9.0

IBM MQ Virtual System Pattern Type

Note

Before using this information and the product it supports, read the information in “Notices” on page 47.

This edition applies to version 9 release 0 of IBM® MQ and to all subsequent releases and modifications until otherwise indicated in new editions.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 2007, 2025.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

IBM MQ Virtual System Pattern Type....5
Overview....7
Restrictions....9
Planning....9
Choosing a lifecycle option for a pattern instance....10
Installing the pattern type....11
Verifying the installation....12
Upgrading the pattern type....13
Software components....14
Configuring....21
Creating patterns....21
Adding MQSC commands to a pattern....29
Deploying patterns....31
Security....32
Verifying a deployed MQ component from a terminal emulator....32
Managing virtual system instances....35
Updating a deployed pattern instance....35
Applying fix packs or interim fixes....36
Installing an IBM MQ interim fix....36
Running IBM MQ operations....38
Migrating....40
Migrating from Pattern Type version 1 to 2....40
Troubleshooting and support....41
Collecting error logs....42
Viewing error logs....44
Reference....44
Script packages....44

Notices....47
Programming interface information....48
Trademarks....48

LTS Linux AIX Overview of IBM MQ Virtual System Pattern Type

IBM MQ Advanced Virtual System Pattern Type for IBM PureApplication® System (referred to in this documentation as IBM MQ Virtual System Pattern Type) contains an IBM MQ plug-in (the virtual system software component) and associated script packages, which you can use to create a virtual system pattern for your IBM MQ environment.

IBM MQ Virtual System Pattern Type is provided as a `.tgz` file, which is named for example `patterntype.com.ibm.vsys.mq-2.0.0.2.tgz`.

IBM MQ Virtual System Pattern Type versions

IBM MQ Virtual System Pattern Type is available for x86-64 Linux® and AIX® platforms:

- Linux

  The pattern type for Linux runs on IBM PureApplication System 2.0 or later and supports IBM MQ 9.0.0 Fix Pack 2 or later.
- AIX

  The pattern type for AIX runs on IBM PureApplication System 2.1.2 and supports IBM MQ 9.0.0 Fix Pack 2 or later.

Table 1. IBM MQ Virtual System Pattern Type versions for x86-64 Linux

Platform	IBM MQ Virtual System Pattern Type version	IBM MQ version	IBM PureApplication System environment
x86-64 Linux	2.0.0.2	9.0.0.0	IBM PureApplication System W2500
		9.0.0.2	

Table 2. IBM MQ Virtual System Pattern Type versions for PowerAIX

Platform	IBM MQ Virtual System Pattern Type version	IBM MQ version	IBM PureApplication System environment
PowerAIX	2.0.0.2	9.0.0.0	IBM PureApplication System W2700
		9.0.0.2	

IBM PureApplication System

IBM PureApplication System helps you to manage virtual applications in a cloud-computing environment, in which data and services are in data centers. PureApplication System can be placed in a data center to dispense applications and topologies into a pool or cloud of virtualized hardware, and to manage these resources. The data and services can then be accessed from any connected devices over the internet. By using IBM PureApplication System, you can access the resources in your cloud and you can manage multiple environments from a single system and remote interface.

IBM PureApplication System installs and configures your software and manages the application runtime by using policies that you define.

For more information about IBM PureApplication System, see:

- Linux The IBM PureApplication System W2500 Documentation.
- AIX The IBM PureApplication System W2700 Documentation.

© Copyright IBM Corp. 2007, 2025

Virtual system patterns

Virtual system patterns enable efficient and repeatable deployments of systems that include one or more virtual machine instances, and the applications that run on them. You can completely automate the deployment and eliminate the need to perform multiple time-consuming manual tasks.

IBM MQ Virtual System Pattern Type contains the IBM MQ software component. You can install the IBM MQ software component onto a system image as part of a virtual system deployment, and can then be managed by the PureApplication System pattern engine.

Included in the virtual system pattern are a number of script packages (compressed files in `.zip` format) that can be used for automation of the Product Lifecycle Management. Script packages can be run when the pattern is deployed as a virtual system, when the virtual system is deleted, or whenever you choose to run the scripts manually.

You can add these artifacts to a blank template to build a customized virtual system pattern for your IBM MQ environment.

For more information about the purpose and usage of virtual patterns, see:

-

Documentation for PureApplication System W2500:
  -

Working with virtual patterns version 2.2.0
  -

Working with virtual patterns version 2.2.1
  -

Working with virtual patterns version 2.2.2
  -

Working with virtual patterns version 2.2.3
  -

Working with virtual patterns version 2.2.4
  -

Working with virtual patterns version 2.2.5
-

Documentation for PureApplication System W2700:
  -

Working with virtual patterns version 2.2.0
  -

Working with virtual patterns version 2.2.1
  -

Working with virtual patterns version 2.2.2
  -

Working with virtual patterns version 2.2.3
  -

Working with virtual patterns version 2.2.4
  -

Working with virtual patterns version 2.2.5

Virtual system patterns in IBM PureApplication System 2.0

"Classic" virtual system patterns are based on hypervisor images. To avoid various problems, including the software (such as IBM MQ) being so closely bound to a hypervisor image, virtual system patterns in IBM PureApplication System 2.0 separate the operating system from the middleware and allow dynamic composition of middleware on top of a specified base operating system image.

For more information about the difference between "classic" patterns that use a hypervisor and virtual system patterns that separate the operating system from the middleware, see the IBM Cloud® (formerly Bluemix) article What's new for virtual system patterns in Pure Application System 2.0.

Related concepts

"Configuring" on page 21

Tasks to help you configure the deployment of IBM MQ Virtual System Pattern Type to IBM PureApplication.

Related tasks

"Installing IBM MQ Virtual System Pattern Type on IBM PureApplication" on page 11
Obtain the IBM MQ Virtual System Pattern Type and upload it to your PureApplication System to install it.

Related information

Overview of IBM PureSystems Redbooks publication

Preparing for IBM PureApplication System: a five part series

LTS Linux AIX Overview of IBM MQ Virtual System Pattern Type

IBM MQ Advanced Virtual System Pattern Type for IBM PureApplication System (referred to in this documentation as IBM MQ Virtual System Pattern Type) contains an IBM MQ plug-in (the virtual system software component) and associated script packages, which you can use to create a virtual system pattern for your IBM MQ environment.

IBM MQ Virtual System Pattern Type is provided as a `.tgz` file, which is named for example `patterntype.com.ibm.vsys.mq-2.0.0.2.tgz`.

IBM MQ Virtual System Pattern Type versions

IBM MQ Virtual System Pattern Type is available for x86-64 Linux and AIX platforms:

- Linux

  The pattern type for Linux runs on IBM PureApplication System 2.0 or later and supports IBM MQ 9.0.0 Fix Pack 2 or later.
- AIX

  The pattern type for AIX runs on IBM PureApplication System 2.1.2 and supports IBM MQ 9.0.0 Fix Pack 2 or later.

Table 3. IBM MQ Virtual System Pattern Type versions for x86-64 Linux

Platform	IBM MQ Virtual System Pattern Type version	IBM MQ version	IBM PureApplication System environment
x86-64 Linux	2.0.0.2	9.0.0.0	IBM PureApplication System W2500
		9.0.0.2	

Table 4. IBM MQ Virtual System Pattern Type versions for PowerAIX

Platform	IBM MQ Virtual System Pattern Type version	IBM MQ version	IBM PureApplication System environment
PowerAIX	2.0.0.2	9.0.0.0	IBM PureApplication System W2700
		9.0.0.2	

IBM PureApplication System

IBM PureApplication System helps you to manage virtual applications in a cloud-computing environment, in which data and services are in data centers. PureApplication System can be placed in a data center to dispense applications and topologies into a pool or cloud of virtualized hardware, and to manage these resources. The data and services can then be accessed from any connected devices over the internet.

By using IBM PureApplication System, you can access the resources in your cloud and you can manage multiple environments from a single system and remote interface.

IBM PureApplication System installs and configures your software and manages the application runtime by using policies that you define.

For more information about IBM PureApplication System, see:

- Linux The IBM PureApplication System W2500 Documentation.
- AIX The IBM PureApplication System W2700 Documentation.

Virtual system patterns

Virtual system patterns enable efficient and repeatable deployments of systems that include one or more virtual machine instances, and the applications that run on them. You can completely automate the deployment and eliminate the need to perform multiple time-consuming manual tasks.

IBM MQ Virtual System Pattern Type contains the IBM MQ software component. You can install the IBM MQ software component onto a system image as part of a virtual system deployment, and can then be managed by the PureApplication System pattern engine.

