systémem Windowsnebo Linux (platformyx86-64) tím, že na správce front a klastry, o které se zajímáte, nasměrujete na IBM WebSphere MQ Explorer . Platformy a úrovně produktu IBM WebSphere MQ , které lze spravovat pomocí produktu IBM WebSphere MQ Explorer , jsou popsány v části [“Vzdálení správci front”](#) na stránce 57.

Chcete-li nakonfigurovat vzdálené správce front produktu IBM WebSphere MQ tak, aby je produkt IBM WebSphere MQ Explorer mohl spravovat, přečtěte si téma [“Předem vyžadovaný software a definice”](#) na stránce 58.

Umožňuje provádět úlohy, které jsou zpravidla přidruženy k nastavení a dokončení pracovního prostředí produktu IBM WebSphere MQlokálně nebo vzdáleně v rámci domény systému Windows nebo Linux (platformyx86 a x86-64) .

V systému Linuxse může produkt IBM WebSphere MQ Explorer spustit, pokud máte více než jednu instalaci platformy Eclipse . Pokud k tomu dojde, spusťte produkt IBM WebSphere MQ Explorer pomocí jiného ID uživatele, který používáte pro další instalaci platformy Eclipse .

Chcete-li produkt IBM WebSphere MQ Explorer spustit v produktu Linuxúspěšně, musíte být schopni zapisovat soubor do svého domovského adresáře a domovský adresář musí existovat.

Co lze dělat s IBM WebSphere MQ Explorer

Jedná se o seznam úloh, které můžete provádět pomocí produktu IBM WebSphere MQ Explorer.

Pomocí Průzkumníka IBM WebSphere MQ můžete:

- Vytvořte a odstraňte správce front (pouze v lokálním počítači).
- Spuštění a zastavení správce front (pouze v lokálním počítači).
- Definujte, zobrazte a pozměňte definice objektů produktu WebSphere MQ , jako jsou například fronty a kanály.
- Procházejte zprávy ve frontě.
- Spuštění a zastavení kanálu.
- Zobrazení informací o stavu kanálu, modulu listener, fronty nebo objektů služeb.
- Zobrazit správce front v klastru.
- Zkontrolujte, které aplikace, uživatelé nebo kanály mají otevřenou konkrétní frontu.
- Vytvořte nový klastr správce front pomocí průvodce *Vytvořit nový klastr* .
- Přidejte správce front do klastru pomocí průvodce *Přidat správce front do klastru* .

- Správa objektu ověřovacích informací, který se používá se zabezpečením kanálu SSL (Secure Sockets Layer).
- Vytvořit a odstranit iniciátory kanálu, monitory spouštěčů a listenery.
- Spusťte nebo zastavte příkazové servery, iniciátory kanálu, monitory spouštěčů a moduly listener.
- Nastavte specifické služby, které mají být spuštěny automaticky při spuštění správce front.
- Upravte vlastnosti správců front.
- Změňte lokálního výchozího správce front.
- Vyvolejte grafické rozhraní ikeyman pro správu certifikátů SSL (secure sockets layer), přidruzte certifikáty ke správcům front a konfigurovat a nastavit úložiště certifikátů (pouze na vašem lokálním počítači).
- Vytvořte objekty JMS z objektů WebSphere MQ a objekty WebSphere MQ z objektů JMS.
- Vytvořte továrnu na připojení rozhraní JMS pro kterýkoli z aktuálně podporovaných typů.
- Upravte parametry pro libovolnou službu, jako je číslo portu TCP pro modul listener, nebo název fronty inicializátoru kanálu.
- Spusťte nebo zastavte trasování služby.

Administrační úlohy provádíte pomocí řady *Dialogů obsahu* a *Dialogová okna vlastností*.

Pohled Obsah

Pohled Obsah je panel, který může zobrazit následující:

- Atributy a administrativní volby týkající se samotného produktu WebSphere MQ .
- Atributy a administrativní volby související s jedním nebo více souvisejícími objekty.
- Atributy a administrativní volby pro klastr.

dialogová okna vlastností

Dialogové okno vlastností je panel, který zobrazuje atributy týkající se objektu v řadě polí, z nichž některé můžete upravovat.

You navigate through the WebSphere MQ Explorer using the *Pohled produktu Navigator*. Modul Navigator vám umožňuje vybrat zobrazení obsahu, které požadujete.

Vzdálení správci front

Pro podporované správce front, ke kterým se můžete připojit, existují dvě výjimky.

V systému Windows nebo Linux (platformy x86 a x86-64) se produkt WebSphere MQ Explorer může připojit ke všem podporovaným správcům front s následujícími výjimkami:

- WebSphere MQ pro správce front z/OS starší než verze 6.0.
- Aktuálně jsou podporovány správci front MQSeries V2 .

Průzkumník IBM WebSphere MQ zpracovává rozdíly ve schopnostech mezi různými úrovněmi příkazů a platformami. Pokud však narazí na atribut, který nerozpoznal, atribut nebude viditelný.

Hodláte-li vzdáleně spravovat správce front V6.0 nebo novější v systému Windows pomocí Průzkumníka IBM WebSphere MQ na počítači WebSphere MQ , musíte nainstalovat opravnou sadu Fix Pack 9 (CSD9) nebo novější na počítač WebSphere MQ for Windows V5.3 .

Hodláte-li vzdáleně spravovat správce front V5.3 v systému iSeries pomocí produktu WebSphere MQ Explorer na počítači WebSphere MQ V6.0 nebo novější, Musíte nainstalovat opravnou sadu 11 (CSD11) nebo novější na počítač WebSphere MQ for iSeries V5.3 . Tato opravná sada opravuje problémy s připojením mezi produktem WebSphere MQ Explorer a správcem front iSeries .

Rozhodování o použití produktu IBM WebSphere MQ Explorer

Při rozhodování o tom, zda má být při instalaci použit produkt IBM WebSphere MQ Explorer , zvažte informace uvedené v tomto tématu.

Musíte si být vědomi následujících bodů:

názvy objektů

Pokud použijete malá písmena pro správce front a jiné objekty pomocí Průzkumníka IBM WebSphere MQ, musíte při práci s objekty pomocí příkazů MQSC uzavřít názvy objektů do jednoduchých uvozovek nebo produkt WebSphere MQ je nerozpoznal.

Správci velkých front

Průzkumník IBM WebSphere MQ funguje nejlépe s malými správci front. Pokud máte na jednom správci front velký počet objektů, může dojít k prodlevám, zatímco produkt WebSphere MQ Explorer extrahuje požadované informace do zobrazení.

Klastry

Klastry WebSphere MQ mohou potenciálně obsahovat stovky nebo tisíce správců front. Průzkumník produktu WebSphere MQ prezentuje správce front v klastru pomocí stromové struktury. Fyzická velikost klastru neovlivňuje rychlost produktu IBM WebSphere MQ Explorer dramaticky, protože Průzkumník IBM WebSphere MQ se nepřipojuje ke správcům front v klastru, dokud je nevyberete.

Nastavení produktu IBM WebSphere MQ Explorer

Tento oddíl popisuje kroky, které je třeba provést při nastavení produktu IBM WebSphere MQ Explorer.

- [“Předem vyžadovaný software a definice” na stránce 58](#)
- [“Zabezpečení” na stránce 58](#)
- [“Zobrazení a skrytí správců front a klastrů” na stránce 62](#)
- [“Členství v klastru” na stránce 63](#)
- [“Převod dat” na stránce 63](#)

Předem vyžadovaný software a definice

Než se pokusíte použít produkt IBM WebSphere MQ Explorer, ujistěte se, že jste splnili následující požadavky.

Produkt IBM WebSphere MQ Explorer se může připojit ke vzdáleným správcům front pouze s použitím komunikačního protokolu TCP/IP.

Zkontrolujte, zda:

1. Příkazový server je spuštěn ve všech vzdáleně spravovaných správci front.
2. V každém vzdáleném správci front musí být spuštěn vhodný objekt modulu listener protokolu TCP/IP. Tento objekt může být modulem listener produktu IBM WebSphere MQ nebo, v systémech UNIX and Linux, démon inetd.
3. Kanál připojení serveru s výchozím názvem SYSTEM.ADMIN.SVRCONN existuje ve všech vzdálených správcích front.

Kanál můžete vytvořit pomocí následujícího příkazu MQSC:

```
DEFINE CHANNEL(SYSTEM.ADMIN.SVRCONN) CHLTYPE(SVRCONN)
```

Tento příkaz vytvoří základní definici kanálu. Chcete-li mít propracovanější definici (například nastavit zabezpečení), potřebujete další parametry. Další informace viz [DEFINE CHANNEL](#).

4. Fronta systému, SYSTEM.MQEXPLORER.REPLY.MODEL, musí existovat.

Zabezpečení

Používáte-li produkt WebSphere MQ v prostředí, ve kterém je důležité, abyste mohli řídit přístup uživatelů ke konkrétním objektům, může být třeba zvážit aspekty zabezpečení použití Průzkumníka IBM WebSphere MQ.

Oprávnění k používání produktu IBM WebSphere MQ Explorer

Kterýkoli uživatel může použít produkt IBM WebSphere MQ Explorer, ale k připojení, přístupu a správě správců front jsou vyžadována určitá oprávnění.

Chcete-li provádět lokální administrativní úlohy pomocí produktu WebSphere MQ Explorer, uživatel musí mít potřebné oprávnění k provádění administrativních úloh. Je-li uživatel členem skupiny mqm, má uživatel oprávnění provádět všechny lokální administrativní úlohy.

Chcete-li se připojit ke vzdálenému správci front a provádět vzdálené administrativní úlohy pomocí produktu WebSphere MQ Explorer, uživatel provádějící program WebSphere MQ Explorer se musí mít k dispozici následující oprávnění:

- Oprávnění CONNECT k objektu cílového správce front
- Oprávnění INQUIRE na cílovém objektu správce front
- Oprávnění DISPLAY k objektu cílového správce front
- Oprávnění INQUIRE k frontě, SYSTEM.MQEXPLORER.REPLY.MODEL
- Oprávnění DISPLAY do fronty, SYSTEM.MQEXPLORER.REPLY.MODEL
- VSTUP (get) oprávnění k frontě, SYSTEM.MQEXPLORER.REPLY.MODEL
- OUTPUT (put), oprávnění k frontě, SYSTEM.ADMIN.COMMAND.QUEUE
- Oprávnění INQUIRE k frontě, SYSTEM.ADMIN.COMMAND.QUEUE
- Oprávnění k provedení vybrané akce

Poznámka: Oprávnění INPUT se vztahuje na vstup uživatele z fronty (operace get). OUTPUT autorita se vztahuje k výstupu z uživatele do fronty (operace put).

Chcete-li se připojit ke vzdálenému správci front v produktu WebSphere MQ for z/OS a provádět vzdálené administrativní úlohy pomocí Průzkumníka IBM WebSphere MQ, je třeba zadat následující údaje:

- Profil RACF pro systémovou frontu SYSTEM.MQEXPLORER.REPLY.MODEL
- Profil RACF pro fronty, AMQ.MQEXPLORER.*

Navíc uživatel, který spouští produkt WebSphere MQ Explorer, je povinen mít následující oprávnění:

- RACFE UPDATE oprávnění k systémové frontě, SYSTEM.MQEXPLORER.REPLY.MODEL
- Oprávnění RACF UPDATE do front, AMQ.MQEXPLORER.*
- Oprávnění CONNECT k objektu cílového správce front
- Oprávnění k provedení vybrané akce
- Oprávnění READ pro všechny profily hlq.DISPLAY.object ve třídě MQCMDS

Další informace o tom, jak udělit oprávnění k objektům WebSphere MQ, viz Udělení přístupu k objektu produktu WebSphere MQ v systémech UNIX nebo Linux a v systému Windows.

Pokusí-li se uživatel o provedení operace, kterou nemají oprávnění k provedení, vyvolá cílový správce front procedury selhání autorizace a operace selže.

Výchozí filtr v Průzkumníku WebSphere MQ Explorer má zobrazit všechny objekty WebSphere MQ. Pokud uživatel nemá žádné objekty WebSphere MQ, ke kterým uživatel nemá oprávnění DISPLAY, dojde k vygenerování selhání autorizace. Pokud se zaznamenávají události oprávnění, omezte rozsah objektů, které se zobrazí, na objekty, ke kterým má uživatel oprávnění DISPLAY.

Zabezpečení pro připojení ke vzdáleným správcům front

Je třeba zabezpečit kanál mezi průzkumníkem IBM WebSphere MQ a každým vzdáleným správcem front.

Průzkumník IBM WebSphere MQ se připojuje ke vzdáleným správcům front jako klientská aplikace MQI. To znamená, že každý vzdálený správce front musí mít definici kanálu připojení serveru a vhodného modulu listener protokolu TCP/IP. Pokud nezabezpečujete kanál připojení k serveru, je možné, aby se zlomyslná aplikace připojovala ke stejnému kanálu připojení k serveru a získala přístup k objektům správce front s neomezeným oprávněním. Chcete-li zabezpečit kanál připojení k serveru, zadejte pro

atribut MCAUSER kanálu buď neprázdnou hodnotu, použijte záznamy ověření kanálu, nebo použijte uživatelskou proceduru pro zabezpečení zprávy.

Výchozí hodnota atributu MCAUSER je ID lokálního uživatele. Pokud jako atribut MCAUSER kanálu připojení serveru zadáte neprázdné jméno uživatele, budou všechny programy, které se připojují ke správci front s použitím tohoto kanálu, spuštěny s identitou uvedeného uživatele a mají stejnou úroveň oprávnění. To se nestane, pokud použijete záznamy ověření kanálu.

Použití ukončení zabezpečení pomocí produktu WebSphere MQ Explorer

Pomocí průzkumníka WebSphere MQ Explorer můžete určit výchozí proceduru zabezpečení a uživatelské procedury zabezpečení specifické pro správce front.

Můžete definovat výchozí proceduru zabezpečení, kterou lze použít pro všechna nová připojení klienta z Průzkumníka produktu WebSphere MQ . Tato výchozí uživatelská procedura může být přepsána při vytvoření připojení. Můžete také definovat proceduru zabezpečení pro jednoho správce front nebo sadu správců front, která bude účinná při vytvoření připojení. Uživatelské procedury můžete zadat pomocí Průzkumníka WebSphere MQ . Další informace najdete v nápovědě k produktu WebSphere MQ .

Použití produktu IBM WebSphere MQ Explorer k připojení ke vzdálenému správci front s použitím kanálů MQI povolených SSL

Produkt IBM WebSphere MQ Explorer se připojuje ke vzdáleným správcům front pomocí kanálu MQI. Chcete-li zabezpečit kanál MQI s použitím zabezpečení SSL, je třeba vytvořit kanál s použitím tabulky definic kanálů klienta.

Informace o způsobu vytvoření kanálu MQI s použitím tabulky definic kanálů klienta naleznete v tématu [Přehled klientů produktu IBM WebSphere MQ MQI](#).

Pokud jste kanál ustanovovali pomocí tabulky definic kanálů klienta, můžete produkt IBM WebSphere MQ Explorer použít k připojení ke vzdálenému správci front s použitím kanálu MQI s povoleným SSL, jak je popsáno v tématu [“Úlohy v systému, který je hostitelem vzdáleného správce front”](#) na stránce 60 a [“Úlohy na systému, který je hostitelem produktu IBM WebSphere MQ Explorer”](#) na stránce 60.

Úlohy v systému, který je hostitelem vzdáleného správce front

V systému, který je hostitelem vzdáleného správce front, proveďte následující úlohy:

1. Definujte připojení k serveru a dvojici připojení klienta a určete příslušnou hodnotu pro proměnnou `SSLCIPH` na připojení serveru na obou kanálech. Další informace o proměnné `SSLCIPH` naleznete v tématu [Ochrana kanálů pomocí SSL](#) .
2. Odešlete tabulku definic kanálů `AMQCLCHL.TAB` , která se nachází v adresáři `@ipcc` správce front, na systém, který je hostitelem produktu IBM WebSphere MQ Explorer.
3. Spusťte modul listener protokolu TCP/IP na určeném portu.
4. Umístěte jak CA tak osobní certifikáty SSL do adresáře SSL správce front:
 - `/var/mqm/qmgrs/+QMNAME+/SSL` pro systémy UNIX and Linux
 - `C:\Program Files\WebSphere MQ\qmgrs\+QMNAME+\SSL` pro systémy Windowskde `+QMNAME+` je token představující název správce front.
5. Vytvořte soubor databáze klíčů typu CMS s názvem `key.kdb` . Založit heslo do souboru buď kontrolou volby v grafickém rozhraní `iKeyman` , nebo pomocí volby `-stash` s příkazy `runmqckm` .
6. Přidejte certifikáty CA do databáze klíčů vytvořené v předchozím kroku.
7. Importujte osobní certifikát pro správce front do databáze klíčů.

Podrobnější informace o práci se zabezpečením SSL na systémech Windows naleznete v tématu [Práce s SSL nebo TLS v systémech UNIX, Linux a Windows](#) .

Úlohy na systému, který je hostitelem produktu IBM WebSphere MQ Explorer

Na systému, který je hostitelem produktu IBM WebSphere MQ Explorer, proveďte následující úlohy:

1. Vytvořte soubor databáze klíčů typu JKS s názvem `key.jks`. Nastaví heslo pro tento soubor databáze klíčů.

Produkt IBM WebSphere MQ Explorer používá soubory úložiště klíčů Java (JKS) pro zabezpečení SSL a soubor úložiště klíčů, který se vytváří pro konfiguraci SSL pro produkt IBM WebSphere MQ Explorer, se musí shodovat s tímto.

2. Přidejte certifikáty CA do databáze klíčů vytvořené v předchozím kroku.
3. Importujte osobní certifikát pro správce front do databáze klíčů.
4. V systémech Windows a Linux spusťte produkt MQ Explorer s použitím systémové nabídky, spustitelného souboru `MQExplorer` nebo příkazu **`strmqcfig`**.
5. Na panelu nástrojů produktu IBM WebSphere MQ Explorer klepněte na volbu **Okno-> Předvolby** a poté rozbalte položku **WebSphere MQ Explorer** a klepněte na volbu **Úložiště certifikátů klienta SSL**. Zadejte jméno a heslo souboru JKS vytvořeného v kroku 1 produktu “Úlohy na systému, který je hostitelem produktu IBM WebSphere MQ Explorer” na stránce 60 jak v úložišti důvěryhodných certifikátů, tak v úložišti osobních certifikátů, poté klepněte na tlačítko **OK**.
6. Zavřete okno **Předvolby** a klepněte pravým tlačítkem myši na položku **Správci front**. Klepněte na volbu **Zobrazit či skrýt správce fronta** poté klepněte na volbu **Přidat** na obrazovce **Zobrazit/skrýt správce front**.
7. Zadejte název správce front a vyberte volbu **Připojit přímo**. Klepněte na volbu Další.
8. Vyberte volbu **Použít tabulku CCDT (Client Channel Definition CCDT)** a zadejte umístění souboru tabulky kanálů, který jste přenesli ze vzdáleného správce front v kroku 2 v produktu “Úlohy v systému, který je hostitelem vzdáleného správce front” na stránce 60 na systému, který je hostitelem vzdáleného správce front.
9. Klepněte na tlačítko **Dokončit**. Nyní můžete přistupovat ke vzdálenému správci front z produktu IBM WebSphere MQ Explorer.

Připojení prostřednictvím jiného správce front

Průzkumník IBM WebSphere MQ vám umožňuje připojit se ke správci front prostřednictvím zprostředkujícího správce front, ke kterému je Průzkumník IBM WebSphere MQ již připojen.

V takovém případě Průzkumník IBM WebSphere MQ vloží zprávy příkazu PCF do intermediačního správce front zadáním následujícího příkazu:

- Parametr *ObjectQMgrName* v deskriptoru objektu (MQOD) jako název cílového správce front. Další informace o rozlišení názvů front naleznete v tématu Rozlišování názvů.
- Parametr *UserIdentifier* v deskriptoru zpráv (MQMD) jako lokální `userId`.

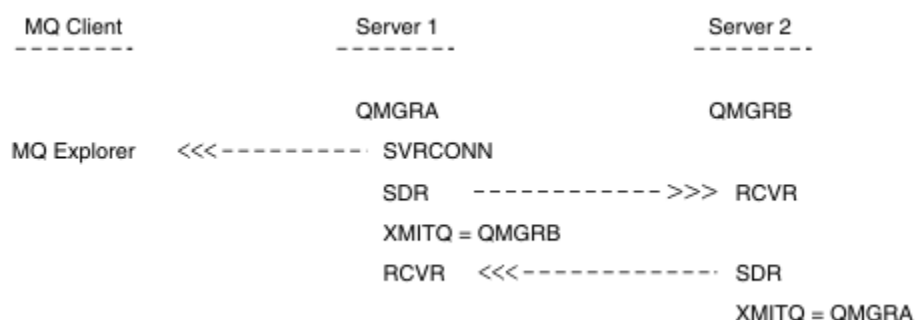
Je-li připojení používáno pro připojení k cílovému správci front prostřednictvím intermediačního správce front, je hodnota `userId` přenášena v parametru *UserIdentifier* deskriptoru zpráv (MQMD) znovu. Aby mohl modul listener MCA v cílovém správci front přijmout tuto zprávu, buď musí být nastaven atribut `MCAUSER`, nebo `userId` již musí existovat s oprávněním k vložení.

Příkazový server v cílovém správci front vkládá zprávy do přenosové fronty s uvedením atributu `userId` v parametru *UserIdentifier* v deskriptoru zpráv (MQMD). V případě úspěšného provedení této funkce musí prvek `userId` již existovat v cílovém správci front s oprávněním vkládání.

Následující příklad ukazuje, jak připojit správce front prostřednictvím zprostředkujícího správce front k Průzkumníku WebSphere MQ.

Vytvořte vzdálené administrační připojení ke správci front. Ověřte, že:

- Správce front na serveru je aktivní a má definován kanál `SVRCONN` (server-connection kanál).
- Modul listener je aktivní.
- Příkazový server je aktivní.
- `SYSTEM.MQ EXPLORER.REPLY.MODEL` fronta byla vytvořena a že máte dostatečné oprávnění.
- Moduly listener správce front, příkazové servery a odesílací kanály jsou spuštěny.



V tomto příkladu platí následující:

- Produkt IBM WebSphere MQ Explorer je připojen ke správci front QMGRA (spuštěn na serveru Server1) s použitím připojení klienta.
- Správce front QMGRB na serveru Server2 může být nyní připojen k produktu IBM WebSphere MQ Explorer prostřednictvím zprostředkujícího správce front (QMGRA).
- Při připojování k produktu QMGRB s produktem WebSphere MQ Explorer vyberte jako zprostředkující správce front položku QMGRA .

V této situaci neexistuje přímé připojení k produktu QMGRB z prostředí IBM WebSphere MQ Explorer; připojení k produktu QMGRB je prostřednictvím produktu QMGRA.

Správce front QMGRB na serveru Server2 je připojen k serveru QMGRA na serveru Server1 s použitím kanálů příjemce odesílatele. Kanál mezi QMGRA a QMGRB musí být nastaven tak, aby vzdálená administrace byla možná; viz [“Příprava kanálů a přenosových front pro vzdálenou administraci”](#) na stránce 106.

Zobrazení a skrytí správců front a klastrů

Průzkumník IBM WebSphere MQ může v daném okamžiku zobrazit více než jednoho správce front. Na panelu Zobrazit/skrýt správce front (který lze vybrat z nabídky pro uzel stromu správce front) můžete zvolit, zda chcete zobrazit informace o jiném (vzdáleném) počítači. Lokální správci front jsou zjišťovány automaticky.

Zobrazení vzdáleného správce front:

1. Klepněte pravým tlačítkem myši na uzel stromu Queue Managers a poté vyberte volbu *Zobrazit/skrýt správce front*
2. Klepněte na tlačítko **Přidat**. Zobrazí se panel Zobrazit či skrýt správce front.
3. Zadejte název vzdáleného správce front a název hostitele nebo adresu IP v poskytnutých polích.
Název hostitele nebo adresa IP se používají k vytvoření připojení klienta ke vzdálenému správci front s použitím výchozího kanálu pro připojení k serveru, SYSTEM.ADMIN.SVRCONN nebo kanál připojení serveru definovaný uživatelem.
4. Klepněte na tlačítko **Dokončit**.

Panel Zobrazit/skrýt správce front také zobrazí seznam všech viditelných správců front. Tento panel můžete použít ke skrytí správců front v navigačním pohledu.

Pokud Průzkumník IBM WebSphere MQ zobrazuje správce front, který je členem klastru, je tento klastr zjištěn a zobrazen automaticky.

Chcete-li exportovat seznam vzdálených správců front z tohoto panelu, postupujte takto:

1. Zavřete panel Zobrazit/skrýt správce front.
2. Klepněte pravým tlačítkem myši na horní uzel stromu **IBM WebSphere MQ** v navigačním podokně Průzkumníku WebSphere MQ a poté vyberte volbu **Exportovat nastavení Průzkumníka MQ** .
3. Klepněte na volbu **PrůzkumníkMQ > Nastavení produktu MQ Explorer** .
4. Vyberte volbu **Informace o připojení > Vzdálení správci front**.

5. Vyberte soubor, do kterého chcete uložit exportovaná nastavení.
6. Nakonec klepněte na tlačítko **Dokončit**, chcete-li exportovat informace o připojení vzdáleného správce front do uvedeného souboru.

Chcete-li importovat seznam vzdálených správců front, postupujte takto:

1. Klepněte pravým tlačítkem myši na horní uzel stromu **IBM WebSphere MQ** v podokně Navigace v Průzkumníku WebSphere MQ a poté vyberte volbu **Importovat nastavení Průzkumníka MQ**.
2. Klepněte na volbu **PrůzkumníkMQ > Nastavení produktu MQ Explorer**.
3. Klepněte na tlačítko **Procházeta** přejděte na cestu k souboru, který obsahuje informace o připojení ke vzdálenému správci front.
4. Klepněte na tlačítko **Otevřít**. Pokud soubor obsahuje seznam vzdálených správců front, je vybráno pole **Informace o připojení > Vzdálené správce front**.
5. Nakonec klepnutím na tlačítko **Dokončit** importujte informace o připojení vzdáleného správce front do produktu WebSphere MQ Explorer.

Členství v klastru

Produkt IBM WebSphere MQ Explorer vyžaduje informace o správcích front, kteří jsou členy klastru.

Je-li správce front členem klastru, bude uzel stromu klastru automaticky naplněn daty.

Pokud se správci front stanou členy klastrů, zatímco je spuštěn produkt IBM WebSphere MQ Explorer, musíte udržovat produkt IBM WebSphere MQ Explorer s daty administrace typu "až do-data" o klastrech tak, aby s nimi mohl efektivně komunikovat a zobrazovat správné informace o klastru, až budete o to požádáni. Aby bylo možné tuto akci provést, produkt WebSphere MQ Explorer vyžaduje následující informace:

- Název správce front úložiště
- Název připojení správce front úložiště v případě, že se nachází ve vzdáleném správci front

S touto informací může produkt WebSphere MQ Explorer provádět následující akce:

- K získání seznamu správců front v klastru použijte správce front úložiště.
- Správa správců front, kteří jsou členy klastru a jsou na podporovaných platformách a úrovních příkazů.

Administrace není možná, pokud:

- Zvolené úložiště se stane nedostupným. Průzkumník produktu WebSphere MQ se automaticky nepřepne na alternativní úložiště.
- Vybrané úložiště nelze kontaktovat prostřednictvím protokolu TCP/IP.
- Zvolené úložiště je spuštěno ve správci front, který je spuštěn na platformě a na úrovni příkazů, které nejsou podporovány produktem WebSphere MQ Explorer.

Členy klastru, které lze spravovat, mohou být lokální, nebo mohou být vzdálené, pokud je lze kontaktovat pomocí protokolu TCP/IP. Produkt IBM WebSphere MQ Explorer se připojuje k lokálním správcům front, kteří jsou členy klastru přímo, bez použití připojení klienta.

Převod dat

Produkt IBM WebSphere MQ Explorer pracuje s CCSID 1208 (UTF-8). To umožní produktu IBM WebSphere MQ Explorer zobrazit data ze vzdálených správců front správně. Bez ohledu na to, zda se připojit ke správci front přímo, nebo pomocí intermediačního správce front, vyžaduje Průzkumník IBM WebSphere MQ všechny příchozí zprávy, které mají být převedeny na CCSID 1208 (UTF-8).

Pokud se pokusíte navázat spojení mezi Průzkumníkem IBM WebSphere MQ a správcem front s identifikátorem CCSID, který produkt IBM WebSphere MQ Explorer nerozpozná, zobrazí se chybová zpráva.

Podporované konverze jsou popsány v části [Převod kódové stránky](#).

Zabezpečení v systému Windows

Průvodce přípravou produktu WebSphere MQ vytvoří speciální uživatelský účet, aby služba Windows mohla být sdílena procesy, které je potřebují používat.

Služba Windows je sdílena mezi klientskými procesy pro instalaci produktu IBM WebSphere MQ . Pro každou instalaci se vytvoří jedna služba. Každá služba má název `MQ_InstallationNamea` má zobrazovaný název IBM WebSphere MQ (`InstallationName`). Před Version 7.1, pouze s jednou instalací na serveru, byla služba Windows pojmenována `MQSeriesServices` se zobrazovaným názvem `IBM MQSeries`.

Vzhledem k tomu, že každá služba musí být sdílena mezi neinteraktivními a interaktivními relacemi přihlášení, musíte je spustit pod zvláštním uživatelským účtem. Pro všechny služby můžete použít jeden speciální uživatelský účet, nebo můžete vytvořit různé speciální uživatelské účty. Každý speciální uživatelský účet musí mít právo uživatele na "Přihlásit se jako služba", abyste získali další informace, viz ["Uživatelská práva požadovaná pro službu IBM WebSphere MQ Windows"](#) na stránce 65. Pokud ID uživatele nemá oprávnění ke spuštění služby, služba se nespustí a vrátí chybu v protokolu událostí systému Windows . Obvykle se spustí Průvodce přípravou produktu IBM WebSphere MQ a ID uživatele je správně nastaveno. Pokud jste však ID uživatele nakonfigurovali ručně, je možné, že budete mít problém, který je třeba vyřešit.

Když instalujete produkt IBM WebSphere MQ a poprvé spustíte průvodce přípravou produktu IBM WebSphere MQ , vytvoří lokální uživatelský účet pro službu s názvem `MUSR_MQADMIN` s požadovanými nastaveními a oprávněními, včetně volby "Logon as a service".

V případě následných instalací vytvoří průvodce přípravou produktu IBM WebSphere MQ uživatelský účet s názvem `MUSR_MQADMINx`, kde x je další dostupné číslo představující ID uživatele, které neexistuje. Heslo pro `MUSR_MQADMINx` se náhodně vygeneruje, když se vytvoří účet, a použije se ke konfiguraci přihlašovacího prostředí pro službu. Vygenerované heslo nevyprší.

Tento účet IBM WebSphere MQ není ovlivněn žádnými zásadami účtu, které jsou nastaveny na systému tak, aby požadovaly, aby byla hesla účtu změněna po určité době.

Heslo není známé mimo toto jednorázové zpracování a je uloženo pomocí operačního systému Windows v zabezpečené části registru.

Použití aktivního adresáře (pouze Windows)

V některých konfiguracích sítě, kde jsou uživatelské účty definovány na řadičích domény, které používají Active Directory, je lokální uživatelský účet, pod kterým je uživatelský účet IBM WebSphere MQ spuštěn, pravděpodobně nemá oprávnění, které potřebuje k dotazování na členství ve skupinách jiných uživatelských účtů domény. Průvodce přípravou produktu IBM WebSphere MQ identifikuje, zda se jedná o případ uskutečňující testy a dotazující se na otázky uživatele týkající se konfigurace sítě.

Pokud účet lokálního uživatele IBM WebSphere MQ , pod kterým je spuštěn, nemá požadované oprávnění, vyzve Průvodce přípravou produktu IBM WebSphere MQ uživatele k zadání podrobností o účtu uživatele domény s konkrétními uživatelskými právy. Uživatelská práva, která uživatelský účet domény vyžaduje, viz ["Uživatelská práva požadovaná pro službu IBM WebSphere MQ Windows"](#) na stránce 65. Jakmile uživatel zadá platné podrobnosti o účtu pro uživatelský účet domény do Průvodce přípravou produktu IBM WebSphere MQ , nakonfiguruje službu IBM WebSphere MQ Windows tak, aby se spouštěla pod novým účtem. Podrobnosti o účtu jsou zadrženy v zabezpečené části registru a uživatelé jej nemohou číst.

Když je služba spuštěna, služba Windows produktu IBM WebSphere MQ se spustí a zůstane spuštěná tak dlouho, jak je služba spuštěna. Administrátor produktu IBM WebSphere MQ , který se přihlásí na server po spuštění služby Windows , může pomocí produktu IBM WebSphere MQ Explorer spravovat správce front na serveru. Tím se připojí IBM WebSphere MQ Explorer k existujícímu procesu služby Windows . Tyto dvě akce vyžadují různé úrovně oprávnění, než budou moci pracovat:

- Proces spuštění vyžaduje oprávnění ke spuštění.
- Administrátor produktu IBM WebSphere MQ vyžaduje přístupové oprávnění.

Uživatelská práva požadovaná pro službu IBM WebSphere MQ Windows

Tabulka v tomto tématu uvádí práva uživatelů požadovaná pro lokální a doménový uživatelský účet, pod kterým je spuštěna služba Windows pro instalaci produktu IBM WebSphere MQ .

Přihlašovat se jako dávková úloha	Umožňuje spuštění služby IBM WebSphere MQ Windows pod tímto uživatelským účtem.
Přihlásit jako služba.	Umožňuje uživatelům nastavit službu IBM WebSphere MQ Windows tak, aby se přihlašuje pomocí konfigurovaného účtu.
Ukončit běh systému	Umožňuje službě IBM WebSphere MQ Windows restartovat server, je-li konfigurován tak, aby při obnově služby došlo k selhání.
Zvýšit kvóty	Požaduje se pro volání operačního systému <code>CreateProcessAsUser</code> .
Vystupovat jako část operačního systému	Požaduje se pro volání operačního systému <code>LogonUser</code> .
Obejít kontrolu procházení	Požaduje se pro volání operačního systému <code>LogonUser</code> .
Zaměnit úroveň procesu	Požaduje se pro volání operačního systému <code>LogonUser</code> .

Poznámka: Práva na ladění programů mohou být zapotřebí v prostředích, kde jsou spuštěny aplikace ASP a IIS.

Váš uživatelský účet domény musí mít tato uživatelská práva Windows nastavená jako účinná uživatelská práva, jak je uvedeno v aplikaci Lokální zásada zabezpečení. Pokud nejsou, nastavte je buď pomocí lokální aplikace zásad zabezpečení lokálně na serveru, nebo pomocí domény Aplikace zabezpečení domény v široké části.

Změna jména uživatele přidruženého ke službě IBM WebSphere MQ

Možná budete muset změnit jméno uživatele přidružené ke službě IBM WebSphere MQ z `MUSR_MQADMIN` na něco jiného. (Pokud je například správce front asociován s produktem DB2, který nepřijímá jména uživatelů s více než 8 znaky, může být třeba tento problém provést.)

Postup

1. Vytvořte nový uživatelský účet (například **NEW_NAME**)
2. K zadání podrobností nového uživatelského účtu použijte Průvodce přípravou produktu IBM WebSphere MQ .

Změna hesla uživatelského účtu služby IBM WebSphere MQ Windows

Informace o této úloze

Chcete-li změnit heslo účtu lokálního uživatele služby IBM WebSphere MQ Windows , proveďte následující kroky:

Postup

1. Identifikujte uživatele, pod kterým je služba spuštěna.
2. Zastavte službu IBM WebSphere MQ produktu IBM z panelu Správa počítače.
3. Změňte požadované heslo stejným způsobem, jako byste změnili heslo jednotlivce.
4. Přejděte na vlastnosti služby produktu IBM WebSphere MQ z panelu Správa počítače.
5. Vyberte stránku **Přihlásit se** .
6. Potvrďte, že se uvedený název účtu shoduje s uživatelem, pro který bylo heslo změněno.

7. Zadejte heslo do polí **Heslo** a **Potvrdit heslo** a klepněte na tlačítko **OK**.

Služba IBM WebSphere MQ Windows pro instalaci spuštěnou pod uživatelským účtem domény

Informace o této úloze

Je-li služba IBM WebSphere MQ Windows pro instalaci spuštěna pod uživatelským účtem domény, můžete také změnit heslo pro účet takto:

Postup

1. Změňte heslo pro účet domény na řadiči domény. Možná budete muset požádat administrátora domény, aby to pro vás udělal.
2. Postupujte podle kroků a upravte stránku **Přihlášení** pro službu IBM WebSphere MQ .

Uživatelský účet, pod kterým je služba IBM WebSphere MQ Windows spuštěna, provádí všechny příkazy MQSC, které jsou vydány aplikacemi uživatelského rozhraní nebo jsou prováděny automaticky při spuštění systému, ukončení práce systému nebo zotavení služby. Tento uživatelský účet musí mít proto oprávnění k administraci produktu IBM WebSphere MQ . Ve výchozím nastavení je přidán do lokální skupiny **mqm** na serveru. Je-li toto členství odebráno, nebude služba IBM WebSphere MQ Windows fungovat. Další informace o uživatelských právech naleznete v tématu [“Uživatelská práva požadovaná pro službu IBM WebSphere MQ Windows”](#) na stránce 65 .

Pokud se vyskytne problém zabezpečení s účtem uživatele, pod kterým je spuštěna služba IBM WebSphere MQ Windows , zobrazí se v protokolu systémových událostí chybové zprávy a popisy.

Související pojmy

“Použití aktivního adresáře (pouzeWindows)” na stránce 64

V některých konfiguracích sítě, kde jsou uživatelské účty definovány na řadičích domény, které používají Active Directory, je lokální uživatelský účet, pod kterým je uživatelský účet IBM WebSphere MQ spuštěn, pravděpodobně nemá oprávnění, které potřebuje k dotazování na členství ve skupinách jiných uživatelských účtů domény. Průvodce přípravou produktu IBM WebSphere MQ identifikuje, zda se jedná o případ uskutečňující testy a dotazující se na otázky uživatele týkající se konfigurace sítě.

IBM WebSphere MQ koordinace s produktem Db2 jako správce prostředků

Pokud spustíte správce front z produktu IBM WebSphere MQ Explorernebo používáte produkt IBM WebSphere MQ V7a budete mít problémy při koordinaci produktu Db2, zkontrolujte protokoly chyb správce front.

Zkontrolujte chybové protokoly správce front, zda neobsahuje chyby, jako jsou následující:

```
23/09/2008 15:43:54 - Process(5508.1) User(MUSR_MQADMIN) Program(amqzxa0.exe)
Host(HOST_1) Installation(Installation1)
VMRF(7.1.0.0) QMgr(A.B.C)
AMQ7604: The XA resource manager 'DB2 MQBankDB database' was not available when called
for xa_open. The queue manager is continuing without this resource manager.
```

Popis: ID uživatele (výchozí název je MUSR_MQADMIN), které spouští proces služby IBM WebSphere MQ amqsvc . exe je stále spuštěn s přístupovým tokenem, který neobsahuje informace o členství ve skupinách pro skupinu DB2USERS.

Řešení: Poté, co jste zajistili, že ID uživatele služby produktu IBM WebSphere MQ je členem DB2USERS, použijte následující posloupnost příkazů:

- Zastavte službu.
- Zastavte všechny ostatní procesy spuštěné pod stejným ID uživatele.
- Restartujte tyto procesy.

Opětovné zavedení počítače by zajistilo předchozí kroky, ale není nutné.

Rozšíření produktu IBM WebSphere MQ Explorer

IBM WebSphere MQ pro systémy Windows a IBM WebSphere MQ for Linux (platformy x86 a x86-64) poskytují administrativní rozhraní s názvem IBM WebSphere MQ Explorer, aby bylo možné provádět administrativní úlohy jako alternativu k použití řídicích příkazů nebo příkazů MQSC.

Tyto informace platí pouze pro produkty WebSphere MQ for Windows a WebSphere MQ for Linux (pouze platformy x86 a x86-64).

Průzkumník IBM WebSphere MQ představuje informace ve stylu, který je konzistentní s rámcem platformy Eclipse a dalšími aplikacemi modulu plug-in, které podporuje platforma Eclipse.

Prostřednictvím rozšíření produktu IBM WebSphere MQ Explorer mají administrátoři systému možnost přizpůsobit produkt WebSphere MQ Explorer tak, aby vylepšil způsob správy produktu WebSphere MQ.

Další informace naleznete v tématu *Rozšíření průzkumníka IBM WebSphere MQ* v dokumentaci produktu IBM WebSphere MQ Explorer.

Použití aplikace Taskbar produktu IBM WebSphere MQ (pouze systémy Windows)

Aplikace IBM WebSphere MQ Taskbar zobrazí ikonu na hlavním panelu systému Windows na serveru. Ikona vám poskytuje aktuální stav produktu IBM WebSphere MQ a nabídku, ze které můžete provádět jednoduché akce.

V systému Oknase ikona WebSphere MQ nachází na hlavním panelu na serveru a je překryta barvou barevně kódovaného stavu, která může mít jeden z následujících významů:

Zelená

Práce správně; žádné výstrahy v současné době

Modrá

Neurčitá; WebSphere MQ se spouští nebo vypíná

Žlutá

Výstraha; jedna nebo více služeb selhávají nebo se již nezdařily

Chcete-li zobrazit nabídku, klepněte pravým tlačítkem myši na ikonu produktu WebSphere MQ. V nabídce můžete provádět následující akce:

- Klepnutím na tlačítko **Otevřít** otevřete monitor výstrah produktu WebSphere MQ.
- Klepnutím na tlačítko **Ukončit** ukončete aplikaci WebSphere MQ Taskbar.
- Klepnutím na volbu **WebSphere MQ Explorer** spustíte Průzkumníka IBM WebSphere MQ.
- Klepnutím na volbu **Zastavit produkt WebSphere MQ** zastavte produkt WebSphere MQ.
- Chcete-li zobrazit informace o monitoru výstrah produktu WebSphere MQ, klepněte na volbu **O produktu WebSphere MQ**.