Included in the virtual system pattern are a number of script packages (compressed files in `.zip` format) that can be used for automation of the Product Lifecycle Management. Script packages can be run when the pattern is deployed as a virtual system, when the virtual system is deleted, or whenever you choose to run the scripts manually.

You can add these artifacts to a blank template to build a customized virtual system pattern for your IBM MQ environment.

For more information about the purpose and usage of virtual patterns, see:

- Linux Documentation for PureApplication System W2500:
  - Linux Working with virtual patterns version 2.2.0
  - Linux Working with virtual patterns version 2.2.1
  - Linux Working with virtual patterns version 2.2.2
  - Linux Working with virtual patterns version 2.2.3
  - Linux Working with virtual patterns version 2.2.4
  - Linux Working with virtual patterns version 2.2.5
- AIX Documentation for PureApplication System W2700:
  - AIX Working with virtual patterns version 2.2.0
  - AIX Working with virtual patterns version 2.2.1
  - AIX Working with virtual patterns version 2.2.2
  - AIX Working with virtual patterns version 2.2.3
  - AIX Working with virtual patterns version 2.2.4
  - AIX Working with virtual patterns version 2.2.5

Virtual system patterns in IBM PureApplication System 2.0

"Classic" virtual system patterns are based on hypervisor images. To avoid various problems, including the software (such as IBM MQ) being so closely bound to a hypervisor image, virtual system patterns in

IBM PureApplication System 2.0 separate the operating system from the middleware and allow dynamic composition of middleware on top of a specified base operating system image.

For more information about the difference between "classic" patterns that use a hypervisor and virtual system patterns that separate the operating system from the middleware, see the IBM Cloud (formerly Bluemix) article What's new for virtual system patterns in Pure Application System 2.0.

Related concepts

“Configuring” on page 21
Tasks to help you configure the deployment of IBM MQ Virtual System Pattern Type to IBM PureApplication.

Related tasks

“Installing IBM MQ Virtual System Pattern Type on IBM PureApplication” on page 11
Obtain the IBM MQ Virtual System Pattern Type and upload it to your PureApplication System to install it.

Related information

Overview of IBM PureSystems Redbooks publication

Preparing for IBM PureApplication System: a five part series

LTS Linux AIX Restrictions for IBM MQ Virtual System Pattern Type

Some assets in IBM MQ Virtual System Pattern Type have restrictions on how you can use them when you build your pattern.

Software component restrictions

IBM MQ software components have the following restrictions:

- Linux At IBM MQ 9.0.0 Fix Pack 2, you can install IBM MQ software components on x86-64 Linux operating systems. Therefore, you should be using IBM PureApplication System W2500. For minimum operating system information for IBM MQ 9.0.0 Fix Pack 2 on Linux, see IBM MQ 9.0.0 Fix Pack 2 Linux system requirements.
- AIX At IBM MQ 9.0.0 Fix Pack 2, you can also install IBM MQ software components on AIX operating systems. Therefore, you should be using IBM PureApplication System W2700. For minimum operating system information for IBM MQ 9.0.0 Fix Pack 2 on AIX, see IBM MQ 9.0.0 Fix Pack 2 AIX system requirements.
- Multiple instances of IBM MQ cannot be installed on a single virtual machine at the same time. The first component on a virtual machine, installs IBM MQ to the default location and, optionally, creates a queue manager. On subsequent occasions, the installed instance of IBM MQ is updated if you try to add a component that contains a later version, otherwise no change is made. Optionally, another queue manager is created.

Related concepts

“IBM MQ Virtual System Pattern Type software components” on page 14

LTS Linux AIX Planning

Consider how you want to use your pattern instances before you deploy them. Find out how to combine the options in the pattern builder with the settings you can specify when you are deploying and managing instances to create an IBM MQ environment that is best suited to your needs.

About this task

You might consider these questions:

Procedure

- Do you want to keep your data after a deployed pattern instance has been deleted?
- Do you want to migrate your pattern instance data after deployment?
- Do you need to reuse data from an existing deployed pattern instance to deploy a new instance?

Results

For detailed information on available options for configuring your patterns and deployed instances, see the subtopic:

Choosing a lifecycle option for a pattern instance

You can choose to manage your IBM MQ log and queue manager data independently from the deployed pattern instance.

About this task

When you create patterns and deploy pattern instances, you can plan the lifecycle of your deployed instance, data management, migration, and reuse of patterns. The following two configuration options are provided to help you to customize your IBM MQ pattern creation and instance deployment:

Manage IBM MQ data manually configuration option
: You can specify a location for your IBM MQ data directory, different from the default one, to prevent your IBM MQ data from being deleted when the deployed instance is deleted. If you do not select the option to manage the IBM MQ data directory manually, the **`deployment id`** is used in the IBM MQ data path at the time of instance creation. When the instance is deleted, the IBM MQ data is deleted too.

Use existing IBM MQ queue manager data directory configuration option
: You can deploy an instance from a pattern and use data that exists from a previously deleted or stopped instance to re-create and resume the IBM MQ queue manager.

Both the **Manage IBM MQ data manually** and the **Use existing IBM MQ queue manager data directory** options are available with `single instance`, `high availability active instance`, and `high availability standby instance` pattern configurations.

The use of the two new configuration options is described in the following instance lifecycle settings:

Procedure

- `Default`

  If you do not select the **Manage IBM MQ data manually** check box when the pattern is created, the **`deployment id`** is included in the IBM MQ log and data path. When a pattern instance that is deployed from this pattern is deleted, the IBM MQ queue manager logs and data that are associated with the pattern instance, are removed. You can use this pattern configuration to deploy long running pattern instances when you want to manage IBM MQ versions through the IBM MQ and IBM PureApplication System maintenance framework. IBM MQ networks with common configurations that are pre-defined in this way, can be deployed quickly. You cannot migrate data or modify a pattern instance that is created with `default` lifecycle settings, to `manual` or `resume` lifecycle settings.

- `Manual`

  Select the **Manage IBM MQ data manually** check box when you create the pattern. The **`deployment id`** is not included in the IBM MQ log and data path, so when a pattern instance that is deployed from this pattern is deleted, the IBM MQ queue manager logs and data that is associated with the pattern instance, remain in the location that you specify. The different location that you choose manually must be on the shared General Parallel File System server. You can use this pattern configuration when the IBM MQ log and data directory path locations must be defined before the pattern instance is deployed, for example, when the directories are registered to an external monitoring service, and when the IBM MQ log and data directories must not be deleted with the pattern instance. If you

know that you might want to use the IBM MQ log and queue manager data from an instance you are deploying now, in subsequent deployments, you must use the **Manage IBM MQ data manually** option. This is a prerequisite for selecting the **Use existing IBM MQ queue manager data directory** option in subsequent deployments.

You are responsible for avoiding IBM MQ queue manager naming collisions in GPFS shared locations. However, if you try to reuse a name of an existing IBM MQ queue manager data directory location for a new instance, the **crtmqm** command transforms the final directory location to establish a unique path. You cannot migrate data or modify a pattern instance that is created with `manual` lifecycle settings, to `default` lifecycle settings.

Note: Consider the impact that any scripts that you create might have on initial and subsequent pattern instance deployments. For example, adding a simple script to create a queue during instance deployment might result in an error in the case where a subsequent deployment uses existing data to resume the queue manager and the queue is already defined. Either modify or remove the script to avoid the error.

- `Resume`

  Select the **Manage IBM MQ data manually** check box and use the **Use existing IBM MQ queue manager data directory** field. You can use this configuration when you want to re-create and resume an IBM MQ queue manager by using data that exists from a previous instance.

  Note: The previous instance must be created with the **Use existing IBM MQ queue manager data directory** field left empty. If the field is not empty, it is evaluated, along with the data path or GPFS shared location, during the deployment to form the final directory for existing IBM MQ queue manager data, for example `/data_path/qmgrs/existing MQ qm data directory`. If the directory location is not found, the deployment fails.

  This field is only valid for deployments that use GPFS shared storage for the IBM MQ queue manager log and data location because directory paths on storage that are tightly bound to the host virtual machine, are removed when the previous pattern instance is deleted. This field is set to the previous IBM MQ queue manager name, for example QM1. However, where multiple IBM MQ queue managers with the same name share a path, the final directory is transformed to establish a unique path, for example QM1.000.

LTS Linux AIX Installing IBM MQ Virtual System Pattern Type on IBM PureApplication

Obtain the IBM MQ Virtual System Pattern Type and upload it to your PureApplication System to install it.

Before you begin

You must be assigned the "Create new catalog content" role or the "Workload resources administration" role with full permissions.

About this task

The IBM MQ Virtual System Pattern Type pattern type is available as a download from IBM Passport Advantage®. The pattern type version 2.0.0.2 contains IBM MQ 9.0.0 Fix Pack 2.

For more information on which IBM MQ version is available with which IBM MQ pattern type version, see “Overview of IBM MQ Virtual System Pattern Type” on page 5.

To load the pattern type to your PureApplication System, import the IBM MQ Virtual System Pattern Type. After loading, provide access for users to work with the patterns. Authorized users can create, deploy, and manage IBM MQ environments in a private cloud environment. You can also use pattern templates to create patterns.

Procedure

1. Obtain the IBM MQ Virtual System Pattern Type version that you want to use.

   When you download the pattern type from IBM Passport Advantage, you must decompress the `.tar.gz` archive to get the `.tgz` file.

   For example, download the `IBM_MQ_ADV_9.0_PURE_LINUX.tar.gz` for Linux or `IBM_MQ_ADV_9.0_PURE_AIX.tar.gz` file for AIX, to obtain the `IBM MQ Virtual System Pattern Type version 2.0.0.2` that contains IBM MQ 9.0.0 Fix Pack 2. You must decompress the downloaded archive file to get the `patterntype.com.ibm.vsys.mq-2.0.0.2.tgz` file, which you need to upload to the PureApplication System to begin installing the pattern type.

   The `.tgz` file is in the `pureapp` directory when you decompress the downloaded file.
2. Upload the pattern type to your PureApplication System.
3. Access the **Pattern Types** menu.
   - In PureApplication 2.0, click **Cloud** > **Pattern Types** from the Workload Console.
   - In PureApplication 2.1, click **Catalog** > **Pattern Types**.
4. To install the IBM MQ Virtual System Pattern Type, click **New**.
5. Upload either a local or a remote file.
   - To upload a local file, click **Browse** on the **Local** tab and select the `.tgz` file that contains the pattern type, for example `patterntype.com.ibm.vsys.mq-2.0.0.2.tgz`.
   - To upload a remote file, specify the URL of the file on the **Remote** tab. If prompted to log in to the remote site to access the file, specify the user name and password.

   Click **OK**.
6. Accept the license agreement.
   a) Click **IBM MQ Advanced Virtual System Pattern Type** and select the most current fix pack.
   b) View the license agreement and accept it.
   c) Enable the pattern by clicking **Status** > **Enable**.

What to do next

Verify that the pattern is installed and available for use.

LTS Linux AIX Verifying IBM MQ Virtual System Pattern Type installation

After installing IBM MQ Virtual System Pattern Type, you can verify that the pattern uploaded correctly to the system, the license agreement is accepted, and the status is available.

Before you begin

Complete the installation steps for IBM MQ Virtual System Pattern Type. See “Installing IBM MQ Virtual System Pattern Type on IBM PureApplication” on page 11.

Procedure

1. Access the **Pattern Types** menu.
   - In PureApplication 2.0, click **Cloud** > **Pattern Types** from the Workload Console.
   - In PureApplication 2.1, click **Catalog** > **Pattern Types**.
2. In the **Pattern Types** list under IBM MQ Advanced Virtual System Pattern Type, select the most current fix pack.
3. View the License Agreement section and verify that the license for the pattern is accepted.
4. View the Status section and verify that the pattern type is in the Available state.

5. Optional: To view a list of plug-ins that comprise the application server pattern, click **Show me all plug-ins in this pattern type**.

   Currently, one plug-in is shown: `plugin.com.ibm.vsys.mq`.

What to do next

You can now work with the verified pattern installation.

LTS Linux AIX Updating IBM MQ Virtual System Pattern Type on IBM PureApplication

The supplied DVD contains the PureApplication `.tgz` file, or you can download it from the product download page.

Before you begin

You must be assigned the "Create new catalog content" role or the "Workload resources administration" role with full permissions.

About this task

The IBM MQ Virtual System Pattern Type is available as an eImage download from IBM Passport Advantage. the download is a `.tgz` file (named `patterntype.com.ibm.vsys.mq-2.0.0.2.tgz`) packaged in a compressed `.tar.gz` archive (named `IBM_MQ_ADV_9.0_PURE_LINUX.tar.gz`).

To load the pattern to PureApplication, import the IBM MQ Virtual System Pattern Type. After loading, provide access for users to work with the patterns. Authorized users can create, deploy, and manage IBM MQ environments in a private cloud environment. You can also use pattern templates to create patterns.

Procedure

1. Download the package to your local machine.
2. Decompress the `.tar.gz` archive (`IBM_MQ_ADV_9.0_PURE_LINUX.tar.gz`) to get to the `.tgz` file (`patterntype.com.ibm.vsys.mq-2.0.0.2.tgz`).
3. Access the **Pattern Types** menu.
   - In PureApplication 2.0, click **Cloud** > **Pattern Types** from the Workload Console.
   - In PureApplication 2.1, click **Catalog** > **Pattern Types**.
4. To install the pattern type IBM MQ Virtual System Pattern Type, click **New**.
5. Upload either a local or a remote file.
   - To upload a local file, click **Browse** on the **Local** tab and select the `.tgz` file that contains the pattern type (`patterntype.com.ibm.vsys.mq-2.0.0.2.tgz`).
   - To upload a remote file, specify the URL of the file on the **Remote** tab. If prompted to log in to the remote site to access the file, specify the user name and password.

   Click **OK**.
6. Accept the license agreement.
   a) Click **IBM MQ Advanced Virtual System Pattern Type** and select the most current fix pack.
   b) View the License Agreement and accept it.
   c) To enable the pattern, click **Status** > **Enable**.

What to do next

Verify that the pattern is installed and available for use.

Related information

Downloading IBM MQ 8.0 from the Passport Advantage web site

Passport Advantage and Passport Advantage Express web site

LTS Linux AIX IBM MQ Virtual System Pattern Type software components

Software components provide function in your patterns. You can configure components to create an IBM MQ environment that is fully operational when you deploy your pattern in the cloud. You can change the properties when you configure or deploy a pattern, and by running a script.

Component configuration parameters

The following table contains component configuration parameters for IBM MQ Virtual System Pattern Type:

Table 5. Component configuration parameters.

Label	Description
Type of IBM MQ configuration	Determines what type of queue manager is created after IBM MQ has been installed on the system. `Installation only`: no queue manager is created and all subsequent parameters are ignored. `Single instance`: a single instance queue manager is created, using the following parameters. `High availability active instance`: a multi-instance active queue manager is created, using the following parameters. `High availability standby instance`: a multi-instance standby queue manager is created, using the following parameters.
Queue manager name	The name of the queue manager. This is required if Type of IBM MQ configuration is not `Installation only`. To avoid confusion, make this name unique in your network.
Queue manager description	A description of the queue manager.
Listener port	The TCP/IP listener port number. This is required if Type of IBM MQ configuration is not `Installation only`.
Dead letter queue	The queue manager dead-letter queue.
Queue manager uses linear logging?	Determines whether the queue manager uses linear logging. If `false` is specified, circular logging is used.
Queue manager log pages	The number of queue manager log pages.
Primary logs	The number of primary IBM MQ log files.
Secondary Logs	The number of secondary IBM MQ log files.
Log path	The file system directory for log data.
Data path	The file system directory for queue manager data.
Error path	The file system directory for error data.
Shared directory	The shared directory in which data and log files are stored. The queue manager must have read and write authorization. This value must be set if Type of IBM MQ configuration is set to `High availability active instance` or `High availability standby instance`.

Table 5. Component configuration parameters. (continued)	
Label	**Description**
Queue manager directory	The name of the queue manager data directory. Sometimes it is different from the queue manager name. The rules for transforming queue manager names into directory names are described in Understanding IBM MQ file names. This value must be set if Type of IBM MQ configuration is set to `High availability active instance`.
Manage IBM MQ data manually	You can choose to manage your data manually. This option ensures that your IBM MQ data is not deleted when the deployed instance is deleted, provided that the location you choose for your data is away from the deployed pattern instance, for example, on the shared GPFS server. If the option to manage the IBM MQ data directory is not selected, the **`deployment id`** is used in the IBM MQ data path at the time of instance creation. When the instance is deleted, the IBM MQ data is deleted too.
Use existing IBM MQ queue manager data directory	You can deploy an instance from a pattern and use data that exists from a previous instance to re-create and resume the IBM MQ queue manager.
Do not install IBM MQ Advanced components	You can choose not to install IBM MQ Advanced components. You might not want the advanced functionality or do not have the required license. For information on which components are included with the IBM MQ Advanced option, see "IBM MQ Virtual System Pattern Type software components for Linux systems" on page 18 and "IBM MQ Virtual System Pattern Type software components for AIX systems" on page 15.