Aplikace pro monitorování výstrah produktu IBM WebSphere MQ (pouze systém Windows).

Monitor výstrah IBM WebSphere MQ je nástroj pro detekci chyb, který identifikuje a zaznamenává problémy s produktem IBM WebSphere MQ na lokálním počítači.

Monitor výstrah zobrazuje informace o aktuálním stavu lokální instalace serveru WebSphere MQ. Také monitoruje rozhraní Windows Advanced Configuration and Power Interface (ACPI) a zajišťuje vynucení nastavení ACPI.

V monitoru výstrah produktu WebSphere MQ můžete provádět následující akce:

- Přístup k produktu IBM WebSphere MQ Explorer přímo
- Zobrazit informace související se všemi neprovedenými výstrahami
- Ukončete činnost služby WebSphere MQ na lokálním počítači.

- Směřovat výstražné zprávy po síti na konfigurovatelný uživatelský účet nebo na pracovní stanici či server systému Windows

Správa lokálních objektů produktu IBM WebSphere MQ

Tato sekce vám sděluje, jak spravovat lokální objekty produktu IBM WebSphere MQ pro podporu aplikačních programů, které používají rozhraní MQI (Message Queue Interface). V tomto kontextu lokální administrace znamená vytváření, zobrazování, změny, kopírování a odstraňování objektů IBM WebSphere MQ.

Kromě přístupů podrobně popsanych v této sekci můžete pomocí Průzkumníka IBM WebSphere MQ spravovat lokální objekty WebSphere MQ; viz [“Administrace pomocí produktu IBM WebSphere MQ Explorer”](#) na stránce 56.

Tento oddíl obsahuje následující informace:

- [Aplikační programy používající rozhraní MQI](#)
- [“Provádění lokálních úloh administrace pomocí příkazů MQSC”](#) na stránce 72
- [“Práce se správci front”](#) na stránce 80
- [“Práce s lokálními frontami”](#) na stránce 82
- [“Práce s frontami aliasů”](#) na stránce 87
- [“Práce s modelové fronty”](#) na stránce 88
- [“Práce se službami”](#) na stránce 95
- [“Správa objektů pro spuštění”](#) na stránce 101

Spuštění a zastavení správce front

Toto téma se používá jako úvod k zastavení a spuštění správce front.

Spuštění správce front

Chcete-li spustit správce front, použijte příkaz `stmqm` následujícím způsobem:

```
stmqm saturn.queue.manager
```

Na systémech WebSphere MQ for Windows a WebSphere MQ for Linux (platformy x86 a x86-64) můžete spustit správce front následujícím způsobem:

1. Otevřete Průzkumníka IBM WebSphere MQ.
2. Vyberte správce front z pohledu Navigator.
3. Klepněte na tlačítko Start. Spustí se správce front.

Pokud spuštění správce front trvá déle než několik sekund, produkt WebSphere MQ vydá informační zprávy přerušovaně podrobně popisující průběh pouštění.

Příkaz `stmqm` nevrací řízení, dokud nebude správce front spuštěn a je připraven přijímat požadavky na připojení.

Automatické spuštění správce front

V produktu WebSphere MQ for Windows lze spustit správce front automaticky při spuštění systému pomocí Průzkumníka produktu WebSphere MQ. Další informace viz [“Administrace pomocí produktu IBM WebSphere MQ Explorer”](#) na stránce 56.

Zastavení správce front

Chcete-li zastavit správce front, použijte příkaz `endmqm`.

Poznámka: Příkaz **endmqm** je třeba použít z instalace přidružené ke správci front, se kterým pracujete. Pomocí příkazu `dspmqr -o installation` můžete zjistit, která instalace správce front je přidružena.

Chcete-li například zastavit správce front s názvem QMB, zadejte následující příkaz:

```
endmqm QMB
```

Na systémech WebSphere MQ for Windows a WebSphere MQ for Linux (platformy x86 a x86-64) můžete správce front zastavit následujícím způsobem:

1. Otevřete Průzkumník IBM WebSphere MQ .
2. Vyberte správce front z pohledu Navigator .
3. Klepněte na tlačítko Stop . . . Zobrazí se panel Ukončení správce front.
4. Vyberte Řízený, nebo Okamžitě.
5. Klepněte na tlačítko OK. Správce front je zastaven.

klidové ukončení

Příkaz **endmqm** standardně provádí klidové ukončení běhu zadaného správce front. Dokončení této operace může trvat určitou dobu. Do klidového stavu bylo ukončeno, dokud se všechny připojené aplikace neodpojí.

Tento typ ukončení práce systému můžete použít k zastavení aplikací. Pokud zadáte:

```
endmqm -c QMB
```

není vám řečeno, kdy byly zastaveny všechny aplikace. (Příkaz `endmqm -c QMB` je ekvivalentní příkazu `endmqm QMB`.)

Nicméně, pokud zadáte:

```
endmqm -w QMB
```

příkaz čeká, dokud všechny aplikace nebudou zastaveny a správce front ukončen.

okamžité ukončení práce systému

Okamžité ukončení činnosti aktuálních volání MQI lze dokončit, ale všechny nové volání se nezdaří. Tento typ ukončení činnosti nečeká, až se aplikace odpojí od správce front.

Chcete-li provést okamžité ukončení práce, zadejte:

```
endmqm -i QMB
```

preventivní ukončení

Poznámka: Tuto metodu nepoužívejte, pokud se nezdaří všechny ostatní pokusy o zastavení správce front pomocí příkazu **endmqm** . Tato metoda může mít nepředvídatelné důsledky pro připojené aplikace.

Pokud okamžité ukončení práce nefunguje, musíte se uchýlit k ukončení *preemptivního* ukončení, přičemž byste měli zadat parametr `-p` . Příklad:

```
endmqm -p QMB
```

To okamžitě zastaví správce front. Pokud tato metoda stále nefunguje, použijte [“Ruční zastavení správce front”](#) na stránce 70 pro alternativní řešení.

Podrobný popis příkazu **endmqm** a jeho voleb naleznete v souboru [endmqm](#).

Máte-li problémy při ukončování práce správce front

Problémy při ukončování činnosti správce front jsou často způsobeny aplikacemi. Například, když aplikace:

- Nezaškrtnli správně návratové kódy MQI
- Nevyžadovat oznámení o uvedení do klidového stavu
- Ukončit bez odpojení od správce front (zadáním volání MQDISC)

Pokud se vyskytne problém při zastavení správce front, můžete příkaz **endmqm** přerušit pomocí Ctrl-C. Poté můžete zadat jiný příkaz **endmqm** , ale tentokrát s příznakem, který určuje typ ukončení práce, který vyžadujete.

Ruční zastavení správce front

Pokud standardní metody pro zastavení správců front selžou, vyzkoušejte zde popsané metody.

Standardní způsob zastavení správců front lze provést pomocí příkazu **endmqm** . Chcete-li správce front zastavit ručně, použijte jeden z postupů popsaných v této části. Podrobnosti o provádění operací ve správcích front pomocí řídicích příkazů naleznete v tématu [Vytvoření a správa správců front](#).

Zastavení správců front v systému Windows

Chcete-li ukončit procesy a službu IBM WebSphere MQ , zastavte správce front v produktu IBM WebSphere MQ for Windows.

Chcete-li zastavit správce front spuštěného v produktu WebSphere MQ for Windows, postupujte takto:

1. Vypište názvy (ID) procesů, které jsou spuštěny, pomocí správce úloh systému Windows .
2. Ukončete procesy pomocí správce úloh systému Windows nebo pomocí příkazu **taskkill** v následujícím pořadí (pokud jsou spuštěny):

AMQZMUCO	Kritický správce procesů
AMQZXMAO	Řadič provádění
AMQZFUMA	Proces OAM
AMQZLAAO	Agenti LQM
AMQZLSAO	Agenti LQM
AMQZMUFO	Správce utility
AMQZMGRO	Řadič procesu
AMQZMURO	Restartovatelný správce procesů
AMQFQPUB	Proces publikování a odběrů
AMQFCXUBA.	Pracovní proces zprostředkovatele
AMQRMPPAX.	Proces sdružování procesu
AMQCRSTA	Proces úlohy odpovídacího modulu bez podprocesů
AMQCRS6B	LU62 -kanál příjemce a připojení klienta
AMQRRMFACITY	Proces úložiště (pro klastry)
AMQZDMAA	Odložené procesor zpráv
AMQPCSEA	Příkazový server
RUNMQTRM	Vyvolat monitor spouštěčů pro server
RUNMQDLQ.	Vyvolání obslužné rutiny fronty nedoručených zpráv
RUNMQCHI.	Proces inicializátoru kanálu

RUNMQLSR	Proces modulu listener kanálu
AMQXSSVNNNAME	Servery sdílené paměti
AMQZTRCN.	Trasovat

3. Zastavte službu WebSphere MQ z aplikace **Administration tools** > **Services** na ovládacím panelu systému Windows .
4. Pokud jste vyzkoušeli všechny metody a správce front nebyl zastaven, znovu zaveďte systém.

Správce úloh systému Windows a příkaz **tasklist** poskytují omezené informace o úlohách. Další informace, které vám pomohou určit, které procesy se týkají konkrétního správce front, zvažte použití nástroje, jako je *Process Explorer* (procexp.exe), dostupného ke stažení na webu společnosti Microsoft na adrese <https://www.microsoft.com>.

Zastavení správců front v systémech UNIX and Linux

Chcete-li ukončit procesy a službu IBM WebSphere MQ , zastavte správce front v produktu IBM WebSphere MQ for UNIX and Linux. Zde můžete vyzkoušet zde popsané metody, pokud standardní metody pro zastavení a odebrání správců front selžou.

Chcete-li zastavit správce front spuštěného v produktu WebSphere MQ for UNIX and Linux , postupujte takto:

1. Pomocí příkazu ps najdete ID procesů programů správce front, které jsou stále spuštěny. Je-li například správce front nazván QMNAME, použijte následující příkaz:

```
ps -ef | grep QMNAME
```

2. Ukončete všechny procesy správce front, které jsou stále spuštěny. Použijte příkaz kill a uveďte ID procesu, která byla zjištěna příkazem ps .

Ukončete procesy v následujícím pořadí:

amqzmuc0	Kritický správce procesů
amqzma0	Řadič provádění
amqzfuma	Proces OAM
amqzlaa0	Agenti LQM
amqzlsa0	Agenti LQM
amqzmuf0	Správce utility
amqzmur0	Restartovatelný správce procesů
amqzmgr0	Řadič procesu
amqfqpub	Proces publikování a odběrů
amqfcxba	Pracovní proces zprostředkovatele
amqrmppa	Proces sdružování procesu
amqcrsta	Proces úlohy odpovídacího modulu bez podprocesů
amqcrs6b	LU62 -kanál příjemce a připojení klienta
amqrrmfa	Proces úložiště (pro klastry)
amqzdmaa	Odložené procesor zpráv
amqpcsea	Příkazový server
runmqtrm	Vyvolat monitor spouštěčů pro server
runmqdlq	Vyvolání obslužné rutiny fronty nedoručených zpráv

runmqchi	Proces inicializátoru kanálu
runmqtsr	Proces modulu listener kanálu

Poznámka: Příkaz **kill -9** můžete použít k ukončení procesů, které se nepodaří zastavit.

Pokud správce front zastavíte ručně, mohou být provedeny soubory FFST a soubory FDC umístěné v produktu `/var/mqm/errors`. nepovažujte za defekt ve správci front.

Správce front se restartuje normálně, a to i poté, co jste ji zastavili s použitím této metody.

Provádění lokálních úloh administrace pomocí příkazů MQSC

Tento oddíl vás seznámí s příkazy MQSC a informuje vás o tom, jak je používat pro některé běžné úlohy.

Používáte-li produkt IBM WebSphere MQ for Windows nebo IBM WebSphere MQ for Linux (platformy x86 a x86-64), můžete rovněž provádět operace popsané v této části pomocí Průzkumníka IBM WebSphere MQ. Další informace viz [“Administrace pomocí produktu IBM WebSphere MQ Explorer”](#) na stránce 56.

Ke správě objektů správce front můžete použít příkazy MQSC včetně samotného správce front, front, definic procesů, kanálů, kanálů připojení klienta, modulů listener, služeb názvů, klastrů a objektů ověřovacích informací. Tato sekce se zabývá správci front, frontami a definicemi procesů. Informace o správě kanálu, kanálu připojení klienta a objektech modulu listener viz [Objekty](#). Informace o všech příkazech MQSC pro správu objektů správce front naleznete v tématu [“Příkazy skriptu \(MQSC\)”](#) na stránce 72.

Příkazy MQSC zadáte do správce front pomocí příkazu `runmqsc`. (Podrobnosti o tomto příkazu naleznete v souboru [runmqsc](#).) Můžete to provést interaktivně, vydávat příkazy z klávesnice nebo můžete přeměrovat standardní vstupní zařízení (`stdin`) ke spuštění posloupnosti příkazů z textového souboru ASCII. V obou případech je formát příkazů stejný. (Informace o spouštění příkazů z textového souboru viz [“Spouštění příkazů MQSC z textových souborů”](#) na stránce 76.)

Příkaz `runmqsc` lze spustit třemi způsoby v závislosti na parametrech nastavených v příkazu:

- Ověřte příkaz bez spuštění, kde jsou příkazy MQSC ověřeny na lokálním správci front, ale nespustí se.
- Spusťte příkaz na lokálním správci front, kde jsou příkazy MQSC spouštěny v lokálním správci front.
- Spusťte příkaz ve vzdáleném správci front, ve kterém jsou spuštěny příkazy MQSC ve vzdáleném správci front.

Chcete-li zobrazit syntaxi, můžete také spustit příkaz následovaný otazníkem.

Atributy objektu uvedené v příkazech MQSC se zobrazí v této sekci velkými písmeny (např. `QMNAME`), i když nejsou citlivé na velikost písmen. Názvy atributů příkazu MQSC jsou omezeny na osm znaků. Příkazy MQSC jsou k dispozici na jiných platformách, včetně souborů IBM i a z/OS.

Příkazy MQSC jsou shrnuty v kolekci témat v sekci [Odkaz na MQSC](#).

Příkazy skriptu (MQSC)

Příkazy MQSC poskytují uniformní metodu zadávání uživatelem čitelných příkazů na platformách WebSphere MQ. Další informace o příkazech *programmable command format* (PCF) naleznete v příručce [“Úvod do formátu programových příkazů”](#) na stránce 9.

Obecný formát příkazů je zobrazen v [příkazech MQSC](#).

Při používání příkazů MQSC byste měli dodržovat následující pravidla:

- Každý příkaz začíná s primárním parametrem (slovesem) a za ním následuje sekundární parametr (podstatné jméno). To je pak následováno názvem nebo generickým názvem objektu (v závorkách), pokud existuje jeden, který se nachází na většině příkazů. Poté mohou parametry obvykle nastat v libovolném pořadí; má-li parametr odpovídající hodnotu, musí se hodnota objevit přímo za parametrem, ke kterému se vztahuje.

- Klíčová slova, závorky a hodnoty mohou být odděleny libovolným počtem mezer a čárkami. Čárka zobrazená v syntaktických diagramech může být vždy nahrazena jednou nebo více mezerami. Před každým parametrem (za primárním parametrem) musí být alespoň jedna mezera (za primárním parametrem).
- Libovolný počet mezer se může vyskytnout na začátku nebo na konci příkazu, a mezi parametry, interpunkcí a hodnotami. Platný je například tento příkaz:

```
ALTER QLOCAL ('Account' )          TRIGDPTH ( 1)
```

Mezery uvnitř páru uvozovek jsou významné.

- Další čárky se mohou objevit kdekoli, kde jsou mezery povoleny a jsou považovány za prázdné (pokud ovšem nejsou uvnitř řetězců ohraničených uvozovkami).
- Opakované parametry nejsou povoleny. Opakující se parametr s verzí "NO" jako v REPLACE NOREPLACE je také nepovolený.
- Řetězce, které obsahují mezery, malé znaky nebo speciální znaky jiné než:
 - Tečka (.)
 - Lomítko (/)
 - Podtržítka (_)
 - Procento (%)

musí být ohraničen jednoduchými uvozovkami, pokud nejsou:

- Generické hodnoty končící hvězdičkou
- Jedna hvězdička (například TRACE (*))
- Specifikace rozsahu obsahující dvojtečku (například CLASS (01:03))

Pokud řetězec sám obsahuje jednoduchou uvozovku, je apostrof představován dvěma jednoduchými uvozovkami. Malá písmena, která nejsou obsažena v uvozovkách, jsou přeskládána na velká písmena.

- Na jiných platformách než z/OS je řetězec neobsahující žádné znaky (tj. dvě jednoduché uvozovky bez mezery mezi) interpretován jako prázdný prostor uzavřený v apostrofech, tj. interpretovaný stejným způsobem jako (""). Výjimkou je, je-li použitý atribut jeden z následujících:
 - TOPICSTR
 - SUB
 - USERDATA
 - SELECTOR

pak jsou dvě jednoduché uvozovky bez mezer interpretovány jako řetězec s nulovou délkou.

- V v7.0 jsou všechny koncové mezery v těchto řetězcových attributech, které jsou založeny na typech MQCHARV, jako např. SELECTOR, sub user data, považovány za významné, což znamená, že 'abc' se nerovná 'abc'.
- Levá závorka následovaná pravou závorkou, bez výrazných informací například mezi, například:

```
NAME ( )
```

je neplatný, není-li výslovně uvedeno jinak.

- Klíčová slova nerozlišují velká a malá písmena: ALTER, alter a ALTER jsou přijatelné. Cokoli, co není obsaženo v uvozovkách, je složené na velká písmena.
- Synonyma jsou definována pro některé parametry. Například DEF je vždy synonymem pro DEFINE, takže DEF QLOCAL je platná. Synonyma nejsou však jen minimální řetězce; DEF není platným synonymem pro DEFINE.

Poznámka: Pro parametr DELETE neexistuje žádné synonymum. Tím se vyhnete náhodnému odstranění objektů při použití DEF, synonym pro DEFINE.

Přehled použití příkazů MQSC pro administraci produktu IBM WebSphere MQ viz [“Provádění lokálních úloh administrace pomocí příkazů MQSC”](#) na stránce 72.

Příkazy MQSC používají určité speciální znaky k tomu, aby měly určité významy. Další informace o těchto speciálních znacích a o jejich použití najdete v tématu [Znaky se speciálním významem](#).

Informace o tom, jak můžete sestavovat skripty pomocí příkazů MQSC, najdete v tématu [Sestavování příkazových skriptů](#).

Úplný seznam příkazů MQSC najdete v tématu [Příkazy MQSC](#).

Související úlohy

[Sestavování příkazových skriptů](#)

Názvy objektů produktu WebSphere MQ

Jak používat názvy objektů v příkazech MQSC.

V příkladech používáme některé dlouhé názvy pro objekty. To vám pomůže identifikovat typ objektu, se kterým se potýkáme.

Když vydáte příkazy MQSC, musíte uvést pouze lokální název fronty. V našem příkladu používáme názvy front jako:

```
ORANGE . LOCAL . QUEUE
```

Část LOCAL . QUEUE názvu slouží k ilustraci toho, že tato fronta je lokální frontou. Pro názvy lokálních front obecně platí *not* .

Název saturn . queue . manager také používáme jako název správce front. Část queue . manager názvu slouží k ilustraci toho, že tento objekt je správce front. U názvů správců front obecně není vyžadována hodnota *not* .

Rozlišování malých a velkých písmen v příkazech MQSC

Příkazy MQSC včetně jejich atributů mohou být zapsány velkými nebo malými písmeny. Názvy objektů v příkazech MQSC jsou složeny na velká písmena (tj. QUEUE a queue nejsou rozlišeny), pokud nejsou názvy uzavřeny v jednoduchých uvozovkách. Pokud uvozovky nejsou použity, objekt se zpracuje s názvem psaným velkými písmeny. Další informace viz [Odkaz na MQSC](#) .

Vyvolání příkazu `runmqsc` , které je společné se všemi řídicími příkazy produktu WebSphere MQ , rozlišuje velikost písmen v některých prostředích produktu WebSphere MQ . Další informace najdete v tématu [Použití řídicích příkazů](#) .

Standardní vstup a výstup

Standardní vstupní zařízení, také nazývané `stdin`, je zařízení, ze kterého se bere vstup do systému. Obvykle se jedná o klávesnici, ale můžete zadat, že vstup má pocházet ze sériového portu nebo z diskového souboru, například. *Standardní výstupní zařízení*, označované také jako `stdout`, je zařízení, na které se posílá výstup ze systému. Obvykle se jedná o obrazovku, ale můžete přesměrovat výstup na sériový port nebo do souboru.

U příkazů operačního systému a řídicích příkazů produktu WebSphere MQ je `<` operátor přesměrování vstupu. Pokud je za operátorem následován název souboru, vezme se tento soubor ze souboru. Podobně platí, že operátor `>` přesměrovává výstup; pokud je tento operátor následován názvem souboru, výstup je směřován do tohoto souboru.

Interaktivní použití příkazů MQSC

Příkazy MQSC můžete použít interaktivně pomocí příkazového okna nebo shellu.

Chcete-li použít příkazy MQSC interaktivně, otevřete příkazové okno nebo shell a zadejte:

```
runmqsc
```

V tomto příkazu nebyl zadán název správce front, takže příkazy MQSC jsou zpracovány výchozím správcem front. Chcete-li použít jiného správce front, zadejte v příkazu **runmqsc** název správce front. Chcete-li například spustit příkazy MQSC ve správci front `jupiter.queue.manager`, použijte následující příkaz:

```
runmqsc jupiter.queue.manager
```

Po tomto stavu jsou všechny příkazy MQSC, které zapíšete, zpracovány tímto správcem front za předpokladu, že jsou ve stejném uzlu a jsou již spuštěny.

Nyní můžete zadat libovolný příkaz MQSC, podle potřeby. Zkuste to například takto:

```
DEFINE QLOCAL (ORANGE.LOCAL.QUEUE)
```

Pro příkazy, které mají příliš mnoho parametrů, aby se vešly na jeden řádek, použijte znaky pokračování k označení toho, že příkaz pokračuje na následujícím řádku:

- Znaménko minus (-) označuje, že příkaz má pokračovat od začátku následujícího řádku.
- Znaménko plus (+) označuje, že příkaz má pokračovat od prvního nemezerových znaků na následujícím řádku.

Vstup příkazu končí posledním znakem neprázdné řádky, která není pokračovacím znakem. Vstup příkazu lze také explicitně ukončit zadáním středníku (;). (To je obzvláště užitečné, pokud omylem zadáte pokračovací znak na konci posledního řádku vstupu příkazu.)

Zpětná vazba z příkazů MQSC

Když vydáte příkazy MQSC, správce front vrátí zprávy operátora, které potvrzují vaše akce nebo vám sdělí o chybách, které jste provedli. Příklad:

```
AMQ8006: WebSphere MQ queue created.
```

Tato zpráva potvrzuje, že byla vytvořena fronta.

```
AMQ8405: Syntax error detected at or near end of command segment below:-
```

```
AMQ8426: Valid MQSC commands are:
```

```
ALTER  
CLEAR  
DEFINE  
DELETE  
DISPLAY  
END  
PING  
REFRESH  
RESET  
RESOLVE  
RESUME  
START  
STOP  
SUSPEND  
4 : end
```

Tato zpráva označuje, že jste provedli chybu syntaxe.

Tyto zprávy jsou odeslány na standardní výstupní zařízení. Pokud jste příkaz nezadali správně, podívejte se do [Odkaz na MQSC](#), kde získáte správnou syntaxi.

Ukončení interaktivního vstupu příkazů MQSC

Chcete-li ukončit práci s příkazy MQSC, zadejte příkaz END.

Případně můžete použít znak EOF pro váš operační systém.

Spouštění příkazů MQSC z textových souborů

Spouštění příkazů MQSC interaktivně je vhodné pro rychlé testy, ale pokud máte velmi dlouhé příkazy nebo opakovaně používáte určitou posloupnost příkazů, zvažte přesměrování `stdin` z textového souboru.

“Standardní vstup a výstup” na stránce 74 obsahuje informace o `stdin` a `stdout`. Chcete-li přesměrovat `stdin` z textového souboru, nejprve vytvořte textový soubor obsahující příkazy MQSC pomocí svého běžného textového editoru. Použijete-li příkaz `runmqsc`, použijte operátory přesměrování. Následující příkaz například spustí posloupnost příkazů obsažených v textovém souboru `myprog.in`:

```
runmqsc < myprog.in
```

Podobně můžete také přesměrovat výstup do souboru. Soubor obsahující příkazy MQSC pro vstup se nazývá *příkazový soubor MQSC*. Výstupní soubor obsahující odpovědi od správce front se nazývá *výstupní soubor*.

Chcete-li přesměrovat jak `stdin`, tak `stdout` na příkaz `runmqsc`, použijte tento tvar příkazu:

```
runmqsc < myprog.in > myprog.out
```

Tento příkaz vyvolá příkazy MQSC obsažené v příkazovém souboru MQSC `myprog.in`. Protože neuvedli jste název správce front, příkazy MQSC se spouštějí na výchozího správce front. Výstup se odešle do textového souboru `myprog.out`. Obrázek 12 na stránce 77 Zobrazí extrakt z příkazového souboru MQSC `myprog.in` a Obrázek 13 na stránce 78 zobrazí odpovídající extrakci výstupu v produktu `myprog.out`.

Chcete-li přesměrovat `stdin` a `stdout` na příkaz `runmqsc`, pro správce front (`saturn.queue.manager`), který není výchozí, použijte tento tvar příkazu:

```
runmqsc saturn.queue.manager < myprog.in > myprog.out
```

příkazové soubory MQSC

Příkazy MQSC jsou napsány v čitelném formátu, tj. v textu ASCII. Obrázek 12 na stránce 77 je extrakt z příkazového souboru MQSC zobrazující příkaz MQSC (`DEFINE QLOCAL`) s jeho atributy. [Odkaz na MQSC](#) obsahuje popis každého příkazu MQSC a jeho syntaxi.

```

.
.
DEFINE QLOCAL(ORANGE.LOCAL.QUEUE) REPLACE +
DESCR(' ') +
PUT(ENABLED) +
DEFPRTY(0) +
DEFPSIST(NO) +
GET(ENABLED) +
MAXDEPTH(5000) +
MAXMSGL(1024) +
DEFSOPT(SHARED) +
NOHARDENBO +
USAGE(NORMAL) +
NOTRIGGER;
.
.

```

Obrázek 12. Extrahovat z příkazového souboru MQSC

Pro přenositelnost mezi prostředími produktu WebSphere MQ omezte délku řádku v příkazových souborech MQSC na 72 znaků. Znaménko plus označuje, že příkaz pokračuje na dalším řádku.

Sestavy příkazů MQSC

Příkaz **runmqsc** vrací zprávu *report*, která je odeslána uživateli stdout. Sestava obsahuje:

- Záhlaví identifikující příkazy MQSC jako zdroj sestavy:

```
Starting MQSC for queue manager jupiter.queue.manager.
```

Kde `jupiter.queue.manager` je název správce front.

- Volitelný číslovaný výpis vydaných příkazů MQSC. Ve výchozím nastavení se text vstupu vypisuje na výstup. V rámci tohoto výstupu má každý příkaz předponu podle pořadového čísla, jak ukazuje [Obrázek 13](#) na stránce 78. Chcete-li výstup potlačit, můžete však použít příznak `-e` příkazu `runmqsc`.
- Chybová zpráva syntaxe pro všechny příkazy, u kterých bylo zjištěno, že jsou chybné.
- *Zpráva operátora* označující výsledek spuštění každého příkazu. Např. zpráva operátora pro úspěšné dokončení příkazu `DEFINE QLOCAL` je:

```
AMQ8006: WebSphere MQ queue created.
```

- Další zprávy, které jsou výsledkem obecných chyb při spuštění skriptového souboru.
- Stručný statistický souhrn sestavy označující počet přečtených příkazů, počet příkazů se syntaxí chyb a počet příkazů, které nelze zpracovat.

Poznámka: Správce front se pokusí zpracovat pouze ty příkazy, které nemají žádné syntaktické chyby.

```

Starting MQSC for queue manager jupiter.queue.manager.
.
.
12:      DEFINE QLOCAL('ORANGE.LOCAL.QUEUE') REPLACE +
:        DESCR(' ') +
:        PUT(ENABLED) +
:        DEFPRTY(0) +
:        DEFPSIST(NO) +
:        GET(ENABLED) +
:        MAXDEPTH(5000) +
:        MAXMSGL(1024) +
:        DEFSOPT(SHARED) +
:        NOHARDENBO +
:        USAGE(NORMAL) +
:        NOTRIGGER;
AMQ8006: WebSphere MQ queue created.
:
.

```

Obrázek 13. Extrahovat z příkazového souboru zpráv MQSC

Spuštění dodaných příkazových souborů MQSC

S produktem WebSphere MQjsou dodávány následující příkazové soubory MQSC:

amqscos0.tst

Definice objektů používaných vzorovými programy.

amqscic0.tst

Definice front pro transakce CICS .

V produktu WebSphere MQ for Windowsjsou tyto soubory umístěny v adresáři `MQ_INSTALLATION_PATH\tools\mqsc\samples`. `MQ_INSTALLATION_PATH` Představuje adresář vysoké úrovně, do kterého je produkt WebSphere MQ nainstalován.

Na systémech UNIX and Linux jsou tyto soubory umístěny v adresáři `MQ_INSTALLATION_PATH/samp`. `MQ_INSTALLATION_PATH` Představuje adresář vysoké úrovně, do kterého je produkt WebSphere MQ nainstalován.

Příkaz, který je spouští, je:

```
runmqsc < amqscos0.tst >test.out
```

Ověření příkazů pomocí příkazu runmqsc

Pomocí příkazu `runmqsc` lze ověřit příkazy MQSC v lokálním správci front, aniž by je skutečně spouštěly. Chcete-li to provést, nastavte příznak `-v` v příkazu `runmqsc` , například:

```
runmqsc -v < myprog.in > myprog.out
```

Když vyvoláte příkaz `runmqsc` pro příkazový soubor MQSC, správce front ověří každý příkaz a vrátí sestavu, aniž by ve skutečnosti spouštěla příkazy MQSC. To vám umožní zkontrolovat syntaxi příkazů ve vašem příkazovém souboru. To je zvláště důležité, pokud:

- Spuštění velkého počtu příkazů z příkazového souboru.
- Je třeba mnohokrát používat příkazový soubor MQSC.

Vrácená sestava je podobná té, která je zobrazena v části [Obrázek 13 na stránce 78](#).

Tuto metodu nelze použít k tomu, abyste mohli ověřit příkazy MQSC vzdáleně. Pokusíte-li se například o tento příkaz:

```
runmqsc -w 30 -v jupiter.queue.manager < myprog.in > myprog.out
```

Příznak -w, který použijete k označení vzdáleného správce front, je ignorován a příkaz se spustí lokálně v režimu verifikace. 30 je počet sekund, po které produkt WebSphere MQ čeká na odpovědi ze vzdáleného správce front.

Spuštění příkazů MQSC z dávkových souborů

Máte-li velmi dlouhé příkazy nebo opakovaně používáte určitou posloupnost příkazů, zvažte přesměrování stdin z dávkového souboru.

Chcete-li provést přesměrování stdin z dávkového souboru, nejprve vytvořte dávkový soubor obsahující příkazy MQSC pomocí svého běžného textového editoru. Použijete-li příkaz runmqsc, použijte operátory přesměrování. Následující příklad:

1. Vytvoří správce testovací fronty, TESTQM
2. Vytvoří odpovídající CLNTCONN a modul listener nastaven na použití portu TCP/IP 1600
3. Vytvoří testovací frontu, TESTQ
4. Přepne zprávu do fronty pomocí ukázkového programu amqspc.

```
export MYTEMPQM=TESTQM
export MYPOR=1600
export MQCHLLIB=/var/mqm/qmgrs/$MYTEMPQM/@ipcc

crtmqm $MYTEMPQM
strmqm $MYTEMPQM
runmqclsr -m $MYTEMPQM -t TCP -p $MYPOR &

runmqsc $MYTEMPQM << EOF
  DEFINE CHANNEL(NLTM) CHLTYPE(SVRCONN) TRPTYPE(TCP)
  DEFINE CHANNEL(NLTM) CHLTYPE(CLNTCONN) QMNAME('$MYTEMPQM') CONNAME('hostname($MYPOR)')
  ALTER CHANNEL(NLTM) CHLTYPE(CLNTCONN)
  DEFINE QLOCAL(TESTQ)
EOF

amqspc TESTQ $MYTEMPQM << EOF
hello world
EOF

endmqm -i $MYTEMPQM
```

Obrázek 14. Ukázkový skript pro spuštění příkazů MQSC z dávkového souboru

Řešení problémů s příkazy MQSC

Pokud nemůžete spustit příkazy MQSC pro spuštění, použijte informace z tohoto tématu, abyste zjistili, zda se na vás některý z těchto běžných problémů vztahuje. Není vždy jasné, jaký problém je, když si přečtete chybu, kterou generuje příkaz.

Pokud nemůžete spustit příkazy MQSC pro spuštění, použijte následující informace, abyste zjistili, zda se na vás některý z těchto běžných problémů vztahuje. Není vždy jasné, jaký problém je, když si přečtete chybu generovanou.

Použijete-li příkaz runmqsc, mějte na paměti následující skutečnosti:

- Použijte operátor < k přesměrování vstupu ze souboru. Pokud tento operátor vynecháte, interpretuje správce front název souboru jako název správce front a zobrazí následující chybovou zprávu:

```
AMQ8118: WebSphere MQ queue manager does not exist.
```

- Pokud přesměrujete výstup do souboru, použijte operátor přesměrování `>` . Při výchozím nastavení je soubor umístěn v aktuálním pracovním adresáři v době vyvolání příkazu `runmqsc` . Uveďte plně kvalifikovaný název souboru pro odeslání výstupu do specifického souboru a adresáře.
- Zkontrolujte, zda jste vytvořili správce front, který bude spouštět příkazy, pomocí následujícího příkazu, který zobrazí všechny správce front:

```
dspmq
```

- Správce front musí být spuštěn. Pokud tomu tak není, spusťte jej; (viz [Spuštění správce front](#)). Pokud se pokusíte spustit správce front, který je již spuštěn, obdržíte chybovou zprávu.
- Pokud jste nedefinovali výchozího správce front nebo jste získali tuto chybu, zadejte v příkazu `runmqsc` název správce front nebo se zobrazí následující chybová zpráva:

```
AMQ8146: WebSphere MQ queue manager not available.
```

- Příkaz MQSC nelze zadat jako parametr příkazu `runmqsc` . Například, toto není platné:

```
runmqsc DEFINE QLOCAL(FRED)
```

- Před zadáním příkazu `runmqsc` nelze zadat příkazy MQSC.
- Příkazy řízení z `runmqsc` nelze spouštět. Například nemůžete vydat příkaz `strmqm` ke spuštění správce front v době, kdy spouštíte příkazy MQSC interaktivně. Pokud tak učiníte, obdržíte chybové zprávy podobné následujícím:

```
runmqsc
.
.
Starting MQSC for queue manager jupiter.queue.manager.

  1 : strmqm saturn.queue.manager
AMQ8405: Syntax error detected at or near end of cmd segment below:-s

AMQ8426: Valid MQSC commands are:
ALTER
CLEAR
DEFINE
DELETE
DISPLAY
END
PING
REFRESH
RESET
RESOLVE
RESUME
START
STOP
SUSPEND
  2 : end
```

Práce se správcí front

Příklady příkazů MQSC, které lze použít k zobrazení nebo změně atributů správce front.

Zobrazení atributů správce front

Chcete-li zobrazit atributy správce front zadaného v příkazu `runmqsc` , použijte následující příkaz MQSC:

```
DISPLAY QMGR
```

Typický výstup z tohoto příkazu je zobrazen v [Obrázek 15 na stránce 81](#)


```

DISPLAY QMGR
  1 : DISPLAY QMGR
AMQ8408: Display Queue Manager details.
  QMNAME(QM1)
  ACCTINT(1800)
  ACCTQ(OFF)
  ACTVCONO (DISABLED)
  ALTDATE(2012-05-27)
  AUTHOREV(DISABLED)
  CHAD(DISABLED)
  CHADEXIT( )
  CLWLDATA( )
  CLWLLEN(100)
  CLWLUSEQ(LOCAL)
  CMDLEVEL(750)
  CONFIGEV(DISABLED)
  CRTIME(16.14.01)
  DEFXMITQ( )
  DISTL(YES)
  IPADDRV(IPV4)
  LOGGEREV(DISABLED)
  MAXHANDS(256)
  MAXPROPL(NOLIMIT)
  MAXUMSGS(10000)
  MONCHL(OFF)
  PARENT( )
  PLATFORM(WINDOWSNT)
  PSNPMSG(DISCARD)
  PSSYNCP(IFPER)
  PSMODE(ENABLED)
  REPOS( )
  ROUTEREC(MSG)
  SCMDSERV(QMGR)
  SSLCRYP( )
  SSLFIPS(NO)
MQData\qmgrs\QM1\ssl\key)
  SSLKEYC(0)
  STATCHL(OFF)
  STATMQI(OFF)
  STRSTPEV(ENABLED)
  TREELIFE(1800)
  ACCTCONO(DISABLED)
  ACCTMQI(OFF)
  ACTIVREC(MSG)
  ACTVTRC(OFF)
  ALTTIME(16.14.01)
  CCSID(850)
  CHADEV(DISABLED)
  CHLEV(DISABLED)
  CLWLEXIT( )
  CLWLNRUC(999999999)
  CMDEV(DISABLED)
  COMMANDQ(SYSTEM.ADMIN.COMMAND.QUEUE)
  CRDATE(2011-05-27)
  DEADQ( )
  DESCR( )
  INHIBTEV(DISABLED)
  LOCALEV(DISABLED)
  MARKINT(5000)
  MAXMSGL(4194304)
  MAXPRTY(9)
  MONACLS(QMGR)
  MONQ(OFF)
  PERFMEV(DISABLED)
  PSRTYCNT(5)
  PSNPRES(NORMAL)
  QMID(QM1_2011-05-27_16.14.01)
  REMOTEEV(DISABLED)
  REPOSNL( )
  SCHINIT(QMGR)
  SSLCRLNL( )
  SSLEV(DISABLED)
  SSLKEYR(C:\Program Files\IBM\WebSphere
  STATACLS(QMGR)
  STATINT(1800)
  STATQ(OFF)
  SYNCP
  TRIGINT(999999999)

```

Obrázek 15. Typický výstup z příkazu DISPLAY QMGR

Parametr ALL je standardní volbou příkazu DISPLAY QMGR . Zobrazí všechny atributy správce front. Výstup má zejména informace o výchozím názvu správce front, názvu fronty nedoručených zpráv a názvu fronty příkazů.

Zadáním následujícího příkazu můžete potvrdit, že tyto fronty existují:

```
DISPLAY QUEUE (SYSTEM.*)
```

Zobrazí se seznam front, které se shodují s kmenem SYSTEM.*. Jsou vyžadovány závorky.

Změna atributů správce front

Chcete-li změnit atributy správce front zadaného v příkazu **runmqsc** , použijte příkaz MQSC ALTER QMGRa zadejte atributy a hodnoty, které chcete změnit. Chcete-li například změnit atributy produktu `jupiter.queue.manager`, použijte následující příkazy:

```
runmqsc jupiter.queue.manager
ALTER QMGR DEADQ (ANOTHERDLQ) INHIBTEV (ENABLED)
```

Příkaz ALTER QMGR mění použitou frontu nedoručených zpráv a umožňuje blokování událostí.

Související odkazy

[Atributy správce front](#)

Práce s lokálními frontami

Tento oddíl obsahuje příklady některých příkazů MQSC, které můžete použít ke správě lokálních, modelů a front aliasů.

Chcete-li získat podrobné informace o těchto příkazech, prohlédněte si příručku [Odkaz na MQSC](#).

Definování lokální fronty

V případě aplikace je lokální správce front správce front, ke kterému je aplikace připojena. Fronty spravované lokálním správcem front se říká, že jsou lokální pro daného správce front.

Chcete-li vytvořit lokální frontu, použijte příkaz MQSC DEFINE QLOCAL. Můžete také použít výchozí definici definovanou ve výchozí definici lokální fronty nebo můžete upravit charakteristiky fronty z výchozích lokálních front.

Poznámka: Výchozí lokální fronta je pojmenována SYSTEM.DEFAULT.LOCAL.QUEUE a byla vytvořena při instalaci systému.

Například příkaz DEFINE QLOCAL, který následuje, definuje frontu s názvem ORANGE.LOCAL.QUEUE s těmito charakteristikami:

- Je povoleno pro získávání, povoleno pro vkládání a pracuje na bázi pořadí priorit.
- Je to *normální* fronta; nejedná se o inicializační frontu nebo přenosovou frontu, a negeneruje zprávy spouštěče.
- Maximální hloubka fronty je 5000 zpráv; maximální délka zprávy je 4194304 bajtů.

```
DEFINE QLOCAL (ORANGE.LOCAL.QUEUE) +
  DESCR('Queue for messages from other systems') +
  PUT (ENABLED) +
  GET (ENABLED) +
  NOTRIGGER +
  MSGDLVSQ (PRIORITY) +
  MAXDEPTH (5000) +
  MAXMSGL (4194304) +
  USAGE (NORMAL);
```

Poznámka:

1. S výjimkou hodnoty pro popis jsou všechny zobrazené hodnoty atributů výchozími hodnotami. Ukázali jsme je zde pro účely ilustrace. Můžete je vynechat, pokud jste si jisti, že výchozí hodnoty jsou to, co chcete nebo které jste nezměnili. Další informace najdete v tématu [“Zobrazení výchozích atributů objektů”](#) na stránce 83.
2. USAGE (NORMAL) označuje, že tato fronta není přenosová fronta.
3. Pokud již máte lokální frontu ve stejném správcí front s názvem ORANGE.LOCAL.QUEUE, tento příkaz selže. Použijte atribut REPLACE, chcete-li přepsat existující definici fronty, ale prohlédněte si také [“Změna atributů lokální fronty”](#) na stránce 84.

Definování fronty nedoručených zpráv

Každý správce front musí mít lokální frontu, která má být použita jako fronta nedoručených zpráv, aby bylo možné uložit zprávy, které nelze doručit na jejich správné místo určení, pro pozdější načtení. Musíte o tom správcí front sdělit informace o frontě nedoručených zpráv.