Related concepts

"Restrictions for IBM MQ Virtual System Pattern Type" on page 9
Some assets in IBM MQ Virtual System Pattern Type have restrictions on how you can use them when you build your pattern.

IBM MQ Virtual System Pattern Type software components for AIX systems

Software components and messages that are installed with the IBM MQ Virtual System Pattern Type for AIX systems.

Components and messages installed in the IBM MQ software component

On AIX each component of IBM MQ is represented by a fileset.

Table 6 on page 16 lists the filesets that are available when installing IBM MQ Virtual System Pattern Type for AIX systems.

Table 7 on page 17 lists the message catalogs for IBM MQ Virtual System Pattern Type for AIX systems.

Table 6. IBM MQ components for AIX systems.			
Component	**Description**	**Fileset name**	**IBM MQ or** **IBM MQ Advanced**
Runtime	Used for application development (server only) and provides support for external applications. **Note:** This component must be installed.	mqm.base.runtime	IBM MQ
Server	You can use the server to run queue managers on your system and connect to other systems over a network. Provides messaging and queuing services to applications, and support for IBM MQ client connections.	mqm.server.rte	IBM MQ
Standard Client	The IBM MQ MQI client is a small subset of IBM MQ, without a queue manager, that uses the queue manager and queues on other (server) systems. It can be used only when the system it is on is connected to another system that is running a full server version of IBM MQ. The client and the server can be on the same system if required.	mqm.client.rte	IBM MQ
SDK	The SDK is required for compiling applications. It includes sample source files, and the bindings (files .H, .LIB, .DLL, and others), that you need to develop applications to run on IBM MQ.	mqm.base.sdk	IBM MQ
Sample programs	The sample application programs are needed if you want to check your IBM MQ installation using the verification procedures.	mqm.base.samples	IBM MQ
Java messaging	The files needed for messaging using Java (includes Java Messaging Service).	mqm.java.rte	IBM MQ
Man pages	UNIX man pages, in US English, for: control commands MQI commands MQSC commands	mqm.man.en_US.data	IBM MQ
Java JRE	A Java Runtime Environment, that is used by those parts of IBM MQ that are written in Java.	mqm.jre.rte	IBM MQ
Message Catalogs	For available languages, see the table of message catalogs that follows.		IBM MQ
IBM Global Security Kit (GSKit)	GSKit 8 Certificate and SSL, or TLS, Base Runtime.	mqm.gskit.rte	IBM MQ
AMQP Service	The AMQP Service makes AMQP channels available. AMQP channels support MQ Light APIs. You can use AMQP channels to give AMQP applications access to the enterprise-level messaging facilities provided by IBM MQ.	mqm.amqp.rte	IBM MQ

Table 6. IBM MQ components for AIX systems. (continued)			
Component	**Description**	**Fileset name**	**IBM MQ or IBM MQ Advanced**
Telemetry Service	MQ Telemetry supports the connection of Internet Of Things (IOT) devices (that is, remote sensors, actuators and telemetry devices) that use the IBM MQ Telemetry Transport (MQTT) protocol. The telemetry service, which is also know as the MQ Extended Reach (MQXR) service, enables a queue manager to act as an MQTT server, and communicate with MQTT client apps. A set of MQTT client libraries is also available in the free download IBM Messaging Telemetry Clients SupportPac. These libraries help you write the MQTT client apps that IOT devices use to communicate with MQTT servers. See also Installing MQ Telemetry.	mqm.xr.service	IBM MQ Advanced
Managed File Transfer	Managed File Transfer transfers files between systems in a managed and auditable way, regardless of file size or the operating systems used. For information about the function of each component, see Managed File Transfer product options.	mqm.ft.agent mqm.ft.base mqm.ft.logger mqm.ft.service mqm.ft.tools	IBM MQ Advanced
Advanced Message Security	Provides a high level of protection for sensitive data flowing through the IBM MQ network, while not impacting the end applications. You must install this component on all IBM MQ installations that host queues you want to protect. You must install the IBM Global Security Kit (GSKit) component on any IBM MQ installation that is used by a program that puts or gets messages to or from a protected queue, unless you are using only Java client connections.	mqm.ams.rte	IBM MQ Advanced

Table 7. IBM MQ message catalogs for AIX systems	
Message catalog language	**Component name**
Brazilian Portuguese	mqm.msg.pt_BR
Czech	mqm.msg.cs_CZ
French	mqm.msg.fr_FR
German	mqm.msg.de_DE
Hungarian	mqm.msg.hu_HU
Italian	mqm.msg.it_IT

Table 7. IBM MQ message catalogs for AIX systems (continued)	
Message catalog language	**Component name**
Japanese	mqm.msg.ja_JP, mqm.msg.Ja_JP
Korean	mqm.msg.ko_KR
Polish	mqm.msg.pl_PL
Russian	mqm.msg.ru_RU
Spanish	mqm.msg.es_ES
Simplified Chinese	mqm.msg.zh_CN, mqm.msg.Zh.CN
Traditional Chinese	mqm.msg.zh_TW, mqm.msg.Zh_TW
U.S. English	mqm.msg.en_US

LTS Linux AIX IBM MQ Virtual System Pattern Type software components for Linux systems

Software components and messages that are installed with the IBM MQ Virtual System Pattern Type for Linux systems.

Components and messages installed in the IBM MQ software component

Table 8 on page 18 lists the IBM MQ components that are installed with IBM MQ Virtual System Pattern Type for Linux systems.

Table 9 on page 20 lists the message catalogs for IBM MQ Virtual System Pattern Type for Linux systems.

Table 8. IBM MQ components for Linux systems			
Component	**Description**	**Component name**	**IBM MQ or IBM MQ Advanced**
Runtime	Used for application development (server only) and provides support for external applications. **Note:** MQSeries®Runtime component must be installed.	MQSeriesRuntime	IBM MQ
Server	You can use the server to run queue managers on your system and connect to other systems over a network. Provides messaging and queuing services to applications, and support for IBM MQ client connections.	MQSeriesServer	IBM MQ
Standard Client	The IBM MQ MQI client is a small subset of IBM MQ, without a queue manager, that uses the queue manager and queues on other (server) systems. It can be used only when the system it is on is connected to another system that is running a full server version of IBM MQ. The client and the server can be on the same system if required.	MQSeriesClient	IBM MQ

Table 8. IBM MQ components for Linux systems (continued)			
Component	**Description**	**Component name**	**IBM MQ or** **IBM MQ Advanced**
SDK	The SDK is required for compiling applications. It includes sample source files, and the bindings (files .H, .LIB, .DLL, and others), that you need to develop applications to run on IBM MQ.	MQSeriesSDK	IBM MQ
Sample programs	The sample application programs are needed if you want to check your IBM MQ installation using the verification procedures.	MQSeriesSamples	IBM MQ
Java messaging	The files needed for messaging using Java (includes Java Messaging Service).	MQSeriesJava	IBM MQ
Man pages	UNIX man pages, in US English, for: control commands MQI commands MQSC commands	MQSeriesMan	IBM MQ
Java JRE	A Java Runtime Environment, version 6.0, that is used by those parts of IBM MQ that are written in Java.	MQSeriesJRE	IBM MQ
Message Catalogs	For available languages, see the table of message catalogs that follows.		IBM MQ
IBM Global Security Kit (GSKit)	GSKit 8 Certificate and SSL Base Runtime.	MQSeriesGSKit	IBM MQ
IBM MQ Explorer	Use the IBM MQ Explorer component of IBM MQ to administer and monitor resources on Linux x86 and x86-64 systems.	MQSeriesExplorer	IBM MQ
AMQP Service	The AMQP Service makes AMQP channels available. AMQP channels support MQ Light APIs. You can use AMQP channels to give AMQP applications access to the enterprise-level messaging facilities provided by IBM MQ.	MQSeriesAMQP	IBM MQ
Telemetry Clients	(64-bit Linux only)	MQSeriesXRClients	IBM MQ Advanced
Telemetry Service	(64-bit Linux only)	MQSeriesXRService	IBM MQ Advanced

Table 8. IBM MQ components for Linux systems (continued)			
Component	**Description**	**Component name**	**IBM MQ or IBM MQ Advanced**
Managed File Transfer	Managed File Transfer transfers files between systems in a managed and auditable way, regardless of file size or the operating systems used. For information about the function of each component, see Managed File Transfer product options.	MQSeriesFTAgent MQSeriesFTBase MQSeriesFTLogger MQSeriesFTService MQSeriesFTTools	IBM MQ Advanced
Advanced Message Security	Provides a high level of protection for sensitive data flowing through the IBM MQ network, while not impacting the end applications. You must install this component on all IBM MQ installations that host queues you want to protect. You must install the IBM Global Security Kit (GSKit) component on any IBM MQ installation that is used by a program that puts or gets messages to or from a protected queue, unless you are using only Java client connections.	MQSeriesAMS	IBM MQ Advanced

Table 9. IBM MQ message catalogs for Linux systems	
Message catalog language	**Component name**
Brazilian Portuguese	MQSeriesMsg_pt
Czech	MQSeriesMsg_cs
French	MQSeriesMsg_fr
German	MQSeriesMsg_de
Hungarian	MQSeriesMsg_hu
Italian	MQSeriesMsg_it
Japanese	MQSeriesMsg_ja
Korean	MQSeriesMsg_ko
Polish	MQSeriesMsg_pl
Russian	MQSeriesMsg_ru
Spanish	MQSeriesMsg_es
Simplified Chinese	MQSeriesMsg_Zh_CN
Traditional Chinese	MQSeriesMsg_Zh_TW
US English	not applicable

LTS Linux AIX Configuring

Tasks to help you configure the deployment of IBM MQ Virtual System Pattern Type to IBM PureApplication.

These tasks will assist you with the installation and deployment of virtual machines and virtual system patterns that include IBM MQ. To configure IBM MQ, see Configuring IBM MQ.

Related concepts

"Security" on page 32

"Managing virtual system instances" on page 35
When you deploy a virtual system pattern into the cloud, the resulting virtual system instance is a working IBM MQ environment. The virtual system instance contains running virtual machines that are created from software components in the pattern.

"Troubleshooting and support" on page 41
Learn how to troubleshoot a problem with your IBM software.

Related tasks

"Installing IBM MQ Virtual System Pattern Type on IBM PureApplication" on page 11
Obtain the IBM MQ Virtual System Pattern Type and upload it to your PureApplication System to install it.

LTS Linux AIX Creating IBM MQ Virtual System Pattern Type patterns

Use the pattern editor to create your own pattern or create a pattern from a template.

Before you begin

You must have access to patterns, access to create patterns, or have Workload resource administration with full permissions to complete this task.

About this task

To deploy IBM MQ, add the IBM MQ software component to a virtual system image. The IBM MQ software component has a number of parameters, described in "IBM MQ Virtual System Pattern Type software components" on page 14.

Procedure

- To create your own pattern using an undefined, blank template, see "Creating IBM MQ Virtual System Pattern Type patterns from blank templates" on page 21.

LTS Linux AIX Creating IBM MQ Virtual System Pattern Type patterns from blank templates

You can use the pattern editor to create your own patterns from blank templates.

Before you begin

You must have access to patterns, access to create patterns, or the Workload resources administration with full permissions.

About this task

Using the pattern editor, you can create your own patterns from a blank template. IBM MQ software components have a number of parameters, described in "IBM MQ Virtual System Pattern Type software components" on page 14.

Procedure

1. Create a new pattern.
   - In PureApplication 2.0, click **Patterns** > **Virtual Systems** > **Create New**.
   - In PureApplication 2.1, click **Patterns** > **Virtual Systems Patterns** > **Create New**.
2. Enter the pattern name and version.

   The default is Version 1.0.
3. From the list of pattern templates, choose the **blank template**. Click **Start Building**.

   The pattern builder launches and an empty canvas is displayed.
4. Select an IBM MQ part from the list of software components and drag the part to the canvas.

   Adding the part to the canvas creates a virtual machine that contains the component.
5. Configure the part.
   a) Enter the name for the part.

      Use a unique name for the operating system part for easier identification during deployment. For example, change the name from the default name (`OS Node`) to `MQNode`. You must create the `root` and `virtuser` passwords for the virtual machine operating system.

   b) Select the number of virtual CPUs to assign to this virtual machine.

   c) Choose whether this part should create a queue manager on the virtual machine.

      If you choose to create a queue manager (selected by default), you must provide a queue manager name and listener port number, for example, `MQ1` and 1414. You can also set other parameters to further configure the queue manager. See "IBM MQ Virtual System Pattern Type software components" on page 14. If you choose not to create a queue manager during this installation process, select **Installation only** option from the **Type of IBM MQ** configuration menu on the **IBM MQ Advanced configuration** panel.

   d)

      Check that you have enough space to install and upgrade IBM MQ.

      On Power/AIX, you might need to mount another disk for `/usr` to provide more space for installing and upgrading IBM MQ. For more information, see "Troubleshooting and support" on page 41.
6. Optional: Select the IBM MQ version to use.

   On the canvas, click the IBM MQ part and select the version that you need.
7. Optional: Add IBM MQ script packages, which run IBM MQ MQSC commands, to further configure your deployment.

   The following script packages are available:

   - `MQExecuteMQSC` runs multiple MQSC commands
   - `MQExecuteMQSCCommand` runs a single MQSC command
8. Optional: If you want to choose to manage IBM MQ data manually, on the canvas, click the IBM MQ part and on the **IBM MQ** attributes panel, select the **Manage IBM MQ data manually** check box.

   This option ensures that your IBM MQ data is not deleted when the deployed instance is deleted, provided that the location you specify for your data is away from the deployed pattern instance, for example, on the shared GPFS server. If the option to manage the IBM MQ data directory is not selected, the **`deployment id`** is used in the IBM MQ data path at the time of instance creation. When the instance is deleted, the IBM MQ data is deleted too.
9. Optional: If you want to choose to use existing IBM MQ queue manager data directory to deploy a subsequent instance from a pattern and resume the existing IBM MQ queue manager, on the canvas, click the IBM MQ part and on the **IBM MQ** attributes panel, select the **Use existing IBM MQ queue manager data directory** check box.

   The previously deployed pattern instance from which you are reusing data must be created with the **Manage IBM MQ data manually** option selected. For more information on deployment lifecycle options, see Choosing a lifecycle option for a pattern instance.
10. Optional: If you do not want to install IBM MQ Advanced components, on the canvas, click the IBM MQ part and on the IBM MQ attributes panel select the **Do not install IBM MQ Advanced** check box.

You might not want the advanced functionality or do not have the required license. For information on which components are included with the IBM MQ Advanced option, see IBM MQ Virtual System Pattern Type software components for Linux systems and IBM MQ Virtual System Pattern Type software components for AIX systems.

Related concepts

"IBM MQ Virtual System Pattern Type software components" on page 14

Related tasks

"Adding MQSC commands to a pattern" on page 29

Related reference

"IBM MQ Virtual System Pattern Type script packages" on page 44

LTS Linux AIX Creating IBM MQ Virtual System Pattern Type patterns from existing templates

You can use the pattern editor to create your own patterns based on existing templates.

Before you begin

- You must have access to patterns, access to create patterns, or the Workload resources administration with full permissions.
- Get General Parallel File System (GPFS) information and the name of the General Parallel File System fileset directory from your PureApplication administrator. For more information, see Administering support for GPFS.

About this task

IBM MQ software components have a number of parameters, described in "IBM MQ Virtual System Pattern Type software components" on page 14.

The following pattern templates are provided for IBM MQ Multi-Instance:

- `IBM MQ Multi-Instance` includes two OS nodes, each has one IBM MQ Advanced software component. One IBM MQ Advanced software component is configured to be the active instance, the other is configured to be a standby instance.
- `IBM MQ Multi-Instance with 2 pairs` includes two OS nodes, each has two IBM MQ Advanced software components. They are configured as two pairs of active and standby instances.

For more information about IBM MQ Multi-Instance, see Multi-instance queue managers.

Procedure

1. Create a new pattern.
   - In PureApplication 2.0, click **Patterns** > **Virtual Systems** > **Create New**.
   - In PureApplication 2.1, click **Patterns** > **Virtual Systems Patterns** > **Create New**.
2. Enter the pattern name and version.
   The default is Version 1.0.
3. From the list of pattern templates, select `IBM MQ Multi-Instance` or `IBM MQ Multi-Instance with 2 pairs` and click **Start Building**.
   The pattern builder opens and a canvas is displayed.
4. Configure the virtual machine.
   a) Enter passwords for users `root` and `virtuser` of node `OS Node A`.
   b) Enter passwords for users `root` and `virtuser` of node `OS Node B`.
5. Configure the GPFS client policy for the nodes `OS Node A` and `OS Node B`.

a) In the first file system mount section, which is selected by default, enter the GPFS information that you obtained from your PureApplication administrator.

b) Enter the GPFS name of the fileset directory that you obtained from your PureApplication administrator.

c) Enter a value for **Storage Maximum**. For example, 1G (1 gigabyte).

d) AIX

Check that you have enough space to install and upgrade IBM MQ.