Chcete-li správcí front sdělit informace o frontě nedoručených zpráv, zadejte název fronty nedoručených zpráv v příkazu `crtmqm` (například `crtmqm -u DEAD.LETTER.QUEUE`) nebo pomocí atributu `DEADQ` v příkazu `ALTER QMGR` určete jeden z nich později. Před použitím fronty nedoručených zpráv je třeba definovat frontu nedoručených zpráv.

Ukázkový dead-letter fronta s názvem SYSTEM.DEAD.LETTER.QUEUE je k dispozici s produktem. Tato fronta je automaticky vytvořena při vytvoření správce front. Tuto definici můžete v případě potřeby upravit a přejmenovat ji.

Fronta nedoručených zpráv nemá žádné speciální požadavky kromě následujících:

- Musí se jednat o lokální frontu
- Jeho atribut MAXMSGL (maximální délka zprávy) musí umožnit frontě tak, aby pojmula největší zprávy, které musí správce front zpracovat **plus** velikost záhlaví nedoručených zpráv (MQDLH)

Produkt WebSphere MQ poskytuje obslužnou rutinu fronty nedoručených zpráv, která umožňuje určit způsob zpracování nebo odebrání zpráv ve frontě nedoručených zpráv. Další informace najdete v tématu [Práce s nedoručené zprávy s manipulátorem fronty WebSphere MQ s dead-letter](#) .

Zobrazení výchozích atributů objektů

Příkaz DISPLAY QUEUE můžete použít k zobrazení atributů, které byly převzaty z výchozího objektu, když byl definován objekt WebSphere MQ .

Definujete-li objekt WebSphere MQ , použije se všechny atributy, které jste neuvedli z výchozího objektu. Když například definujete lokální frontu, zdědí tato fronta všechny atributy, které jste vynechali v definici z výchozí lokální fronty, která se nazývá SYSTEM.DEFAULT.LOCAL.QUEUE. Chcete-li vidět přesně, jaké jsou tyto atributy, použijte následující příkaz:

```
DISPLAY QUEUE (SYSTEM.DEFAULT.LOCAL.QUEUE)
```

Syntaxe tohoto příkazu se liší od syntaxe odpovídajícího příkazu DEFINE. V příkazu DISPLAY můžete zadat pouze název fronty, zatímco v příkazu DEFINE je třeba určit typ fronty, která je QLOCAL, QALIAS, QMODEL nebo QREMOTE.

Můžete selektivně zobrazovat atributy tak, že je určíte jednotlivě. Příklad:

```
DISPLAY QUEUE (ORANGE.LOCAL.QUEUE) +
    MAXDEPTH +
    MAXMSGL +
    CURDEPTH;
```

Tento příkaz zobrazí tři uvedené atributy, jak je uvedeno:

```
AMQ8409: Display Queue details.
    QUEUE (ORANGE.LOCAL.QUEUE)          TYPE (QLOCAL)
    CURDEPTH (0)                          MAXDEPTH (5000)
    MAXMSGL (4194304)
```

CURDEPTH je aktuální hloubka fronty, tj. počet zpráv ve frontě. Jedná se o užitečný atribut, který se má zobrazit, protože pomocí monitorování hloubky fronty můžete zajistit, že se fronta nebude zaplněna.

Kopírování definice lokální fronty

Definici fronty můžete zkopírovat pomocí atributu LIKE v příkazu DEFINE.

Příklad:

```
DEFINE QLOCAL (MAGENTA.QUEUE) +
    LIKE (ORANGE.LOCAL.QUEUE)
```

Tento příkaz vytvoří frontu se stejnými atributy jako naše původní fronta ORANGE.LOCAL.QUEUE místo těch, které mají výchozí lokální frontu systému. Zadejte název fronty, která má být zkopírována **přesně** , jak byla zadána při vytvoření fronty. Pokud název obsahuje malá písmena, uzavřete název do jednoduchých uvozovek.

Tento tvar příkazu DEFINE můžete také použít ke kopírování definice fronty, ale nahradit jednu nebo více změn atributů původního objektu. Příklad:

```
DEFINE QLOCAL (THIRD.QUEUE) +
```

```
LIKE (ORANGE.LOCAL.QUEUE) +  
MAXMSGL(1024);
```

Tento příkaz zkopíruje atributy fronty ORANGE.LOCAL.QUEUE do fronty THIRD.QUEUE, ale určuje, že maximální délka zprávy v nové frontě má být 1024 bajtů, nikoli 4194304.

Poznámka:

1. Použijete-li atribut LIKE v příkazu DEFINE, kopírujete pouze atributy fronty. Nekopírujete zprávy do fronty.
2. Pokud definujete lokální frontu, aniž byste uvedli LIKE, je to stejné jako DEFINE LIKE (SYSTEM.DEFAULT.LOCAL.QUEUE).

Změna atributů lokální fronty

Atributy fronty můžete změnit dvěma způsoby pomocí příkazu ALTER QLOCAL nebo příkazu DEFINE QLOCAL s atributem REPLACE.

V produktu [“Definování lokální fronty”](#) na stránce 82 jde o frontu s názvem ORANGE.LOCAL.QUEUE byla definována. Předpokládejme například, že chcete snížit maximální délku zprávy v této frontě na 10,000 bajtů.

- Použití příkazu ALTER:

```
ALTER QLOCAL (ORANGE.LOCAL.QUEUE) MAXMSGL(10000)
```

Tento příkaz změní jednotlivý atribut, který z maximální délky zprávy; všechny ostatní atributy zůstanou stejné.

- Použití příkazu DEFINE s volbou REPLACE, například:

```
DEFINE QLOCAL (ORANGE.LOCAL.QUEUE) MAXMSGL(10000) REPLACE
```

Tento příkaz mění nejen maximální délku zprávy, ale také všechny ostatní atributy, kterým jsou uvedeny jejich výchozí hodnoty. Fronta je nyní povolena, zatímco dříve byla zablokována. Povolené operace Put enabled je výchozí, jak je určeno frontou SYSTEM.DEFAULT.LOCAL.QUEUE.

Pokud **snížujete** maximální délku zprávy na existující frontě, stávající zprávy nebudou ovlivněny. Nové zprávy však musí splňovat nová kritéria.

Vymazání lokální fronty

K vyčištění lokální fronty můžete použít příkaz CLEAR.

Chcete-li odstranit všechny zprávy z lokální fronty s názvem MAGENTA.QUEUE, použijte následující příkaz:

```
CLEAR QLOCAL (MAGENTA.QUEUE)
```

Poznámka: Neexistuje žádný náznak, který vám umožní změnit svou mysl; jakmile stisknete klávesu Enter, zprávy se ztratí.

Frontu nelze vymazat, pokud:

- Existují nepotvrzené zprávy, které byly vloženy do fronty pod bodem synchronizace.
- Aplikace má aktuálně otevřenou frontu.

Odstranění lokální fronty

K odstranění lokální fronty můžete použít příkaz MQSC DELETE QLOCAL.

Frontu nelze odstranit, pokud na ní obsahuje nepotvrzené zprávy. Pokud však má fronta jednu nebo více potvrzených zpráv a žádné nepotvrzené zprávy, lze ji odstranit pouze v případě, že uvedete volbu PURGE. Příklad:

```
DELETE QLOCAL (PINK.QUEUE) PURGE
```

Zadání parametru NOURGE místo PURGE zajišťuje, že fronta nebude odstraněna, pokud obsahuje nějaké potvrzené zprávy.

Procházení front

Produkt WebSphere MQ nabízí ukázkový prohlížeč front, který můžete použít k prohlížení obsahu zpráv ve frontě. Prohlížeč je dodáván ve zdrojovém i spustitelném formátu.

MQ_INSTALLATION_PATH představuje adresář vysoké úrovně, do kterého je produkt WebSphere MQ nainstalován.

V produktu WebSphere MQ for Windows jsou názvy souborů a cesty pro prohlížeč ukázkových front následující:

Zdroj

```
MQ_INSTALLATION_PATH\tools\c\samples\
```

Spustitelné

```
MQ_INSTALLATION_PATH\tools\c\samples\bin\amqsbcg.exe
```

V produktu WebSphere MQ for UNIX and Linux jsou názvy souborů a cesty následující:

Zdroj

```
MQ_INSTALLATION_PATH/samp/amqsbcg0.c
```

Spustitelné

```
MQ_INSTALLATION_PATH/samp/bin/amqsbcg
```

Ukázka vyžaduje dva vstupní parametry, název fronty a název správce front. Příklad:

```
amqsbcg SYSTEM.ADMIN.QMGREVENT.tpp01 saturn.queue.manager
```

Typické výsledky z tohoto příkazu jsou zobrazeny v [Obrázek 16](#) na stránce 86.

Informace o plánování velikosti úložiště, které potřebujete pro fronty, naleznete na webu IBM WebSphere MQ pro sestavy výkonu specifické pro danou platformu:

<https://www.ibm.com/software/integration/ts/mqseries/>

Práce s frontami aliasů

Můžete definovat alias fronty tak, aby nepřímo odkazovala na jinou frontu nebo téma.

V 7.5.0.8



Upozornění: Distribuční seznamy nepodporují použití alias fronta, které odkazují na objekty tématu. Od Version 7.5.0, Fix Pack 8, jestliže alias fronta odkazuje na objekt tématu v distribučním seznamu, IBM WebSphere MQ vrací MQRC_ALIAS_BASE_Q_TYPE_ERROR.

Fronta, na kterou odkazuje alias fronta, může být některá z následujících:

- Lokální fronta (viz [“Definování lokální fronty”](#) na stránce 82).
- Lokální definice vzdálené fronty (viz [“Vytvoření lokální definice vzdálené fronty”](#) na stránce 111).
- Téma.

Alias fronta není skutečná fronta, ale definice, která se za běhu interpretuje jako skutečná (nebo cílová) fronta. Definice alias fronty určuje cílovou frontu. Když aplikace provede volání MQOPEN do alias fronty, správce front tento alias interpretuje na název cílové fronty.

Alias frontu nelze převést na jinou lokálně definovanou alias frontu. Alias fronty však může interpretovat alias fronty, které jsou definovány na jiném místě v klastrech, jejichž členem je lokální správce front. Další informace viz [Rozlišení názvů](#).

Alias fronty jsou užitečné pro:

- Poskytuje různým aplikacím různé úrovně přístupových oprávnění k cílové frontě.
- Umožňuje různým aplikacím pracovat se stejnou frontou různými způsoby. (Možná budete chtít přiřadit jiné výchozí priority nebo jiné výchozí hodnoty perzistence.)
- Zjednodušení údržby, migrace a vyrovnávání pracovní zátěže. (Možná budete chtít změnit název cílové fronty, aniž byste museli změnit aplikaci, která nadále používá alias.)

Předpokládejme například, že byla vyvinuta aplikace pro vložení zpráv do fronty s názvem MY.ALIAS.QUEUE. Uvádí název této fronty, když vytváří požadavek MQOPEN, a nepřímo, pokud vkládá zprávu do této fronty. Aplikace si není vědoma, že fronta je alias fronty. Pro každé volání MQI používající tento alias správce front interpretuje skutečný název fronty, což může být buď lokální fronta, nebo vzdálená fronta definovaná v tomto správcí front.

Změnou hodnoty atributu TARGET můžete přesměrovat volání MQI do jiné fronty, případně do jiného správce front. To je užitečné pro údržbu, migraci a vyrovnávání zátěže.

Definování alias fronty

Následující příkaz vytvoří alias frontu:

```
DEFINE QALIAS (MY.ALIAS.QUEUE) TARGET (YELLOW.QUEUE)
```

Tento příkaz přesměruje volání MQI, která určují MY.ALIAS.QUEUE do fronty YELLOW.QUEUE. Příkaz nevytvoří cílovou frontu; volání MQI se nezdaří, pokud fronta YELLOW.QUEUE za běhu neexistuje.

Změníte-li definici aliasu, můžete přesměrovat volání MQI do jiné fronty. Příklad:

```
ALTER QALIAS (MY.ALIAS.QUEUE) TARGET (MAGENTA.QUEUE)
```

Tento příkaz přesměruje volání MQI do jiné fronty, MAGENTA.QUEUE.

Můžete také použít alias fronty, aby se jedna fronta (cílová fronta) zdala mít různé atributy pro různé aplikace. To provedete definováním dvou aliasů, jednoho pro každou aplikaci. Předpokládejme, že existují dvě aplikace:

- Aplikace ALPHA může vkládat zprávy na YELLOW.QUEUE, ale není povoleno z ní získávat zprávy.
- Aplikace BETA může získat zprávy z YELLOW.QUEUE, ale není povoleno do ní vkládat zprávy.

Následující příkaz definuje alias, který je povolen a je zakázán pro aplikaci ALPHA:

```
DEFINE QALIAS (ALPHAS.ALIAS.QUEUE) +  
  TARGET (YELLOW.QUEUE) +  
  PUT (ENABLED) +  
  GET (DISABLED)
```

Následující příkaz definuje alias, který je zakázán a je povolen pro aplikaci BETA:

```
DEFINE QALIAS (BETAS.ALIAS.QUEUE) +  
  TARGET (YELLOW.QUEUE) +  
  PUT (DISABLED) +  
  GET (ENABLED)
```

ALPHA používá název fronty ALPHAS.ALIAS.QUEUE ve svých voláních MQI; BETA používá název fronty BETAS.ALIAS.QUEUE. Oba přistupují ke stejné frontě, ale různými způsoby.

Atributy LIKE a REPLACE můžete použít při definování aliasů fronty stejným způsobem, jakým používáte tyto atributy s lokálními frontami.

Použití jiných příkazů s frontami aliasů

Příslušné příkazy MQSC můžete použít k zobrazení nebo změně atributů alias fronty nebo k odstranění objektu alias fronty. Příklad:

Pomocí následujícího příkazu zobrazte atributy alias fronty:

```
DISPLAY QALIAS (ALPHAS.ALIAS.QUEUE)
```

Použijte následující příkaz ke změně názvu základní fronty, na kterou se alias interpretuje, kde volba `force` vynutí změnu, i když je fronta otevřená:

```
ALTER QALIAS (ALPHAS.ALIAS.QUEUE) TARGET(ORANGE.LOCAL.QUEUE) FORCE
```

K odstranění tohoto aliasu fronty použijte následující příkaz:

```
DELETE QALIAS (ALPHAS.ALIAS.QUEUE)
```

Alias fronty nelze odstranit, pokud má aplikace aktuálně otevřenou frontu. Další informace o tomto a dalších příkazech alias fronty naleznete v tématu [Odkaz na MQSC](#).

Práce s modelové fronty

Správce front vytvoří *dynamickou frontu* v případě, že přijme volání MQI z aplikace určující název fronty, který byl definován jako modelová fronta. Název nové dynamické fronty je generován správcem front při vytvoření fronty. *Modelová fronta* je šablona, která uvádí atributy všech dynamických front vytvořených z této fronty. Modelové fronty poskytují vhodnou metodu pro aplikace k vytvoření front podle potřeby.

Definování modelové fronty

Frontu modelu definujete spolu se sadou atributů stejným způsobem, jakým definujete lokální frontu. Modelové fronty a lokální fronty mají stejnou sadu atributů, kromě toho, že u modelových front můžete určit, zda jsou vytvořené dynamické fronty dočasné nebo trvalé. (Trvalé fronty se udržují ve všech restartech správce front, dočasné fronty nikoli.) Příklad:

```
DEFINE QMODEL (GREEN.MODEL.QUEUE) +
  DESCR('Queue for messages from application X') +
  PUT (DISABLED) +
  GET (ENABLED) +
  NOTRIGGER +
  MSGDLVSQ (FIFO) +
  MAXDEPTH (1000) +
  MAXMSGL (2000) +
  USAGE (NORMAL) +
  DEFTYPE (PERMDYN)
```

Tento příkaz vytvoří definici modelové fronty. Z atributu DEFTYPE můžete vidět, že skutečné fronty vytvořené z této šablony jsou trvalé dynamické fronty. Jakékoli uvedené atributy se automaticky nezkopírují z `SYSTEM.DEFAULT.MODEL.QUEUE` výchozí fronta.

Atributy `LIKE` a `REPLACE` můžete použít při definování modelových front stejným způsobem, jako je používáte s lokálními frontami.

Použití jiných příkazů s modelové fronty

Můžete použít příslušné příkazy `MQSC` k zobrazení nebo změně atributů modelové fronty nebo k odstranění objektu modelové fronty. Příklad:

Chcete-li zobrazit atributy modelové fronty, použijte následující příkaz:

```
DISPLAY QUEUE (GREEN.MODEL.QUEUE)
```

Pomocí následujícího příkazu můžete změnit model, který má povolit vložení do jakékoli dynamické fronty vytvořené z tohoto modelu:

```
ALTER QMODEL (BLUE.MODEL.QUEUE) PUT(ENABLED)
```

K odstranění této modelové fronty použijte tento příkaz:

```
DELETE QMODEL (RED.MODEL.QUEUE)
```

Práce s administrativními tématy

Spravujete-li administrativní témata, použijte příkazy `MQSC`.

Podrobné informace o těchto příkazech najdete v příručce [Odkaz na MQSC](#).

Související pojmy

[Objekty administrativního tématu](#)

[“Definování administrativního tématu” na stránce 90](#)

Použijte příkaz `MQSC DEFINE TOPIC` k vytvoření administrativního tématu. Při definování administrativního tématu můžete volitelně nastavit každý atribut tématu.

[“Zobrazení atributů objektu administrativního tématu” na stránce 90](#)

Chcete-li zobrazit objekt administrativního tématu, použijte příkaz `MQSC DISPLAY TOPIC`.

[“Změna atributů administrativního tématu” na stránce 91](#)

Atributy témat můžete změnit dvěma způsoby pomocí příkazu `ALTER TOPIC` nebo příkazu `DEFINE TOPIC` s atributem `REPLACE`.

[“Kopírování definice administrativního tématu”](#) na stránce 91

Definici tématu můžete kopírovat s použitím atributu LIKE v příkazu **DEFINE** .

[“Odstranění definice administrativního tématu”](#) na stránce 92

K odstranění administrativního tématu můžete použít příkaz MQSC **DELETE TOPIC** .

Definování administrativního tématu

Použijte příkaz MQSC **DEFINE TOPIC** k vytvoření administrativního tématu. Při definování administrativního tématu můžete volitelně nastavit každý atribut tématu.

Jakýkoli atribut tématu, který není explicitně nastaven, je zděděn z výchozího administrativního tématu SYSTEM.DEFAULT.TOPIC, který byl vytvořen při instalaci instalace systému.

Například, příkaz **DEFINE TOPIC** , který následuje, definuje téma s názvem **ORANGE . TOPIC** s těmito charakteristikami:

- Interpretuje řetězec tématu ORANGE. Informace o tom, jak lze používat řetězce témat, najdete v tématu [Kombinování řetězců témat](#).
- Jakýkoli atribut, který je nastaven na ASPARENT, používá atribut tak, jak je definován nadřazeným tématem tohoto tématu. Tato akce se opakuje stromem témat, pokud jde o kořenové téma, SYSTEM.BASE.TOPIC je nalezeno. Další informace o stromech témat najdete v tématu [Stromy témat](#).

```
DEFINE TOPIC (ORANGE.TOPIC) +  
        TOPICSTR (ORANGE) +  
        DEFPRTY(ASPARENT) +  
        NPMSGDLV(ASPARENT)
```

Poznámka:

- S výjimkou hodnoty řetězce tématu jsou všechny zobrazené hodnoty atributů výchozími hodnotami. Zobrazují se zde pouze jako ilustrace. Můžete je vynechat, pokud jste si jisti, že výchozí hodnoty jsou to, co chcete nebo které jste nezměnili. Další informace najdete v tématu [“Zobrazení atributů objektu administrativního tématu”](#) na stránce 90.
- Pokud již máte administrativní téma na stejném správci front s názvem ORANGE.TOPIC, tento příkaz selže. Použijte atribut REPLACE, chcete-li přepsat existující definici tématu, ale prohlédněte si také [“Změna atributů administrativního tématu”](#) na stránce 91

Zobrazení atributů objektu administrativního tématu

Chcete-li zobrazit objekt administrativního tématu, použijte příkaz MQSC **DISPLAY TOPIC** .

Chcete-li zobrazit všechna témata, použijte:

```
DISPLAY TOPIC(ORANGE.TOPIC)
```

Můžete selektivně zobrazovat atributy tak, že je určíte jednotlivě.Příklad:

```
DISPLAY TOPIC(ORANGE.TOPIC) +  
        TOPICSTR +  
        DEFPRTY +  
        NPMSGDLV
```

Tento příkaz zobrazí tři uvedené atributy, jak je uvedeno:

```
AMQ8633: Display topic details.  
TOPIC(ORANGE.TOPIC)                TYPE(LOCAL)  
TOPICSTR(ORANGE)                    DEFPRTY(ASPARENT)  
NPMSGDLV(ASPARENT)
```

Chcete-li zobrazit hodnoty ASPARENT tak, jak jsou použity v době běhu programu, použijte příkaz **DISPLAY TPSTATUS**. Např. použijte:

```
DISPLAY TPSTATUS(ORANGE) DEFPRTY NPMSGDLV
```

Příkaz zobrazí následující podrobnosti:

```
AMQ8754: Display topic status details.  
TOPICSTR(ORANGE) DEFPRTY(0)  
NPMSGDLV(ALLAVAIL)
```

Definujete-li administrativní téma, vezme všechny atributy, které explicitně nezadáte, explicitně z výchozího administrativního tématu, které se nazývá SYSTEM.DEFAULT.TOPIC. Chcete-li zjistit, jaké jsou tyto výchozí atributy, použijte následující příkaz:

```
DISPLAY TOPIC (SYSTEM.DEFAULT.TOPIC)
```

Změna atributů administrativního tématu

Atributy témat můžete změnit dvěma způsoby pomocí příkazu **ALTER TOPIC** nebo příkazu **DEFINE TOPIC** s atributem **REPLACE**.

Chcete-li například změnit výchozí prioritu zpráv doručených na téma s názvem ORANGE.TOPIC, aby bylo 5, použijte jeden z následujících příkazů.

- Pomocí příkazu **ALTER** :

```
ALTER TOPIC(ORANGE.TOPIC) DEFPRTY(5)
```

Tento příkaz mění jeden atribut, který má výchozí prioritu zprávy doručené na toto téma do 5; všechny ostatní atributy zůstanou stejné.

- Pomocí příkazu **DEFINE** :

```
DEFINE TOPIC(ORANGE.TOPIC) DEFPRTY(5) REPLACE
```

Tento příkaz změní výchozí prioritu zpráv doručených do tohoto tématu. Všechny ostatní atributy mají své výchozí hodnoty.

Změníte-li prioritu zpráv odeslaných do tohoto tématu, nebudou ovlivněny existující zprávy. Jakákoli nová zpráva však použije uvedenou prioritu, pokud ji nezajišťuje publikující aplikace.

Kopírování definice administrativního tématu

Definici tématu můžete kopírovat s použitím atributu LIKE v příkazu **DEFINE**.

Příklad:

```
DEFINE TOPIC (MAGENTA.TOPIC) +  
LIKE (ORANGE.TOPIC)
```

Tento příkaz vytvoří téma MAGENTA.TOPIC, se stejnými atributy jako původní téma, ORANGE.TOPIC, spíše než výchozí administrativní téma systému. Zadejte název tématu, které má být zkopírováno přesně tak, jak bylo zadáno při vytváření tématu. Pokud název obsahuje malá písmena, uzavřete název do jednoduchých uvozovek.

Tento tvar příkazu **DEFINE** můžete také použít ke kopírování definice tématu, ale provést změny v attributech původní. Příklad:

```
DEFINE TOPIC(BLUE.TOPIC) +  
TOPICSTR(BLUE) +  
LIKE(ORANGE.TOPIC)
```

Také můžete kopírovat atributy tématu BLUE.TOPIC na téma GREEN.TOPIC a zadejte, že když nemohou být publikace doručeny do jejich správné fronty odběratele, nebudou umístěny do fronty nedoručených zpráv. Příklad:

```
DEFINE TOPIC(GREEN.TOPIC) +  
TOPICSTR(GREEN) +
```

Odstranění definice administrativního tématu

K odstranění administrativního tématu můžete použít příkaz MQSC **DELETE TOPIC** .

```
DELETE TOPIC (ORANGE.TOPIC)
```

Aplikace již nebudou moci otevřít téma pro publikování nebo vytvořit nové odběry s použitím názvu objektu, ORANGE.TOPIC. Publikování aplikací, které mají otevřeno téma, je schopno pokračovat v publikování vyřešeného řetězce tématu. Všechny odběry, které již byly v tomto tématu provedeny, budou nadále přijímat publikování po odstranění tématu.

Aplikace, které se na tento objekt tématu neodkazují, ale používají interpretovaný řetězec tématu, který tento objekt tématu představoval, 'ORANGE' v tomto příkladu pokračuje v práci. V takovém případě dědí vlastnosti z objektu tématu ve stromu témat. Další informace o stromech témat najdete v tématu [Stromy témat](#).

Práce s odběry

Správa odběrů pomocí příkazů MQSC.

Odběry mohou být jedním ze tří typů, definovaných v atributu **SUBTYPE** :

ADMIN

Administrativně definovaný uživatelem.

PROXY

Interně vytvořený odběr pro směrování publikací mezi správci front.

Rozhraní API

Vytvořeno programově, například pomocí volání MQSUB MQI.

Chcete-li získat podrobné informace o těchto příkazech, prohlédněte si příručku [Odkaz na MQSC](#) .

Související pojmy

[“Definování administrativního odběru” na stránce 92](#)

Použijte příkaz MQSC **DEFINE SUB** k vytvoření administrativního odběru. Můžete také použít výchozí definici definovanou ve výchozí lokální definici odběru. Nebo můžete upravit charakteristiky odběru z výchozích lokálních odběrů, SYSTEM.DEFAULT.SUB , která byla vytvořena při instalaci systému.

[“Zobrazení atributů odběrů” na stránce 93](#)

Pomocí příkazu **DISPLAY SUB** lze zobrazit konfigurované atributy všech odběrů známých správci front.

[“Změna atributů lokálního odběru” na stránce 94](#)

Atributy odběru můžete změnit dvěma způsoby pomocí příkazu **ALTER SUB** nebo příkazu **DEFINE SUB** s atributem **REPLACE** .

[“Kopírování definice lokálního odběru” na stránce 94](#)

Definici odběru můžete kopírovat pomocí atributu **LIKE** u příkazu **DEFINE** .

[“Odstranění odběru” na stránce 95](#)

Chcete-li odstranit lokální odběr, můžete použít příkaz MQSC **DELETE SUB** .

Definování administrativního odběru

Použijte příkaz MQSC **DEFINE SUB** k vytvoření administrativního odběru. Můžete také použít výchozí definici definovanou ve výchozí lokální definici odběru. Nebo můžete upravit charakteristiky odběru z výchozích lokálních odběrů, SYSTEM.DEFAULT.SUB , která byla vytvořena při instalaci systému.

Například příkaz **DEFINE SUB** , který následuje, definuje odběr s názvem ORANGE s těmito charakteristikami:

- Trvalý odběr, což znamená, že přetrvává po restartování správce front s neomezeným vypršením platnosti.
- Obdrží publikace vytvořené v řetězci tématu ORANGE s prioritami zpráv nastaveným publikujícími aplikacemi.
- Publikace doručené pro tento odběr se odesílají do lokální fronty SUBQ, tato fronta musí být definována před definicí odběru.

```
DEFINE SUB (ORANGE) +
  TOPICSTR (ORANGE) +
  DESTCLAS (PROVIDED) +
  DEST (SUBQ) +
  EXPIRY (UNLIMITED) +
  PUBPTY (AS PUB)
```

Poznámka:

- Název odběru a názvu řetězce tématu se nemusí shodovat.
- S výjimkou hodnot popisu a řetězce tématu jsou všechny zobrazené hodnoty atributů výchozími hodnotami. Zobrazují se zde pouze jako ilustrace. Můžete je vynechat, pokud jste si jisti, že výchozí hodnoty jsou to, co chcete nebo které jste nezměnili. Další informace najdete v tématu [“Zobrazení atributů odběrů”](#) na stránce 93.
- Pokud již máte lokální odběr u stejného správce front s názvem TEST, tento příkaz selže. Použijte atribut **REPLACE**, chcete-li přepsat existující definici fronty, ale prohlédněte si také [“Změna atributů lokálního odběru”](#) na stránce 94.
- Pokud fronta SUBQ neexistuje, tento příkaz selže.

Zobrazení atributů odběrů

Pomocí příkazu **DISPLAY SUB** lze zobrazit konfigurované atributy všech odběrů známých správci front.

Např. použijte:

```
DISPLAY SUB (ORANGE)
```

Můžete selektivně zobrazovat atributy tak, že je určíte jednotlivě. Příklad:

```
DISPLAY SUB (ORANGE) +
  SUBID +
  TOPICSTR +
  DURABLE
```

Tento příkaz zobrazí tři uvedené atributy, jak je uvedeno:

```
AMQ8096: WebSphere MQ subscription inquired.
  SUBID(414D5120414141202020202020202020EE921E4E20002A03)
  SUB(ORANGE) TOPICSTR(ORANGE)
  DURABLE(YES)
```

TOPICSTR je vyřešený řetězec tématu, na kterém je tento odběratel v provozu. Je-li definován odběr pro použití objektu tématu, bude řetězec tématu z tohoto objektu použit jako předpona k řetězci tématu poskytovaného při vytváření odběru. SUBID je jedinečný identifikátor přiřazený správcem front při vytvoření odběru. Jedná se o užitečný atribut, který se má zobrazit, protože některé názvy odběrů mohou být dlouhé nebo v různých znakových sadách, pro které by se mohlo stát nepraktické.

Alternativní metodou pro zobrazení odběrů je použití ID SUBID:

```
DISPLAY SUB +
  SUBID(414D5120414141202020202020202020EE921E4E20002A03) +
  TOPICSTR +
  DURABLE
```

Tento příkaz vydá stejný výstup jako předtím:

```
AMQ8096: WebSphere MQ subscription inquired.  
SUBID(414D51204141412020202020202020EE921E4E20002A03)  
SUB(ORANGE) TOPICSTR(ORANGE)  
DURABLE(YES)
```

Proxy odběry na správci front se při výchozím nastavení nezobrazují. Chcete-li je zobrazit, zadejte **SUBTYPE PROXY** nebo **ALL**.

Příkaz `DISPLAY SBSTATUS` můžete použít k zobrazení atributů běhového prostředí. Použijte například příkaz:

```
DISPLAY SBSTATUS(ORANGE) NUMMSGs
```

Zobrazí se následující výstup:

```
AMQ8099: WebSphere MQ subscription status inquired.  
SUB(ORANGE)  
SUBID(414D51204141412020202020202020EE921E4E20002A03)  
NUMMSGs(0)
```

Definujete-li administrativní odběr, vezme se v něm všechny atributy, které explicitně neurčujete z výchozího odběru, který se nazývá `SYSTEM.DEFAULT.SUB`. Chcete-li zjistit, jaké jsou tyto výchozí atributy, použijte následující příkaz:

```
DISPLAY SUB (SYSTEM.DEFAULT.SUB)
```

Změna atributů lokálního odběru

Atributy odběru můžete změnit dvěma způsoby pomocí příkazu **ALTER SUB** nebo příkazu **DEFINE SUB** s atributem **REPLACE**.

Chcete-li například změnit prioritu zpráv doručených na odběr s názvem `ORANGE` na hodnotu 5, použijte některý z následujících příkazů:

- Použití příkazu **ALTER**:

```
ALTER SUB(ORANGE) PUBPRTY(5)
```

Tento příkaz změní jeden atribut, který má prioritu zpráv doručených do tohoto odběru na 5; všechny ostatní atributy zůstanou stejné.

- Použití příkazu **DEFINE**:

```
DEFINE SUB (ORANGE) PUBPRTY(5) REPLACE
```

Tento příkaz mění nejen prioritu zpráv doručených do tohoto odběru, ale všechny ostatní atributy, kterým jsou uvedeny jejich výchozí hodnoty.

Pokud změníte prioritu zpráv odeslaných na tento odběr, nebudou ovlivněny existující zprávy. Jakékoli nové zprávy však mají zadanou prioritu.

Kopírování definice lokálního odběru

Definici odběru můžete kopírovat pomocí atributu **LIKE** u příkazu **DEFINE**.

Příklad:

```
DEFINE SUB (BLUE) +  
LIKE (ORANGE)
```

Také můžete zkopírovat atributy `subREAL` do `sub THIR.D.SUBa` uvést, že `correlID` doručených publikací je třetí, spíše než vydavatelé `correlID`. Příklad:

```
DEFINE SUB(THIRD.SUB) +  
LIKE(BLUE) +  
DETCORL(ORANGE)
```

Odstranění odběru

Chcete-li odstranit lokální odběr, můžete použít příkaz MQSC **DELETE SUB** .

```
DELETE SUB(ORANGE)
```

Odběr můžete odstranit také pomocí ID SUBID:

```
DELETE SUB SUBID(414D51204141412020202020202020EE921E4E20002A03)
```

Kontrola zpráv v odběru

Informace o této úloze

Je-li definován odběr, je přidružen ke frontě. Publikované zprávy odpovídající tomuto odběru jsou vloženy do této fronty.

Všimněte si, že následující příkazy **runmqsc** zobrazují pouze ty odběry, které obdržely zprávy.

Chcete-li zkontrolovat zprávy, které jsou momentálně ve frontě pro odběr, proveďte následující kroky:

Postup

1. Chcete-li zkontrolovat zprávy ve frontě pro typ odběru **DISPLAY SBSTATUS(<sub_name> NUMMSGS**, prohlédněte si téma [“Zobrazení atributů odběrů”](#) na stránce 93.
2. Je-li hodnota **NUMMSGS** větší než nula, identifikujte frontu přidruženou k odběru zadáním příkazu **DISPLAY SUB(<sub_name>)DEST**.
3. Při použití názvu fronty, který je vrácen, můžete zprávy zobrazit pomocí postupu popsaného v tématu [“Procházení front”](#) na stránce 85.

Práce se službami

Servisní objekty představují prostředky, pomocí kterých mohou být další procesy spravovány jako součást správce front. Se službami můžete definovat programy, které jsou spuštěny a zastaveny při spuštění a ukončení správce front. Služby produktu IBM WebSphere MQ jsou vždy spuštěny pod ID uživatele, který spustil správce front.

Objekty služby mohou být buď následujících typů:

Server

Server je objekt služby, který má parametr **SERVTYPE** určený jako **SERVER**. Objekt služby serveru je definice programu, který se provádí, když je spuštěn zadaný správce front. Objekty služeb serveru definují programy, které se obvykle spouští po dlouhou dobu. Objekt služby serveru lze například použít k provedení procesu monitoru spouštěčů, jako je například **runmqtrm**.

Souběžně může být spuštěna pouze jedna instance objektu služby serveru. Stav spuštěných objektů služeb serveru může být monitorován pomocí příkazu MQSC, **DISPLAY SVSTATUS**.

Příkaz

Příkaz je objekt služby, který má parametr **SERVTYPE** zadaný jako **COMMAND**. Objekty služby příkazu jsou podobné objektům služby serveru, ale souběžně může být spuštěno více instancí objektu příkazového služby a jejich stav nemůže být monitorován pomocí příkazu MQSC **DISPLAY SVSTATUS**.

Pokud příkaz MQSC, **STOP SERVICE**, neprovede žádnou kontrolu, aby se zjistilo, zda program spuštěný příkazem MQSC, **START SERVICE**, je stále aktivní před provedením ukončovacího programu.

Definování objektu služby

Objekt služby definujete s různými atributy.

Atributy jsou následující:

SERVTYPE

Definuje typ objektu služby. Možné hodnoty jsou následující:

SERVER

Objekt služby serveru.

V daném okamžiku může být spuštěna pouze jedna instance objektu služby serveru. Stav objektů služby serveru může být monitorován pomocí příkazu MQSC, DISPLAY SVSTATUS.

PŘÍKAZ

Objekt služby příkazu.

Souběžně lze provádět více instancí objektu služby příkazů. Stav objektů služby příkazů nelze monitorovat.

STARTCMD

Program, který se spustí ke spuštění služby. Musí být zadána úplná cesta k programu.

STARTARG

Argumenty předané do počátečního programu.

STDERR

Uvádí cestu k souboru, do kterého by měla být přesměrována standardní chyba (stderr) servisního programu.

STDOUT

Uvádí cestu k souboru, do kterého by měl být přesměrován standardní výstup (stdout) servisního programu.

STOPCMD

Program, který se provádí za účelem zastavení služby. Musí být zadána úplná cesta k programu.

STOPARG

Argumenty předané do ukončovacího programu.

CONTROL

Uvádí, jak se má služba spustit a zastavit:

RUČNÍ

Služba se nespustí automaticky nebo automaticky zastavovat. Je řízen pomocí příkazů START SERVICE a STOP SERVICE. Toto je výchozí hodnota.

QMGR

Definovaná služba má být spuštěna a zastavena současně s tím, jak je spuštěn a zastaven správce front.

POUZE SPUŠTĚNÍ

Služba se spustí ve stejnou dobu, kdy je správce front spuštěn, ale není požadováno zastavení při zastavení správce front.

Související pojmy

[“Správa služeb” na stránce 96](#)

Pomocí parametru CONTROL může být instance objektu služby buď spuštěna a automaticky zastavena správcem front, nebo spuštěna a zastavena pomocí příkazů MQSC START SERVICE a STOP SERVICE.

Správa služeb

Pomocí parametru CONTROL může být instance objektu služby buď spuštěna a automaticky zastavena správcem front, nebo spuštěna a zastavena pomocí příkazů MQSC START SERVICE a STOP SERVICE.

Když je spuštěna instance objektu služby, zpráva se zapíše do protokolu chyb správce front obsahující název objektu služby a ID procesu spuštěného procesu. Níže uvedená položka protokolu pro objekt služby serveru začíná takto:

```
02/15/2005 11:54:24 AM - Process(10363.1) User(mqm) Program(amqzmgr0)
Host(HOST_1) Installation(Installation1)
VRMF(7.1.0.0) QMgr(A.B.C)
AMQ5028: The Server 'S1' has started. ProcessId(13031).

EXPLANATION:
The Server process has started.
ACTION:
None.
```

Níže uvedená položka protokolu pro objekt služby příkazů začíná takto:

```
02/15/2005 11:53:55 AM - Process(10363.1) User(mqm) Program(amqzmgr0)
Host(HOST_1) Installation(Installation1)
VRMF(7.1.0.0) QMgr(A.B.C)
AMQ5030: The Command 'C1' has started. ProcessId(13030).

EXPLANATION:
The Command has started.
ACTION:
None.
```

Když se služba instance serveru zastaví, zpráva se zapíše do protokolů chyb správce front obsahující název služby a ID procesu ukončení procesu. Následuje příklad položky protokolu pro zastavovací objekt služby serveru:

```
02/15/2005 11:54:54 AM - Process(10363.1) User(mqm) Program(amqzmgr0)
Host(HOST_1) Installation(Installation1)
VRMF(7.1.0.0) QMgr(A.B.C)
AMQ5029: The Server 'S1' has ended. ProcessId(13031).

EXPLANATION:
The Server process has ended.
ACTION:
None.
```

Související odkazy

[“Další proměnné prostředí” na stránce 97](#)

Při spuštění služby je prostředí, ve kterém je proces služby spuštěn, zděděno z prostředí správce front. Je možné definovat další proměnné prostředí, které se mají nastavit v prostředí procesu služby, přidáním proměnných, které chcete definovat, do jednoho ze souborů potlačení prostředí `service.env`.

Další proměnné prostředí

Při spuštění služby je prostředí, ve kterém je proces služby spuštěn, zděděno z prostředí správce front. Je možné definovat další proměnné prostředí, které se mají nastavit v prostředí procesu služby, přidáním proměnných, které chcete definovat, do jednoho ze souborů potlačení prostředí `service.env`.

Poznámka:

Existují dva možné soubory, do kterých můžete přidat proměnné prostředí:

- Soubor `service.env` rozsahu počítače, který je umístěn v adresáři `/var/mqm` na systémech UNIX and Linux nebo v datovém adresáři vybraném během instalace na systémech Windows.
- Soubor `service.env` oboru správce front, který je umístěn v datovém adresáři správce front. Například umístění souboru potlačení prostředí pro správce front s názvem QMNAME je:
 - Na systémech UNIX and Linux `/var/mqm/qmgrs/QMNAME/service.env`
 - Na systémech Windows `C:\Program Files\IBM\WebSphere MQ\qmgrs\QMNAME\service.env`

Oba soubory jsou zpracovány, jsou-li k dispozici, s definicemi v souboru oboru správce front, které mají přednost před definicemi v souboru oboru počítače.

V souboru `service.env` lze zadat libovolnou proměnnou prostředí. Pokud například služba IBM WebSphere MQ spouští určitý počet příkazů, může být užitečné nastavit uživatelskou proměnnou `PATH` v souboru `service.env`. Hodnoty, které nastavíte, nemohou být proměnnými prostředí; například `CLASSPATH=%CLASSPATH%` je chybný. Podobně by na Linux `PATH=$PATH:/opt/mqm/bin` poskytovaly neočekávané výsledky.

`CLASSPATH` musí být velkými písmeny a příkaz cesty ke třídě může obsahovat pouze literály. Některé služby (například Telemetrie) nastavují vlastní cestu ke třídám. Přidá se k ní cesta `CLASSPATH` definovaná v souboru `service.env`.

Formát proměnných definovaných v souboru `service.env`, je seznam dvojic proměnných názvu a hodnoty. Každá proměnná musí být definována na novém řádku a každá proměnná je převzata tak, jak je explicitně definována, včetně mezer. Následuje příklad souboru `service.env`:

```
#####  
##  
## <N_OCO_COPYRIGHT> ##  
## Licensed Materials - Property of IBM ##  
## ##  
## 63H9336 ##  
## (C) Copyright IBM Corporation 2005, 2024. ##  
## ##  
## <NOC_COPYRIGHT> ##  
## ##  
#####  
#####  
## Module Name: service.env ##  
## Type : WebSphere MQ service environment file ##  
## Function : Define additional environment variables to be set ##  
## for SERVICE programs. ##  
## Usage : <VARIABLE>=<VALUE> ##  
## ##  
#####  
MYLOC=/opt/myloc/bin  
MYTMP=/tmp  
TRACEDIR=/tmp/trace  
MYINITQ=ACCOUNTS.INITIATION.QUEUE
```

Související odkazy

“Nahraditelné vložené do definic služeb” na stránce 98

V definici objektu služby je možné nahradit tokeny. Symboly, které jsou nahrazovány, jsou při provádění servisního programu automaticky nahrazeny jejich rozbaleným textem. Náhradní tokeny lze odebrat z následujícího seznamu obecných tokenů nebo z libovolných proměnných, které jsou definovány v souboru `service.env`.