On Power/AIX, you might need to mount another disk for `/usr` to provide more space for installing and upgrading IBM MQ. For more information, see "Troubleshooting and support" on page 41.

Each queue manager requires just over 100M of space when created but this storage requirement will increase with further configuration and runtime processing of messages. The same GPFS partition might be used in multiple deploys, so you need to allow enough storage for the workload you require.

6. Optional: Select the operating system version to use.

   On the canvas, click the operating system part and select the version that you want.

7. Optional: Select the IBM MQ version to use.

   On the canvas, click the IBM MQ part and select the version that you want.

8. Optional: If you want to choose to manage IBM MQ data manually, on the canvas, click the IBM MQ part and on the **IBM MQ** attributes panel, select the **Manage IBM MQ data manually** check box.

   This option ensures that your IBM MQ data is not deleted when the deployed instance is deleted, provided that the location you specify for your data is away from the deployed pattern instance, for example, on the shared GPFS server. If the option to manage the IBM MQ data directory is not selected, the **`deployment id`** is used in the IBM MQ data path at the time of instance creation. When the instance is deleted, the IBM MQ data is deleted too.

9. Optional: If you want to choose to use existing IBM MQ queue manager data directory to deploy a subsequent instance from a pattern and resume the existing IBM MQ queue manager, on the canvas, click the IBM MQ part and on the **IBM MQ** attributes panel, select the **Use existing IBM MQ queue manager data directory** check box.

   The previously deployed pattern instance from which you are reusing data must be created with the **Manage IBM MQ data manually** option selected. For more information on deployment lifecycle options, see Choosing a lifecycle option for a pattern instance.

10. Optional: If you do not want to install IBM MQ Advanced components, on the canvas, click the IBM MQ part and on the IBM MQ attributes panel select the **Do not install IBM MQ Advanced** check box.

    You might not want the advanced functionality or do not have the required license. For information on which components are included with the IBM MQ Advanced option, see IBM MQ Virtual System Pattern Type software components for Linux systems and IBM MQ Virtual System Pattern Type software components for AIX systems.

Related tasks

"Adding IBM MQ Multi-Instance support" on page 25
You can add IBM MQ Multi-Instance support using different virtual machines in your pattern.

LTS Linux AIX Adding multiple software components on the same virtual machine

Generally, each software component must be on a dedicated virtual machine. However, you can add specific software components to the same virtual machine.

Before you begin

Create a pattern from a blank template. For more information, see "Creating IBM MQ Virtual System Pattern Type patterns from blank templates" on page 21.

About this task

If the software components are not of the same version, the latest version is installed. For example, if you install 9.0.0.0 and 9.0.0.2 software components on the same virtual machine, V9.0.0.2 is installed.

Procedure

1. Drag an IBM MQ Advanced software component onto the canvas.

   Adding the component to the canvas creates a virtual machine that contains the component.
2. Drag another IBM MQ Advanced software component onto the same virtual machine.
3. Configure the parameters of each component.

   The port number and queue manager name must be unique on a virtual machine. Deployments with port numbers and queue manager names that are not unique, will fail.

LTS Linux AIX Adding IBM MQ Multi-Instance support

You can add IBM MQ Multi-Instance support using different virtual machines in your pattern.

Before you begin

- Create a pattern from a blank template. For more information, see “Creating IBM MQ Virtual System Pattern Type patterns from blank templates” on page 21.
- Get General Parallel File System (GPFS) information and the name of the GPFS fileset directory from your PureApplication administrator. For more information, see Administering support for GPFS.

About this task

By using the pattern editor, you can create your own pattern with IBM MQ Multi-Instance support. Such a pattern includes two operating system nodes. Each node has a single IBM MQ software component. One IBM MQ software component is configured to be the active instance, the other is configured to be a standby instance. For more information about IBM MQ Multi-Instance, see Multi-instance queue managers.

Procedure

Add the component to the canvas.

1. Drag an IBM MQ software component onto the canvas.

   Adding the component to the canvas creates a virtual machine that contains the component.

Configure the virtual machine.

2. Double-click the default name, `OS Node`, and enter a new name.

   Use a unique name for easier identification during deployment. For example, you might change the name to `MQNodeActive`.
3. Select the number of virtual CPUs to assign to this virtual machine.
4. Enter the required memory size.

   Because you will be using a GPFS policy in subsequent steps, set the memory size to at least 4096 MB.
5. Create passwords for `root` and `virtuser`.

AIX Check that you have enough space to install and upgrade IBM MQ.

6. On Power/AIX, check whether you need to mount another disk for `/usr` to provide more space for installing and upgrading IBM MQ.

   For more information, see “Troubleshooting and support” on page 41.

Configure the IBM MQ software component to be the active instance.

7. Double-click the name of the IBM MQ software component, `IBM MQ Advanced` by default, and enter a new name.

   Use a unique name for the IBM MQ software component for easier identification during configuration and deployment. For example, you might change the name to `MQQueueManagerActive`.
8. Select `High availability active instance` for **Type of IBM MQ configuration**.
9. Enter the path for the shared directory. For example, `/opt/MQShare`.
10. Enter the queue manager name. For example, `QM1`.
11. Enter the listener port number. For example, `1414`.
12. Optional: If you do not want to accept the default values, enter appropriate values for the dead letter queue, queue manager log pages, primary logs, secondary logs, and the error path.
13. Optional: If you want to choose to manage IBM MQ data manually, on the canvas, click the IBM MQ part and on the **IBM MQ** attributes panel, select the **Manage IBM MQ data manually** check box.

    This option ensures that your IBM MQ data is not deleted when the deployed instance is deleted, provided that the location you specify for your data is away from the deployed pattern instance, for example, on the shared GPFS server. If the option to manage the IBM MQ data directory is not selected, the **`deployment id`** is used in the IBM MQ data path at the time of instance creation. When the instance is deleted, the IBM MQ data is deleted too.
14. Optional: If you want to choose to use existing IBM MQ queue manager data directory to deploy a subsequent instance from a pattern and resume the existing IBM MQ queue manager, on the canvas, click the IBM MQ part and on the **IBM MQ** attributes panel, select the **Use existing IBM MQ queue manager data directory** check box.

    The previously deployed pattern instance from which you are reusing data must be created with the **Manage IBM MQ data manually** option selected. For more information on deployment lifecycle options, see Choosing a lifecycle option for a pattern instance.
15. Optional: If you do not want to install IBM MQ Advanced components, on the canvas, click the IBM MQ part and on the IBM MQ attributes panel select the **Do not install IBM MQ Advanced** check box.

    You might not want the advanced functionality or do not have the required license. For information on which components are included with the IBM MQ Advanced option, see IBM MQ Virtual System Pattern Type software components for Linux systems and IBM MQ Virtual System Pattern Type software components for AIX systems.

Add a GPFS policy to the virtual machine.

16. Click **Add a Component Policy** on the virtual machine node.
17. Select the **GPFS Client Policy** checkbox.

Configure the GPFS client policy.

18. Select `GPFS Client Install and Configuration` for **GPFS Client Configurations**.
19. Select the first file system mount section.
20. Enter the GPFS information that you have obtained from your PureApplication administrator.
21. Enter the GPFS name of the fileset directory that you have obtained from your PureApplication administrator.
22. Enter a value for **Storage Maximum**. For example, `1G` (1 gigabyte).
23. Enter the path for the shared directory to link to on your local system.

    This is the path you entered in step "9" on page 26. For example, `/opt/MQShare`.

Create and configure another virtual machine.

24. Drag another IBM MQ software component onto the canvas.

    Another virtual machine is created, containing the new component.
25. Double-click the default name, `OS Node`, and enter a new name.

    Use a unique name for easier identification during deployment. For example, you might change the name to `MQNodeStandby`.
26. Select the number of virtual CPUs to assign to this virtual machine.
27. Enter the required memory size.

Because you will be using a GPFS policy in subsequent steps, set the memory size to at least 4096 MB.

28. Create passwords for `root` and `virtuser`.

Configure the IBM MQ software component to be the standby instance.

29. Double-click the name of the IBM MQ software component, `IBM MQ Advanced` by default, and enter a new name.

    Use a unique name for the IBM MQ software component for easier identification during configuration and deployment. For example, change the name to `MQQueueManagerStandby`.