Nahraditelné vložené do definic služeb

V definici objektu služby je možné nahradit tokeny. Symboly, které jsou nahrazovány, jsou při provádění servisního programu automaticky nahrazeny jejich rozbaleným textem. Náhradní tokeny lze odebrat z následujícího seznamu obecných tokenů nebo z libovolných proměnných, které jsou definovány v souboru `service.env`.

Zde jsou uvedeny běžné tokeny, které lze použít k nahrazení tokenů v definici objektu služby:

INSTALAČNÍ_CESTA MQ_INSTALL_PATH

Umístění, kde je nainstalován produkt WebSphere MQ .

MQ_DATA_PATH

Umístění datového adresáře produktu WebSphere MQ :

- V systémech UNIX and Linux je umístění datového adresáře produktu WebSphere MQ `/var/mqm/` .
- V systémech Windows je umístěním datového adresáře produktu WebSphere MQ datový adresář vybraný během instalace produktu WebSphere MQ .

QMNAME

Název aktuálního správce front.

NÁZEV_SLUŽBY_MQ

Název služby.

ID_PČ_SERVERU

Tento token může být použit pouze argumenty STOPARG a STOPCCMD.

Pro objekty služby serveru je tento token nahrazen ID procesu procesu spuštěným argumenty STARTCMD a STARTTARG. Jinak bude tento token nahrazen hodnotou 0.

MQ_Q_MGR_DATA_PATH

Umístění datového adresáře správce front.

MQ_MGR_DATA_NAME

Přetřansformovaný název správce front. Další informace o transformaci názvu naleznete v tématu [Základní informace o názvech souborů produktu WebSphere MQ](#).

To use replaceable inserts, insert the token within + characters into any of the STARTCMD, STARTARG, STOPCMD, STOPARG, STDOUT or STDERR strings. Příklady této informace viz [“Příklady použití objektů služeb”](#) na stránce 99.

Příklady použití objektů služeb

Služby v této sekci jsou zapsány se znaky oddělovače cesty ve stylu systému UNIX , není-li uvedeno jinak.

Použití objektu služby serveru

Tento příklad ukazuje, jak definovat, použít a změnit objekt služby serveru ke spuštění monitoru spouštěčů.

1. Objekt služby serveru je definován pomocí následujícího příkazu MQSC:

```
DEFINE SERVICE(S1) +
  CONTROL(QMGR) +
  SERVTYPE(SERVER) +
  STARTCMD('+MQ_INSTALL_PATH+bin/!unmqtm') +
  STARTARG('-m +QMNAME+ -q ACCOUNTS.INITIATION.QUEUE') +
  STOPCMD('+MQ_INSTALL_PATH+bin/amqsstop') +
  STOPARG('-m +QMNAME+ -p +MQ_SERVER_PID+')
```

Kde:

+MQ_INSTALL_PATH+ je token, který představuje instalační adresář.

+QMNAME+ je token představující název správce front.

ACCOUNTS.INITIATION.QUEUE je inicializační fronta.

amqsstop je ukázkový program dodávaný s produktem WebSphere MQ , který požaduje, aby správce front přerušl všechna připojení pro ID procesu. Příkaz amqsstop generuje příkazy PCF, proto musí být spuštěn příkazový server.

+MQ_SERVER_PID+ je token představující ID procesu předaný do programu zastavení.

Seznam obecných tokenů viz [“Nahraditelné vložené do definic služeb”](#) na stránce 98 .

2. Instance objektu služby serveru bude spuštěna při příštím spuštění správce front. Instance objektu služby serveru se však spustí okamžitě s následujícím příkazem MQSC:

```
START SERVICE(S1)
```

3. Stav procesu služby serveru se zobrazí pomocí následujícího příkazu MQSC:

```
DISPLAY SVSTATUS(S1)
```

4. Tento příklad nyní ukazuje, jak změnit objekt služby serveru a aktualizace jsou převzaty ručním restartováním procesu služby serveru. Objekt služby serveru je změněn tak, že inicializační fronta je zadána jako JUPITER . INITIATION . QUEUE . Je použit následující příkaz MQSC:

```
ALTER SERVICE(S1) +  
  STARTARG('-m +QMNAME+ -q JUPITER.INITIATION.QUEUE')
```

Poznámka: Spuštěná služba nevybere žádné aktualizace své definice služby, dokud nebude restartována.

5. Proces služby serveru je restartován, aby došlo k úpravě změn pomocí následujících příkazů MQSC:

```
STOP SERVICE(S1)
```

Následovány:

```
START SERVICE(S1)
```

Proces služby serveru je restartován a vyzvedne změny provedené v produktu “4” na stránce 100.

Poznámka: Příkaz MQSC, STOP SERVICE, lze použít pouze tehdy, je-li v definici služby uveden argument STOPCMD.

Použití objektu služby příkazu

Tento příklad uvádí, jak definovat objekt služby příkazu pro spuštění programu, který zapisuje záznamy do systémového protokolu operačního systému, když je správce front spuštěn nebo zastaven.

1. Objekt služby příkazu je definován pomocí následujícího příkazu MQSC:

```
DEFINE SERVICE(S2) +  
  CONTROL(QMGR) +  
  SERVTYPE(COMMAND) +  
  STARTCMD('/usr/bin/logger') +  
  STARTARG('Queue manager +QMNAME+ starting') +  
  STOPCMD('/usr/bin/logger') +  
  STOPARG('Queue manager +QMNAME+ stopping')
```

Kde:

logger je systémem dodaný příkaz UNIX and Linux pro zápis do systémového protokolu.
+QMNAME+ je token představující název správce front.

Použití objektu služby příkazu při ukončení pouze správce front

Tento příklad uvádí, jak definovat objekt služby příkazu pro spuštění programu, který zapisuje záznamy do systémového protokolu operačního systému, když je správce front zastaven pouze.

1. Objekt služby příkazu je definován pomocí následujícího příkazu MQSC:

```
DEFINE SERVICE(S3) +  
  CONTROL(QMGR) +  
  SERVTYPE(COMMAND) +  
  STOPCMD('/usr/bin/logger') +  
  STOPARG('Queue manager +QMNAME+ stopping')
```

Kde:

logger je ukázkový program dodávaný s produktem WebSphere MQ , který může zapisovat záznamy do systémového protokolu operačního systému.
+QMNAME+ je token představující název správce front.

Další informace o předávání argumentů

Tento příklad ukazuje, jak definovat objekt služby serveru pro spuštění programu nazvaného `runserv`, když je spuštěn správcem `front`.

Tento příklad je napsán s oddělovacím znakem cesty ve stylu systému Windows.

Jedním z argumentů, které mají být předány do spouštěcího programu, je řetězec obsahující mezeru. Tento argument musí být předán jako jediný řetězec. Chcete-li toho dosáhnout, jsou použity dvojité uvozovky, jak je zobrazeno v následujícím příkazu, abyste definovali objekt služby příkazu:

1. Objekt služby serveru je definován pomocí následujícího příkazu MQSC:

```
DEFINE SERVICE(S1) SERVTYPE(SERVER) CONTROL(QMGR) +
  STARTCMD('C:\Program Files\Tools\runserv.exe') +
  STARTARG('-m +QMNAME+ -d "C:\Program Files\Tools\'') +
  STDOUT('C:\Program Files\Tools\+MQ_SERVICE_NAME+.out')

DEFINE SERVICE(S4) +
  CONTROL(QMGR) +
  SERVTYPE(SERVER) +
  STARTCMD('C:\Program Files\Tools\runserv.exe') +
  STARTARG('-m +QMNAME+ -d "C:\Program Files\Tools\'') +
  STDOUT('C:\Program Files\Tools\+MQ_SERVICE_NAME+.out')
```

Kde:

+QMNAME+ je token představující název správce `front`.

"C:\Program Files\Tools\'" je řetězec obsahující mezeru, která bude předána jako jediný řetězec.

Automatické spuštění služby

Tento příklad ukazuje, jak definovat objekt služby serveru, který může být použit k automatickému spuštění monitoru spouštěčů při spuštění správce `front`.

1. Objekt služby serveru je definován pomocí následujícího příkazu MQSC:

```
DEFINE SERVICE(TRIG_MON_START) +
  CONTROL(QMGR) +
  SERVTYPE(SERVER) +
  STARTCMD('runmqtm') +
  STARTARG('-m +QMNAME+ -q +IQNAME+')
```

Kde:

+QMNAME+ je token představující název správce `front`.

+IQNAME+ je proměnná prostředí definovaná uživatelem v jednom ze souborů `service.env` představujících název inicializační fronty.

Správa objektů pro spuštění

Produkt WebSphere MQ vám umožňuje automaticky spustit aplikaci, když jsou splněny určité podmínky ve frontě. Například můžete chtít spustit aplikaci, když se počet zpráv ve frontě dosáhne zadaného čísla. Tento mechanismus se nazývá *spuštění*. Je třeba definovat objekty, které podporují spuštění.

Spuštění v tématu [Spuštění aplikací produktu WebSphere MQ pomocí spouštěčů](#) je popsáno podrobně.

Definování fronty aplikací pro spuštění

Fronta aplikací je lokální fronta, kterou využívají aplikace pro zaslání zpráv prostřednictvím rozhraní MQI. Spouštěcí impuls vyžaduje, aby byl definován počet atributů fronty ve frontě aplikací.

Spouštěcí impuls je povolen atributem *Trigger* (TRIGGER v příkazech MQSC). V tomto příkladě se má generovat událost triggeru, když je 100 zpráv s prioritou 5 nebo vyšší na lokální frontě MOTOR.INSURANCE.QUEUE:

```
DEFINE QLOCAL (MOTOR.INSURANCE.QUEUE) +  
  PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS) +  
  MAXMSGL (2000) +  
  DEFPSIST (YES) +  
  INITQ (MOTOR.INS.INIT.QUEUE) +  
  TRIGGER +  
  TRIGTYPE (DEPTH) +  
  TRIGDPTH (100)+  
  TRIGMPRI (5)
```

kde:

QLOCAL (MOTOR.INSURANCE.QUEUE)

Je název definované aplikační fronty.

PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS)

Je jméno definice procesu, která definuje aplikaci, která má být spuštěna programem monitoru spouštěčů.

MAXMSGL (2000)

Je maximální délka zpráv ve frontě.

DEFPSIST (YES)

Určuje, že zprávy v této frontě jsou standardně trvalé.

INITQ (MOTOR.INS.INIT.QUEUE)

Jedná se o název inicializační fronty, v níž má správce front vložit zprávu spouštěče.

TRIGGER

je hodnota atributu spouštěče.

TRIGTYPE (DEPTH)

Uvádí, že událost triggeru je generována, když se počet zpráv požadované priority (TRIGMPRI) dostane do počtu uvedeného v TRGDPTH.

TRIGDPTH (100)

Počet zpráv požadovaných k vygenerování události spouštěče.

TRIGMPRI (5)

Je priorita zpráv, které mají být počítány správcem front při rozhodování o tom, zda má být generována událost spouštěče. Spočítají se pouze zprávy s prioritou 5 nebo vyšší.

Definování inicializační fronty

Dojde-li k události spouštěče, správce front vloží do inicializační fronty zadané v definici fronty aplikací zprávu spouštěče. Inicializační fronty nemají žádná speciální nastavení, ale můžete použít následující definici lokální fronty MOTOR.INS.INIT.QUEUE jako vodítko:

```
DEFINE QLOCAL(MOTOR.INS.INIT.QUEUE) +  
  GET (ENABLED) +  
  NOSHARE +  
  NOTRIGGER +  
  MAXMSGL (2000) +  
  MAXDEPTH (1000)
```

Definování procesu

Použijte příkaz DEFINE PROCESS k vytvoření definice procesu. Definice procesu definuje aplikaci, která má být použita ke zpracování zpráv z aplikační fronty. Definice definice fronty aplikací pojmenovává proces, který má být použit, a tím asociuje aplikační frontu s aplikací, která má být použita ke zpracování svých zpráv. To se provádí přes atribut PROCESS na

frontě aplikací MOTOR.INSURANCE.QUEUE. Následující příkaz MQSC definuje požadovaný proces, MOTOR.INSURANCE.QUOTE.PROCESS, identifikovaný v tomto příkladu:

```
DEFINE PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS) +
DESCR ('Insurance request message processing') +
APPLTYPE (UNIX) +
APPLICID ('/u/admin/test/IRMP01') +
USERDATA ('open, close, 235')
```

Kde:

MOTOR.INSURANCE.QUOTE.PROCESS

Jedná se o název definice procesu.

DESCR ('Insurance request message processing')

Popisuje aplikační program, k němuž se tato definice vztahuje. Tento text se zobrazí, použijete-li příkaz DISPLAY PROCESS. To vám může pomoci identifikovat, co proces dělá. Pokud v řetězci použijete mezery, musíte řetězec uzavřít do jednoduchých uvozovek.

APPLTYPE (UNIX)

Je typ aplikace, která má být spuštěna.

APPLICID ('/u/admin/test/IRMP01')

Jedná se o název spustitelného souboru aplikace, který je zadán jako úplný název souboru. V systémech Windows by obvyklá hodnota APPLICID byla c:\appl\test\irmp01.exe.

USERDATA ('open, close, 235')

Jsou uživatelem definovaná data, která mohou být použita aplikací.

Zobrazení atributů definice procesu

Použijte příkaz DISPLAY PROCESS, abyste přezkoumali výsledky své definice. Příklad:

```
DISPLAY PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS)

24 : DISPLAY PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS) ALL
AMQ8407: Display Process details.
DESCR ('Insurance request message processing')
APPLICID ('/u/admin/test/IRMP01')
USERDATA (open, close, 235)
PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS)
APPLTYPE (UNIX)
```

Příkaz MQSC ALTER PROCESS můžete také použít ke změně existující definice procesu a k odstranění definice procesu pomocí příkazu DELETE PROCESS.

Správa vzdálených objektů produktu IBM WebSphere MQ

Tato sekce vám sděluje, jak spravovat objekty produktu IBM WebSphere MQ ve vzdáleném správci front pomocí příkazů MQSC a jak používat objekty vzdálených front k řízení cíle zpráv a zpráv odpovědí.

Tento oddíl popisuje:

- [“Kanály, klastry a vzdálené fronty” na stránce 103](#)
- [“Vzdálená administrace z lokálního správce front” na stránce 105](#)
- [“Vytvoření lokální definice vzdálené fronty” na stránce 111](#)
- [“Použití definic vzdálených front jako aliasů” na stránce 113](#)
- [“Převod dat mezi kódovanými znakovými sadami” na stránce 113](#)

Kanály, klastry a vzdálené fronty

Správce front komunikuje s jiným správcem front odesláním zprávy a, je-li to nutné, přijímá odpověď zpět. Přijímající správce front může být:

- Na stejném počítači
- Na jiném počítači ve stejném umístění (nebo dokonce i na druhé straně světa)
- Spuštění na stejné platformě jako lokální správce front
- Spuštění na jiné platformě podporované produktem WebSphere MQ

Tyto zprávy mohou pocházet z:

- Uživatelem napsané aplikační programy, které přenášejí data z jednoho uzlu do jiného.
- Uživatelem napsané administrační aplikace, které používají příkazy PCF nebo rozhraní MQAI
- Průzkumník IBM WebSphere MQ .
- Odesílání správců front:
 - Zprávy událostí instrumentace pro jiného správce front
 - Příkazy MQSC zadané z příkazu `runmqsc` v nepřímém režimu (kde jsou příkazy spuštěny v jiném správci front).

Než bude možné zprávu odeslat vzdálenému správci front, potřebuje lokální správce front mechanismus pro zjišťování příchodu zpráv a přenosu těchto zpráv:

- Alespoň jeden kanál
- Přenosová fronta
- Inicializátor kanálu

Aby vzdálený správce front obdržel zprávu, je vyžadován modul listener.

Kanál je jednosměrný komunikační spoj mezi dvěma správci front a může přenášet zprávy určené pro libovolný počet front ve vzdáleném správci front.

Každý konec kanálu má samostatnou definici. Je-li například jeden konec odesilatelem nebo serverem, druhý konec musí být příjemce nebo žadatel. Jednoduchý kanál se skládá z *definice odesílacího kanálu* na konci lokálního správce front a z *definice přijímacího kanálu* na konci vzdáleného správce front. Obě definice musí mít stejný název a dohromady tvoří jeden kanál zpráv.

Chcete-li, aby vzdálený správce front odpovídal na zprávy odeslané lokálním správcem front, nastavte druhý kanál tak, aby odesílal odpovědi zpět na lokálního správce front.

K definování kanálů použijte příkaz MQSC `DEFINE CHANNEL`. V tomto oddílu příklady týkající se kanálů používají výchozí atributy kanálu, není-li určeno jinak.

Na každém konci kanálu se nachází agent MCA (Message Channel Agent), který řídí odesílání a příjem zpráv. MCA přijímá zprávy z přenosové fronty a vkládá je na komunikační spojení mezi správci front.

Přenosová fronta je specializovaná lokální fronta, která dočasně uchovává zprávy dříve, než je program MCA vyzvedne a odešle je do vzdáleného správce front. Do *definice vzdálené fronty* zadejte název přenosové fronty.

Chcete-li povolit přenos zpráv pomocí více podprocesů, můžete program MCA povolit. Tento proces je znám jako *pipelining*. Potrubí umožňuje agentovi MCA efektivněji přenášet zprávy a zvyšovat výkon kanálů. Viz Atributy kanálů , kde naleznete podrobnosti o tom, jak konfigurovat kanál tak, aby používal rozhraní *pipelining*.

“Příprava kanálů a přenosových front pro vzdálenou administraci” na stránce 106 vám říká, jak používat tyto definice k nastavení vzdálené administrace.

Další informace o nastavení distribuovaných front obecně najdete v tématu Distribuované komponenty řazení do front.

Vzdálená administrace pomocí klastrů

V síti produktu WebSphere MQ s použitím distribuovaných front je každý správce front nezávislý. Pokud jeden správce front potřebuje odeslat zprávy jinému správci front, musí definovat přenosovou frontu, kanál ke vzdálenému správci front a definici vzdálené fronty pro každou frontu, do níž chce odesílat zprávy.

Klastr je skupina správců front nastavená takovým způsobem, že správci front mohou navzájem komunikovat přes jednu síť bez složité fronty přenosových front, kanálů a front. Klastry lze snadno nastavit a obvykle obsahují správce front, kteří spolu logicky souvisejí a které potřebují sdílet data nebo aplikace. Dokonce i ten nejmenší klastr snižuje náklady na administraci systému.

Vytvoření sítě správců front v klastru zahrnuje méně definic než vytváření tradičního prostředí distribuovaných front. S menšími definicemi můžete rychle a snadno nastavit nebo změnit svou síť a snížit riziko, že se v definicích vyskytne chyba.

Chcete-li nastavit klastr, potřebujete definice jednoho odesílatele klastru (CLUSSDR) a jednoho příjemce klastru (CLUSRCVR) pro každého správce front. Nepotřebujete žádné definice přenosových front nebo definice vzdálených front. Zásady vzdálené správy jsou stejné při použití v klastru, ale definice samy o sobě jsou značně zjednodušené.

Další informace o klastrech, jejich attributech a o tom, jak je nastavit, najdete v tématu [Klastry správců front](#).

Vzdálená administrace z lokálního správce front

Tento oddíl informuje o tom, jak spravovat vzdáleného správce front z lokálního správce front pomocí příkazů MQSC a PCF.

Příprava front a kanálů jsou v podstatě stejné pro příkazy MQSC a PCF. V této sekci jsou v příkladech uvedeny příkazy MQSC, protože je snazší je pochopit. Další informace o psaní administračních programů pomocí příkazů PCF naleznete v části [“Použití uživatelem programovatelných formátů příkazů”](#) na stránce 10.

Příkazy MQSC můžete odeslat do vzdáleného správce front buď interaktivně, nebo z textového souboru obsahujícího tyto příkazy. Vzdálený správce front se může nacházet ve stejném počítači nebo v jiném případě na jiném počítači. Vzdáleně můžete spravovat správce front v jiných prostředích produktu WebSphere MQ, včetně systémů UNIX and Linux, systémů Windows, IBM i a z/OS.

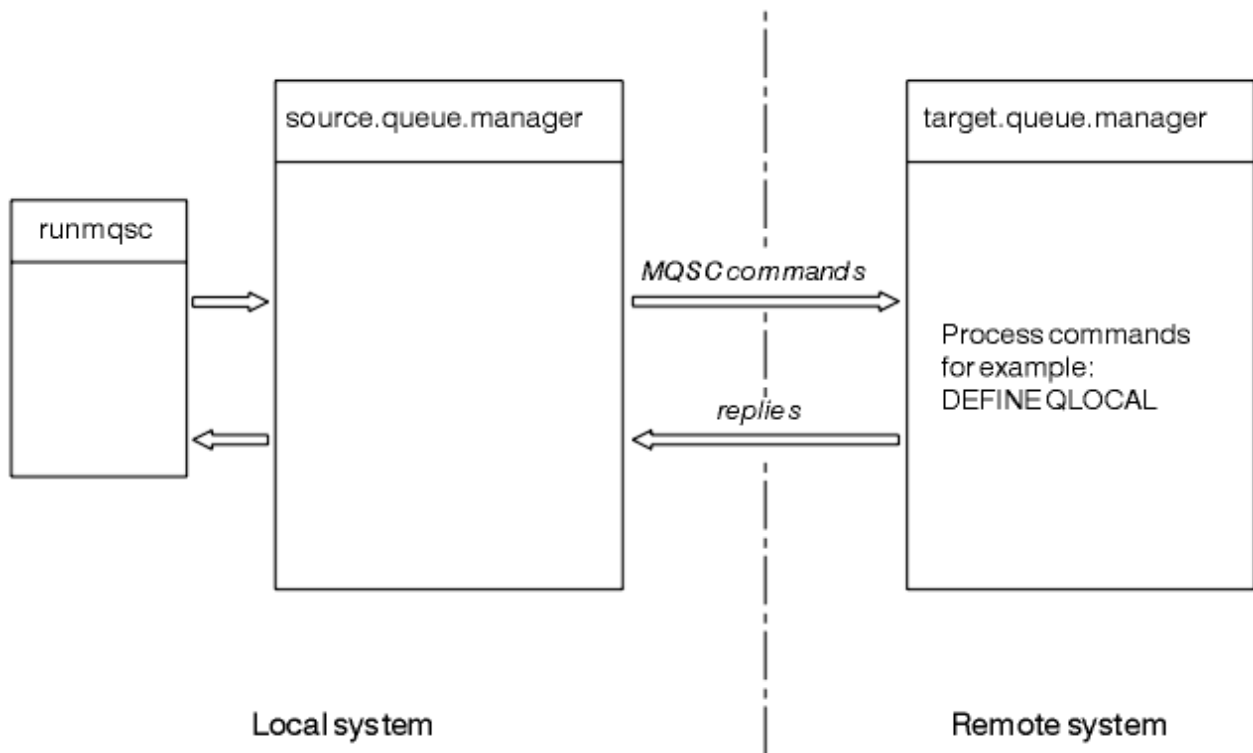
Chcete-li implementovat vzdálenou administraci, musíte vytvořit specifické objekty. Pokud nemáte specializované požadavky, jsou výchozí hodnoty (například pro maximální délku zprávy) dostatečné.

Příprava správců front pro vzdálenou administraci

Jak používat příkazy MQSC pro přípravu správců front pro vzdálenou administraci.

Obrázek 17 na stránce 106 zobrazuje konfiguraci správců front a kanálů, které potřebujete pro vzdálenou administraci pomocí příkazu **runmqsc**. Objekt `source.queue.manager` je zdrojový správce front, ze kterého můžete vydávat příkazy MQSC a pro které jsou vráceny výsledky těchto příkazů (operátorské zprávy). Objekt `target.queue.manager` je název cílového správce front, který zpracovává příkazy a generuje zprávy operátora.

Poznámka: Používáte-li **runmqsc** s volbou `-w, source.queue.manager` **musí** být výchozím správcem front. Další informace o vytváření správce front naleznete v souboru [crtmqm](#).



Obrázek 17. Vzdálená administrace pomocí příkazů MQSC

V obou systémech, pokud jste tak dosud neučinili, postupujte takto:

- Vytvořte správce front a výchozí objekty pomocí příkazu `crtmqm`.
- Spusťte správce front pomocí příkazu `strtmqm`.

V cílovém správci front proveďte následující akce:

- Fronta příkazů, `SYSTEM.ADMIN.COMMAND.QUEUE`, musí být přítomna. Tato fronta je standardně vytvořena při vytvoření správce front.

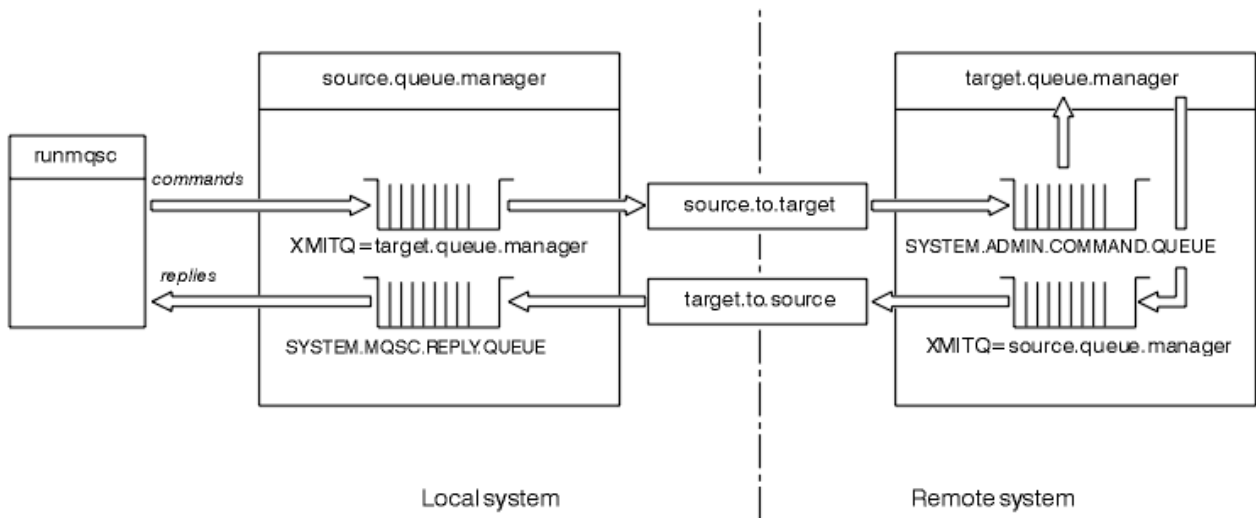
Tyto příkazy musíte spustit lokálně nebo přes síťovou službu, jako je například Telnet.

Příprava kanálů a přenosových front pro vzdálenou administraci

Jak používat příkazy MQSC pro přípravu kanálů a přenosových front pro vzdálenou administraci.

Chcete-li spustit příkazy MQSC vzdáleně, nastavte dva kanály, jeden pro každý směr a jejich přidružené přenosové fronty. Tento příklad předpokládá, že používáte protokol TCP/IP jako typ transportu a že znáte danou adresu TCP/IP.

Kanál `source . to . target` je určen pro odesílání příkazů MQSC ze zdrojového správce front do cílového správce front. Jeho odesílatel je v `source . queue . manager` a jeho příjemce je v `target . queue . manager`. Kanál `target . to . source` je určen k vrácení výstupu z příkazů a jakýchkoli zpráv operátora, které byly generovány do správce zdrojové fronty. Pro každý kanál musíte také definovat přenosovou frontu. Tato fronta je lokální frontou, která je přidělena k názvu přijímajícího správce front. Název `XMITQ` se musí shodovat s názvem vzdáleného správce front, aby byla vzdálená administrace funkční, pokud nepoužíváte alias správce front. [Obrázek 18 na stránce 107](#) shrnuje tuto konfiguraci.



Obrázek 18. Nastavení kanálů a front pro vzdálenou administraci

Další informace o nastavení kanálů najdete v tématu [Připojování aplikací pomocí distribuovaných front](#).

Definování kanálů, modulů listener a přenosových front

Na zdrojovém správci front (`source.queue.manager`) zadejte následující příkazy MQSC pro definování kanálů, modulu listener a přenosové fronty:

1. Definujte odesílací kanál na zdrojovém správci front:

```
DEFINE CHANNEL ('source.to.target') +
  CHLTYPE(SDR) +
  CONNAME (RHX5498) +
  XMITQ ('target.queue.manager') +
  TRPTYPE(TCP)
```

2. Definujte přijímací kanál na zdrojovém správci front:

```
DEFINE CHANNEL ('target.to.source') +
  CHLTYPE(RCVR) +
  TRPTYPE(TCP)
```

3. Definujte modul listener na zdrojovém správci front:

```
DEFINE LISTENER ('source.queue.manager') +
  TRPTYPE (TCP)
```

4. Definujte přenosovou frontu ve zdrojovém správci front:

```
DEFINE QLOCAL ('target.queue.manager') +
  USAGE (XMITQ)
```

Vydejte následující příkazy na cílovém správci front (`target.queue.manager`), abyste vytvořili kanály, listener a přenosovou frontu:

1. Definujte odesílací kanál v cílovém správci front:

```
DEFINE CHANNEL ('target.to.source') +
  CHLTYPE(SDR) +
  CONNAME (RHX7721) +
  XMITQ ('source.queue.manager') +
  TRPTYPE(TCP)
```

2. Definujte přijímací kanál v cílovém správci front:

```
DEFINE CHANNEL ('source.to.target') +
  CHLTYPE(RCVR) +
  TRPTYPE(TCP)
```

3. Definujte modul listener na cílovém správci front:

```
DEFINE LISTENER ('target.queue.manager') +
  TRPTYPE (TCP)
```

4. Definujte přenosovou frontu v cílovém správci front:

```
DEFINE QLOCAL ('source.queue.manager') +
  USAGE (XMITQ)
```

Poznámka: Názvy připojení TCP/IP uvedené pro atribut CONNAME v definicích odesílacího kanálu jsou pouze pro ilustraci. Jedná se o název sítě počítače na *druhém* konci připojení. Použijte hodnoty vhodné pro vaši síť.

Spouštění listenerů a kanálů

Jak používat příkazy MQSC pro spouštění modulů listener a kanálů.

Spusťte oba listenery pomocí následujících příkazů MQSC:

1. Spusťte modul listener na zdrojovém správci front (`source.queue.manager`) zadáním následujícího příkazu MQSC:

```
START LISTENER ('source.queue.manager')
```

2. Spusťte modul listener na cílovém správci front (`target.queue.manager`) zadáním následujícího příkazu MQSC:

```
START LISTENER ('target.queue.manager')
```

Oba odesílací kanály lze spustit pomocí následujících příkazů MQSC:

1. Spusťte odesílací kanál ve zdrojovém správci front, `source.queue.manager`, zadáním následujícího příkazu MQSC:

```
START CHANNEL ('source.to.target')
```

2. Spusťte odesílací kanál v cílovém správci front `target.queue.manager` zadáním následujícího příkazu MQSC:

```
START CHANNEL ('target.to.source')
```

Automatická definice kanálů

Automatickou definici příjemce a definic připojení k serveru můžete povolit aktualizací objektu správce front pomocí příkazu MQSC, `ALTER QMGR` (nebo příkazu `PCF Change Queue Manager`).

Pokud produkt WebSphere MQ obdrží požadavek na připojení typu Inbound a nemůže najít odpovídající příjemce nebo kanál připojení serveru, vytvoří kanál automaticky. Automatické definice jsou založeny na dvou výchozích definicích dodaných s produktem WebSphere MQ: `SYSTEM.AUTO.RECEIVER` a `SYSTEM.AUTO.SVRCONN`.

Další informace o automatickém vytváření definic kanálů naleznete v tématu [Příprava kanálů](#). Informace o automatickém definování kanálů pro klastry naleznete v tématu [Automatická definice kanálů klastru](#).

Správa příkazového serveru pro vzdálenou administraci

Jak spustit, zastavit a zobrazit stav příkazového serveru. Příkazový server je povinný pro veškerou administraci zahrnující příkazy PCF, rozhraní MQAI a také pro vzdálenou administraci.

Ke každému správci front může být přidružen příkazový server. Příkazový server zpracovává všechny příchozí příkazy ze vzdálených správců front nebo příkazy PCF z aplikací. Tento příkaz prezentuje příkazy správci front pro zpracování a vrací kód dokončení nebo zprávu operátora v závislosti na původu příkazu.

Poznámka: Pro vzdálenou administraci se ujistěte, že je cílový správce front spuštěn. Jinak nemohou zprávy obsahující příkazy opustit správce front, ze kterého jsou vydávány. Místo toho jsou tyto zprávy zařazeny do fronty v lokální přenosové frontě, která obsluhuje vzdáleného správce front. Vyhněte se této situaci.

Pro spuštění a zastavení příkazového serveru existují samostatné řídicí příkazy. Po spuštění příkazového serveru mohou uživatelé produktu WebSphere MQ for Windows nebo WebSphere MQ for Linux (platformy x86 a x86-64) provádět operace popsané v následujících sekcích pomocí Průzkumníka IBM WebSphere MQ. Další informace viz [“Administrace pomocí produktu IBM WebSphere MQ Explorer”](#) na stránce 56.

Spuštění příkazového serveru

V závislosti na hodnotě atributu správce front, *SCMDSERV*, je příkazový server buď spuštěn automaticky při spuštění správce front, nebo musí být spuštěn ručně. Hodnotu atributu správce front lze změnit pomocí příkazu `MQSC ALTER QMGR`, který určuje parametr *SCMDSERV*. Při výchozím nastavení je příkazový server spuštěn automaticky.

Je-li parametr *SCMDSERV* nastaven na hodnotu `MANUAL`, spusťte příkazový server pomocí příkazu:

```
stmqcsv saturn.queue.manager
```

kde `saturn.queue.manager` je správce front, pro který je spuštěn příkazový server.

Zobrazení stavu příkazového serveru

Pro vzdálenou administraci zkontrolujte, zda je spuštěn příkazový server v cílovém správci front. Pokud není spuštěn, vzdálené příkazy nelze zpracovat. Všechny zprávy obsahující příkazy jsou zařazeny do fronty v cílové frontě správce front.

Chcete-li zobrazit stav příkazového serveru pro správce front, zadejte následující příkaz `MQSC`:

```
DISPLAY QMSTATUS CMDSERV
```

Zastavení příkazového serveru

Chcete-li ukončit příkazový server spuštěný v předchozím příkladu, použijte následující příkaz:

```
endmqcsv saturn.queue.manager
```

Příkazový server můžete zastavit dvěma způsoby:

- Pro řízené ukončení použijte příkaz `endmqcsv` s příznakem `-c`, který je výchozí.
- Chcete-li provést okamžité zastavení, použijte příkaz `endmqcsv` s příznakem `-i`.

Poznámka: Zastavení správce front také ukončí příkazový server, který je k němu přidružen.

Vydávání příkazů MQSC ve vzdáleném správci front

Ke spuštění příkazů `MQSC` ve vzdáleném správci front můžete použít konkrétní tvar příkazu `runmqsc`.

Příkazový server **musí** být spuštěn na cílovém správci front, pokud se bude zpracovávat vzdáleně příkazy MQSC. (To není nutné ve zdrojovém správci front). Informace o tom, jak spustit příkazový server ve správci front, najdete v tématu [“Správa příkazového serveru pro vzdálenou administraci”](#) na stránce 109.

Ve zdrojovém správci front pak můžete příkazy MQSC spustit interaktivně v nepřímém režimu zadáním příkazu:

```
runmqsc -w 30 target.queue.manager
```

Tento tvar příkazu runmqsc s příznakem -w spouští příkazy MQSC v nepřímém režimu, kde jsou příkazy vloženy (v modifikovaném tvaru) do vstupní fronty příkazového serveru a jsou prováděny v pořadí.

Zadáte-li příkaz MQSC, bude přesměrován na vzdáleného správce front v tomto případě `target.queue.manager`. Časový limit je nastaven na 30 sekund; pokud není odpověď přijata během 30 sekund, vygeneruje se následující zpráva na lokálním (zdrojovém) správci front:

```
AMQ8416: MQSC timed out waiting for a response from the command server.
```

Po ukončení vydávání příkazů MQSC lokální správce front zobrazí všechny odpovědi, které přišly a zahodí všechny další odpovědi.

Zdrojový správce front je standardně nastaven na výchozího lokálního správce front. Zadáte-li v příkazu **runmqsc** volbu `-m LocalQmgrName`, můžete příkazy směřovat tak, aby byly vydány prostřednictvím libovolného lokálního správce front.

V nepřímém režimu můžete také spustit příkazový soubor MQSC ve vzdáleném správci front. Příklad:

```
runmqsc -w 60 target.queue.manager < mycomds.in > report.out
```

kde `mycomds.in` je soubor obsahující příkazy MQSC a `report.out` je soubor sestavy.

Navrhovaná metoda pro vzdálené zadávání příkazů

Při vydávání příkazů ve vzdáleném správci front zvažte použití následujícího přístupu:

1. Zadejte příkazy MQSC, které mají být spuštěny na vzdáleném systému v příkazovém souboru.
2. Ověřte lokální příkazy MQSC zadáním příznaku `-v` v příkazu `runmqsc`.
Produkt `runmqsc` nelze použít k ověření příkazů MQSC v jiném správci front.
3. Zkontrolujte, zda je příkazový soubor spuštěn lokálně bez chyby.
4. Spusťte příkazový soubor ve vzdáleném systému.

Pokud máte problémy s použitím příkazů MQSC vzdáleně

Máte-li potíže se spuštěním příkazů MQSC vzdáleně, ujistěte se, že máte:

- Byl spuštěn příkazový server v cílovém správci front.
- Definována platná přenosová fronta.
- Definovány jsou dva konce kanálů zpráv pro oba tyto kanály:
 - Kanál, na kterém jsou odesílány příkazy.
 - Kanál, na kterém mají být vráceny odpovědi.
- Zadejte správný název připojení (CONNNAME) v definici kanálu.
- Moduly listener byly spuštěny před spuštěním kanálů zpráv.
- Zaškrtnuto, že interval odpojení nevypršel, například, pokud byl kanál spuštěn, ale po nějaké době byl vypnut. To je obzvláště důležité, pokud spouštíte kanály ručně.

- Odeslané požadavky ze zdrojového správce front, které nemají smysl pro cílového správce front (například požadavky obsahující parametry, které nejsou podporovány ve vzdáleném správci front).

Další informace najdete v tématu [“Řešení problémů s příkazy MQSC”](#) na stránce 79.

Vytvoření lokální definice vzdálené fronty

Lokální definice vzdálené fronty je definice v lokálním správci front, který odkazuje na frontu ve vzdáleném správci front.

Nemusíte definovat vzdálenou frontu z lokální pozice, ale výhodou je, že aplikace mohou odkazovat na vzdálenou frontu lokálně definovaným názvem místo toho, aby museli zadávat jméno kvalifikované pomocí ID správce front, ve kterém je umístěna vzdálená fronta.

Základní informace o tom, jak fungují lokální definice vzdálených front

Aplikace se připojuje k lokálnímu správci front a poté vyvolá volání MQOPEN . V otevřeném volání je určený název fronty uveden v definici vzdálené fronty v lokálním správci front. Definice vzdálené fronty poskytuje názvy cílové fronty, cílového správce front a volitelně přenosové fronty. Chcete-li vložit zprávu do vzdálené fronty, aplikace vydá volání MQPUT , přičemž uvede manipulátor vrácený z volání MQOPEN . Správce front používá název vzdálené fronty a název vzdáleného správce front v záhlaví přenosu na začátku zprávy. Tyto informace se používají ke směrování zprávy na její správné místo určení v síti.

Jako administrátor můžete řídit místo určení zprávy změnou definice vzdálené fronty.

Následující příklad uvádí, jak aplikace vloží zprávu do fronty vlastněné vzdáleným správcem front. Aplikace se připojuje ke správci front, například saturn . queue . manager . Cílová fronta je vlastněna jiným správcem front.

Při volání MQOPEN aplikace uvádí tato pole:

Hodnota pole	Popis
<i>ObjectName</i> CYAN.REMOTE.QUEUE	Určuje lokální název objektu vzdálené fronty. To definuje cílovou frontu a cílového správce front.
<i>ObjectType</i> (Fronta)	Identifikuje tento objekt jako frontu.
<i>ObjectQmgrName</i> Mezera nebo saturn . queue . manager	Toto pole je volitelné. Je-li tato hodnota prázdná, použije se název lokálního správce front. (Jedná se o správce front, v němž existuje definice vzdálené fronty.)

Poté aplikace vydá zprávu MQPUT , aby do této fronty vložila zprávu.

V lokálním správci front je možné vytvořit lokální definici vzdálené fronty pomocí následujících příkazů MQSC:

```
DEFINE QREMOTE (CYAN.REMOTE.QUEUE) +
DESCR ('Queue for auto insurance requests from the branches') +
RNAME (AUTOMOBILE.INSURANCE.QUOTE.QUEUE) +
RQMNAME (jupiter.queue.manager) +
XMITQ (INQUOTE.XMIT.QUEUE)
```

kde:

QREMOTE (CYAN.REMOTE.QUEUE)

Určuje lokální název objektu vzdálené fronty. Jedná se o název, který aplikace připojené k tomuto správci front musí ve volání MQOPEN určit, aby otevřela frontu AUTOMOBILE.INSURANCE.QUOTE.QUEUE na vzdáleném správci front jupiter . queue . manager .

DESCR ('Queue for auto insurance requests from the branches')

Poskytuje další text, který popisuje použití fronty.