30. Select `High availability standby instance` for **Type of IBM MQ configuration**.
31. Add parameter reference. You can do this in one of two ways:

    Add reference for all parameters.

    a. Click the blue point which is next to the IBM MQ standby instance software component (for example, `MQQueueManagerStandby`) and drag the cursor to draw a line to the IBM MQ active instance software component (for example, `MQQueueManagerActive`).

    b. Add references for all the parameters listed in the pop-up window.

    c. Click **OK**.

    Add reference for each parameter individually.

    a. Click **Configure Data Dependencies** next to the **Shared directory** input area, to add a reference:

       i) In the pop-up window, click the component-level parameter tab.

       ii) Select the name of the IBM MQ active instance software component. This is the name you entered in step "7" on page 26. For example, `MQQueueManagerActive`.

       iii) Select `ha_standby_shared_dir` for the **Output** attribute.

       iv) Click **ADD** and then **OK**.

    b. Click **Configure Data Dependencies** next to the **Queue manager name** input area, to add a reference:

       i) In the pop-up window, click the component-level parameter tab.

       ii) Select the name of the IBM MQ active instance software component. This is the name you entered in step "7" on page 26.

       iii) Select `ha_standby_name` for the **Output** attribute.

       iv) Click **ADD** and then **OK**.

    c. Click **Configure Data Dependencies** next to the input area of **Listener port** to add a reference.

       i) In the pop-up window, click the component-level parameter tab.

       ii) Select the name of the IBM MQ active instance software component. This is the name you entered in step "7" on page 26.

       iii) Select `ha_standby_port` for the **Output** attribute.

       iv) Click **ADD** and then **OK**.

    d. Click **Configure Data Dependencies** next to the input area of **Error path** to add a reference.

       i) In the pop-up window, click the component-level parameter tab.

       ii) Select the name of the IBM MQ active instance software component. This is the name you entered in step "7" on page 26.

       iii) Select `ha_standby_error_path` for the **Output** attribute.

       iv) Click **ADD** and then **OK**.

e. Click **Configure Data Dependencies** next to the input area of **Queue manager directory** to add a reference.

i) In the pop-up window, click component-level parameter tab.

ii) Select the name of the IBM MQ active instance software component. This is the name you entered in step "7" on page 26.

iii) Select `ha_standby_directory` for the **`Output`** attribute.

iv) Click **ADD** and then **OK**.

f. Click **Configure Data Dependencies** next to the **Manage IBM MQ data manually** input area, to add a reference:

i) In the pop-up window, click the component-level parameter tab.

ii) Select the name of the IBM MQ active instance software component. This is the name you entered in step "7" on page 26.

iii) Select `ha_manage_mq_data_manually` for the **`Output`** attribute.

iv) Click **ADD** and then **OK**.

g. Click **Configure Data Dependencies** next to the **Use existing IBM MQ queue manager data directory** input area, to add a reference:

i) In the pop-up window, click the component-level parameter tab.

ii) Select the name of the IBM MQ active instance software component. This is the name you entered in step "7" on page 26.

iii) Select `ha_existing_data_path` for the **`Output`** attribute.

iv) Click **ADD** and then **OK**.

h. Click **Configure Data Dependencies** next to the **Do not install IBM MQ Advanced** input area, to add a reference:

i) In the pop-up window, click the component-level parameter tab.

ii) Select the name of the IBM MQ active instance software component. This is the name you entered in step "7" on page 26.

iii) Select `ha_do_not_install_mq_advanced_components` for the **`Output`** attribute.

iv) Click **ADD** and then **OK**.

Add a GPFS policy to the virtual machine.

32. Click **Add a Component Policy** on the virtual machine node.
33. Select the **GPFS Client Policy** checkbox.

Configure the GPFS client policy.

34. Select `GPFS Client Install and Configuration` for **GPFS Client Configurations**.
35. Select the first file system mount section.
36. Enter the GPFS information that you have obtained from your PureApplication administrator.
37. Enter the GPFS name of the fileset directory that you have obtained from your PureApplication administrator.
38. Enter a value for **Storage Maximum**. For example, `1G` (1 gigabyte).
39. Enter the path for the shared directory to link to on your local system.

    This is the path you entered in step "9" on page 26. For example, `/opt/MQShare`.

Select the operating system version and IBM MQ version.

40. On the canvas, click the operating system part and select the version that you want to use.
41. On the canvas, click the IBM MQ part and select the version that you want to use.

LTS Linux AIX Adding MQSC commands to a pattern

You can add one or more files of IBM MQ commands to customize a pattern that contains IBM MQ software components. The commands run when the pattern is deployed as a virtual system, when the virtual system is deleted, or whenever you choose to run the commands manually.

About this task

You can use MQSC commands to customize IBM MQ queue managers that have been deployed in patterns. Two script packages are provided which run MQSC commands against queue managers:

MQExecuteMQSCCommand
The **MQExecuteMQSCCommand** script package provides a way for you to run a single MQSC command against a queue manager. This command is suitable for creating or changing a single IBM MQ object or for running a single query against queue manager objects. To use this script package, pass the MQSC command directly in the **COMMAND** parameter to the script package.

MQExecuteMQSC
The **MQExecuteMQSC** script package provides a way for you to run MQSC commands in one or more MQSC command files. You can make the MQSC command files available to the virtual machine in two ways:

- You can copy the `.mqsc` script files to a directory on the virtual machine. For example, you can create a custom operating system image which contains the files, or you can load them in a separate script package. You can then pass the name of the directory containing the `.mqsc` files in the **MQSC_DIRECTORY** parameter to the script package. When the script package runs it executes all of the `.mqsc` command files in the directory.
- You can make a copy of the `MQExecuteMQSC.zip` archive file and add the `.mqsc` command files to the archive and leave the directory name parameter on the script package blank. When the script package runs it executes all of the `.mqsc` command files in the archive file. See “Adding MQSC command files to the MQExecuteMQSC script package” on page 29 for details of this procedure.

In both cases, specify the required queue manager in the **QMGR_NAME** parameter.

What to do next

Add more parts or scripts to the pattern, including multiple instances of the IBM MQ software component, and then deploy the pattern.

Related reference

“IBM MQ Virtual System Pattern Type script packages” on page 44

LTS Linux AIX Adding MQSC command files to the MQExecuteMQSC script package

You can make a copy of the `MQExecuteMQSC.zip` archive file and add `.mqsc` command files, which run when the pattern is deployed as a virtual system, when the virtual system is deleted, or whenever you choose to run the commands manually.

Before you begin

- Check that you have `Create new catalog content` permission.

  From the appliance user interface, click **System** > **Users** and in the list of users, click your user name.
- Add the IBM MQ pattern type to the PureApplication System. See “Installing IBM MQ Virtual System Pattern Type on IBM PureApplication” on page 11.
- Add an IBM MQ software component to the pattern. See “Creating IBM MQ Virtual System Pattern Type patterns from blank templates” on page 21 and “Adding multiple software components on the same virtual machine” on page 24.

- Prepare one or more files of MQSC commands to customize an IBM MQ software component in the pattern.

Procedure

Add one or more files of IBM MQ commands to a copy of the `MQExecuteMQSC` script package. Attach the script package to the IBM MQ software component in a pattern.

1. Create a script package by cloning the **MQExecuteMQSC** script package that is provided.
   a) Open the **Script Packages** interface.
      Click **Catalog** > **Script Packages**. The **Script Packages** window opens.
   b) In the list of script packages, click **MQExecuteMQSC**.
   c) Click **Clone**.
   d) Type a unique name for the copy of the script package in the **Name** field and click **OK**.
2. Add one or more files of MQSC commands to the script package.
   a) In the **Script package files** section of the canvas, under "The script package is in `MQExecuteMQSC-1.0.0.1.zip`", click Download .
   b) Save the downloaded script package on your workstation.

      Tip: Change the name of the script package when you download it to avoid confusion with other script packages you might download. For example, give it the name of the script package.

      Restriction: If you add multiple script packages that contain MQSC command files to an IBM MQ software component, you must modify the `cbscript.json` file in each script package.

      Change the temporary directory in the line, `"location": "/tmp/mq/mqsc"` to `"location": "/tmp/mq/`*`dir`*`"`, where *`dir`* is the name you choose for a new directory. The appliance creates the new temporary directory, which contains the MQSC command files from this script package. It then runs all the commands from the command files in the directory. If you do not create a different temporary directory for each script package, the command files accumulate in the single directory. Each time the appliance processes a script package, it runs commands from all the accumulated files in the single temporary directory, which might give unexpected results.
   c) Add one or more MQSC files with the extension `.mqsc` to the downloaded script package that you are editing.

      With some compressed files tools you, can add files directly to the downloaded zipped package. With other tools, extract the files from the .zip, and re-create it after adding your MQSC file.
   d) In the **Script package files** section of the canvas, in the **Browse...** field, click or type the path to the modified `MQExecuteMQSC-1.0.0.1.zip` file. Then click **Upload**.

      If you renamed `MQExecuteMQSC-1.0.0.1.zip`, you can upload the file with its new name.
3. Choose when to run the script by selecting an option in the **Executes** drop-down list on the canvas.
4. From the appliance user interface, open the **Script Packages** window.
   - From the menu bar, click **Catalog** > **Script Packages**.
   - From the **Welcome** page, under **Setting up your private cloud**, click **Add script packages**.

   The **Script Packages** window opens.
5. Open a pattern that contains an IBM MQ basic part for editing.
   a) Find the pattern in the list of Virtual Systems Patterns.

      A draft icon that follows the pattern name signifies that the pattern is editable.

      The pattern opens in the properties window.
   b) Click the edit icon in the menu bar.

The pattern editor opens, which has lists of parts, scripts, and add-ons in a navigation pane, and an editing canvas.

6. Click the **Scripts** tab in the navigator

   Available scripts are listed.
7. Drag your script package onto the part, which is on the editing canvas.
8. Click **OK** to close the window.
9. Click **Done Editing** to finish editing the pattern.

   The pattern is saved.

What to do next

Add more parts or scripts to the pattern, including multiple instances of the IBM MQ software component, and then deploy the pattern.

Related reference

"IBM MQ Virtual System Pattern Type script packages" on page 44

LTS Linux AIX Deploying IBM MQ Virtual System Pattern Type patterns

When you deploy an IBM MQ Advanced software component on an operating system image within a virtual system pattern, IBM MQ is installed and any defined queue managers are started.

Before you begin

To deploy a virtual system pattern, you must first create a pattern from a blank template. See "Creating IBM MQ Virtual System Pattern Type patterns from blank templates" on page 21.

About this task

A deployed pattern is a virtual system instance. The deployment process creates and starts virtual machines for defined parts. The amount of time required for the deployment depends on the complexity of the pattern that you are deploying.

Procedure

To deploy a pattern, select the pattern to deploy and specify required parameters for the virtual system instance.

1. Click **Patterns** > **Virtual Systems**.
2. Select the pattern to deploy and click **Deploy**.
3. Under **Configure**, enter the required information.

   The attributes that are required differ depending on the defined configuration and any associated script packages. Complete the following required attributes:

   - For **Name**, enter a name for the virtual system instance.
   - Under **Environment Profile**, select the type of environment profile to which you are deploying the pattern.
   - Under **Component Attributes**, specify the attributes for the components in your pattern.

What to do next

You can view the status of your instance from the Virtual System Instances window. When the deployment is complete, you can administer the instance.

For more information about the Deploy feature, see:

Linux IBM PureApplication System W2500, version 2.2.2 documentation: 'Deploying virtual system patterns'

AIX IBM PureApplication System W2700, version 2.1.2 documentation: 'Deploying virtual system patterns'

LTS Linux AIX Security

When an IBM MQ software component is added to a virtual system pattern, or when it is deployed, you can customize IBM MQ security settings before any users log on.

Default security of the IBM MQ software component

When the IBM MQ software component is first deployed in an IBM MQ virtual system instance, the deployment process creates the user `mqm` with the primary group `mqm`. No password is assigned, so by default you cannot log on with the `mqm` user ID. The deployment process runs as the `mqm` user to create and configure the queue manager.

By default, the mqm group UID and GID are both set to 1414. If the mqm group already exists, the UID and GID are not changed.

The deployment process creates two other users when the virtual machine is deployed, `root` and `virtuser`. If these users run the `sudo -u mqm` *command* command, they are authorized to run *command* with the authority of the `mqm` group. The users themselves are not members of the `mqm` group.

In virtual system images supplied by IBM, `virtuser` has permission to run the **sudo** and to run commands under the `mqm` account. If you create you own virtual system image, you can grant this permission by adding `virtuser ALL = (mqm : mqm) ALL` to the `/etc/sudoers` file. For example:

```
echo "virtuser  ALL = (mqm : mqm) ALL" >> /etc/sudoers
```

LTS Linux AIX Verifying a deployed IBM MQ software component from an SSH terminal emulator

Verify an IBM MQ software component locally, without authorizing remote users to the `virtuser` user ID. Test the deployment from an SSH terminal emulator.

Before you begin

- Deploy the pattern successfully; see “Deploying IBM MQ Virtual System Pattern Type patterns” on page 31.
- Open an SSH terminal emulator window on the client workstation you are using to connect to a virtual machine in the virtual system instance.
  - On UNIX and Linux, run the **ssh** command.
  - On Windows, obtain an SSH terminal emulator such as PuTTY. The PuTTY website is PuTTY: A Free Telnet/SSH Client. When a connection is established with the deployed virtual machine, the steps in the task are the same.

About this task

Verify the deployment of an IBM MQ software component by running the **put** and **get** sample programs on the virtual machine where the IBM MQ software component is deployed.

The user ID `virtuser` is not a member of the `mqm` group. The example demonstrates the use of the **sudo** command to make `virtuser` a temporary member of the `mqm` group for the duration of the command. The example also sets up the minimum set of permissions for `virtuser` to run the put and get sample

programs, and then removes them. As an alternative, the example also shows running the **sudo** command to run the **put** and **get** sample programs.

Procedure

1. Obtain the host name or IP address of the virtual machine that contains a deployed IBM MQ software component.
   a) In the appliance user interface, click **Instances** > **Virtual Systems**. Click the virtual system instance in the navigator.

      The property window for the virtual system instance opens.

   b) In the property window, click the plus sign next to **Virtual machines** to expand the list of virtual machines.

      A list of the virtual machines in the system instance opens. Each virtual machine has a plus sign and an icon to show that it is running: .

   c) Click the plus sign next to the virtual machine you want to connect to.

      The properties of the virtual machine are listed.

   The public IP address is displayed in the list of virtual machines.
2. Connect the SSH terminal emulator you are running to the virtual machine.
   - On UNIX and Linux:
     a. In a command shell window type

     ```
     ssh virtuser@ hostname-IPaddress
     ```

     . where *hostname-IPaddress* is either a host name or IP address.

     The system responds with a password prompt.

     Note: The first time you connect an SSH session, you must respond to the host authentication prompt. To continue, respond `yes`.

     b. Type the password of the `virtuser` user ID.
   - On Windows:
     a. Open the **PuTTY** configuration window by running the **PuTTY** command.
     b. In the Session tab, type the **Host Name (or IP address)** and click **Open**. Leave the **Port** as 22, and the **Connection type** as SSH.

     The system responds by opening a window and the prompt, `login as:`

     c. Type `virtuser`.

     The system responds with a password prompt.

     Note: The first time you connect an SSH session, you must respond to the host authentication prompt. To continue, respond `yes`.

     d. Type the password of the `virtuser` user ID.
3. Verify the version of IBM MQ.

   Check the version of IBM MQ that is returned by the following command:

   ```
   su mqm -c 'dspmqver'
   ```
4. Authorize the user ID `virtuser` to run the verification application.
   a) Give `virtuser` permission to connect to the queue manager.

   ```
   sudo -u mqm setmqaut -m QmgrName -t qmgr -p virtuser +connect
   ```

   The system responds with:

```
[sudo] password for virtuser:
```

Type the password for the user ID `virtuser`.

The system responds with:

```
The setmqaut command completed successfully.
```

b) Give `virtuser` permission to use put, get, and inquire MQI calls against the queue used for verification.

```
sudo -u mqm setmqaut -m QmgrName -n SYSTEM.DEFAULT.LOCAL.QUEUE -t queue -p virtuser +put
+get +inq
```

The system responds with:

```
The setmqaut command completed successfully.
```

5. Put a message on `SYSTEM.DEFAULT.LOCAL.QUEUE`.

```
/opt/mqm/samp/bin/amqsput SYSTEM.DEFAULT.LOCAL.QUEUE QmgrName
```

The system response:

```
Sample AMQSPUT0 start

target queue is qName
```

6. Type in a message followed by two new lines

```
Hello world
```

The system response:

```
Sample AMQSPUT0 end
```

7. Get the message from `SYSTEM.DEFAULT.LOCAL.QUEUE`.

```
/opt/mqm/samp/bin/amqsget SYSTEM.DEFAULT.LOCAL.QUEUE QmgrName
```

The system response:

```
Sample AMQSGET0 start

message <Hello world>

15 seconds delay

no more messages

Sample AMQSGET0 end
```

Results

You demonstrated the queue manager is working correctly.

What to do next

You have completed