RNAME (AUTOMOBILE.INSURANCE.QUOTE.QUEUE)

Určuje název cílové fronty ve vzdáleném správci front. Jedná se o skutečnou cílovou frontu pro zprávy odeslané aplikacemi, které uvádějí název fronty CYAN.REMOTE.QUEUE. Fronta AUTOMOBILE.INSURANCE.QUOTE.QUEUE musí být definována jako lokální fronta ve vzdáleném správci front.

RQMNAME (jupiter.queue.manager)

Určuje název vzdáleného správce front, který vlastní cílovou frontu AUTOMOBILE.INSURANCE.QUOTE.QUEUE.

XMITQ (INQUOTE.XMIT.QUEUE)

Uvádí název přenosové fronty. Tento parametr je volitelný; pokud není zadán název přenosové fronty, použije se fronta se stejným názvem, jako je vzdálený správce front.

V každém případě musí být příslušná přenosová fronta definována jako lokální fronta s atributem *Usage*, který uvádí, že se jedná o přenosovou frontu (USAGE (XMITQ) v příkazech MQSC).

Alternativní způsob vkládání zpráv do vzdálené fronty

Použití lokální definice vzdálené fronty není jediným způsobem, jak umístit zprávy do vzdálené fronty. Aplikace mohou určit úplný název fronty včetně názvu vzdáleného správce front jako součást volání MQOPEN. V takovém případě nepotřebujete lokální definici vzdálené fronty. Znamená to však, že aplikace musí buď znát nebo mít přístup k názvu vzdáleného správce front za běhu programu.

Použití jiných příkazů se vzdálenými frontami

Pomocí příkazů MQSC můžete zobrazit nebo změnit atributy objektu vzdálené fronty, nebo můžete objekt vzdálené fronty odstranit. Příklad:

- Chcete-li zobrazit atributy vzdálené fronty, postupujte takto:

```
DISPLAY QUEUE (CYAN.REMOTE.QUEUE)
```

- Chcete-li změnit vzdálenou frontu na povolení vložení. To nemá vliv na cílovou frontu, pouze aplikace, které určují tuto vzdálenou frontu:

```
ALTER QREMOTE (CYAN.REMOTE.QUEUE) PUT(ENABLED)
```

- K odstranění této vzdálené fronty. To nemá vliv na cílovou frontu, pouze její lokální definice:

```
DELETE QREMOTE (CYAN.REMOTE.QUEUE)
```

Poznámka: Když odstraníte vzdálenou frontu, odstraníte pouze lokální reprezentaci vzdálené fronty. Vzdálenou frontu jako takovou, ani žádné zprávy, neodstraníte.

Definování přenosové fronty

Přenosová fronta je lokální fronta, která se používá, když správce front předává zprávy vzdálenému správci front prostřednictvím kanálu zpráv.

Kanál poskytuje jednosměrné propojení se vzdáleným správcem front. Zprávy se zařadí do fronty v přenosové frontě, dokud je kanál nebude moci přijmout. Při definování kanálu je třeba určit název přenosové fronty na odesílajícím konci kanálu zpráv.

Atribut USAGE příkazu MQSC definuje, zda je fronta přenosová fronta nebo normální fronta.

Výchozí přenosové fronty

Když správce front odesílá zprávy vzdálenému správci front, identifikuje přenosovou frontu pomocí následující posloupnosti:

1. Přenosová fronta uvedená na atributu XMITQ lokální definice vzdálené fronty.
2. Přenosová fronta se stejným názvem jako má správce cílové fronty. (Tato hodnota je výchozí hodnotou atributu XMITQ lokální definice vzdálené fronty.)
3. Přenosová fronta uvedená na atributu DEFXMITQ lokálního správce front.

Například následující příkaz MQSC vytvoří výchozí přenosovou frontu v systému `source.queue.manager` pro zprávy, které se předávají produktu `target.queue.manager`:

```
DEFINE QLOCAL ('target.queue.manager') +
DESCR ('Default transmission queue for target qm') +
USAGE (XMITQ)
```

Aplikace mohou vkládat zprávy přímo do přenosové fronty nebo nepřímo prostřednictvím definice vzdálené fronty. Další informace najdete v tématu [“Vytvoření lokální definice vzdálené fronty”](#) na stránce [111](#).

Použití definic vzdálených front jako aliasů

Kromě vyhledání fronty v jiném správci front můžete také použít lokální definici vzdálené fronty pro alias správce front a alias fronty odpovědí na frontu. Oba typy aliasů jsou interpretována prostřednictvím lokální definice vzdálené fronty. Musíte nastavit vhodné kanály, aby zpráva dorazila na místo určení.

Alias správce front

Alias je proces, při kterém je název cílového správce front, jak je uveden ve zprávě, změněn správcem front na trase zpráv. Alias správce front jsou důležité, protože je lze použít k řízení cíle zpráv v rámci sítě správců front.

To provedete změnou definice vzdálené fronty na správci front v místě řízení. Odesílající aplikace si není vědoma toho, že uvedený název správce front je alias.

Další informace o aliasech správce front najdete v tématu [Co jsou aliasy?](#).

Alias fronty pro odpověď

Volitelně může aplikace uvést název fronty pro odpovědi, když umístí *zprávu požadavku* do fronty.

Pokud aplikace, která zpracovává zprávu, extrahuje název fronty pro odpovědi, ví, kam poslat *zprávu odpovědi*, je-li to požadováno.

Alias fronty pro odpověď je proces, jehož prostřednictvím správce front na trase zpráv provádí změnu fronty pro odpověď, jak je uvedeno ve zprávě vzniklé při zpracování požadavku. Odesílající aplikace si není vědoma toho, že zadané jméno fronty pro odpověď je alias.

Alias fronty pro odpověď vám umožňuje změnit název fronty pro odpovědi a volitelně i jeho správce front. To vám na oplátku umožňuje řídit, která trasa se bude používat pro zprávy odpovědi.

Další informace o zprávách požadavků, zprávách odpovědí a frontách odpovědí na fronty naleznete v tématu [Typy zpráv](#) a [Fronta pro odpovědi a správce front](#).

Další informace o alias fronty odpovědi naleznete v tématu [Alias fronty odpovědi a klastry](#).

Převod dat mezi kódovanými znakovými sadami

Data zprávy v definovaných formátech produktu WebSphere MQ (známá také jako *vestavěné formáty*) mohou být převedena správcem front z jedné kódované znakové sady do jiné za předpokladu, že obě znakové sady souvisí s jedním jazykem nebo se skupinou podobných jazyků.

Například převod mezi kódovanými znakovými sadami s identifikátory (CCSID) 850 a 500 je podporován, protože se oba používají pro západoevropské jazyky.

Pro konverze znaků EBCDIC newline (NL) na ASCII si prohlédněte [Všichni správci front](#).

Podporované konverze jsou definovány v části [Převod dat](#).

Pokud správce front nemůže převést zprávy ve vestavěných formátech

Správce front nemůže automaticky převádět zprávy ve vestavěných formátech, pokud jejich identifikátory CCSID reprezentují různé skupiny národních jazyků. Například konverze mezi CCSID 850 a CCSID 1025 (která je kódovanou znakovou sadou EBCDIC pro jazyky pomocí cyrilice) není podporována, protože mnoho znaků v jedné kódované znakové sadě nemůže být reprezentováno v druhém. Máte-li síť správců front pracujících v různých národních jazycích a převod dat mezi některými z kódovaných znakových sad není podporován, můžete povolit výchozí převod. Výchozí převod dat je popsán v části [“Výchozí převod dat”](#) na stránce 114.

Soubor ccsid.tbl

Soubor ccsid.tbl se používá pro následující účely:

- V produktu WebSphere MQ for Windows se zaznamenává všechny podporované kódové sady.
- Na platformách AIX a HP-UX jsou podporované kódové sady interně drženy operačním systémem.
- Pro všechny ostatní platformy UNIX and Linux jsou podporované kódové sady umístěny v tabulkách pro převod, které jsou k dispozici v produktu WebSphere MQ.
- Určuje libovolné další kódové sady. Chcete-li určit další kódové sady, je třeba upravit soubor ccsid.tbl (návod, jak to provést v souboru).
- Určuje jakýkoli výchozí převod dat.

Můžete aktualizovat informace zaznamenané v souboru ccsid.tbl; můžete tak učinit například v případě, že budoucí vydání operačního systému podporuje další kódované znakové sady.

V produktu WebSphere MQ for Windows se ccsid.tbl standardně nachází v adresáři C:\Program Files\IBM\WebSphere MQ\conv\table.

V produktu WebSphere MQ for UNIX and Linux se soubor ccsid.tbl nachází v adresáři /var/mqm/conv/table.

Výchozí převod dat

Nastavíte-li kanály mezi dvěma počítači, na kterých není konverze dat normálně podporována, musíte povolit výchozí převod dat pro kanály, které chcete pracovat.

Chcete-li povolit výchozí převod dat, upravte soubor ccsid.tbl tak, aby určoval předvolený CCSID EBCDIC a výchozí CCSID ASCII. Pokyny, jak to provést, jsou obsaženy v souboru. To musíte provést na všech počítačích, které budou připojeny pomocí kanálů. Restartujte správce front, aby se změna projevila.

Předvolený proces konverze dat je následující:

- Není-li konverze mezi zdrojovým a cílovým CCSID podporována, ale identifikátory CCSID zdrojového a cílového prostředí jsou buď EBCDIC, nebo oba ASCII, znaková data se předají cílové aplikaci bez konverze.
- Pokud jeden CCSID reprezentuje znakovou sadu ASCII a druhá představuje kódovanou znakovou sadu EBCDIC, produkt WebSphere MQ převede data s použitím výchozích CCSID pro převod dat definovaných v souboru ccsid.tbl.

Poznámka: Pokuste se omezit převáděné znaky na ty, které mají stejné hodnoty kódu v kódované znakové sadě uvedené pro zprávu a ve výchozí kódované znakové sadě. Použijete-li pouze sadu znaků, která je platná pro názvy objektů produktu WebSphere MQ (jak je definováno v poli [Pojmenování objektů IBM WebSphere MQ](#)), tento požadavek obecně uspokojí. Výjimky se vyskytují s identifikátory CCSID EBCDIC

290, 930, 1279 a 5026 použité v Japonsku, kde malá písmena mají odlišné kódy od těch, které jsou použity v jiných CCSID EBCDIC.

Převod zpráv v uživatelem definovaných formátech

Správce front nemůže převést zprávy v uživatelem definovaných formátech z jedné kódované znakové sady na jinou. Potřebujete-li převést data ve formátu definovaném uživatelem, musíte pro každý takový formát zadat uživatelské procedury pro převod dat. Nepoužívejte výchozí CCSID pro převod znakových dat v uživatelem definovaných formátech. Další informace o převádění dat v uživatelem definovaných formátech a o zápisu uživatelských procedur pro převod dat naleznete v tématu [Zápis uživatelských procedur pro převod dat](#).

Změna identifikátoru CCSID správce front

Pokud jste použili atribut CCSID příkazu ALTER QMGR ke změně CCSID správce front, zastavte a znovu spusťte správce front, abyste se ujistili, že všechny spuštěné aplikace, včetně příkazového serveru a programů kanálu, jsou zastaveny a restartovány.

To je nezbytné, protože všechny aplikace spuštěné v případě změny CCSID správce front budou nadále používat existující CCSID.

Správa serveru IBM WebSphere MQ Telemetry

IBM WebSphere MQ Telemetry se podává pomocí IBM WebSphere MQ Explorer nebo na příkazovém řádku. Průzkumník můžete použít ke konfiguraci kanálů telemetrie, k řízení služby telemetrie a k monitorování klientů MQTT připojených k produktu IBM WebSphere MQ. Nakonfigurujte zabezpečení produktu IBM WebSphere MQ Telemetry pomocí služeb JAAS, SSL a správce oprávnění objektu IBM WebSphere MQ .

Administrace pomocí produktu IBM WebSphere MQ Explorer

Průzkumník můžete použít ke konfiguraci kanálů telemetrie, k řízení služby telemetrie a k monitorování klientů MQTT připojených k produktu IBM WebSphere MQ. Nakonfigurujte zabezpečení produktu IBM WebSphere MQ Telemetry pomocí služeb JAAS, SSL a správce oprávnění objektu IBM WebSphere MQ .

Administrace pomocí příkazového řádku

IBM WebSphere MQ Telemetry lze kompletně spravovat na příkazovém řádku pomocí příkazů IBM WebSphere MQ [MQSC](#) .

Dokumentace produktu IBM WebSphere MQ Telemetry také obsahuje ukázkové skripty, které demonstrují základní použití aplikace MQ Telemetry Transport v3 Client.

Před použitím si prostudujte ukázky v [ukázkových programech IBM WebSphere MQ Telemetry](#) v části [Vyvíjení aplikací pro produkt IBM WebSphere MQ Telemetry](#) .

Související pojmy

WebSphere MQ Telemetry

[“Konfigurace distribuovaných front pro odesílání zpráv klientům MQTT” na stránce 119](#)

Aplikace produktu IBM WebSphere MQ mohou odesílat zprávy klientů MQTT v3 publikováním do odběru vytvořeného klientem nebo odesláním zprávy přímo. Je-li použita metoda Whichever, je zpráva vložena do produktu SYSTEM.MQTT.TRANSMIT.QUEUE a odeslána klientovi prostřednictvím služby telemetrie (MQXR). Existuje celá řada způsobů, jak umístit zprávu do SYSTEM.MQTT.TRANSMIT.QUEUE.

[“identifikace klienta MQTT, autorizace a ověření” na stránce 122](#)

[“Ověření kanálu telemetrie pomocí zabezpečení SSL” na stránce 128](#)

[“Soukromí zveřejňování kanálů telemetrie” na stránce 130](#)

[“Konfigurace zabezpečení SSL pro klienty MQTT a kanály telemetrie” na stránce 131](#)

[“Konfigurace kanálu JAAS kanálu telemetrie” na stránce 136](#)

Nakonfigurujte službu JAAS tak, aby ověřoval identitu `Jméno` uživatele odeslané klientem.

[“Démon IBM WebSphere MQ Telemetry pro koncepcce zařízení” na stránce 138](#)

Démon IBM WebSphere MQ Telemetry pro zařízení je rozšířenou aplikací klienta MQTT V3 . Použijte jej k ukládání a předávání zpráv z jiných klientů MQTT. Připojuje se k produktu IBM WebSphere MQ jako ke klientovi MQTT, ale k němu se můžete připojit i k dalším klientům MQTT.

Související úlohy

[“Konfigurace správce front pro telemetrie v systémech Linux a AIX” na stránce 116](#)

Chcete-li správce front spustit pro spuštění produktu IBM WebSphere MQ Telemetry, postupujte podle těchto pokynů. Automatizovanou proceduru můžete spustit, chcete-li nastavit jednodušší konfiguraci pomocí podpory produktu IBM WebSphere MQ Telemetry pro produkt IBM WebSphere MQ Explorer.

[“Konfigurace správce front pro telemetrie v produktu Windows” na stránce 118](#)

Chcete-li správce front spustit pro spuštění produktu IBM WebSphere MQ Telemetry, postupujte podle těchto pokynů. Automatizovanou proceduru můžete spustit, chcete-li nastavit jednodušší konfiguraci pomocí podpory produktu IBM WebSphere MQ Telemetry pro produkt IBM WebSphere MQ Explorer.

Související odkazy

[Vlastnosti MQXR](#)

Konfigurace správce front pro telemetrie v systémech Linux a AIX

Chcete-li správce front spustit pro spuštění produktu IBM WebSphere MQ Telemetry, postupujte podle těchto pokynů. Automatizovanou proceduru můžete spustit, chcete-li nastavit jednodušší konfiguraci pomocí podpory produktu IBM WebSphere MQ Telemetry pro produkt IBM WebSphere MQ Explorer.

Než začnete

1. Informace o tom, jak instalovat produkt IBM WebSphere MQ a funkci IBM WebSphere MQ Telemetry naleznete v tématu [Instalace produktu IBM WebSphere MQ Telemetry](#) .
2. Vytvořte a spusťte správce front. Na správce front se v této úloze odkazuje jako na *qMgr* .
3. Jako součást této úlohy můžete konfigurovat službu telemetrie (MQXR). Nastavení vlastností MQXR se ukládají do souboru vlastností specifického pro platformu: `mqxr_unix.properties`. Soubor vlastností MQXR obvykle nemusíte upravovat přímo, protože téměř všechna nastavení mohou být konfigurována prostřednictvím příkazů administrace MQSC nebo MQ Explorer. Pokud se rozhodnete soubor upravit přímo, zastavte správce front před tím, než provedete změny. Viz [vlastnosti MQXR](#).

Informace o této úloze

Podpora IBM WebSphere MQ Telemetry pro IBM WebSphere MQ Explorer obsahuje průvodce a vzorovou příkazovou proceduru `sampleMQM`. Nastavení počáteční konfigurace pomocí ID uživatele "guest"; viz téma [Ověření instalace produktu IBM WebSphere MQ Telemetry pomocí ukázkových programů IBM WebSphere MQ Explorer a IBM WebSphere MQ Telemetry ukázkových programů](#).

Postupujte podle kroků v této úloze a nakonfigurujte produkt IBM WebSphere MQ Telemetry ručně pomocí různých schémat autorizace.

Postup

1. Otevřete příkazové okno v adresáři ukázek telemetrie.
Adresář ukázek telemetrie je `/opt/mqm/mqxr/samples`.
2. Vytvořte přenosovou frontu telemetrie.

```
echo "DEFINE QLOCAL('SYSTEM.MQTT.TRANSMIT.QUEUE') USAGE(XMITQ) MAXDEPTH(100000)" | runmqsc qMgr
```

Je-li služba telemetrie (MQXR) spuštěna poprvé, vytvoří `SYSTEM.MQTT.TRANSMIT.QUEUE`.

Je vytvořen ručně v této úloze, protože produkt `SYSTEM.MQTT.TRANSMIT.QUEUE` musí existovat před spuštěním služby telemetrie (MQXR), aby bylo možné k němu autorizovat přístup.

3. Nastavení výchozí přenosové fronty

Je-li služba telemetrie (MQXR) poprvé spuštěna, neupraví správce front, aby učinil `SYSTEM.MQTT.TRANSMIT.QUEUE` výchozí přenosovou frontou.

Chcete-li nastavit `SYSTEM.MQTT.TRANSMIT.QUEUE` jako výchozí přenosovou frontu, změňte výchozí vlastnost přenosové fronty. Upravte vlastnost pomocí příkazu IBM WebSphere MQ Explorer nebo pomocí příkazu v následujícím příkladu:

```
echo "ALTER QMGR DEFEXMITQ('SYSTEM.MQTT.TRANSMIT.QUEUE') " | runmqsc qMgr
```

Změna výchozí přenosové fronty může kolidovat s vaší stávající konfigurací. Důvodem změny výchozí přenosové fronty do produktu `SYSTEM.MQTT.TRANSMIT.QUEUE` je usnadnění odesílání zpráv přímo klientům MQTT. Bez změny výchozí přenosové fronty je třeba přidat definici vzdálené fronty pro každého klienta, který přijímá zprávy produktu IBM WebSphere MQ ; viz [“Přímé odeslání zprávy klientovi”](#) na stránce 121.

4. Chcete-li vytvořit jedno nebo více uživatelských jmen, postupujte podle pokynů v části [“Autorizování klientů MQTT pro přístup k objektům produktu WebSphere MQ”](#) na stránce 123 . ID uživatelů mají oprávnění publikovat, odebírat a odesílat publikace do klientů MQTT.

5. Nainstalujte službu telemetrie (MQXR)

```
cat /opt/<install_dir>/mqxr/samples/installMQXRService_unix.mqsc | runmqsc qMgr
```

Viz také příklad kódu v souboru [Obrázek 19](#) na stránce 117.

6. Spustit službu

```
echo "START SERVICE(SYSTEM.MQXR.SERVICE) " | runmqsc qMgr
```

Služba telemetrie (MQXR) se spouští automaticky při spuštění správce front.

Je spuštěn ručně v této úloze, protože správce front je již spuštěn.

7. Pomocí produktu IBM WebSphere MQ Explorer lze konfigurovat kanály telemetrie pro příjem připojení od klientů MQTT.

Kanály telemetrie musí být nakonfigurovány tak, aby jejich identity byly jedním z ID uživatelů definovaných v kroku 4.

Viz také termín [DEFINE CHANNEL \(MQTT\)](#).

8. Ověřte konfiguraci spuštěním ukázkového klienta.

Aby mohl ukázkový klient pracovat s vaším kanálem telemetrie, musí tento kanál autorizovat klienta k publikování, odběru a příjmu publikací. Ukázkový klient se při výchozím nastavení připojuje k kanálu telemetrie na portu 1883. Viz též [IBM WebSphere MQ Telemetry sample programs](#).

Příklad

[Obrázek 19](#) na stránce 117 ukazuje příkaz `runmqsc` k manuálnímu vytvoření `SYSTEM.MQXR.SERVICE` na Linux.

```
DEF SERVICE(SYSTEM.MQXR.SERVICE) +
  CONTROL(QMGR) +
  DESCR('Manages clients using MQXR protocols such as MQTT') +
  SERVTYPE(SERVER) +
  STARTCMD('+MQ_INSTALL_PATH+/mqxr/bin/runMQXRService.sh') +
  STARTARG('-m +QMNAME+ -d "+MQ_Q_MGR_DATA_PATH+" -g "+MQ_DATA_PATH+"') +
  STOPCMD('+MQ_INSTALL_PATH+/mqxr/bin/endMQXRService.sh') +
  STOPARG('-m +QMNAME+') +
  STDOUT('+MQ_Q_MGR_DATA_PATH+/mqxr.stdout') +
  STDERR('+MQ_Q_MGR_DATA_PATH+/mqxr.stderr')
```

Obrázek 19. `installMQXRService_unix.mqsc`

Konfigurace správce front pro telemetrie v produktu Windows

Chcete-li správce front spustit pro spuštění produktu IBM WebSphere MQ Telemetry, postupujte podle těchto pokynů. Automatizovanou proceduru můžete spustit, chcete-li nastavit jednodušší konfiguraci pomocí podpory produktu IBM WebSphere MQ Telemetry pro produkt IBM WebSphere MQ Explorer.

Než začnete

1. Informace o tom, jak instalovat produkt IBM WebSphere MQ a funkci IBM WebSphere MQ Telemetry naleznete v tématu [Instalace produktu IBM WebSphere MQ Telemetry](#).
2. Vytvořte a spusťte správce front. Na správce front se v této úloze odkazuje jako na *qMgr*.
3. Jako součást této úlohy můžete konfigurovat službu telemetrie (MQXR). Nastavení vlastností MQXR se ukládají do souboru vlastností specifického pro platformu: `mqxr_win.properties`. Soubor vlastností MQXR obvykle nemusíte upravovat přímo, protože téměř všechna nastavení mohou být konfigurována prostřednictvím příkazů administrace MQSC nebo MQ Explorer. Pokud se rozhodnete soubor upravit přímo, zastavte správce front před tím, než provedete změny. Viz [vlastnosti MQXR](#).

Informace o této úloze

Podpora IBM WebSphere MQ Telemetry pro IBM WebSphere MQ Explorer obsahuje průvodce a vzorovou příkazovou proceduru `sampleMQM`. Nastavení počáteční konfigurace pomocí ID uživatele "guest"; viz téma [Ověření instalace produktu IBM WebSphere MQ Telemetry pomocí ukázkových programů IBM WebSphere MQ Explorer a IBM WebSphere MQ Telemetry ukázkových programů](#).

Postupujte podle kroků v této úloze a nakonfigurujte produkt IBM WebSphere MQ Telemetry ručně pomocí různých schémat autorizace.

Postup

1. Otevřete příkazové okno v adresáři ukázek telemetrie.

Adresář ukázek telemetrie je `WMQ program installation directory\mqxr\samples`.

2. Vytvořte přenosovou frontu telemetrie.

```
echo DEFINE QLOCAL('SYSTEM.MQTT.TRANSMIT.QUEUE') USAGE(XMITQ) MAXDEPTH(100000) | runmqsc qMgr
```

Je-li služba telemetrie (MQXR) spuštěna poprvé, vytvoří `SYSTEM.MQTT.TRANSMIT.QUEUE`.

Je vytvořen ručně v této úloze, protože produkt `SYSTEM.MQTT.TRANSMIT.QUEUE` musí existovat před spuštěním služby telemetrie (MQXR), aby bylo možné k němu autorizovat přístup.

3. Nastavení výchozí přenosové fronty

```
echo ALTER QMGR DEFXMITQ('SYSTEM.MQTT.TRANSMIT.QUEUE') | runmqsc qMgr
```

Obrázek 20. Nastavit výchozí přenosovou frontu

Je-li služba telemetrie (MQXR) poprvé spuštěna, neupraví správce front, aby učinil `SYSTEM.MQTT.TRANSMIT.QUEUE` výchozí přenosovou frontu.

Chcete-li nastavit `SYSTEM.MQTT.TRANSMIT.QUEUE` jako výchozí přenosovou frontu, změňte výchozí vlastnost přenosové fronty. Upravte vlastnost pomocí příkazu IBM WebSphere MQ Explorer nebo pomocí příkazu v produktu [Obrázek 20 na stránce 118](#).

Změna výchozí přenosové fronty může kolidovat s vaší stávající konfigurací. Důvodem změny výchozí přenosové fronty do produktu `SYSTEM.MQTT.TRANSMIT.QUEUE` je usnadnění odesílání zpráv přímo klientům MQTT. Bez změny výchozí přenosové fronty je třeba přidat definici vzdálené fronty pro každého klienta, který přijímá zprávy produktu IBM WebSphere MQ; viz ["Přímé odeslání zprávy klientovi"](#) na stránce 121.

- Chcete-li vytvořit jedno nebo více uživatelských jmen, postupujte podle pokynů v části “Autorizování klientů MQTT pro přístup k objektům produktu WebSphere MQ” na stránce 123 . ID uživatelů mají oprávnění publikovat, odebírat a odesílat publikace do klientů MQTT.
- Nainstalujte službu telemetrie (MQXR)

```
type
installMQXRService_win.mqsc | runmqsc qMgr
```

- Spustit službu

```
echo START SERVICE(SYSTEM.MQXR.SERVICE) | runmqsc qMgr
```

Služba telemetrie (MQXR) se spouští automaticky při spuštění správce front.

Je spuštěn ručně v této úloze, protože správce front je již spuštěn.

- Pomocí produktu IBM WebSphere MQ Explorer lze konfigurovat kanály telemetrie pro příjem připojení od klientů MQTT.

Kanály telemetrie musí být nakonfigurovány tak, aby jejich identity byly jedním z ID uživatelů definovaných v kroku 4.

Viz také termín [DEFINE CHANNEL \(MQTT\)](#).

- Ověřte konfiguraci spuštěním ukázkového klienta.

Aby mohl ukázkový klient pracovat s vaším kanálem telemetrie, musí tento kanál autorizovat klienta k publikování, odběru a příjmu publikací. Ukázkový klient se při výchozím nastavení připojuje k kanálu telemetrie na portu 1883. Viz též [IBM WebSphere MQ Telemetry sample programs](#).

Ruční vytvoření SYSTEM.MQXR.SERVICE

Obrázek 21 na stránce 119 ukazuje příkaz `runmqsc` k manuálnímu vytvoření SYSTEM.MQXR.SERVICE na Windows.

```
DEF SERVICE(SYSTEM.MQXR.SERVICE) +
CONTROL(QMGR) +
DESCR('Manages clients using MQXR protocols such as MQTT') +
SERVTYPE(SERVER) +
STARTCMD('+MQ_INSTALL_PATH+\mqxr\bin\runMQXRService.bat') +
STARTARG('-m +QMNAME+ -d "+MQ_Q_MGR_DATA_PATH+\.' -g "+MQ_DATA_PATH+\.'') +
STOPCMD('+MQ_INSTALL_PATH+\mqxr\bin\endMQXRService.bat') +
STOPARG('-m +QMNAME+') +
STDOUT('+MQ_Q_MGR_DATA_PATH+\mqxr.stdout') +
STDERR('+MQ_Q_MGR_DATA_PATH+\mqxr.stderr')
```

Obrázek 21. `installMQXRService_win.mqsc`

Konfigurace distribuovaných front pro odesílání zpráv klientům MQTT

Aplikace produktu IBM WebSphere MQ mohou odesílat zprávy klientů MQTT v3 publikováním do odběru vytvořeného klientem nebo odesláním zprávy přímo. Je-li použita metoda Whichever, je zpráva vložena do produktu SYSTEM.MQTT.TRANSMIT.QUEUE a odeslána klientovi prostřednictvím služby telemetrie (MQXR). Existuje celá řada způsobů, jak umístit zprávu do SYSTEM.MQTT.TRANSMIT.QUEUE.

Publikování zprávy v odezvě na odběr klienta MQTT

Služba telemetrie (MQXR) vytváří odběr v zastoupení klienta MQTT. Klient je místem určení pro všechny publikace, které odpovídají odběru odesílanému klientem. Telemetrické služby předávají zpět odpovídající publikace zpět klientovi.

Klient MQTT je připojen k produktu WebSphere MQ jako správce front s jeho názvem správce front nastaveným na hodnotu `ClientIdentifier`. Místo určení pro publikace, které mají být odeslány na klienta, je přenosová fronta, SYSTEM.MQTT.TRANSMIT.QUEUE. Služba telemetrie předává zprávy v produktu SYSTEM.MQTT.TRANSMIT.QUEUE klientům MQTT s použitím názvu cílového správce front jako klíče ke specifickému klientovi.

Služba telemetrie (MQXR) otevře přenosovou frontu pomocí příkazu `ClientIdentifier` jako název správce front. Služba telemetrie (MQXR) předává obslužnou rutinu objektu fronty k volání produktu MQSUB , aby přesměroval publikování, která odpovídají odběru klienta. V rozlišení názvu objektu je `ClientIdentifier` vytvořen jako název vzdáleného správce front a přenosová fronta musí být interpretovaný jako `SYSTEM.MQTT.TRANSMIT.QUEUE`. Při použití standardního rozlišení názvu objektu WebSphere MQ se produkt `ClientIdentifier` řeší následujícím způsobem, viz [Tabulka 6 na stránce 120](#).

1. `ClientIdentifier` neodpovídá ničemu.

`ClientIdentifier` je název vzdáleného správce front. Neshoduje se s názvem lokálního správce front, aliasem správce front nebo názvem přenosové fronty.

Název fronty není definován. V současné době služba telemetrie (MQXR) nastavuje `SYSTEM.MQTT.PUBLICATION.QUEUE` jako název fronty. Klient MQTT v3 nepodporuje fronty, takže vyřešený název fronty je klientem ignorován.

Vlastnost lokálního správce front, Výchozí přenosová fronta, musí být název nastaven na `SYSTEM.MQTT.TRANSMIT.QUEUE`, aby bylo publikování odesláno na `SYSTEM.MQTT.TRANSMIT.QUEUE`, aby bylo odesláno klientovi.

2. `ClientIdentifier` se shoduje s aliasem správce front nazvaným `ClientIdentifier`.

`ClientIdentifier` je název vzdáleného správce front. Shoduje se s názvem aliasu správce front. Alias správce front musí být definován spolu s `ClientIdentifier` jako název vzdáleného správce front.

Nastavením názvu přenosové fronty v definici alias správce front není nutné, aby byl výchozí přenos nastaven na `SYSTEM.MQTT.TRANSMIT.QUEUE`.

Tabulka 6. Rozpoznání názvu alias správce front MQTT					
	Vstup		Výstup		
ClientIdentifier	Název správce front	Název fronty	Název správce front	Název fronty	Přenosová fronta
Neodpovídá ničemu	<code>ClientIdentifier</code>	<i>nedefinováno</i>	<code>ClientIdentifier</code>	<i>nedefinováno</i>	Výchozí přenosová fronta. <code>SYSTEM.MQTT.TRANSMIT.QUEUE</code>
Odpovídá aliasu správce front s názvem <code>ClientIdentifier</code>	<code>ClientIdentifier</code>	<i>nedefinováno</i>	<code>ClientIdentifier</code>	<i>nedefinováno</i>	<code>SYSTEM.MQTT.TRANSMIT.QUEUE</code>

Další informace o rozlišování názvů naleznete v tématu [Rozlišování názvů](#).

Všechny programy produktu WebSphere MQ mohou publikovat do stejného tématu. Publikování je odesláno svým odběratelům, včetně klientů MQTT v3 , kteří mají předplatné tématu.

Je-li v klastru vytvořeno administrativní téma s atributem `CLUSTER(clusterName)`, může každá aplikace v klastru publikovat na klienta; například:

```
echo DEFINE TOPIC('MQTTExamples') TOPICSTR('MQTT Examples') CLUSTER(MQTT) REPLACE | runmqsc qMgr
```

Obrázek 22. Definování tématu klastru v systému Windows

Poznámka: Neposkytujte SYSTEM.MQTT.TRANSMIT.QUEUE atribut klastru.

Odběratelé klienta MQTT a vydavatelé se mohou připojovat k různým správcům front. Odběratelé a vydavatelé mohou být součástí stejného klastru nebo připojeni k hierarchii publikování/odběru. Publikování je doručeno od vydavatele k odběrateli pomocí produktu WebSphere MQ.

Přímé odeslání zprávy klientovi

Alternativou ke klientovi, který vytváří odběr a přijímá publikování, která odpovídá tématu odběru, odešle přímo klientovi MQTT v3. Klientské aplikace MQTT V3 nemohou odesílat zprávy přímo, ale jiná aplikace, jako např. aplikace WebSphere MQ.

Aplikace WebSphere MQ musí znát `ClientIdentifier` klienta MQTT v3. Protože klienti MQTT v3 nemají fronty, je jako název tématu předán název cílové fronty do metody `messageArrived` klienta aplikace MQTT v3. Například v programu MQI vytvořte deskriptor objektu s klientem jako názvem `ObjectQmgrName`:

```
MQOD.ObjectQmgrName = ClientIdentifier;  
MQOD.ObjectName = name;
```

Obrázek 23. Deskriptor objektu MQI pro odeslání zprávy do cíle klienta MQTT v3

Je-li aplikace napsána pomocí platformy JMS, vytvořte místo určení typu point-to-point, například:

```
javax.jms.Destination jmsDestination =  
    (javax.jms.Destination)jmsFactory.createQueue  
    ("queue://ClientIdentifier/name");
```

Obrázek 24. Místo určení rozhraní JMS pro odeslání zprávy do klienta MQTT v3

Chcete-li odeslat nevyžádaná zpráva do klienta MQTT, použijte definici vzdálené fronty. Název vzdáleného správce front musí být resolevdo `ClientIdentifier` klienta. Přenosová fronta musí být interpretovaný jako SYSTEM.MQTT.TRANSMIT.QUEUE; viz [Tabulka 7 na stránce 121](#). Název vzdálené fronty může být cokoli. Klient jej přijme jako řetězec tématu.

Vstup		Výstup		
Název fronty	Název správce front	Název fronty	Název správce front	Přenosová fronta
Název definice vzdálené fronty	Název prázdného nebo lokálního správce front	Název vzdálené fronty použitý jako řetězec tématu	<code>ClientIdentifier</code>	SYSTEM.MQTT.TRANSMIT.QUEUE

Je-li klient připojen, je zpráva odeslána přímo klientovi MQTT, který volá metodu `messageArrived`; viz [messageArrived method](#).

Pokud byl klient odpojen s trvalou relací, je tato zpráva uložena v produktu SYSTEM.MQTT.TRANSMIT.QUEUE; viz [služba MQTT bez stavu a stavové relace](#). Je postoupen klientovi, když se klient znovu připojí k relaci.

Pokud odešlete netrvalou zprávu, odešle se klientovi s "nejvýše jednou" kvalitou služby, `QoS=0`. Pokud odešlete trvalou zprávu přímo klientovi, standardně se odešle s kvalitou služby "přesně jednou", `QoS=2`. Vzhledem k tomu, že klient nemusí mít mechanismus perzistence, může klient snížit kvalitu služby, kterou přijímá pro přímé odeslání zpráv. Chcete-li snížit kvalitu služby pro zprávy odesílané přímo na klienta, vytvořte odběr tématu DEFAULT. `QoS`. Určete maximální kvalitu služby, kterou klient může podporovat.

identifikace klienta MQTT, autorizace a ověření

Služba telemetrie (MQXR) publikuje nebo odebírá témata WebSphere MQ v zastoupení klientů MQTT s použitím kanálů MQTT. Administrátor produktu WebSphere MQ konfiguruje identitu kanálu MQTT, která se používá pro autorizaci produktu WebSphere MQ. Administrátor může definovat společnou identitu pro kanál nebo použít `Jméno uživatele` nebo `ClientIdentifier` klienta připojeného k danému kanálu.

Služba telemetrie (MQXR) může ověřit identitu klienta pomocí tokenu `Username` dodaného klientem nebo pomocí certifikátu klienta. `Jméno uživatele` je ověřováno pomocí hesla poskytnutého klientem.

Shrnutí: Identifikace klienta je výběrem identity klienta. V závislosti na kontextu je klient identifikován pomocí `ClientIdentifier`, `Username`, obecné identity klienta vytvořené administrátorem nebo klientským certifikátem. Identifikátor klienta používaný pro kontrolu pravosti nemusí být stejný jako identifikátor, který se používá pro autorizaci.

Klientské programy MQTT nastavují `Jméno uživatele` a `Heslo`, které jsou odeslány na server pomocí kanálu MQTT. Mohou také nastavit vlastnosti zabezpečení SSL, které jsou vyžadovány pro šifrování a ověření připojení. Administrátor rozhodne, zda má být ověřen kanál MQTT a jak ověřit kanál.

Chcete-li autorizovat klienta protokolu MQTT pro přístup k objektům produktu WebSphere MQ, autorizujte hodnotu `ClientIdentifier` nebo `Username` klienta nebo autorizujte společnou identitu klienta. Chcete-li povolit klientovi připojit se k produktu WebSphere MQ, ověřit identitu `Jméno uživatele` nebo použít certifikát klienta. Nakonfigurujte službu JAAS pro ověření uživatele `Jméno uživatele` a nakonfigurujte zabezpečení SSL pro ověření certifikátu klienta.

Pokud na straně klienta nastavíte `Heslo`, buď zašifrujte připojení pomocí sítě VPN, nebo nakonfigurujte kanál MQTT tak, aby používal zabezpečení SSL, aby bylo heslo nastaveno jako soukromé.

Certifikáty klientů je obtížné spravovat. Z tohoto důvodu, pokud jsou rizika spojená s ověřením hesla přijatelná, ověření hesla se často používá k ověření klientů.

Pokud je k dispozici bezpečný způsob správy a ukládání certifikátu klienta, je možné spolehnout se na ověření certifikátu. Avšak zřídka je možné bezpečně spravovat certifikáty v typech prostředí, ve kterých se používá telemetrie. Místo toho je ověření zařízení používajících certifikáty klientů doplněno ověřením hesel klienta na serveru. Z důvodu další složitosti je použití certifikátů klienta omezeno na vysoce citlivé aplikace. Použití dvou forem ověření se nazývá dvoufaktorové ověření. Musíte znát jeden z faktorů, jako je heslo, a mít druhý takový certifikát, jako je například certifikát.

Ve vysoce citlivé aplikaci, jako je například čip a poziční zařízení, je toto zařízení během výroby zablokováno, aby se zabránilo manipulaci s vnitřním hardwarem a softwarem. Důvěryhodný, časově omezený, klientský certifikát je zkopírován do zařízení. Zařízení je implementováno do umístění, kde má být použito. Další ověřování se provádí pokaždé, když je zařízení používáno, buď pomocí hesla, nebo jiným certifikátem z čipové karty.

Identita a autorizace klienta MQTT

K získání přístupu k objektům WebSphere MQ použijte `ClientIdentifier`, `Username` nebo společnou identitu klienta.

Administrátor produktu WebSphere MQ má tři volby pro výběr identity kanálu MQTT. Administrátor provede výběr při definování nebo úpravě kanálu MQTT používaného klientem. Identita se používá k autorizaci přístupu k tématům týkajícím se produktu WebSphere MQ. K dispozici jsou tyto volby:

1. Identifikátor klienta.
2. Identita, kterou administrátor poskytuje pro kanál.
3. `Jméno uživatele` předané z klienta MQTT.

`Jméno uživatele` je atribut třídy voleb `MqttConnect`. Musí být nastavena dříve, než se klient připojí ke službě. Jeho výchozí hodnota je `null`.

Pomocí příkazu WebSphere MQ **setmqaut** můžete vybrat, které objekty a které akce mají být použity identitou přidružené k kanálu MQTT. Chcete-li například autorizovat určitou identitu kanálu, MQTTClient, kterou poskytuje administrátor správce front, QM1:

```
setmqaut -m QM1 -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p MQTTClient -all +put  
setmqaut -m QM1 -t topic -n SYSTEM.BASE.TOPIC -p MQTTClient -all +pub +sub
```

Autorizování klientů MQTT pro přístup k objektům produktu WebSphere MQ

Chcete-li autorizovat klienty MQTT k publikování a přihlášení k odběru objektů WebSphere MQ, postupujte podle následujících kroků. Postup sleduje čtyři alternativní vzory řízení přístupu.

Než začnete

Klienti MQTT jsou autorizováni přistupovat k objektům v produktu WebSphere MQ prostřednictvím přiřazení identity při připojení k telemetrickým kanálům. Produkt WebSphere MQ Administrator konfiguruje kanál telemetrie pomocí nástroje WebSphere MQ Explorer k poskytnutí jednoho ze tří typů identit:

1. ClientIdentifier
2. Jméno uživatele
3. Název, který administrátor přiřadí ke kanálu.

Je-li použit typ Whichever, musí být identita definována pro produkt WebSphere MQ jako činitel instalovanými autorizační službou. Výchozí autorizační služba na systému Windows nebo Linux se nazývá OAM (Object Authority Manager). Používáte-li OAM, musí být identita definována jako ID uživatele.

Použijte identitu, chcete-li poskytnout klientovi nebo kolekci klientů oprávnění k publikování nebo odběru témat definovaných v produktu WebSphere MQ. Pokud se klient MQTT přihlásil k odběru tématu, použijte k získání oprávnění pro příjem výsledných publikování jeho identitu.

Je těžké spravovat systém s desítkami tisíc klientů MQTT, přičemž každý z nich vyžaduje individuální přístupová oprávnění. Jedním řešením je definovat společné identity a přidružit jednotlivé klienty MQTT k jedné z obecných identit. Definujte tolik obecných identit, kolik jich potřebujete k definování různých kombinací oprávnění. Dalším řešením je napsat svou vlastní autorizační službu, která se může snadněji vypořádat s tisíci uživateli, než je operační systém.

Klienty MQTT můžete kombinovat do společných identit dvěma způsoby pomocí OAM:

1. Definujte více kanálů telemetrie, každý s jiným ID uživatele, který administrátor přiděluje pomocí Průzkumníka produktu WebSphere MQ. Klienti, kteří se připojují pomocí různých čísel portů TCP/IP, jsou přidruženi k různým kanálům telemetrie a přiřazují se různým identitám.
2. Definujte jeden kanál telemetrie, ale každý klient vyberte volbu Jméno uživatele z malé sady ID uživatelů. Administrátor konfiguruje kanál telemetrie tak, aby jako svou identitu vybral klienta Jméno uživatele.

V této úloze se identita kanálu telemetrie nazývá *mqttUser*, bez ohledu na to, jak je nastavena. Pokud kolekce klientů používají různé identity, použijte více *mqttUsers*, jeden pro každou kolekci klientů. Protože úloha používá nástroj OAM, každý *mqttUser* musí být ID uživatele.

Informace o této úloze

V této úloze máte na výběr ze čtyř modelů řízení přístupu, které můžete upravit na specifické požadavky. Vzory se liší v granularitě řízení přístupu.

- [“Bez řízení přístupu” na stránce 124](#)
- [“Hrubé řízení přístupu” na stránce 124](#)
- [“Řízení přístupu s vysokou úrovní granularit” na stránce 124](#)
- [“Detailní řízení přístupu” na stránce 124](#)

Výsledkem těchto modelů je přiřadit sady oprávnění produktu *mqttUsers* k publikování a odběru produktu WebSphere MQa přijímat publikování z produktu WebSphere MQ.

Bez řízení přístupu

Klienti MQTT mají k dispozici administrativní oprávnění produktu WebSphere MQ a mohou provádět jakékoli akce s libovolným objektem.

Postup

1. Vytvořte ID uživatele *mqttUser* , který bude fungovat jako identita všech klientů MQTT.
2. Přidejte *mqttUser* do skupiny *mqm* ; viz [Přidání uživatele do skupiny na systému Windows](#) nebo [Přidání uživatele do skupiny na systému Linux](#)

Hrubé řízení přístupu

Klienti MQTT mají oprávnění k publikování a odběru a k odesílání zpráv na klienty MQTT. Nemají oprávnění k provádění jiných akcí nebo k přístupu k jiným objektům.

Postup

1. Vytvořte ID uživatele *mqttUser* , který bude fungovat jako identita všech klientů MQTT.
2. Autorizovat produkt *mqttUser* k publikování a odběru všech témat a k odesílání publikací klientům MQTT.

```
setmqaut -m qMgr -t topic -n SYSTEM.BASE.TOPIC -p mqttUser -all +pub +sub  
setmqaut -m qMgr -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p mqttUser -all +put
```

Řízení přístupu s vysokou úrovní granularity

Klienti MQTT jsou rozděleni do různých skupin pro publikování a přihlášení k různým sadám témat a k odesílání zpráv na klienty MQTT.

Postup

1. Vytvořte více uživatelských jmen, *mqttUsers* a více administrativních témat ve stromu témat publikování/odběru.
2. Autorizovat různé *mqttUsers* pro různá témata.

```
setmqaut -m qMgr -t topic -n topic1 -p mqttUserA -all +pub +sub  
setmqaut -m qMgr -t topic -n topic2 -p mqttUserB -all +pub +sub
```

3. Vytvořte skupinu *mqtt* a přidejte všechny *mqttUsers* do skupiny.
4. Autorizovat *mqtt* k odesílání témat klientům MQTT.

```
setmqaut -m qMgr -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p mqtt -all +put
```

Detailní řízení přístupu

Klienti MQTT jsou vestavěni do existujícího systému řízení přístupu, který autorizuje skupiny k provádění akcí na objektech.

Informace o této úloze

ID uživatele je přiřazeno k jedné nebo více skupinám operačního systému v závislosti na oprávněních, která vyžaduje. Pokud jsou aplikace WebSphere MQ publikující a odebírající se ke stejnému prostoru tématu jako klienti MQTT, použijte tento model. Na skupiny se odkazuje jako na PublishX, SubscribeYa mqtt

PublishX

Členové skupin produktu PublishX mohou publikovat do produktu *topicX*.

SubscribeY

Členové skupin produktu SubscribeY se mohou přihlásit k odběru produktu *topicY*.

mqtt

Členové skupiny *mqtt* mohou odesílat publikování na klienty MQTT.

Postup

1. Vytvořte více skupin, PublishX a SubscribeY, které jsou přiděleny na více administrativních témat ve stromu témat publikování/odběru.
2. Vytvořte skupinu *mqtt*.
3. Vytvořte více ID uživatelů, *mqttUsers* a přidejte uživatele do kterékoli ze skupin v závislosti na tom, k čemu mají oprávnění.
4. Autorizujte různé skupiny PublishX a SubscribeX pro různá témata a autorizujte skupinu *mqtt* k odesílání zpráv klientům MQTT.

```
setmqaut -m qMgr -t topic -n topic1 -p PublishX -all +pub
setmqaut -m qMgr -t topic -n topic1 -p SubscribeX -all +pub +sub
setmqaut -m qMgr -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p mqtt -all +put
```

Ověření klienta MQTT pomocí hesla

Ověřte Jméno uživatele pomocí hesla klienta. Klienta můžete ověřit pomocí jiné identity k totožnosti, která se používá k autorizaci klienta pro publikování a odběr témat.

Služba telemetrie (MQXR) používá službu JAAS k ověření klienta Username. Volba JAAS používá heslo dodané klientem MQTT.

Administrátor produktu WebSphere MQ rozhoduje, zda ověřit identitu Jméno uživatele, nebo ne provést ověření ve všech, konfiguracích kanálu MQTT, ke kterému se klient připojuje. Klienti mohou být přiřazeni k různým kanálům a každý kanál lze nakonfigurovat tak, aby ověřoval klienty různými způsoby. Pomocí JAAS můžete nakonfigurovat, které metody musí ověřit klienta a které mohou volitelně autentizovat klienta.

Výběr identity pro ověření nemá vliv na volbu identity pro autorizaci. Možná budete chtít nastavit obecnou identitu pro autorizaci pro administrativní pohodlí, ale autentizovat každého uživatele k použití této identity. Následující postup popisuje kroky k ověření totožnosti jednotlivých uživatelů při používání společné identity:

1. Administrátor produktu WebSphere MQ nastavuje identitu kanálu MQTT na libovolný název, jako je například produkt MQTTClientUser, pomocí Průzkumníka produktu WebSphere MQ.
2. Administrátor produktu WebSphere MQ autorizuje příkaz MQTTClient k publikování a odběru libovolného tématu:

```
setmqaut -m QM1 -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p MQTTClient -all +put
setmqaut -m QM1 -t topic -n SYSTEM.BASE.TOPIC -p MQTTClient -all +pub +sub
```

3. Vývojář aplikací klienta MQTT vytvoří objekt MqttConnectOptions a nastaví Jméno uživatele a Heslo, než se připojí k serveru.
4. Vývojář zabezpečení vytvoří JAAS LoginModule, aby ověřil Jméno uživatele s heslem Heslo a zahrne jej do konfiguračního souboru JAAS.
5. Administrátor produktu WebSphere MQ konfiguruje kanál MQTT k ověření identity uživatele UserName klienta pomocí JAAS.

Ověření klienta MQTT pomocí SSL

Připojení mezi klientem MQTT a správcem front jsou vždy iniciována klientem MQTT. Klient MQTT je vždy klientem SSL. Ověření klienta serveru a ověření serveru klienta MQTT jsou volitelná.

Poskytnutím klienta se soukromým podepsaným digitálním certifikátem můžete ověřit klienta MQTT na serveru IBM WebSphere MQ. Administrátor serveru IBM WebSphere MQ může přinutit klienty MQTT, aby se ověřovali ve správci front pomocí protokolu SSL. Ověření klienta lze požadovat pouze jako součást vzájemného ověření.

Jako alternativu k používání SSL ověřují některé druhy VPN (Virtual Private Network), jako je IPsec, koncové body připojení TCP/IP. VPN šifruje každý IP paket, který se posílá po síti. Jakmile je navázáno spojení s VPN, vytvořili jste důvěryhodnou síť. Klienty MQTT je možné připojit ke kanálům telemetrie pomocí protokolu TCP/IP v síti VPN.

Ověření klienta pomocí SSL spoléhá na to, že klient má tajný klíč. Tajný klíč je soukromý klíč klienta v případě certifikátu podepsaného držitelem nebo klíč poskytnutý certifikační autoritou. Klíč se používá k podepsání digitálního certifikátu klienta. Každý, kdo má k dispozici odpovídající veřejný klíč, může ověřit digitální certifikát. Certifikáty mohou být důvěryhodné, nebo pokud jsou zřetězeny, jsou trasovány zpět prostřednictvím řetězu certifikátů s důvěryhodným kořenovým certifikátem. Ověření klienta odešle všechny certifikáty v řetězu certifikátů, které poskytuje klient na server. Server kontroluje řetěz certifikátů, dokud nenajde certifikát, kterému důvěřuje. Důvěryhodný certifikát je buď veřejný certifikát generovaný z certifikátu podepsaného držitelem, nebo kořenový certifikát, který je obvykle vydán certifikační autoritou. Jako finální, volitelný, krok lze důvěryhodný certifikát porovnat s "aktivním" seznamem odvolaných certifikátů.

Důvěryhodný certifikát může být vydán certifikační autoritou a je již obsažen v úložišti certifikátů JRE. Může se jednat o certifikát podepsaný držitelem nebo o certifikát, který byl přidán do úložiště klíčů kanálu telemetrie jako důvěryhodný certifikát.

Poznámka: Kanál telemetrie má kombinované úložiště klíčů/úložiště údajů o důvěryhodnosti, které drží soukromé klíče jednomu nebo více kanálům telemetrie, a všechny veřejné certifikáty potřebné k ověření klientů. Vzhledem k tomu, že kanál SSL musí mít úložiště klíčů a jedná se o stejný soubor jako úložiště údajů o důvěryhodnosti kanálu, není nikdy na úložiště certifikátů JRE odkazováno. Z toho vyplývá, že pokud ověření klienta vyžaduje kořenový certifikát CA, musíte umístit kořenový certifikát do úložiště klíčů pro kanál i v případě, že je již kořenový certifikát CA v úložišti certifikátů JRE. Na úložiště certifikátů JRE se nikdy neodkazuje.

Přemýšlejte o hrozbách, s nimiž má ověřování klienta bojovat, a rolích, které klient a server hrají při potlačování hrozeb. Ověřování samotného certifikátu klienta není dostatečné k zabránění neoprávněnému přístupu do systému. Pokud má zařízení klienta v držení nějaký jiný uživatel, nemusí zařízení klienta nutně jednat s oprávněním patřícím držiteli certifikátu. Nikdy se proti nechtěným útokům nespolehejte na jedinou obranu. Přínejmenším použijte dvoufaktorovou metodu ověření a doplňte držení certifikátu znalostmi o soukromých informacích. Můžete například použít službu JAAS a ověřit klienta pomocí hesla vydaného serverem.

Primární hrozbou pro certifikát klienta je to, že se dostane do špatných rukou. Certifikát je uchovávan v heslem chráněném úložišti klíčů u klienta. Jak se dostane do úložiště klíčů? Jakým způsobem klient MQTT získá heslo k úložišti klíčů? Jak je bezpečná je ochrana pomocí hesla? Telemetrická zařízení jsou často snadno odnímatelná, a pak mohou být někde v soukromí napadnuta. Musí být hardware zařízení odolný proti neoprávněné manipulaci? Obtíže s rozdělením a ochranou certifikátů na straně klienta jsou známy, označují se jako problém se správou klíčů.

Sekundární hrozbou je, že zařízení může být zneužito pro přístup k serverům nezamýšlenými způsoby. Například když je manipulováno s aplikací MQTT, lze využít slabinu v konfiguraci serveru s použitím identity ověřeného klienta.

Chcete-li ověřit klienta MQTT pomocí protokolu SSL, nakonfigurujte kanál telemetrie a klienta.

-
-

Konfigurace kanálu telemetrie pro ověření klienta MQTT pomocí SSL

Administrátor produktu IBM WebSphere MQ konfiguruje kanály telemetrie na serveru. Každý kanál je konfigurován tak, aby přijímal připojení TCP/IP na jiném čísle portu. Kanály SSL jsou konfigurovány tak, aby přístup k souborům s klíči byl chráněn přístupovou frází. Je-li kanál SSL definován bez přístupové fráze nebo bez souboru s klíči, kanál připojení SSL neakceptuje.

Nastavte vlastnost `com.ibm.mq.MQTT.ClientAuth` kanálu telemetrie SSL na `REQUIRED`, chcete-li přinutit všechny klienty, kteří se připojují k tomuto kanálu, aby poskytli důkaz o tom, že mají ověřené digitální certifikáty. Certifikáty klientů jsou ověřovány pomocí certifikátů od certifikačních autorit, což vede k důvěryhodnému kořenovému certifikátu. Je-li certifikát klienta podepsán sám sebou nebo

je podepsán certifikátem, který pochází z certifikační autority, musí být bezpečně uloženo veřejně podepsané certifikáty klienta nebo certifikační autority na serveru.

Umístěte certifikát klienta s veřejným podpisem nebo certifikát z certifikační autority do úložiště klíčů kanálu telemetrie. Na serveru jsou veřejně podepsané certifikáty uloženy ve stejném souboru s klíči jako soukromě podepsané certifikáty, spíše než v odděleném úložišti údajů o důvěryhodnosti.

Server ověřuje podpis všech klientských certifikátů, které se posílají pomocí všech veřejných certifikátů a šifrovacích sad, které má. Server ověřuje řetěz klíčů. Správce front může být konfigurován tak, aby testoval certifikát na seznamu odvolaných certifikátů (CRL). Vlastnost seznamu názvů odvolání správce front je SSLCRLNL.

Je-li některý z certifikátů, který klient odešle, ověřen certifikátem v úložišti klíčů serveru, je klient autentizován.

Administrátor produktu WebSphere MQ může nakonfigurovat stejný kanál telemetrie, aby používal službu JAAS ke kontrole klienta `UserName` nebo `ClientIdentifier` klienta s klientem `Password`.

Pro více kanálů telemetrie můžete použít stejné úložiště klíčů.

Ověření alespoň jednoho digitálního certifikátu v úložišti klíčů klienta chráněného heslem na zařízení ověřuje klienta na serveru. Digitální certifikát se používá pouze pro ověření produktem WebSphere MQ. Nepoužívá se k ověření adresy TCP/IP klienta, nebo k nastavení identity klienta pro autorizaci nebo účtování. Identita klienta adoptovaného serverem je buď `Jméno uživatele`, nebo `ClientIdentifier` klienta, nebo identita vytvořená administrátorem produktu WebSphere MQ.

Šifrovací sady zabezpečení SSL můžete také použít pro ověřování klienta. Zde je abecední seznam šifrovacích sad SSL, které jsou momentálně podporované:

- SSL_DH_anon_EXPORT_WITH_DES40_CBC_SHA
- SSL_DH_anon_EXPORT_WITH_RC4_40_MD5
- SSL_DH_anon_WITH_3DES_EDE_CBC_SHA
- SSL_DH_anon_WITH_AES_128_CBC_SHA
- SSL_DH_anon_WITH_DES_CBC_SHA
- SSL_DH_anon_WITH_RC4_128_MD5
- SSL_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA
- SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA
- SSL_DHE_DSS_WITH_AES_128_CBC_SHA
- SSL_DHE_DSS_WITH_DES_CBC_SHA
- SSL_DHE_DSS_WITH_RC4_128_SHA
- SSL_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA
- SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA
- SSL_DHE_RSA_WITH_AES_128_CBC_SHA
- SSL_DHE_RSA_WITH_DES_CBC_SHA
- SSL_KRB5_EXPORT_WITH_DES_CBC_40_MD5
- SSL_KRB5_EXPORT_WITH_DES_CBC_40_SHA
- SSL_KRB5_EXPORT_WITH_RC4_40_MD5
- SSL_KRB5_EXPORT_WITH_RC4_40_SHA
- SSL_KRB5_WITH_3DES_EDE_CBC_MD5
- SSL_KRB5_WITH_3DES_EDE_CBC_SHA
- SSL_KRB5_WITH_DES_CBC_MD5
- SSL_KRB5_WITH_DES_CBC_SHA
- SSL_KRB5_WITH_RC4_128_MD5
- SSL_KRB5_WITH_RC4_128_SHA

- SSL_RSA_EXPORT_WITH_DES40_CBC_SHA
- SSL_RSA_EXPORT_WITH_RC4_40_MD5
- SSL_RSA_FIPS_WITH_3DES_EDE_CBC_SHA
- **V 7.5.0.2** SSL_RSA_FIPS_WITH_AES_128_CBC_SHA256
- **V 7.5.0.2** SSL_RSA_FIPS_WITH_AES_256_CBC_SHA256
- SSL_RSA_FIPS_WITH_DES_CBC_SHA
- SSL_RSA_WITH_3DES_EDE_CBC_SHA
- SSL_RSA_WITH_AES_128_CBC_SHA
- **V 7.5.0.2** SSL_RSA_WITH_AES_128_CBC_SHA256
- **V 7.5.0.2** SSL_RSA_WITH_AES_256_CBC_SHA256
- SSL_RSA_WITH_DES_CBC_SHA
- SSL_RSA_WITH_NULL_MD5
- SSL_RSA_WITH_NULL_SHA
- **V 7.5.0.2** SSL_RSA_WITH_NULL_SHA256
- SSL_RSA_WITH_RC4_128_MD5
- SSL_RSA_WITH_RC4_128_SHA

V 7.5.0.2 Plánujete-li používat šifrovací sady produktu SHA-2 , viz [Systémové požadavky pro použití šifrovacích sad produktu SHA-2 s kanály MQTT](#).

Související pojmy

“Konfigurace kanálu telemetrie pro ověření kanálu pomocí zabezpečení SSL” na stránce 129
 Administrátor produktu IBM WebSphere MQ konfiguruje kanály telemetrie na serveru. Každý kanál je konfigurován tak, aby přijímal připojení TCP/IP na jiném čísle portu. Kanály SSL jsou konfigurovány tak, aby přístup k souborům s klíči byl chráněn přístupovou frází. Je-li kanál SSL definován bez přístupové fráze nebo bez souboru s klíči, kanál připojení SSL neakceptuje.

[CipherSpecs a CipherSuites](#)

Související odkazy

[DEFINOVAT KANÁL \(MQTT\)](#)

[ZMĚNIT KANÁL \(MQTT\)](#)

Ověření kanálu telemetrie pomocí zabezpečení SSL

Připojení mezi klientem MQTT a správcem front jsou vždy iniciována klientem MQTT. Klient MQTT je vždy klientem SSL. Ověření klienta serveru a ověření serveru klienta MQTT jsou volitelná.

Klient se vždy pokusí o ověření serveru, pokud není nakonfigurován pro použití specifikace CipherSpec, která podporuje anonymní připojení. Pokud se ověření nezdaří, připojení není navázáno.

Jako alternativu k používání SSL ověřují některé druhy VPN (Virtual Private Network), jako je IPsec, koncové body připojení TCP/IP. VPN šifruje každý IP paket, který se posílá po síti. Jakmile je navázáno spojení s VPN, vytvořili jste důvěryhodnou síť. Klienty MQTT je možné připojit ke kanálům telemetrie pomocí protokolu TCP/IP v síti VPN.

Ověření serveru pomocí SSL ověřuje server, na který chcete zasílat důvěrné informace. Klient provádí kontroly, které odpovídají certifikátům odeslaným ze serveru, proti certifikátům umístěným v úložišti údajů o důvěryhodnosti nebo ve svém úložišti prostředí JRE cacerts .

Úložiště certifikátů JRE je soubor JKS, cacerts. Nachází se v adresáři JRE InstallPath\lib\security\. Je nainstalován s výchozím heslem changeit. Můžete buď uložit certifikáty, kterým důvěřujete, do úložiště certifikátů JRE, nebo do úložiště údajů o důvěryhodnosti klienta. Nelze používat obě úložiště. Úložiště údajů o důvěryhodnosti klienta použijte v případě, že chcete

zachovat veřejné certifikáty, kterým klient důvěřuje, od certifikátů, které používají jiné aplikace Java. Úložiště certifikátů JRE používejte v případě, že chcete používat společné úložiště certifikátů pro všechny aplikace Java spuštěné v rámci klienta. Rozhodnete-li se použít úložiště certifikátů JRE, zkontrolujte certifikáty, které obsahuje, abyste se ujistili, že jim důvěřujete.

Konfiguraci JSSE můžete upravit zadáním jiného poskytovatele důvěryhodnosti. Poskytovatele důvěryhodnosti můžete upravit tak, aby prováděl různé kontroly certifikátů. V některých prostředích OGSAN, které používají klienta MQTT, prostředí poskytuje jiného poskytovatele důvěryhodnosti.

Chcete-li ověřit kanál telemetrie pomocí SSL, nakonfigurujte server a klienta.

-
-

Konfigurace kanálu telemetrie pro ověření kanálu pomocí zabezpečení SSL

Administrátor produktu IBM WebSphere MQ konfiguruje kanály telemetrie na serveru. Každý kanál je konfigurován tak, aby přijímal připojení TCP/IP na jiném čísle portu. Kanály SSL jsou konfigurovány tak, aby přístup k souborům s klíči byl chráněn přístupovou frází. Je-li kanál SSL definován bez přístupové fráze nebo bez souboru s klíči, kanál připojení SSL neakceptuje.

Uložte digitální certifikát serveru, podepsaný svým soukromým klíčem, v úložišti klíčů, které bude kanál telemetrie používat na serveru. Uložte všechny certifikáty ve svém klíčovém řetězci v úložišti klíčů, chcete-li přenést klíč řetězce do klienta. Konfigurujte kanál telemetrie pomocí průzkumníka WebSphere MQ pro použití zabezpečení SSL. Zadejte ji s cestou k úložišti klíčů a přístupové heslo pro přístup k úložišti klíčů. Pokud nenastavíte číslo portu TCP/IP kanálu, číslo portu kanálu telemetrie SSL se standardně nastaví na hodnotu 8883.

Šifrovací sady zabezpečení SSL můžete také použít pro ověření kanálu. Zde je abecední seznam šifrovacích sad SSL, které jsou momentálně podporované:

- SSL_DH_anon_EXPORT_WITH_DES40_CBC_SHA
- SSL_DH_anon_EXPORT_WITH_RC4_40_MD5
- SSL_DH_anon_WITH_3DES_EDE_CBC_SHA
- SSL_DH_anon_WITH_AES_128_CBC_SHA
- SSL_DH_anon_WITH_DES_CBC_SHA
- SSL_DH_anon_WITH_RC4_128_MD5
- SSL_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA
- SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA
- SSL_DHE_DSS_WITH_AES_128_CBC_SHA
- SSL_DHE_DSS_WITH_DES_CBC_SHA
- SSL_DHE_DSS_WITH_RC4_128_SHA
- SSL_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA
- SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA
- SSL_DHE_RSA_WITH_AES_128_CBC_SHA
- SSL_DHE_RSA_WITH_DES_CBC_SHA
- SSL_KRB5_EXPORT_WITH_DES_CBC_40_MD5
- SSL_KRB5_EXPORT_WITH_DES_CBC_40_SHA
- SSL_KRB5_EXPORT_WITH_RC4_40_MD5
- SSL_KRB5_EXPORT_WITH_RC4_40_SHA
- SSL_KRB5_WITH_3DES_EDE_CBC_MD5
- SSL_KRB5_WITH_3DES_EDE_CBC_SHA
- SSL_KRB5_WITH_DES_CBC_MD5

- SSL_KRB5_WITH_DES_CBC_SHA
- SSL_KRB5_WITH_RC4_128_MD5
- SSL_KRB5_WITH_RC4_128_SHA
- SSL_RSA_EXPORT_WITH_DES40_CBC_SHA
- SSL_RSA_EXPORT_WITH_RC4_40_MD5
- SSL_RSA_FIPS_WITH_3DES_EDE_CBC_SHA
- **V 7.5.0.2** SSL_RSA_FIPS_WITH_AES_128_CBC_SHA256
- **V 7.5.0.2** SSL_RSA_FIPS_WITH_AES_256_CBC_SHA256
- SSL_RSA_FIPS_WITH_DES_CBC_SHA
- SSL_RSA_WITH_3DES_EDE_CBC_SHA
- SSL_RSA_WITH_AES_128_CBC_SHA
- **V 7.5.0.2** SSL_RSA_WITH_AES_128_CBC_SHA256
- **V 7.5.0.2** SSL_RSA_WITH_AES_256_CBC_SHA256
- SSL_RSA_WITH_DES_CBC_SHA
- SSL_RSA_WITH_NULL_MD5
- SSL_RSA_WITH_NULL_SHA
- **V 7.5.0.2** SSL_RSA_WITH_NULL_SHA256
- SSL_RSA_WITH_RC4_128_MD5
- SSL_RSA_WITH_RC4_128_SHA

V 7.5.0.2 Plánujete-li používat šifrovací sady produktu SHA-2 , viz [Systémové požadavky pro použití šifrovacích sad produktu SHA-2 s kanály MQTT](#).

Související pojmy

“Konfigurace kanálu telemetrie pro ověření klienta MQTT pomocí SSL” na stránce 126

Administrátor produktu IBM WebSphere MQ konfiguruje kanály telemetrie na serveru. Každý kanál je konfigurován tak, aby přijímal připojení TCP/IP na jiném čísle portu. Kanály SSL jsou konfigurovány tak, aby přístup k souborům s klíči byl chráněn přístupovou frází. Je-li kanál SSL definován bez přístupové fráze nebo bez souboru s klíči, kanál připojení SSL neakceptuje.

[CipherSpecs a CipherSuites](#)

Související odkazy

[DEFINOVAT KANÁL \(MQTT\)](#)

[ZMĚNIT KANÁL \(MQTT\)](#)

Soukromí zveřejňování kanálů telemetrie

Soukromí publikování protokolu MQTT odeslané v obou směrech přes kanály telemetrie jsou zabezpečeny pomocí SSL pro šifrování přenosů přes připojení.

Klienti MQTT, kteří se připojují k telemetrickým kanálům, používají zabezpečení SSL k zabezpečení soukromých informací přenášených na kanálu pomocí symetrického šifrování klíče. Vzhledem k tomu, že koncové body nejsou ověřeni, nelze samotné šifrování kanálu důvěřovat. Kombinuje zabezpečení ochrany soukromí se serverem nebo vzájemným autentizací.

Jako alternativu k používání SSL ověřují některé druhy VPN (Virtual Private Network), jako je IPsec, koncové body připojení TCP/IP. VPN šifruje každý IP paket, který se posílá po síti. Jakmile je navázáno spojení s VPN, vytvořili jste důvěryhodnou síť. Klienty MQTT je možné připojit ke kanálům telemetrie pomocí protokolu TCP/IP v síti VPN.

Pro typickou konfiguraci, která šifruje kanál a ověřuje server, nahlédněte do [“Ověření kanálu telemetrie pomocí zabezpečení SSL”](#) na stránce 128.

Šifrování připojení SSL bez ověření serveru odkrývá připojení k útokům typu "man-in-the-middle-middle". Ačkoli informace, které vyměňujete, jsou chráněny proti odposlouchávání, nevíte, s kým si ji vyměňujete. Pokud neovládnete síť, vystavujete se někomu, kdo zadrží vaše IP přenosy a maskuje se jako koncový bod.

Můžete vytvořit šifrované připojení SSL bez ověření serveru pomocí výměny klíčů Diffie-Hellman CipherSpec, která podporuje anonymní SSL. Hlavní utajený údaj, sdílený mezi klientem a serverem a používaný k šifrování přenosů SSL, je vytvořen bez výměny soukromě podepsaného certifikátu serveru.

Vzhledem k tomu, že anonymní připojení jsou nezabezpečená, většina implementací SSL nestandardně používá anonymní CipherSpecs. Pokud je požadavek klienta na připojení SSL přijat kanálem telemetrie, kanál musí mít úložiště klíčů chráněné heslem. Ve výchozím nastavení, protože implementace SSL nepoužívají anonymní CipherSpecs, musí úložiště klíčů obsahovat soukromě podepsaný certifikát, který může klient ověřit.

Pokud použijete anonymní CipherSpecs, úložiště klíčů serveru musí existovat, ale nemusí obsahovat žádné soukromě podepsané certifikáty.

Jiným způsobem, jak navázat šifrované spojení, je nahradit poskytovatele důvěryhodnosti na klientovi vlastním implementací. Váš poskytovatel důvěryhodnosti by neověřil certifikát serveru, ale připojení by bylo šifrováno.

Konfigurace zabezpečení SSL pro klienty MQTT a kanály telemetrie

Klienti MQTT a služba WebSphere MQ Telemetry (MQXR) používají k připojení telemetrických kanálů pomocí zabezpečení SSL Java Secure Socket Extension (JSSE). Démon IBM WebSphere MQ Telemetry pro zařízení nepodporuje SSL.

Konfigurujte zabezpečení SSL pro ověření kanálu telemetrie, klienta MQTT a šifrování přenosu zpráv mezi klienty a telemetrickým kanálem.

Jako alternativu k používání SSL ověřují některé druhy VPN (Virtual Private Network), jako je IPsec, koncové body připojení TCP/IP. VPN šifruje každý IP paket, který se posílá po síti. Jakmile je navázáno spojení s VPN, vytvořili jste důvěryhodnou síť. Klienty MQTT je možné připojit ke kanálům telemetrie pomocí protokolu TCP/IP v síti VPN.

Můžete nakonfigurovat připojení mezi klientem Java MQTT a kanálem telemetrie, chcete-li používat protokol SSL přes TCP/IP. To, co je zabezpečeno, závisí na tom, jak nakonfigurovat SSL pro použití JSSE. Počínaje nejbezpečnějšími konfiguracemi můžete nakonfigurovat tři různé úrovně zabezpečení:

1. Povolte připojení pouze důvěryhodných klientů MQTT. Připojte klienta MQTT pouze k důvěryhodnému kanálu telemetrie. Šifrování zpráv mezi klientem a správcem front; viz [“Ověření klienta MQTT pomocí SSL”](#) na stránce 125
2. Připojte klienta MQTT pouze k důvěryhodnému kanálu telemetrie. Šifrování zpráv mezi klientem a správcem front; viz [“Ověření kanálu telemetrie pomocí zabezpečení SSL”](#) na stránce 128.
3. Šifrování zpráv mezi klientem a správcem front; viz [“Soukromí zveřejňování kanálů telemetrie”](#) na stránce 130.

Konfigurační parametry JSSE

Upravte parametry JSSE, abyste změnili způsob, jakým je nakonfigurováno připojení SSL. Konfigurační parametry JSSE jsou uspořádány do tří sad:

1. [Kanál telemetrie IBM WebSphere MQ](#)
2. [Klient MQTT Java](#)
3. [JRE](#)

Konfigurujte parametry kanálu telemetrie pomocí Průzkumníka IBM WebSphere MQ. Nastavte parametry klienta protokolu MQTT Java v atributu `MqttConnectionOptions.SSLProperties`.

Upravte parametry zabezpečení prostředí JRE úpravou souborů v adresáři zabezpečení prostředí JRE jak na klientu, tak na serveru.

IBM WebSphere MQ kanál telemetrie

Nastavte všechny parametry zabezpečení SSL kanálu telemetrie pomocí Průzkumníka produktu WebSphere MQ .

ChannelName

ChannelName je povinný parametr u všech kanálů.

Název kanálu identifikuje kanál přidružený k určitému číslu portu. Kanály názvů, které vám pomohou spravovat sady klientů MQTT.

PortNumber

PortNumber je volitelný parametr na všech kanálech. Výchozí hodnota je 1883 pro kanály TCP a 8883 pro kanály SSL.

Číslo portu TCP/IP přidružené k tomuto kanálu. Klienti MQTT jsou připojeni k kanálu zadáním portu definovaného pro kanál. Má-li kanál vlastnosti SSL, musí se klient připojit pomocí protokolu SSL, například:

```
MQTTClient mqttClient = new MqttClient( "ssl://www.example.org:8884", "clientId");
mqttClient.connect();
```

KeyFile

Volba KeyFileName je požadovaným parametrem pro kanály SSL. Musí být vynechán pro kanály TCP.

KeyFileName je cesta k úložišti klíčů Java obsahující digitální certifikáty, které poskytujete. Použijte JKS, JCEKS nebo PKCS12 jako typ úložiště klíčů na serveru.

Identifikujte typ úložiště klíčů pomocí jednoho z následujících přípon souborů:

- .jks
- .jceks
- .p12
- .pkcs12

Předpokládá se, že úložiště klíčů JKS je úložištěm klíčů s jinou příponou souboru.

Na serveru můžete kombinovat jeden typ úložiště klíčů s jinými typy úložiště klíčů na straně klienta.

Uložte soukromý certifikát serveru do úložiště klíčů. Certifikát je znám jako certifikát serveru. Certifikát může být podepsán sám sebou nebo může být součástí řetězu certifikátů, který je podepsán podpisovým orgánem.

Používáte-li řetěz certifikátů, umístěte přidružené certifikáty do úložiště klíčů serveru.

Certifikát serveru a všechny certifikáty ve svém řetězu certifikátů jsou odeslány klientům za účelem ověření identity serveru.

Pokud jste nastavili ClientAuth na Required, úložiště klíčů musí obsahovat všechny certifikáty nezbytné k ověření klienta. Klient odešle certifikát podepsaný svým držitelem nebo řetěz certifikátů a klient je autentizován prvním ověřením tohoto materiálu proti certifikátu v úložišti klíčů. Pomocí řetězu certifikátů může jeden certifikát ověřit mnoho klientů, i když je vydáván s různými klientskými certifikáty.

PassPhrase

PassPhrase je povinný parametr pro kanály SSL. Musí být vynechán pro kanály TCP.

Přístupová fráze se používá k ochraně úložiště klíčů.

ClientAuth

Parametr `ClientAuth` je volitelným parametrem zabezpečení SSL. Výchozí hodnota je bez ověření klienta. Musí být vynechán pro kanály TCP.

Nastavte volbu `ClientAuth`, pokud chcete, aby služba telemetrie (MQXR) ověřila klienta, před tím, než povolíte klientovi připojení k telemetrickým kanálům.

Nastavíte-li `ClientAuth`, klient se musí připojit k serveru pomocí SSL a ověřit server. V odpovědi na nastavení `ClientAuth` odešle klient svůj digitální certifikát na server a všechny ostatní certifikáty ve svém úložišti klíčů. Jeho digitální certifikát je známý jako certifikát klienta. Tyto certifikáty jsou ověřovány proti certifikátům uchovávaným v úložišti klíčů kanálu a v úložišti prostředí `JRE cacerts`.

CipherSuite

Hodnota `CipherSuite` je volitelným parametrem zabezpečení SSL. Výchozí nastavení je vyzkoušet všechny povolené specifikace `CipherSpecs`. Musí být vynechán pro kanály TCP.

Chcete-li použít konkrétní `CipherSpec`, nastavte `CipherSuite` na název `CipherSpec`, který musí být použit k ustanovení připojení SSL.

Služba telemetrie a klient MQTT vyjednáva obecnou specifikaci `CipherSpec` ze všech specifikací `CipherSpecs`, které jsou povoleny na každém konci. Je-li na obou koncích připojení zadána specifická hodnota `CipherSpec`, musí se shodovat s položkou `CipherSpec` na druhém konci.

Nainstalujte další šifry přidáním dalších poskytovatelů do JSSE.

Federální standardy zpracování informací (FIPS)

Standard FIPS je volitelné nastavení. Ve výchozím nastavení není nastavena.

Buď na panelu vlastností správce front, nebo pomocí obslužného programu `runmqsc` nastavte `SSLFIPS`. Volba `SSLFIPS` určuje, zda mají být použity pouze algoritmy s certifikací FIPS.

Seznam názvů revokace

Seznam názvů revoze je volitelné nastavení. Ve výchozím nastavení není nastavena.

Buď na panelu vlastností správce front, nebo pomocí obslužného programu `runmqsc` nastavte `SSLCRLNL`. `SSLCRLNL` uvádí seznam názvů objektů ověřovacích informací, které se používají k poskytnutí umístění odvolaných certifikátů.

Nejsou použity žádné další parametry správce front, které nastavují vlastnosti zabezpečení SSL.

Klient MQTT Java

Nastavte vlastnosti SSL pro klienta Java v souboru `MqttConnectionOptions.SSLProperties`; například:

```
java.util.Properties sslClientProperties = new Properties();
sslClientProperties.setProperty("com.ibm.ssl.keyStoreType", "JKS");
com.ibm.micro.client.mqttv3.MqttConnectOptions conOptions = new MqttConnectOptions();
conOptions.setSSLProperties(sslClientProperties);
```

Názvy a hodnoty specifických vlastností jsou popsány ve třídě `MqttConnectOptions`. Pro odkazy na dokumentaci ke klientským rozhraním API pro knihovny klienta MQTT viz [Odkaz na programování klienta MQTT](#).

Protokol

Protokol je volitelný.

Protokol je vybrán při vyjednávání s telemetrickým serverem. Požadujete-li určitý protokol, můžete jej vybrat. Pokud telemetrický server nepodporuje protokol, připojení selže.

ContextProvider

`ContextProvider` je volitelný.

KeyStore

KeyStore je volitelné. Nakonfigurujte ji, pokud je na serveru nastaveno ClientAuth , aby se vynutila autentizace klienta.

Umístěte digitální certifikát klienta, podepsaný pomocí svého soukromého klíče do úložiště klíčů. Zadejte cestu k úložišti klíčů a heslo. Typ a poskytovatel jsou volitelné. JKS je výchozí typ a IBMJCE je výchozí poskytovatel.

Určete jiného poskytovatele úložiště klíčů, který bude odkazovat na třídu, která přidává nového poskytovatele úložiště klíčů. Chcete-li vytvořit instanci produktu KeyManagerFactory , nastavte název algoritmu používaného poskytovatelem úložiště klíčů tak, že nastavíte název správce klíčů.

TrustStore

Parametr TrustStore je volitelný. Všechny certifikáty, které důvěřujete, můžete umístit do úložiště JRE cacerts .

Konfigurujte úložiště údajů o důvěryhodnosti, chcete-li mít pro klienta jiné úložiště údajů o důvěryhodnosti. Úložiště údajů o důvěryhodnosti byste neměli konfigurovat, pokud server používá certifikát vydaný dobře známou CA, která již má svůj kořenový certifikát uložený v produktu cacerts.

Přidejte veřejně podepsaný certifikát serveru nebo kořenového certifikátu do úložiště údajů o důvěryhodnosti a uveďte cestu k úložišti údajů o důvěryhodnosti a heslo. JKS je výchozí typ a IBMJCE je výchozí poskytovatel.

Uveďte jiného poskytovatele úložiště údajů o důvěryhodnosti, který bude odkazovat na třídu, která přidává nového poskytovatele úložiště údajů o důvěryhodnosti. Předejte název algoritmu používaného poskytovatelem úložiště údajů o důvěryhodnosti pro vytvoření instance správce TrustManagerFactory nastavením názvu správce důvěryhodnosti.

JRE

Další aspekty zabezpečení produktu Java , které mají vliv na chování zabezpečení SSL na straně klienta i serveru, jsou v prostředí JRE konfigurovány. Konfigurační soubory na Windows jsou v *Java Installation Directory\jre\lib\security*. Pokud používáte prostředí JRE dodané s produktem IBM WebSphere MQ , cesta je uvedena v následující tabulce:

Platforma	Cesta k souboru
Windows	<i>WMQ Installation Directory\java\jre\lib\security</i>
Ostatní platformy UNIX and Linux	<i>WMQ Installation Directory/java/jre64/jre/lib/security</i>

Dobře známá certifikační autority

Soubor cacerts obsahuje kořenové certifikáty známých certifikačních autorit. cacerts se standardně použije, pokud neuvédete úložiště údajů o důvěryhodnosti. Pokud používáte úložiště cacerts nebo neposkytujete úložiště údajů o důvěryhodnosti, musíte přezkoumat a upravit seznam podepisujících subjektů v produktu cacerts tak, aby splňoval vaše požadavky na zabezpečení.

Produkt cacerts můžete otevřít pomocí příkazu WebSphere MQ `stirmqikm`, který spustí obslužný program správy klíčů IBM . Otevřete soubor cacerts jako soubor JKS s použitím hesla `changeit`. Upravte heslo tak, aby byl soubor zabezpečen.

Konfigurace tříd zabezpečení

Použijte soubor `java.security` k registraci dalších poskytovatelů zabezpečení a dalších výchozích vlastností zabezpečení.

Oprávnění

Použijte soubor `java.policy` k úpravě oprávnění udělených prostředkům. `javaws.policy` uděluje oprávnění pro `javaws.jar`

Odolnost šifrování

Některá prostředí JRE se dodávají se sníženým šifrováním odolnosti. Pokud nemůžete importovat klíče do úložišť klíčů, může být příčinou snížení odolnosti šifrování. Bud' zkuste spustit příkaz **ikeyman** pomocí příkazu **strmqikm**, nebo stáhněte silné, ale omezené soubory jurisdikcí z [IBM Developer Kit, Security information](#).

Důležité: Vaše země původu by mohla mít omezení týkající se dovozu, držení, užívání nebo přeexportování do jiné země, šifrovacího softwaru. Před stažením nebo používáním neomezených souborů zásad je třeba zkontrolovat zákony vaší země. Zkontrolujte jeho předpisy a zásady týkající se dovozu, držení, užívání a zpětného exportu šifrovacího softwaru za účelem zjištění, zda je tento šifrovací software povolen.

Upravte poskytovatele důvěryhodnosti tak, aby se klient mohl připojit k libovolnému serveru.

Následující příklad ilustruje, jak přidat poskytovatele důvěryhodnosti a odkázat jej z kódu klienta MQTT. Tento příklad nezajišťuje žádné ověření klienta nebo serveru. Výsledné připojení SSL je šifrováno, aniž bylo ověřováno.

Úsek kódu v produktu [Obrázek 25](#) na stránce 135 nastavuje poskytovatele důvěry `AcceptAllProviders` a správce důvěryhodnosti pro klienta MQTT.

```
java.security.Security.addProvider(new AcceptAllProvider());
java.util.Properties sslClientProperties = new Properties();
sslClientProperties.setProperty("com.ibm.ssl.trustManager", "TrustAllCertificates");
sslClientProperties.setProperty("com.ibm.ssl.trustStoreProvider", "AcceptAllProvider");
conOptions.setSSLProperties(sslClientProperties);
```

Obrázek 25. Úsek kódu klienta MQTT

```
package com.ibm.mq.id;
public class AcceptAllProvider extends java.security.Provider {
    private static final long serialVersionUID = 1L;
    public AcceptAllProvider() {
        super("AcceptAllProvider", 1.0, "Trust all X509 certificates");
        put("TrustManagerFactory.TrustAllCertificates",
            AcceptAllTrustManagerFactory.class.getName());
    }
}
```

Obrázek 26. `AcceptAllProvider.java`

```
protected static class AcceptAllTrustManagerFactory extends
    javax.net.ssl.TrustManagerFactorySpi {
    public AcceptAllTrustManagerFactory() {}
    protected void engineInit(java.security.KeyStore keystore) {}
    protected void engineInit(
        javax.net.ssl.ManagerFactoryParameters parameters) {}
    protected javax.net.ssl.TrustManager[] engineGetTrustManagers() {
        return new javax.net.ssl.TrustManager[] { new AcceptAllX509TrustManager() };
    }
}
```

Obrázek 27. `AcceptAllTrustManagerFactory.java`

```

protected static class AcceptAllX509TrustManager implements
    javax.net.ssl.X509TrustManager {
    public void checkClientTrusted(
        java.security.cert.X509Certificate[] certificateChain,
        String authType) throws java.security.cert.CertificateException {
        report("Client authtype=" + authType);
        for (java.security.cert.X509Certificate certificate : certificateChain) {
            report("Accepting:" + certificate);
        }
    }
    public void checkServerTrusted(
        java.security.cert.X509Certificate[] certificateChain,
        String authType) throws java.security.cert.CertificateException {
        report("Server authtype=" + authType);
        for (java.security.cert.X509Certificate certificate : certificateChain) {
            report("Accepting:" + certificate);
        }
    }
    public java.security.cert.X509Certificate[] getAcceptedIssuers() {
        return new java.security.cert.X509Certificate[0];
    }
    private static void report(String string) {
        System.out.println(string);
    }
}

```

Obrázek 28. AcceptAllX509TrustManager.java

Konfigurace kanálu JAAS kanálu telemetrie

Nakonfigurujte službu JAAS tak, aby ověřoval identitu Jméno uživatele odeslané klientem.

Administrátor produktu WebSphere MQ konfiguruje, které kanály MQTT vyžadují ověření klienta pomocí služby JAAS. Určete název konfigurace JAAS pro každý kanál, který má provést ověření JAAS . Kanály mohou používat stejnou konfiguraci JAAS nebo mohou používat různé konfigurace služby JAAS . Konfigurace jsou definovány v produktu *WMQData directory\qmgrs\qMgrName\mqxr\jaas.config*.

Soubor *jaas.config* je uspořádán podle názvu konfigurace JAAS . Pod každým názvem konfigurace je uveden seznam konfigurací přihlášení, viz [Obrázek 29 na stránce 137](#).

JAAS poskytuje čtyři standardní přihlašovací moduly. Standardní přihlašovací moduly NT a UNIX mají omezenou hodnotu.

Modul JndiLogin

Ověřuje se proti adresářové službě konfigurované pod rozhráním JNDI (Java Naming and Directory Interface).

Krb5LoginModule

Ověřuje použití protokolů Kerberos .

NTLoginModule

Provádí ověření pomocí informací o zabezpečení NT pro aktuálního uživatele.

Modul UnixLogin

Provádí ověření pomocí informací o zabezpečení systému UNIX pro aktuálního uživatele.

Problém s používáním produktu NTLoginModule nebo UnixLoginModule je, že služba telemetrie (MQXR) je spuštěna s identitou mqm , nikoli s identitou kanálu MQTT. mqm je identita předávaná produktu NTLoginModule nebo UnixLoginModule pro ověřování, nikoli identita klienta.

Chcete-li tento problém odstranit, napište svůj vlastní přihlašovací modul nebo použijte jiné standardní přihlašovací moduly. Ukázka JAASLoginModule .java je dodávána spolu s produktem WebSphere MQ Telemetry. Jedná se o implementaci rozhraní produktu *javax.security.auth.spi.LoginModule* . Použijte jej k vytvoření vlastní metody ověření.

Všechny nové třídy LoginModule , které zadáte, musí být na cestě ke třídě služby telemetrie (MQXR). Neumísťujte své třídy do adresářů produktu WebSphere MQ , které jsou v cestě ke třídám. Vytvořte své vlastní adresáře a definujte celou cestu ke třídám pro službu telemetrie (MQXR).

Cestu ke třídě používanou službou telemetrie (MQXR) můžete rozšířit nastavením cesty ke třídám v souboru `service.env`. `CLASSPATH` musí být velkými písmeny, a příkaz cesty ke třídě může obsahovat pouze literály. Proměnné v proměnné `CLASSPATH` nelze použít; například `CLASSPATH=%CLASSPATH%` je chybná. Služba telemetrie (MQXR) nastavuje vlastní cestu ke třídám. Hodnota `CLASSPATH` definovaná v produktu `service.env` je přidána k ní.

Služba telemetrie (MQXR) poskytuje dvě zpětná volání, která vrací hodnoty `Jméno uživatele` a `Heslo` pro klienta připojeného k kanálu MQTT. Hodnoty `Jméno uživatele` a `Heslo` jsou nastaveny v objektu `MqttConnectOptions`. Příklad přístupu k `Username` a `Password` najdete v tématu [Obrázek 30 na stránce 137](#).

Příklady

Příklad konfiguračního souboru JAAS s jednou pojmenovanou konfigurací, `MQXRConfig`.

```
MQXRConfig {
  samples.JAASLoginModule required debug=true;
  //com.ibm.security.auth.module.NTLoginModule required;
  //com.ibm.security.auth.module.Krb5LoginModule required
  //      principal=principal@your_realm
  //      useDefaultCcache=TRUE
  //      renewTGT=true;
  //com.sun.security.auth.module.NTLoginModule required;
  //com.sun.security.auth.module.UnixLoginModule required;
  //com.sun.security.auth.module.Krb5LoginModule required
  //      useTicketCache="true"
  //      ticketCache="$ {user.home} ${}/tickets";
};
```

Obrázek 29. Ukázkový soubor `jaas.config`

Příklad modulu přihlášení JAAS kódovaného tak, aby přijímal hodnoty `Jméno uživatele` a `Heslo` poskytnuté klientem MQTT.

```
public boolean login()
  throws javax.security.auth.login.LoginException {
  javax.security.auth.callback.Callback[] callbacks =
    new javax.security.auth.callback.Callback[2];
  callbacks[0] = new javax.security.auth.callback.NameCallback("NameCallback");
  callbacks[1] = new javax.security.auth.callback.PasswordCallback(
    "PasswordCallback", false);
  try {
    callbackHandler.handle(callbacks);
    String username = ((javax.security.auth.callback.NameCallback) callbacks[0])
      .getName();
    char[] password = ((javax.security.auth.callback.PasswordCallback) callbacks[1])
      .getPassword();
    // Accept everything.
    if (true) {
      loggedIn = true;
    } else
      throw new javax.security.auth.login.FailedLoginException("Login failed");

    principal= new JAASPrincipal(username);

  } catch (java.io.IOException exception) {
    throw new javax.security.auth.login.LoginException(exception.toString());
  } catch (javax.security.auth.callback.UnsupportedCallbackException exception) {
    throw new javax.security.auth.login.LoginException(exception.toString());
  }
  return loggedIn;
}
```

Obrázek 30. Ukázková metoda `JAASLoginModule.Login()`

Démon IBM WebSphere MQ Telemetry pro koncepce zařízení

Démon IBM WebSphere MQ Telemetry pro zařízení je rozšířenou aplikací klienta MQTT V3 . Použijte jej k ukládání a předávání zpráv z jiných klientů MQTT. Připojuje se k produktu IBM WebSphere MQ jako ke klientovi MQTT, ale k němu se můžete připojit i k dalším klientům MQTT.

Démon je zprostředkovatel publikování a odběru. Klienti MQTT V3 se k němu připojují, aby mohli publikovat a odebírat témata, pomocí řetězců témat pro publikování a filtry témat k odběru. Řetězec tématu je hierarchický, s úrovněmi témat děleno /. Filtry témat jsou řetězce témat, které mohou obsahovat zástupné znaky pro jednu úroveň + a zástupný znak více úrovní # jako poslední část řetězce tématu.

Poznámka: Zástupné znaky v démonu se řídí přísnějšími pravidly produktu WebSphere Message Broker, v6. IBM WebSphere MQ se liší. Podporuje víceúrovňové zástupné znaky více úrovní; zástupné znaky mohou stát v libovolném počtu úrovní hierarchie, kdekoli v řetězci tématu.

K démonu se připojuje více klientů MQTT v3 pomocí portu modulu listener. Výchozí port modulu listener je upravitelný. Můžete definovat více portů modulu listener a přidělit jim různé obory názvů, viz [“Démon WebSphere MQ Telemetry pro porty modulu listener zařízení”](#) na stránce 145. Démon je sám o sobě klientem MQTT v3 . Nakonfigurujte připojení mostu démona k připojení démona k portu modulu listener jiného démona nebo ke službě WebSphere MQ Telemetry (MQXR).

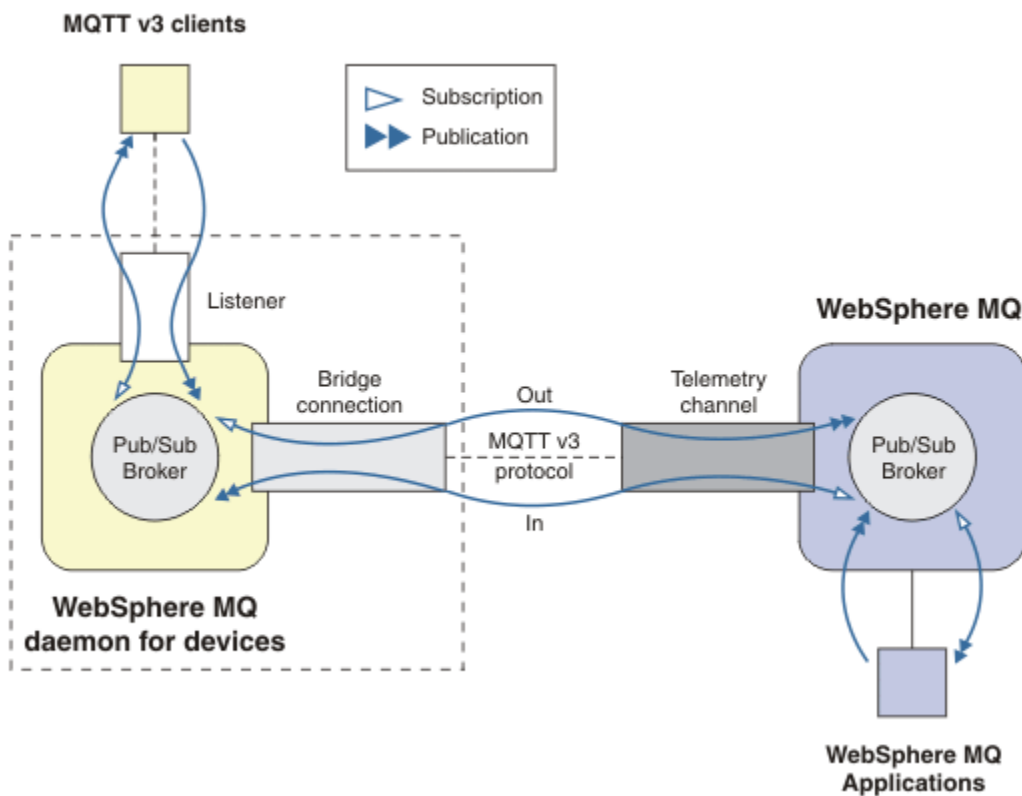
Pro zařízení lze konfigurovat více mostů pro démona WebSphere MQ Telemetry. Mosty slouží k připojení sítě démonů, které si mohou vyměňovat publikace.

Každý most může publikovat a odebírat témata na svém lokálním démonu. Může také publikovat a odebírat témata z jiného démona, zprostředkovatele publikování a odběru WebSphere MQ nebo jakéhokoli jiného zprostředkovatele MQTT v3 , ke kterému je připojen. Pomocí filtru témat můžete vybrat publikace k šíření z jednoho zprostředkovatele do jiného. Publikace lze šířit v obou směrech. Můžete propagovat veřejné služby z lokálního démona do každého z připojených vzdálených zprostředkovatelů nebo z libovolného z připojených zprostředkovatelů na místní démona; viz [“Démon IBM WebSphere MQ Telemetry pro zařízení mostů”](#) na stránce 138.

Démon IBM WebSphere MQ Telemetry pro zařízení mostů

Most démona IBM WebSphere MQ Telemetry pro zařízení spojuje dva zprostředkovatele publikování/ odběru s použitím protokolu MQTT v3 . Most šíří publikace od jednoho zprostředkovatele do druhého, a to v obou směrech. Na jednom konci je démon WebSphere MQ Telemetry pro připojení mostu zařízení a v druhém případě může být správce front nebo jiný démon. Správce front je připojen k připojení mostu pomocí kanálu telemetrie. Démon je připojen k mostu pomocí modulu listener démona.

Démon IBM WebSphere MQ Telemetry pro zařízení podporuje jedno nebo více souběžných připojení k jiným zprostředkovatelům. Připojení z démona se nazývají mosty a jsou definována položkami připojení v konfiguračním souboru démona. Připojení k produktu IBM WebSphere MQ jsou prováděna pomocí kanálů telemetrie IBM WebSphere MQ , jak je znázorněno na následujícím obrázku:



Obrázek 31. Připojování IBM WebSphere MQ Telemetry daemon for devices k IBM WebSphere MQ

Most připojuje démona k jinému zprostředkovateli jako klient MQTT v3 . Parametry mostu zrcadlí atributy klienta MQTT v3 .

Most je více než připojení. Působí jako agent publikování a odběru umístěný mezi dvěma zprostředkovateli publikování/odběru. Lokální zprostředkovatel je démon IBM WebSphere MQ Telemetry pro zařízení a vzdálený zprostředkovatel je libovolný zprostředkovatel publikování/odběru, který podporuje protokol MQTT v3 . Vzdáleného zprostředkovatele je obvykle jiný démon nebo IBM WebSphere MQ.

Úloha mostu je pro šíření publikování mezi dvěma zprostředkovateli. Most je obousměrný. Šíří publikace v obou směrech. [Obrázek 31 na stránce 139](#) ilustruje způsob, jakým most připojuje démona IBM WebSphere MQ Telemetry pro zařízení k produktu IBM WebSphere MQ. Produkt [“Příklad nastavení tématu pro most”](#) na stránce 140 používá příklady pro ilustraci způsobu použití parametru tématu ke konfiguraci mostu.

Šipky In a Out v [Obrázek 31 na stránce 139](#) označují obousměrný most mostu. Na jednom konci šipky se vytvoří odběr. Publikace, které odpovídají odběru, jsou publikovány do zprostředkovatele na opačném konci šipky. Šipka je označena v závislosti na toku publikací. Publications flow In to the daemon and Out from the daemon. Význam popisků je použit v syntaxi příkazu. Nezapomeňte, že hodnoty In a Out odkazují na tok publikování, a nikoli na místo, kam se má odběr odeslat.

Ostatní klienti, aplikace nebo zprostředkovatelé mohou být připojeni buď k produktu IBM WebSphere MQ , nebo k démonu WebSphere MQ Telemetry pro zařízení. Publikují se a přihlašují se k tématům na zprostředkovateli, ke kterým jsou připojeni. Je-li zprostředkovatel IBM WebSphere MQ, témata mohou být klastrovaná nebo distribuovaná a nejsou explicitně definována v lokálním správci front.

Použití mostů

Spojte démony společně pomocí připojení mostu a modulů listener. Spojte démony a správce front dohromady pomocí připojení mostu a kanálů telemetrie. Když spojíte více zprostředkovatelů

dohromady, je možné vytvářet smyčky. Buďte opatrní: Publikování by mohly cirkulovat kolem cyklu brokerů, nezjištěné.

Některé z důvodů používání démonů přemostěných do produktu IBM WebSphere MQ jsou následující:

Snižte počet připojení klienta protokolu MQTT k produktu WebSphere MQ .

Pomocí hierarchie démonů můžete připojit mnoho klientů k produktu WebSphere MQ; více klientů než kolik se může připojit k jednomu správci frontou současně.

Uložit a předat zprávy mezi klienty MQTT a produktem WebSphere MQ

Mezi klienty a produktem IBM WebSphere MQ můžete používat ukládání a přesměrování, pokud klienti nemají své vlastní úložiště. Můžete použít více typů připojení mezi klientem MQTT a produktem WebSphere MQ; viz téma [Koncepty a scénáře telemetrie k monitorování a řízení](#).

Filtrování publikací vyměněných mezi klienty MQTT a produktem WebSphere MQ

Publikace se dělí pouze na zprávy, které se zpracovávají lokálně, a zprávy, které zahrnují jiné aplikace. Lokální publikace mohou zahrnovat řídicí toky mezi senzory a regulátory a vzdálené publikace zahrnují požadavky na odečty, stav a konfigurační příkazy.

Změna prostorů témat publikací

Vyvarujte se řetězcům témat z klientů připojených k různým portům modulu listener z kolidujících s jinými porty modulu listener. Tento příklad používá démona k označení odečtů měřidel pocházejících z různých budov; viz [Oddělení prostorů témat různých skupin klientů](#).

Příklad nastavení tématu pro most

Publikovat vše do vzdáleného zprostředkovatele-použití výchozích hodnot

Výchozí směr se nazývá outa most publikuje témata do vzdáleného zprostředkovatele. Parametr topic určuje, která témata mají být šířena s použitím filtrů témat.

Most používá parametr topic v produktu [Obrázek 32 na stránce 140](#) k přihlášení k odběru všeho, který je publikován na lokální démon klientem MQTT nebo jinými zprostředkovateli. Most publikuje témata ke vzdálenému zprostředkovateli, který je připojen k mostu.

```
connection Daemon1
topic #
```

Obrázek 32. Publikovat vše ve vzdáleném zprostředkovateli

Publikovat vše do vzdáleného zprostředkovatele-explicitní

Nastavení topic v následujícím fragmentu kódu poskytuje stejný výsledek jako použití výchozích nastavení. Jediný rozdíl je, že parametr **direction** je explicitní. Pomocí směru out se přihlaste k odběru lokálního zprostředkovatele, démona a publikujte na vzdáleném zprostředkovateli. Publikace vytvořené na lokálním démonu, k jejichž odběru je tento most odebírán, jsou publikovány na vzdáleném zprostředkovateli.

```
connection Daemon1
topic # out
```

Obrázek 33. Publikovat vše do vzdáleného zprostředkovatele-explicitní

Publikovat vše pro lokální zprostředkovatele

Místo použití směru out můžete nastavit opačný směr, in. Následující fragment kódu konfiguruje most k přihlášení k odběru všeho publikovaného ve vzdáleném zprostředkovateli, který je připojen k mostu. Most publikuje témata do lokálního zprostředkovatele, démona.

```
connection Daemon1
topic # in
```

Obrázek 34. Publikovat vše pro lokální zprostředkovatele

Publikovat vše z tématu exportu na lokálním zprostředkovateli do tématu importu na vzdáleném zprostředkovateli

Pomocí dvou dalších parametrů témat, **local_prefix** a **remote_prefix**, upravte filtr témat, # v předchozích příkladech. Jeden parametr se používá k úpravě filtru témat použitého v odběru a druhý parametr se používá k úpravě tématu, na které je publikování publikováno. Výsledkem je nahrazení začátku řetězce tématu použitého v jednom zprostředkovateli jiným řetězcem tématu na druhém zprostředkovateli.

V závislosti na směru příkazu tématu se význam hodnot **local_prefix** a **remote_prefix** vrací. Je-li směr out, použije se standardní hodnota **local_prefix** jako část odběru tématu a **remote_prefix** nahradí část **local_prefix** řetězce tématu ve vzdálené publikaci. Je-li směr in, stane se **remote_prefix** součástí vzdáleného odběru a **local_prefix** nahradí část **remote_prefix** řetězce tématu.

První část řetězce tématu se často považuje za definici prostoru tématu. Použijte další parametry ke změně prostoru tématu, na který je téma publikováno. Toto můžete provést, chcete-li se vyhnout kolizím tématu s jiným tématem na cílovém zprostředkovateli nebo odebrat řetězec tématu bodu připojení.

Jako příklad, v následujícím fragmentu kódu jsou všechny publikace k řetězci tématu export/# v démonu znovu publikovány na import/# ve vzdáleném zprostředkovateli.

```
topic # out export/ import/
```

Obrázek 35. Publikovat vše z tématu exportu na lokálním zprostředkovateli do tématu importu na vzdáleném zprostředkovateli

Publikovat vše do tématu importu v lokálním zprostředkovateli z tématu exportu ve vzdáleném zprostředkovateli

Následující fragment kódu ukazuje obrácenou konfiguraci. Most se přihlásí k odběru všeho publikovaného pomocí řetězce tématu produktu export/# na vzdáleném zprostředkovateli a publikuje jej na serveru import/# na lokálním zprostředkovateli.

```
connection Daemon1
topic # in import/ export/
```

Obrázek 36. Publikovat vše do tématu importu v lokálním zprostředkovateli z tématu exportu ve vzdáleném zprostředkovateli

Publikovat vše z bodu připojení produktu 1884/ do vzdáleného zprostředkovatele s původními řetězci témat

V následujícím fragmentu kódu se most přihlásí k odběru všeho publikovaného klienty připojenými k bodu připojení 1884/ na lokálním démonu. Most publikuje vše publikované na místo připojení ke vzdálenému zprostředkovateli. Řetězec bodu připojení 1884/ je odebrán z témat publikovaných na vzdáleném zprostředkovateli. Hodnota *local_prefix* je stejná jako řetězec bodu připojení 1884/ a *vzdálená_předpona* je prázdný řetězec.

```
listener 1884
mount_point 1884/
connection Daemon1
topic # out 1884/ ""
```

Obrázek 37. Publikujte vše z bodu připojení serveru 1884/ do vzdáleného zprostředkovatele s původními řetězci témat.

Oddělování prostorů témat různých klientů připojených k různým démonům

Aplikace je napsána pro elektrická měřidla, aby bylo možné publikovat odečty měřidla pro budovu. Odečty jsou publikovány s použitím klientů MQTT k hostiteli hostovanému ve stejné budově. Téma vybrané pro publikace je power. Stejná aplikace je implementována na celou řadu budov v komplexu. Pro monitorování lokalit a datové úložiště jsou odečty ze všech budov agregovány pomocí připojení mostu. Připojení propojují démony budovy s produktem WebSphere MQ v centrálním umístění.

Klientské aplikace v každé budově jsou identické, ale data musí být rozlišena podle budovy. Každé čtení má téma power a musí mít před sebou předponu budovy, aby ji bylo možné rozlišit. Most z první budovy v komplexu používá předponu meters/building01/, od sestavení dvou předpon je meters/building02/. Odečty z ostatních budov sledují stejný vzorek. Produkt WebSphere MQ přijímá odečty s tématy jako meters/building01/power.

Příklad je ve stavu contrived; v praxi bude prostor tématu, který aplikace publikuje, pravděpodobně konfigurovatelný.

Konfigurační soubor pro každý démon má příkaz topic, který následuje za vzorem v následujícím fragmentu kódu:

```
connection Daemon1
topic power out "" meters/building01/
```

Obrázek 38. Oddělit prostory témat klientů připojených k různým démonům

Zadejte prázdný řetězec jako zástupný symbol pro nepoužívaný parametr local_prefix.

Oddělit prostory témat klientů připojených ke stejnému démonu

Předpokládejme, že k připojení všech měřičů napájení se používá jeden démon. Za předpokladu, že v aplikaci lze nakonfigurovat připojení k různým portům, můžete tyto budovy rozlišit připojením měřidel z různých budov k různým portům modulu listener, jako v následujícím fragmentu kódu. Opět se jedná o příklad, který je možné použít; ilustruje, jak lze použít body připojení.

```
listener 1884
mount_point meters/building01/
listener 1885
mount_point meters/building02/
connection Daemon1
topic meters/+power out
```

Obrázek 39. Oddělit prostory témat klientů připojených ke stejnému démonu

Přemapování různých témat pro publikace proudící v obou směrech

V konfiguraci v následujícím fragmentu kódu se most přihlásí k odběru jednoho tématu b ze vzdáleného zprostředkovatele a předává publikace o produktu b lokálnímu démonu, čímž se změní téma na a. Most se také přihlašuje k odběru jediného tématu x na lokálním zprostředkovateli a předává publikace o produktu x vzdáleným zprostředkovatelem, přičemž se změní téma na y.

```
connection Daemon1
topic "" in a b
topic "" out x y
```

Obrázek 40. Přemapování různých témat pro publikace proudící v obou směrech

Důležitým bodem v tomto příkladu je to, že různá témata jsou přihlášena k odběru a publikována v obou zprostředkovatelů. Prostory témat u obou zprostředkovatelů jsou disjunktní.

Přemapování stejných témat u publikací tekoucích v obou směrech (cyklus)

Na rozdíl od předchozího příkladu má konfigurace v produktu [Obrázek 41 na stránce 143](#) obecně výsledky ve smyčce. V příkazu tématu `topic "" in a b` se most přihlásí k odběru b vzdáleně a publikuje se lokálně do produktu a. V druhém příkazu tématu se most přihlásí k odběru lokálně z produktu a a publikuje se do produktu b vzdáleně. Stejná konfigurace může být napsána tak, jak je zobrazeno v [Obrázek 42 na stránce 143](#).

Obecný výsledek je takový, že pokud se klient publikuje do b vzdáleně, publikování se přeneso na lokální démona jako publikování na téma a. Avšak při publikování z mostu do lokálního démona na téma a se publikování shoduje s odběrem, který byl proveden z mostu do lokálního tématu a. Odběr je `topic "" out a b`. V důsledku toho je publikování převedeno zpět na vzdálený zprostředkovatel jako publikování na téma b. Most je nyní přihlášen k odběru vzdáleného tématu ba cyklus začne znovu.

Někteří zprostředkovatelé implementují detekci smyčky, aby se zabránilo tomu, že se smyčka děje. Ale mechanismus detekce smyčky musí fungovat, když jsou k sobě propojeny různé typy zprostředkovatelů. Detekce smyčky nebude fungovat, pokud je produkt WebSphere MQ přemostěný do démona WebSphere MQ Telemetry pro zařízení. Je funkční, pokud je dva démon IBM WebSphere MQ Telemetry pro zařízení propojen mostem dohromady. Ve výchozím nastavení je detekce smyčky zapnutá; viz [try_private](#).

```
connection Daemon1
topic "" in a b
topic "" out a b
```

Obrázek 41. !Přemapování stejných témat u publikací tekoucích v obou směrech

```
connection Daemon1
topic "" both a b
```

Obrázek 42. !Přemapujte stejná témata pro publikace, které tečou v obou směrech, pomocí produktu both.

Konfigurace v produktu [Obrázek 40 na stránce 143](#) je stejná jako [Obrázek 41 na stránce 143](#).

Dostupnost připojení mostu IBM WebSphere MQ Telemetry daemon for devices

Nakonfigurovat více adres připojení mostu IBM WebSphere MQ Telemetry daemon for devices pro připojení k prvnímu dostupnému vzdálenému zprostředkovateli. Je-li zprostředkovatelem správce front s více instancemi, zadejte obě jeho adresy TCP/IP. Nakonfigurujte primární připojení pro připojení nebo opětovné připojení k primárnímu serveru, je-li k dispozici.

Parametr mostu připojení `addresses` je seznam adres soketů TCP/IP. Most se pokusí o připojení k každé adrese postupně, dokud se nestane úspěšným připojením. Parametry připojení `round_robin` a `start_type` řídí, jak se adresy použijí, jakmile se provede úspěšné připojení.

Má-li parametr `start_type` hodnotu `auto`, `manual` nebo `lazy`, pokus o připojení se nezdaří, pokud se připojení nezdaří, a to se pokusí o nové připojení. Každou adresu používá postupně, s přibližně 20 sekundovou prodlevou při každém pokusu o připojení. Má-li parametr `start_type` hodnotu `once`, nedojde k automatickému pokusu o nové připojení, pokud se připojení nezdaří.

Pokud má parametr `round_robin` hodnotu `true`, pokusy o připojení mostu začínají na první adrese v seznamu a v seznamu se pokusí o každou adresu v seznamu. Je-li seznam vyčerpán, spustí se znovu na první adrese. Je-li v seznamu pouze jedna adresa, zkusí ji znovu každých 20 sekund.

Má-li parametr `round_robin` hodnotu `false`, je dána přednost první adrese v seznamu, která se nazývá primární server. Pokud se první pokus o připojení k primárnímu serveru nezdaří, bude most pokračovat v pokusu o nové připojení k primárnímu serveru na pozadí. Současně se most pokouší připojit pomocí jiných adres v seznamu. Když se pokusy o připojení k primárnímu serveru pokusí o úspěch, odpojí se most od aktuálního připojení a přepne se na připojení k primárnímu serveru.

Pokud je připojení odpojeno dobrovolně, například zadáním příkazu **`connection_stop`**, pak se znovu pokusí o použití stejné adresy znovu. Je-li připojení odpojeno z důvodu selhání připojení nebo vzdáleného zprostředkovatele, který ruší připojení, most počká 20 sekund. Pak se pokusí připojit k další adrese v seznamu, nebo stejnou adresu, pokud je v seznamu pouze jedna adresa.

Připojení ke správci front s více instancemi

V konfiguraci správce front s více instancemi je správce front spuštěn na dvou různých serverech s různými adresami IP. Kanály telemetrie jsou obvykle konfigurovány bez konkrétní adresy IP. Jsou konfigurovány pouze s číslem portu. Když je kanál telemetrie spuštěn, standardně vybere první dostupnou síťovou adresu na lokálním serveru.

Konfigurujte parametr `addresses` v rámci připojení mostu se dvěma adresami IP používanými správcem front. Nastavte parametr `round_robin` na hodnotu `true`.

Dojde-li k selhání aktivní instance správce front, správce front se přepne na instanci v pohotovostním režimu. Démon zjistí, že připojení k aktivní instanci se přerušilo a pokouší se znovu navázat spojení s rezervní instancí. Používá druhou adresu IP v seznamu adres konfigurovaných pro připojení mostu.

Správce front, k němuž je připojen most, je stále stejného správce front. Správce front obnoví svůj vlastní stav. Je-li `cleansession` nastaveno na hodnotu `false`, relace připojení mostu se obnoví do stejného stavu jako před překonáním selhání. Připojení bude pokračovat po prodlevě. Zprávy, které mají "alespoň jednou" nebo "nejvýše jednou" kvalitu služby, se neztratí a odběry budou pokračovat v práci.

Doba opětovného připojení závisí na počtu kanálů a klientů, které se restartují při spuštění instance v pohotovostním režimu, a počet zpráv, které byly zainfraly. Připojení mostu se může pokusit o opakované připojení k oběma adresám IP před opětovným usídlým připojením.

Nekonfigurujte telemetrický kanál správce front s více instancemi s konkrétní adresou IP. Adresa IP je platná pouze na jednom serveru.

Používáte-li alternativní řešení vysoké dostupnosti, které spravuje adresu IP, může být správné konfigurovat kanál telemetrie se specifickou adresou IP.

cleansession

Připojení mostu je relací klienta MQTT v3. Můžete řídit, zda připojení spustí novou relaci, nebo zda obnoví existující relaci. Pokud obnoví existující relaci, připojení mostu zachová odběry a zachované publikace z předchozí relace.

Nenastavujte `cleansession` na `false`, pokud adresa zobrazuje více IP adres a IP adresy se připojují k telemetrickým kanálům umístěným různými správci front nebo k různým telemetrickým démonům. Stav relace se nepřenáší mezi správci front nebo démony. Pokus o restartování existující relace v jiném správci front nebo démonu vede ke spuštění nové relace. Nejisté zprávy jsou ztraceny a odběry se nemusí chovat podle očekávání.

oznámení

Aplikace může sledovat, zda je připojení mostu spuštěno pomocí oznámení. Oznámení je publikace, která má hodnotu 1, připojená nebo 0, odpojená. Je publikován do souboru *topicString*, který je definován parametrem *notification_topic*. Výchozí hodnota *topicString* je `$$SYS/broker/connection/clientIdentifier/state`. Výchozí hodnota *topicString* obsahuje předponu `$$SYS`. Přihlaste se k odběru témat začínajících řetězcem `$$SYS` definováním filtru tématu začínajícího řetězcem `$$SYS`. Filtr témat `#se` přihlásí k odběru všeho, nepřihlásí se k odběru témat začínajících řetězcem `$$SYS` na démonu. Představte si `$$SYS` jako definování speciálního prostoru tématu systému, který se liší od prostoru tématu aplikace.

Volba Oznámení umožňuje produktu IBM WebSphere MQ Telemetry daemon for devices upozornit klienty protokolu MQTT v případě, že je k mostu připojen nebo odpojen.

keepalive_interval

Parametr připojení mostu *keepalive_interval* nastavuje interval mezi mostem, který odesílá příkaz ping protokolu TCP/IP na vzdálený server. Výchozí interval je 60 sekund. Testování spojení zabraňuje tomu, aby byla relace TCP/IP uzavřena vzdáleným serverem nebo brána firewall, která zjišťuje dobu nečinnosti připojení.

CLIENTID

Připojení mostu je relací klienta MQTT v3 a má *clientIdentifier* nastavovacího parametru připojení mostu *clientid*. Chcete-li obnovit předchozí relaci tak, že nastavíte parametr *cleansession* na hodnotu `false`, bude hodnota proměnné prostředí *clientIdentifier* použita v každé relaci shodná. Výchozí hodnota *clientid* je *hostname.connectionName*, která zůstává stejná.

Instalace, ověření, konfigurace a řízení démona produktu WebSphere MQ Telemetry pro zařízení

Instalace, konfigurace a řízení démona jsou založené na souborech.

Nainstalujte démona zkopírováním sady SDK (Software Development Kit) na zařízení, na kterém chcete spustit démona.

Jako příklad spusťte obslužný program klienta protokolu MQTT a připojte se k démonu WebSphere MQ Telemetry pro zařízení jako zprostředkovatel publikování/odběru. Další informace naleznete v tématu [Publikování zprávy do specifického klienta MQTT v3](#).

Konfigurujte démona vytvořením konfiguračního souboru, viz [WebSphere MQ Telemetry démon pro konfigurační soubor zařízení](#).

Řídit běžící démona vytvořením příkazů v souboru `amqtd.d.upd`. Každých 5 sekund démon čte soubor, spouští příkazy a odstraňuje soubor; viz [WebSphere MQ Telemetry démon pro příkazový soubor zařízení](#).

Démon WebSphere MQ Telemetry pro porty modulu listener zařízení

Připojte klienty MQTT V3 k démonu WebSphere MQ Telemetry pro zařízení používající porty modulu listener. Můžete kvalifikovat port modulu listener s bodem připojení a maximálním počtem připojení.

Port modulu listener musí odpovídat číslu portu uvedenému v metodě klienta MQTT `connect(serverURI)` klienta připojujícího se k tomuto portu. Ve výchozím nastavení je na klientovi i na démona nastaven 1883.

Výchozí port pro démona můžete změnit nastavením globální definice `port` v konfiguračním souboru démona. Specifické porty můžete nastavit přidáním definice posluchače do konfiguračního souboru démona.

Pro každý port modulu listener, který není výchozím portem, můžete určit místo připojení pro izolaci klientů. Klienti s připojením k portu s bodem připojení jsou izolováni od ostatních klientů; viz [“Démon WebSphere MQ Telemetry pro body připojení zařízení”](#) na stránce 146.

Můžete omezit počet klientů, kteří se mohou připojit k libovolnému portu. Nastavením globální definice `max_connections` omezíte připojení na výchozí port, nebo kvalifikujte každý port modulu listener parametrem `max_connections`.

Příklad

Příklad konfiguračního souboru, který mění výchozí port z 1883 na 1880, a omezuje připojení na port 1880 na 10000. Připojení k portu 1884 jsou omezena na 1000. Klienti připojené k portu 1884 jsou izolováni od klientů připojených k jiným portům.

```
port 1880
max_connections 10000
listener 1884
mount_point 1884/
max_connections 1000
```

Démon WebSphere MQ Telemetry pro body připojení zařízení

K portu modulu listener, který používají klienti MQTT k připojení k démonu WebSphere MQ Telemetry pro zařízení, můžete přidružit bod připojení s portem modulu listener. Bod připojení izoluje publikování a odběry vyměňované klienty MQTT pomocí jednoho portu modulu listener od klientů MQTT připojených k jinému portu modulu listener.

Klienti připojené k portu modulu listener s bodem připojení nemohou nikdy přímo vyměňovat témata s klienty připojenými k žádným jiným portům modulu listener. Klienti připojené k portu modulu listener bez bodu připojení mohou publikovat nebo odebírat témata libovolného klienta. Klienti si nejsou vědomi toho, zda jsou připojeny k bodu připojení, nebo ne; nečiní žádný rozdíl v řetězcích témat vytvořených klienty.

Bod připojení je řetězec textu, který je vložen před řetězec tématu publikování a odběrů. Je předponou o všech řetězcích témat vytvořených klienty připojenými k portu modulu listener s bodem připojení. Řetězec textu je odebrán ze všech řetězců témat odeslaných klientům připojeným k portu modulu listener.

Pokud port modulu listener nemá bod připojení, řetězce témat publikování a odběrů vytvořených a přijímaných klienty připojenými k portu se nezmění.

Vytvořte řetězce bodu připojení s koncovým /. Tímto způsobem je bod připojení nadřazeným tématem stromu témat pro bod připojení.

Příklad

Konfigurační soubor obsahuje následující porty modulu listener:

```
listener 1883
mount_point 1883/
listener 1884 127.0.0.1
mount_point 1884/
listener 1885
```

Klient, připojený k portu 1883, vytváří odběr produktu `MyTopic`. Démon registruje odběr jako `1883/MyTopic`. Jiný klient připojený k portu 1883 publikuje zprávu na téma `MyTopic`. Démon změní řetězec tématu na `1883/MyTopic` a vyhledá odpovídající odběry. Odběratel na portu 1883 obdrží publikování s původním řetězcem tématu `MyTopic`. Démon odebral předponu bodu připojení z řetězce tématu.

Jiný klient, připojený k portu 1884, také publikuje na téma `MyTopic`. Tento čas démon registruje téma jako `1884/MyTopic`. Odběratel na portu 1883 neobdržel publikování, protože jiný bod připojení má za následek odběr s jiným řetězcem tématu.

Klient, připojený k portu 1885, publikuje na téma, `1883/MyTopic`. Démon nemění řetězec tématu. Odběratel na portu 1883 přijímá publikování do produktu `MyTopic`.

Démon WebSphere MQ Telemetry pro kvalitu zařízení služby, trvalé odběry a zachované publikace

Nastavení kvality služby se vztahuje pouze na spuštěného démona. Pokud se démon zastaví, ať už řízeným způsobem, nebo kvůli selhání, stav inflatěových zpráv je ztracen. Doručování zprávy alespoň jednou, nebo nanejvýš jednou, nelze zaručit, pokud se démon zastaví. Démon WebSphere MQ Telemetry pro zařízení podporuje omezené trvání. Nastavte konfigurační parametr **retained_persistence** pro uložení zachovaných publikování a odběrů, pokud je démon vypnut.

Na rozdíl od produktu WebSphere MQ démon WebSphere MQ Telemetry pro zařízení nežurnálovat trvalá data. Stav relace, stav zprávy a zachované publikace nejsou ukládány transakčně. Démon ve výchozím nastavení vyřazuje všechna data, když se zastaví. Můžete nastavit volbu pro periodické check-checkpoint a zachované publikace. Stav zprávy se vždy ztratí, když se démon zastaví. Všechny nezachované publikace jsou ztraceny.

Nastavte volbu konfigurace démona `Retained_persistence` na `true`, chcete-li ukládat zachovaná publikování pravidelně do souboru. Když se démon restartuje, zachované publikace, které byly naposledy uloženy automaticky, byly obnoveny. Ve výchozím nastavení nejsou zachované zprávy vytvořené klienty obnoveny při restartu démona.

Nastavte volbu konfigurace démona `Retained_persistence` na `true`, chcete-li uložit odběry vytvořené v trvalé relaci pravidelně do souboru. Je-li volba `Retained_persistence` nastavena na hodnotu `true`, dojde k obnovení odběrů, které klienti vytvářejí v relaci s volbou `CleanSession` nastaveným na hodnotu `false`, "trvalou relací". Démon obnoví odběry, když se restartuje, což začne přijímat publikování. Klient přijme publikace, když se restartuje s volbou `CleanSession` na `false`. Stav relace klienta se standardně neukládá, když se démon zastaví, a proto se neobnoví odběry, a to ani v případě, že klient nastaví `CleanSession` na `false`.

`Retained_persistence` je mechanismus automatického ukládání. Nemohou uložit nejnovější zachované publikace nebo odběry. Můžete měnit způsob ukládání zachovaných publikování a odběrů. Nastavte interval mezi uloženími nebo počet změn mezi uloženími, pomocí voleb konfigurace `autosave_on_changes` a `autosave_interval`.

Ukázková konfigurace pro nastavení perzistence

```
# Sample configuration
# Daemon listens on port 1882 with persistence in /tmp
# Autosave every minute
port 1882
persistence_location /tmp/
retained_persistence true
autosave_on_changes false
autosave_interval 60
```

Démon WebSphere MQ Telemetry pro zabezpečení zařízení

Démon WebSphere MQ Telemetry pro zařízení může ověřovat klienty, kteří se k němu připojují, používat pověření k připojení k jiným zprostředkovatelům a řídit přístup k tématům. Zabezpečení démona je omezeno tím, že je sestavováno pomocí klienta WebSphere MQ Telemetry C, který neposkytuje podporu SSL. Připojení k démonu a od něj proto nejsou šifrovány a nelze je ověřovat pomocí certifikátů.

Ve výchozím nastavení není zapnuto žádné zabezpečení.

Ověřování klientů

Klienti MQTT mohou nastavit jméno uživatele a heslo pomocí metod `MqttConnectOptions.setUsername` a `MqttConnectOptions.setPassword`.

Ověřte klienta, který se připojuje k démonu, kontrolou jména uživatele a hesla poskytnutého klientem proti položkám v souboru hesel. Chcete-li povolit ověření, vytvořte soubor hesel a nastavte parametr `password_file` v konfiguračním souboru démona; viz `password_file`.

Nastavte parametr `allow_anonymous` v konfiguračním souboru démona tak, aby se klienti připojující bez uživatelských jmen nebo hesel připojovali k démonu, který kontroluje ověření; viz [allow_anonymous](#). Pokud klient poskytuje jméno uživatele nebo heslo, je vždy zkontrolováno proti souboru hesel, je-li nastaven parametr `password_file`.

Nastavte parametr `clientity_prefixes` v konfiguračním souboru démona a omezte připojení na konkrétní klienty. Klienti musí mít `clientIdentifiers`, které začínají jednou z předpon uvedených v parametru `clientid_prefixes`; viz [clientid_prefixes](#).

Zabezpečení připojení mostu

Každý démon WebSphere MQ Telemetry pro připojení mostu zařízení je klient MQTT V3. Můžete nastavit jméno uživatele a heslo pro každé připojení mostu jako parametr připojení mostu v konfiguračním souboru démona; viz [username](#) a [password](#). Most se poté může ověřit u zprostředkovatele.

Řízení přístupu k tématům

Jsou-li klienti ověřováni, může démon také poskytovat řízení přístupu k tématům pro každého uživatele. Démon uděluje řízení přístupu na základě shody s tématem, ke kterému klient buď publikuje, nebo se přihlašuje s řetězcem tématu přístupu v souboru řízení přístupu; viz [acl_file](#).

Seznam přístupových práv má dvě části. První část řídí přístup pro všechny klienty, včetně anonymních klientů. Druhá část má sekci pro libovolného uživatele v souboru hesel. Vypisuje specifické řízení přístupu pro každého uživatele.

Příklad

Parametry zabezpečení jsou zobrazeny v následujícím příkladu.

```
acl_file c:\WMQTDaemon\config\acl.txt
password_file c:\WMQTDaemon\config\passwords.txt
allow_anonymous true
connection Daemon1
username daemon1
password deamonpassword
```

Obrázek 43. Konfigurační soubor démona

```
Fred:Fredpassword
Barney:Barneypassword
```

Obrázek 44. Soubor hesel, `passwords.txt`

```
topic home/public/#
topic read meters/#
user Fred
topic write meters/fred
topic home/fred/#
user Barney
topic write meters/barney
topic home/barney/#
```

Obrázek 45. Soubor řízení přístupu, `acl.txt`

Administrace výběrového vysílání

Tyto informace použijte k seznámení se s úlohami administrace výběrového vysílání produktu WebSphere MQ, jako je například snížení velikosti zpráv výběrového vysílání a povolení konverze dat.

Začínáme s výběrovým vysíláním

Tyto informace použijte k zahájení práce s tématy výběrového vysílání WebSphere MQ a s objekty s informacemi o komunikaci.

Informace o této úloze

Systém zpráv výběrového vysílání produktu WebSphere MQ používá síť k doručování zpráv pomocí mapování témat na adresy skupin. Následující úlohy jsou rychlým způsobem, jak otestovat, zda je požadovaná adresa IP a port správně nakonfigurovány pro systém zpráv výběrového vysílání.

Vytvoření objektu COMMINFO pro výběrové vysílání

Objekt s informacemi o komunikaci (COMMINFO) obsahuje atributy přidružené k přenosu výběrového vysílání. Další informace o parametrech objektu COMMINFO najdete v tématu [DEFINE COMMINFO](#).

K definování objektu COMMINFO pro výběrové vysílání použijte následující příklad příkazového řádku:

```
DEFINE COMMINFO(MC1) GRPADDR(group address) PORT(port number)
```

kde *MC1* je název vašeho objektu COMMINFO, *adresa skupiny* je vaše adresa IP skupinového vysílání nebo název DNS a *číslo portu* je port, na kterém má být přenos vysílán (Výchozí hodnota je 1414).

Vytvoří se nový objekt COMMINFO s názvem *MC1*; tento název je název, který musíte zadat při definování objektu TOPIC v následujícím příkladu.

Vytvoření objektu TOPIC pro výběrové vysílání

Téma je předmětem informací publikovaných ve zprávě publikování/odběru a téma je definováno vytvořením objektu TOPIC. Objekty TOPIC mají dva parametry, které definují, zda mohou být použity s výběrovým vysíláním nebo ne. Tyto parametry jsou: **COMMINFO** a **MCAST**.

- **COMMINFO** Tento parametr uvádí název objektu informací o komunikaci výběrového vysílání. Další informace o parametrech objektu COMMINFO najdete v tématu [DEFINE COMMINFO](#).
- **MCAST** Tento parametr uvádí, zda je výběrové vysílání povoleno na této pozici ve stromu témat.

K definování objektu TOPIC pro výběrové vysílání použijte následující příklad příkazového řádku:

```
DEFINE TOPIC(ALLSPORTS) TOPICSTR('Sports') COMMINFO(MC1) MCAST(ENABLED)
```

Vytvoří se nový objekt TOPIC s názvem *ALLSPORTS*. Má řetězec tématu *Sports*, jeho související komunikační informační objekt se nazývá *MC1* (což je název, který jste zadali při definování objektu COMMINFO v předchozím příkladu) a výběrové vysílání je povoleno.

Testování publikování/odběru výběrového vysílání

Po vytvoření objektů TOPIC a COMMINFO lze tyto objekty testovat pomocí ukázky `amqspubc` a ukázky `amqssubc`. Další informace o těchto ukázkách naleznete v tématu [Ukázkové programy publikování/odběru](#).

1. Otevřete dvě okna příkazového řádku; První příkazový řádek je určen pro ukázkou publikování `amqspubc` a druhý příkazový řádek je určen pro ukázkou odběru `amqssubc`.
2. Zadejte následující příkaz na příkazový řádek 1:

```
amqspubc Sports QM1
```

kde *Sports* je řetězec tématu objektu TOPIC definovaný v předchozím příkladu, a *QM1* je název správce front.

3. Zadejte následující příkaz na příkazový řádek 2:

```
amqssubc Sports QM1
```

kde *Sports* a *QM1* jsou stejné jako použité v kroku "2" na stránce 149.

4. Zadejte příkaz `Hello world` na příkazový řádek 1. Pokud je port a adresa IP, které jsou uvedeny v objektu `COMMINFO`, správně nakonfigurovány; vzorek `amqssubc`, který naslouchá na portu pro publikace z uvedené adresy, výstupů `Hello world` na příkazovém řádku 2.

Topologie témat výběrového vysílání produktu IBM WebSphere MQ

Tento příklad se používá k pochopení topologie témat výběrového vysílání produktu IBM WebSphere MQ.

Podpora skupinového vysílání produktu IBM WebSphere MQ vyžaduje, aby každý podstrom má svou vlastní skupinu výběrového vysílání a datový proud v rámci celkové hierarchie.

Schéma tvorby adres IP *classful network* má určený adresní prostor pro adresy pro výběrové vysílání. Úplný rozsah adres IP pro výběrové vysílání je `224.0.0.0` až `239.255.255.255`, ale některé z těchto adres jsou vyhrazeny. Pro seznam rezervovaných adres se obraťte na administrátora systému nebo se podívejte na další informace v protokolu [IPv4 Multicast Address Space Registry](#). Doporučuje se používat lokálně vymezenou adresu výběrového vysílání v rozsahu `239.0.0.0` až `239.255.255.255`.

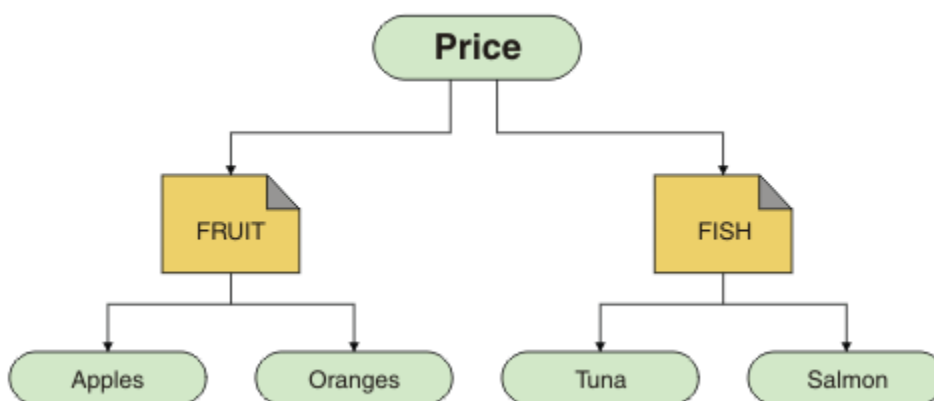
V následujícím diagramu jsou k dispozici dva možné proudy dat výběrového vysílání:

```
DEF COMMINFO(MC1) GRPADDR(239.XXX.XXX.XXX
)
DEF COMMINFO(MC2) GRPADDR(239.YYY.YYY.YYY)
```

kde `239.XXX.XXX.XXX` a `239.YYY.YYY.YYY` jsou platné adresy výběrového vysílání.

Tyto definice tématu se používají k vytvoření stromu témat, jak je zobrazeno v následujícím diagramu:

```
DEFINE TOPIC(FRUIT) TOPICSTRING('Price/FRUIT') MCAST(ENABLED) COMMINFO(MC1)
DEFINE TOPIC(FISH) TOPICSTRING('Price/FISH') MCAST(ENABLED) COMMINFO(MC2)
```



Každý objekt pro komunikaci výběrového vysílání (`COMMINFO`) představuje jiný proud dat, protože jejich adresy skupin jsou odlišné. V tomto příkladě je téma `FRUIT` definováno pro použití objektu `COMMINFO MC1`, téma `FISH` je definováno pro použití objektu `COMMINFO MC2` a uzel `Price` nemá žádné definice výběrového vysílání.

Výběrové vysílání produktu WebSphere MQ má pro řetězce témat mezní hodnotu 255 znaků. Toto omezení znamená, že je třeba dbát na názvy uzlů a listových uzlů ve stromu; pokud jsou názvy uzlů a koncových uzlů příliš dlouhé, může řetězec tématu překročit 255 znaků a vrátit kód příčiny [2425 \(0979\) \(RC2425\): MQRC_TOPIC_STRING_ERROR](#). Doporučuje se, aby řetězce témat byly co nejkratší, protože delší řetězce témat mohou mít škodlivý vliv na výkon.

Řízení velikosti zpráv výběrového vysílání

Tyto informace použijte, chcete-li se dozvědět více o formátu zpráv produktu WebSphere MQ a snížit velikost zpráv WebSphere MQ.

Zprávy produktu WebSphere MQ mají k sobě přidružené číselné atributy, které jsou obsaženy v deskriptoru zpráv. Pro malé zprávy mohou tyto atributy představovat většinu datového provozu a mohou mít významný nepříznivý vliv na přenosovou rychlost. Výběrové vysílání produktu WebSphere MQ umožňuje uživateli konfigurovat, který z těchto atributů je přenášen spolu se zprávou.

Přítomnost atributů zpráv jiných než řetězec tématu závisí na tom, zda objekt COMMINFO uvádí, že musí být odeslány nebo nikoli. Není-li atribut přenesen, použije přijímající aplikace výchozí hodnotu. Výchozí hodnoty MQMD nejsou nutně stejné jako hodnota MQMD_DEFAULT, a jsou popsány v [Tabulka 9 na stránce 151](#).

Objekt COMMINFO obsahuje atribut MCPROP, který řídí, kolik z polí MQMD a vlastností uživatele proudí se zprávou. Nastavením hodnoty tohoto atributu na odpovídající úroveň můžete řídit velikost zpráv výběrového vysílání produktu WebSphere MQ :

MCPROP

Vlastnosti výběrového vysílání určují, kolik vlastností MQMD a uživatelských vlastností bude přenášeno se zprávami.

ALL

Zasílají se všechny uživatelské vlastnosti a všechna pole MQMD.

Odpověď

Přenášejí se pouze uživatelské vlastnosti a pole MQMD, která souvisejí s odpovídáním na zprávy. Jde o následující vlastnosti:

- MsgType
- MessageId
- CorrelId
- ReplyToQ
- ReplyToQmgr

UŽIVATEL

Přenášejí se pouze uživatelské vlastnosti.

ŽÁDNÉ

Nepřenášejí se žádné uživatelské vlastnosti ani pole MQMD.

COMPAT

This value causes the transmission of the message to be done in a compatible mode to RMM, which allows some inter-operation with the current XMS applications and WebSphere Message Broker RMM applications.

Atributy zpráv výběrového vysílání

Atributy zprávy mohou pocházet z různých míst, jako je MQMD, polí v MQRFH2a vlastností zprávy.

Následující tabulka ukazuje, co se stane, když se zprávy odesílají s výhradou hodnoty [MCPROP](#) a výchozí hodnoty použité, když se atribut neodešle.

<i>Tabulka 9. Atributy systému zpráv a informace o tom, jak se vztahují k</i>		
Atribut	Akce při použití výběrového vysílání	Předvolba, pokud není přenesena
TopicString	Vždy zahrnout	Nelze použít
MQMQ StrucId	Nepřeneseno	Nelze použít
Verze MQMD	Nepřeneseno	Nelze použít
Sestava	Zahrnuto, pokud není výchozí	0
MsgType	Zahrnuto, pokud není výchozí	MQM_DATAGRAM
Vypršení	Zahrnuto, pokud není výchozí	0

Tabulka 9. Atributy systému zpráv a informace o tom, jak se vztahují k (pokračování)

Atribut	Akce při použití výběrového vysílání	Předvolba, pokud není přenesena
Zpětná vazba	Zahrnuto, pokud není výchozí	0
Kódování	Zahrnuto, pokud není výchozí	MQENC_NORMAL (equiv).
CodedCharSetId	Zahrnuto, pokud není výchozí	1208
Formát	Zahrnuto, pokud není výchozí	MQRFH2
Priorita	Zahrnuto, pokud není výchozí	4
Trvání	Zahrnuto, pokud není výchozí	MQPER_NOT_PERSISTENT
MsgId	Zahrnuto, pokud není výchozí	Null
CorrelId	Zahrnuto, pokud není výchozí	Null
BackoutCount	Zahrnuto, pokud není výchozí	0
ReplyToQ	Zahrnuto, pokud není výchozí	Prázdný
ReplyToQMgr	Zahrnuto, pokud není výchozí	Prázdný
UserIdentifier	Zahrnuto, pokud není výchozí	Prázdný
AccountingToken	Zahrnuto, pokud není výchozí	Null
Typ ITType PutApp	Zahrnuto, pokud není výchozí	MQAT_JAVA
PutAppINName	Zahrnuto, pokud není výchozí	Prázdný
PutDate	Zahrnuto, pokud není výchozí	Prázdný
PutTime	Zahrnuto, pokud není výchozí	Prázdný
ApplOriginData	Zahrnuto, pokud není výchozí	Prázdný
GroupID	Vyloučeno	Nelze použít
MsgSeqNumber	Vyloučeno	Nelze použít
Offset	Vyloučeno	Nelze použít
MsgFlags	Vyloučeno	Nelze použít
OriginalLength	Vyloučeno	Nelze použít
UserProperties	Zahrnuto	Nelze použít

Související odkazy

[ZMĚNIT KOMMINFO](#)

[DEFINOVAT COMMINFO](#)

Povolení konverze dat pro systém zpráv výběrového vysílání

Pomocí těchto informací můžete pochopit, jak převod dat pracuje pro systém zpráv výběrového vysílání WebSphere MQ Multicast.

WebSphere MQ Multicast je sdílený, bezspojový protokol, a proto není možné, aby každý klient mohl provádět specifické požadavky na konverzi dat. Každý klient přihlášený k odběru stejného toku výběrového vysílání přijímá stejná binární data; proto, je-li zapotřebí konverze dat produktu WebSphere MQ, provede se konverze lokálně na každém klientovi.

V instalaci se smíšenými platformami může být to, že většina klientů vyžaduje data ve formátu, který není nativním formátem pro přenášeč aplikaci. V této situaci lze hodnoty **CCSID** a **ENCODING** objektu výběrového vysílání **COMMINFO** použít k definování kódování přenosu zpráv pro efektivitu.

Výběrové vysílání produktu WebSphere MQ podporuje převod dat informačního obsahu zprávy pro následující vestavěné formáty:

- MQADMIN
- UDÁLOST MQEVENT
- MQPCF
- MQRFH.
- MQRFH2
- FUNKCE MQSTR

Kromě těchto formátů můžete také definovat vlastní formáty a použít uživatelskou proceduru pro převod dat MQDXP-Data-conversion exit parameter .

Informace o převezech programovacích dat naleznete v tématu Převod dat v rozhraní MQI pro systém zpráv výběrového vysílání.

Další informace o konverzi dat najdete v tématu Převod dat.

Další informace o uživatelských procedurách pro převod dat a produktu `ClientExitPath` naleznete v části ClientExitPath stanza konfiguračního souboru klienta.

Monitorování aplikace výběrového vysílání

Tyto informace použijte k seznámení se s administrací a monitorováním výběrového vysílání WebSphere MQ .

Stav aktuálních vydavatelů a odběratelů pro provoz výběrového vysílání (například počet odeslaných a přijatých zpráv nebo počet ztracených zpráv) se pravidelně přenáší na server z klienta. Když je přijat stav, atribut `COMMEV` objektu `COMMINFO` uvádí, zda správce front vloží do `SYSTEM.ADMIN.PUBSUB.EVENT`. Zpráva o události obsahuje přijaté informace o stavu. Tyto informace jsou neocenitelnou diagnostickou pomůckou při hledání zdroje problému.

Chcete-li zobrazit informace o připojení aplikací připojených ke správci front, použijte příkaz `MQSC DISPLAY CONN` . Další informace o příkazu **DISPLAY CONN** naleznete v části DISPLAY CONN.

Použijte příkaz `MQSC DISPLAY TPSTATUS` , abyste zobrazili stav vašich vydavatelů a odběratelů. Další informace o příkazu **DISPLAY TPSTATUS** naleznete v části DISPLAY TPSTATUS.

COMPMEV a indikátor spolehlivosti zpráv výběrového vysílání

Indikátor spolehlivosti používaný ve spojení s atributem **COMMEV** objektu `COMMINFO` je klíčovým prvkem monitorování vydavatelů a odběratelů výběrového vysílání WebSphere MQ . Indikátor spolehlivosti (pole **MSGREL** , které je vráceno ve stavových příkazech publikování nebo odběru) je indikátor WebSphere MQ , který ilustruje procentní část přenosů, které nemají žádné chyby. Někdy je nutné zprávy znovu přenést kvůli chybě přenosu, která se odrazí v hodnotě **MSGREL** . Potenciální příčiny chyb přenosu zahrnují pomalé odběratele, přetížené sítě a výpadky sítě. **COMMEV** kontroluje, zda jsou zprávy událostí generovány pro výběrové vysílání, které jsou vytvořeny pomocí objektu `COMMINFO` a jsou nastaveny na jednu ze tří možných hodnot:

VYPNUTO

Zprávy událostí se nezapíší.

POVOLENO

Zprávy událostí se vždy zapíší, s frekvencí definovanou v parametru `COMMINFO MONINT` .

Výjimka

Zprávy událostí jsou zapisovány v případě, že spolehlivost klesne pod prahovou hodnotu spolehlivosti. Úroveň spolehlivosti zpráv 90% nebo méně označuje, že se může vyskytnout problém s konfigurací sítě nebo že jedna nebo více aplikací Publikování/Odběru běží příliš pomalu:

- Hodnota **MSGREL (100, 100)** udává, že v krátké době nebo v dlouhodobém časovém rámci nedošlo k žádným problémům.
- Hodnota **MSGREL (80, 60)** udává, že 20% zpráv má v současné době problémy, ale že je to také zlepšení dlouhodobé hodnoty 60.

Klienti mohou pokračovat v přenášení a přijímání dat výběrového vysílání i v případě, že je přerušeno připojení typu unicast do správce front, a proto mohou být data zastaralá.

spolehlivost zpráv výběrového vysílání

V této části se dozvíte, jak nastavit odběr výběrového vysílání WebSphere MQ a historii zpráv.

Klíčovým prvkem pro překonání selhání přenosu s výběrovým vysíláním je WebSphere MQ¹ s vyrovnávací pamětí přenášených dat (historie zpráv, které mají být uchovány na vysílajícím konci odkazu). Tento proces znamená, že ukládání zpráv do vyrovnávací paměti není vyžadováno při vkládání aplikačního procesu, protože produkt WebSphere MQ poskytuje spolehlivost. Velikost této historie se konfiguruje prostřednictvím objektu informací o komunikaci (COMMINFO), jak je popsáno v následujících informacích. Větší přenosová vyrovnávací paměť znamená, že se v případě potřeby přenáší více historie přenosu, ale vzhledem k povaze výběrového vysílání 100% zajištěného doručení nelze podporovat.

Historie zpráv výběrového vysílání WebSphere MQ je řízena v objektu informací o komunikaci (COMMINFO) pomocí atributu **MSGHIST** :

MSGHIST

Tato hodnota představuje množství historie zpráv v kilobajtech, které uchovává systém k obsluze opakovaných přenosů v případě NACKs (negativní potvrzení).

Hodnota 0 udává nejnižší úroveň spolehlivosti. Výchozí hodnota je 100 KB.

Historie nových odběrů výběrového vysílání produktu WebSphere MQ je řízena v objektu informace o komunikaci (COMMINFO) atributem **NSUBHIST** :

NSUBHIST

Historie nového odběratele určuje, zda odběratel, který se připojuje k proudu publikování, obdrží veškerá data, která jsou momentálně k dispozici, nebo jen publikace zveřejněné od okamžiku přihlášení k odběru.

ŽÁDNÉ

Hodnota NONE způsobí, že se vysílač přenesou pouze publikování provedené od okamžiku odběru. Výchozí hodnota je NONE .

ALL

Hodnota ALL způsobí, že se vysílač znovu přenesou, jak je známo, o historii tématu. Za určitých okolností může tato situace vyvolat podobné chování u zachovaných publikování.

Poznámka: Použití hodnoty ALL může mít nepříznivý vliv na výkon, pokud existuje rozsáhlá historie témat, protože všechny historie témat jsou znovu přeneseny.

Související odkazy

[DEFINOVAT COMMINFO](#)

[ZMĚNIT KOMMINFO](#)

Rozšířené úlohy výběrového vysílání

V těchto informacích naleznete informace o rozšířených úlohách administrace výběrového vysílání produktu WebSphere MQ , jako je například konfigurování souborů produktu .ini a interoperability s produktem WebSphere MQ LLM.

Aspekty zabezpečení v instalaci výběrového vysílání naleznete v tématu [Zabezpečení výběrového vysílání](#) .

Přemostění mezi doménami pro výběrové vysílání a non-multicast

Pomocí těchto informací můžete pochopit, co se stane, když vydavatel, který není výběrového vysílání, publikuje do tématu WebSphere MQ Multicast enabled.

Pokud vydavatel, který není výběrového vysílání, publikuje do tématu, které je definováno jako **MCAST** povoleno, a povolíte **BRIDGE**, správce front předá zprávu prostřednictvím výběrového vysílání přímo všem odběratelům, kteří mohou naslouchat. Vydavatel výběrového vysílání nemůže publikovat témata, která nejsou aktivována pro výběrové vysílání.

Existující témata mohou být multicast zapnutá nastavením parametrů **MCAST** a **COMMINFO** objektu tématu. Další informace o těchto parametrech naleznete v tématu [Základní koncepce výběrového vysílání](#).

Atribut **COMMINFO** objektu **BRIDGE** řídí publikace z aplikací, které nepoužívají výběrové vysílání. Je-li parametr **BRIDGE** nastaven na hodnotu **ENABLED** a parametr **MCAST** tématu je také nastaven na hodnotu **ENABLED**, budou z aplikací, které nepoužívají výběrové vysílání, připojeno k aplikacím, které nepoužívají výběrové vysílání. Další informace o parametru **BRIDGE** naleznete v tématu [DEFINE COMMINFO](#).

Konfigurace souborů .ini pro výběrové vysílání

Tyto informace použijte k porozumění polím výběrového vysílání produktu WebSphere MQ v souborech .ini.

Další konfiguraci výběrového vysílání produktu WebSphere MQ lze provést v souboru ini. Specifický soubor ini, který musíte použít, je závislý na typu aplikací:

- Klient: Nakonfigurujte soubor *MQ_DATA_PATH/mqclient.ini*.
- Správce front: Konfigurujte soubor *MQ_DATA_PATH/qmgrs/QMNAME/qm.ini*.

kde *MQ_DATA_PATH* je umístění datového adresáře WebSphere MQ (*/var/mqm/mqclient.ini*) a *QMNAME* je název správce front, na který se vztahuje soubor .ini.

Soubor .ini obsahuje pole používaná k dokončování chování výběrového vysílání produktu WebSphere MQ :

```
Multicast:
Protocol           = IP | UDP
IPVersion          = IPV4 | IPV6 | ANY | BOTH
LimitTransRate     = DISABLED | STATIC | DYNAMIC
TransRateLimit     = 100000
SocketTTL          = 1
Batch              = NO
Loop               = 1
Interface          = <IPaddress>
FeedbackMode       = ACK | NACK | WAIT1
HeartbeatTimeout   = 20000
HeartbeatInterval  = 2000
```

Protokol

UDP

V tomto režimu jsou pakety odesílány pomocí protokolu UDP. Síťové prvky nemohou poskytovat asistenci v distribuci výběrového vysílání, protože je to v režimu IP. Formát paketu zůstává kompatibilní s PGM. Toto je výchozí hodnota.

IP

V tomto režimu vysílač odesílá prvotní IP pakety. Síťové prvky s podporou PGM pomáhají v spolehlivé výběrové distribuci paketů. Tento režim je plně kompatibilní se standardem PGM.

Verze protokolu IP

IPV4

Komunikujte pouze pomocí protokolu IPv4. Toto je výchozí hodnota.

IPV6

Komunikujte pouze pomocí protokolu IPv6.

ANY

Komunikujte pomocí protokolu IPv4, IPv6 nebo obou, podle toho, který protokol je k dispozici.

obojí

Podporuje komunikaci pomocí protokolů IPv4 i IPv6.

LimitTrans-Rychlost**VYPNUTO**

K dispozici není žádná kontrola přenosové rychlosti. Toto je výchozí hodnota.

STATICÝ

Implementuje řídicí prvek statické přenosové rychlosti. Vysílač nebude přenášet rychlostí, která přesahuje rychlost zadanou parametrem TransRateLimit.

DYNAMICKÝ

Vysílač adaptuje svou přenosovou rychlost podle zpětné vazby, kterou dostává od přijímačů.

V tomto případě nesmí být limit přenosové rychlosti vyšší než hodnota uvedená v parametru LimitTransRate. Vysílač se snaží dosáhnout optimální přenosové rychlosti.

Limit TransRate

Limit přenosové rychlosti v Kbps.

SocketTTL

Hodnota parametru SocketTTL určuje, zda může výběrový provoz projít přes směrovač, nebo počet směrovačů, které může projít.

Dávka

Určuje, zda jsou zprávy v dávce nebo jsou odeslány okamžitě. Jsou zde 2 možné hodnoty:

- *NE* Zprávy nejsou dávkové zpracování, jsou odeslány okamžitě.
- *ANO* Zprávy jsou v dávce.

Cyklus

Chcete-li povolit cyklus výběrového vysílání, nastavte hodnotu na 1. Smyčka skupinového vysílání definuje, zda jsou data odesílána do hostitele zpět do hostitele nebo ne.

Rozhraní

Adresa IP rozhraní, na kterém probíhá výběrové vysílání. Další informace a odstraňování problémů najdete v tématu: [Testování aplikací pro výběrové vysílání v síti bez výběrového vysílání a Nastavení příslušné sítě pro provoz výběrového vysílání](#)

FeedbackMode**NACK**

Zpětná vazba pro negativní potvrzení. Toto je výchozí hodnota.

ACK

Zpětná vazba prostřednictvím pozitivních potvrzení.

WAIT1

Zpětná vazba od pozitivních potvrzení, kde vysílač čeká pouze 1 ACK od některého z přijímačů.

HeartbeatTimeout

Časový limit prezenčního signálu v milisekundách. Hodnota 0 označuje, že události časového limitu prezenčního signálu nejsou generovány přijímačem nebo přijímači daného tématu. Výchozí hodnota je 20000.

HeartbeatInterval

Interval prezenčního signálu v milisekundách. Hodnota 0 znamená, že nejsou odeslány žádné synchronizační signály. Interval prezenčního signálu musí být podstatně menší než hodnota **HeartbeatTimeout**, aby nedošlo k nepravdivému výskytu události časového limitu prezenčního signálu. Výchozí hodnota je 2000.

interoperabilita výběrového vysílání s produktem WebSphere MQ Low Latency Messaging

Tyto informace vám pomohou pochopit interoperabilitu produktů WebSphere MQ Multicast a WebSphere MQ Low Latency Messaging (LLM).

Základní přenos informačního obsahu je možný pro aplikaci pomocí LLM, s jinou aplikací používající výběrové vysílání pro výměnu zpráv v obou směrech. Ačkoli multicast používá technologii LLM, samotný produkt LLM není zabudovaný. Proto je možné instalovat služby LLM i WebSphere MQ Multicast a pracovat a obsluhovat tyto dva produkty odděleně.

Aplikace LLM, které komunikují s výběrovým vysíláním, mohou vyžadovat odeslání a přijetí vlastností zprávy. Vlastnosti zprávy produktu WebSphere MQ a pole MQMD se přenášejí jako vlastnosti zprávy LLM se specifickými kódy vlastností zprávy LLM, jak je uvedeno v následující tabulce:

Tabulka 10. Vlastnosti zprávy produktu WebSphere MQ na mapování vlastností LLM produktu WebSphere MQ

Vlastnost produktu WebSphere MQ	Typ vlastnosti WebSphere MQ LLM	Druh nemovitosti LLM	Kód vlastnosti LLM
MQMD.Report	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1001
MQMD.MsgType	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1002
MQMD.Expiry	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1003
MQMD.Feedback	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1004
MQMD.Encoding	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1005
MQMD.CodedCharSetId	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1006
MQMD.Format	RMM_MSG_PROP_BYTES	LM_PROP_PROP_PROP_Řetězec	-1007
MQMD.Priority	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1008
MQMD.Persistence	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1009
MQMD.MsgId	RMM_MSG_PROP_BYTES	LLM_PROP_KIND_ByteArray	-1010
MQMD.BackoutCount	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1012
MQMD.ReplyToQ	RMM_MSG_PROP_BYTES	LM_PROP_PROP_PROP_Řetězec	-1013
MQMD.ReplyToQMger	RMM_MSG_PROP_BYTES	LM_PROP_PROP_PROP_Řetězec	-1014
MQMD.PutDate	RMM_MSG_PROP_BYTES	LM_PROP_PROP_PROP_Řetězec	-1020
MQMD.PutTime	RMM_MSG_PROP_BYTES	LM_PROP_PROP_PROP_Řetězec	-1021
MQMD.ApplOriginData	RMM_MSG_PROP_BYTES	LM_PROP_PROP_PROP_Řetězec	-1022
MQPubOptions	RMM_MSG_PROP_INT32	LLM_PROP_KIND_int32	-1053

Další informace o LLM naleznete v dokumentaci k produktu LLM: [WebSphere MQ Low Latency Messaging](#).

Správa serveru HP Integrity NonStop Server

V této části najdete informace o administrativních úlohách pro klienta IBM WebSphere MQ pro produkt HP Integrity NonStop Server.

Pro vás jsou k dispozici dvě administrativní úlohy:

1. Ruční spuštění TMF/Gateway z Pathway.
2. Zastavení TMF/Gateway z Pathway.

Ruční spuštění TMF/Gateway z Pathway

Můžete povolit cestu k automatickému spuštění TMF/Gateway na prvním požadavku na zařazení do seznamu, nebo můžete ručně spustit TMF/Gateway z Pathway.

Postup

Chcete-li ručně spustit TMF/Gateway z Pathway, zadejte následující příkaz PATHCOM:

```
START SERVER <server_class_name>
```

Pokud klientská aplikace provede požadavek na zařazení do seznamu, než TMF/Gateway dokončí obnovu neověřených transakcí, bude požadavek pozastaven až po dobu 1 sekundy. Pokud se obnova v tomto čase nedokončí, je zařazení do seznamu zamítnuto. Klient pak obdrží chybu MQRC_UOW_ENLISTMENT_ERROR z použití transakčního rozhraní MQI.

Zastavení TMF/Gateway z Pathway

Tato úloha popisuje, jak zastavit TMF/Gateway z Pathway, a jak restartovat TMF/Gateway poté, co ji zastavíte.

Postup

1. Chcete-li zabránit tomu, aby byly do TMF/Gateway provedeny nové požadavky na zařazení do seznamu, zadejte následující příkaz:

```
FREEZE SERVER <server_class_name>
```

2. Chcete-li spustit TMF/Gateway k dokončení jakýchkoli in-flight operací a ukončit, zadejte tento příkaz:

```
STOP SERVER <server_class_name>
```

3. Chcete-li povolit automatické restartování TMF/Gateway buď automaticky při prvním zařazování nebo ručním provedením následujících kroků [1](#) a [2](#), zadejte tento příkaz:

```
THAW SERVER <server_class_name>
```

Aplikace nemůže provádět nové požadavky na zařazení a není možné vydat příkaz **START**, dokud nezádáte příkaz **THAW**.

Tyto informace byly vyvinuty pro produkty a služby poskytované v USA.

Společnost IBM nemusí nabízet produkty, služby nebo funkce uvedené v tomto dokumentu v jiných zemích. Informace o produktech a službách, které jsou ve vaší oblasti aktuálně dostupné, získáte od místního zástupce společnosti IBM. Odkazy na produkty, programy nebo služby společnosti IBM v této publikaci nejsou míněny jako vyjádření nutnosti použití pouze uvedených produktů, programů či služeb společnosti IBM. Místo toho lze použít jakýkoli funkčně ekvivalentní produkt, program nebo službu, které neporušují žádná práva k duševnímu vlastnictví IBM. Za vyhodnocení a ověření činnosti libovolného produktu, programu či služby od jiného typu než IBM je však zodpovědný uživatel.

Společnost IBM může vlastnit patenty nebo nevyřízené žádosti o patenty zahrnující předměty popsané v tomto dokumentu. Vlastnictví tohoto dokumentu neposkytuje licenci k těmto patentům. Dotazy týkající se licencí můžete posílat písemně na adresu:

IBM Director of Licensing IBM Corporation North Castle Drive Armonk, NY 10504-1785 U.S.A.

Odpovědi na dotazy týkající se licencí pro dvoubajtové znakové sady (DBCS) získáte od oddělení IBM Intellectual Property Department ve vaší zemi, nebo tyto dotazy můžete zasílat písemně na adresu:

Intellectual Property Licensing Legal and Intellectual Property Law IBM Japan, Ltd. 19-21, Nihonbashi-Hakozakicho, Chuo-ku Tokyo 103-8510, Japan

Následující odstavec se netýká Velké Británie nebo kterékoliv jiné země, kde taková opatření odporují místním zákonům: SPOLEČNOST INTERNATIONAL BUSINESS MACHINES CORPORATION TUTO PUBLIKACI POSKYTUJE TAKOVOU, "JAKÁ JE", BEZ JAKÝCHKOLIV ZÁRUK, VYJÁDŘENÝCH VÝSLOVNĚ NEBO VYPLÝVAJÍCÍCH Z OKOLNOSTÍ, VČETNĚ, A TO ZEJMÉNA, ZÁRUK NEPORUŠENÍ PRÁV TŘETÍCH STRAN, PRODEJNOSTI NEBO VHODNOSTI PRO URČITÝ ÚČEL VYPLÝVAJÍCÍCH Z OKOLNOSTÍ. Některé právní řády u určitých transakcí nepřipouštějí vyloučení záruk výslovně vyjádřených nebo vyplývajících z okolností, a proto se na vás toto omezení nemusí vztahovat.

Uvedené údaje mohou obsahovat technické nepřesnosti nebo typografické chyby. Údaje zde uvedené jsou pravidelně upravovány a tyto změny budou zahrnuty v nových vydáních této publikace. Společnost IBM může kdykoli bez upozornění provádět vylepšení nebo změny v produktech či programech popsaných v této publikaci.

Jakékoliv odkazy v této publikaci na webové stránky jiných společností než společnosti IBM jsou poskytovány pouze pro pohodlí uživatele a nemohou být žádným způsobem vykládány jako doporučení těchto webových stránek. Materiály uvedené na tomto webu nejsou součástí materiálů pro tento produkt IBM a použití uvedených stránek je pouze na vlastní nebezpečí.

Společnost IBM může použít nebo distribuovat jakékoli informace, které jí sdělíte, libovolným způsobem, který společnost považuje za odpovídající, bez vyžádání vašeho svolení.

Vlastníci licence k tomuto programu, kteří chtějí získat informace o možnostech (i) výměny informací s nezávisle vytvořenými programy a jinými programy (včetně tohoto) a (ii) oboustranného využití vyměňovaných informací, mohou kontaktovat informační středisko na adrese:

IBM Corporation Software Interoperability Coordinator, Department 49XA 3605 Highway 52 N Rochester, MN 55901 U.S.A.

Poskytnutí takových informací může být podmíněno dodržením určitých podmínek a požadavků zahrnujících v některých případech uhrazení stanoveného poplatku.

IBM poskytuje licencovaný program popsaný v těchto informacích a veškeré dostupné licencované materiály na základě podmínek smlouvy IBM Customer Agreement, IBM International Program License Agreement nebo jiné ekvivalentní smlouvy mezi námi.

Jakékoli údaje o výkonnosti obsažené v této publikaci byly zjištěny v řízeném prostředí. Výsledky získané v jakémkoli jiném operačním prostředí se proto mohou výrazně lišit. Některá měření mohla být prováděna

na vývojových verzích systémů a není zaručeno, že tato měření budou stejná i na běžně dostupných systémech. Některá měření mohla být navíc odhadnuta pomocí extrapolace. Skutečné výsledky mohou být jiné. Čtenáři tohoto dokumentu by měli zjistit použitelné údaje pro své specifické prostředí.

Informace ohledně produktů jiných společností než IBM byly získány od dodavatelů těchto produktů, z jimi publikovaných prohlášení či jiných veřejně přístupných zdrojů. Společnost IBM neprováděla testování těchto produktů a nemůže potvrdit přesnost informací ohledně jejich výkonnosti a kompatibility či správnost jiných prohlášení týkajících se produktů jiných společností než IBM. Další případné otázky ve věci produktů, které nejsou dodávány společností IBM, směřujte přímo na dodavatele těchto produktů.

Všechna prohlášení týkající se budoucího směru vývoje nebo záměrů IBM se mohou změnit nebo mohou být zrušena bez předchozího upozornění a představují pouze cíle a záměry.

Tyto údaje obsahují příklady dat a sestav používaných v běžných obchodních operacích. Aby byla představa úplná, používají se v příkladech jména osob a názvy společností, značek a produktů. Všechna tato jména a názvy jsou fiktivní a jejich podobnost se jmény, názvy a adresami používanými ve skutečnosti je zcela náhodná.

LICENČNÍ INFORMACE:

Tyto informace obsahují ukázkové aplikační programy ve zdrojovém jazyce ilustrující programovací techniky na různých operačních platformách. Tyto ukázkové programy můžete bez závazků vůči společnosti IBM jakýmkoli způsobem kopírovat, měnit a distribuovat za účelem vývoje, používání, odbytu či distribuce aplikačních programů odpovídajících rozhraní API pro operační platformu, pro kterou byly ukázkové programy napsány. Tyto příklady nebyly plně testovány za všech podmínek. Společnost IBM proto nemůže zaručit spolehlivost, upotřebitelnost nebo funkčnost těchto programů.

Při prohlížení těchto dokumentů v elektronické podobě se nemusí zobrazit všechny fotografie a barevné ilustrace.

Informace o programovacím rozhraní

Informace programátorských rozhraní, je-li poskytnuta, vám pomohou vytvořit aplikační software pro použití s tímto programem.

Tato příručka obsahuje informace o zamýšlených programovacích rozhraních, které umožňují zákazníkům psát programy za účelem získání služeb produktu IBM WebSphere MQ.

Tyto informace však mohou obsahovat i diagnostické údaje a informace o úpravách a ladění. Informace o diagnostice, úpravách a vyladění jsou poskytovány jako podpora ladění softwarových aplikací.

Důležité: Nepoužívejte tyto informace o diagnostice, úpravách a ladění jako programátorské rozhraní, protože se mohou měnit.

Ochranné známky

IBM, logo IBM, ibm.com jsou ochranné známky společnosti IBM Corporation, registrované v mnoha jurisdikcích po celém světě. Aktuální seznam ochranných známek IBM je k dispozici na webu na stránce "Copyright and trademark information" www.ibm.com/legal/copytrade.shtml. Ostatní názvy produktů a služeb mohou být ochrannými známkami společnosti IBM nebo jiných společností.

Microsoft a Windows jsou ochranné známky společnosti Microsoft Corporation ve Spojených státech a případně v dalších jiných zemích.

UNIX je registrovaná ochranná známka skupiny The Open Group ve Spojených státech a případně v dalších jiných zemích.

Linux je registrovaná ochranná známka Linuse Torvaldse ve Spojených státech a případně v dalších jiných zemích.

Tento produkt obsahuje software vyvinutý v rámci projektu Eclipse Project (<http://www.eclipse.org/>).

Java a všechny ochranné známky a loga založené na termínu Java jsou ochranné známky nebo registrované ochranné známky společnosti Oracle anebo příbuzných společností.



Číslo položky:

(1P) P/N: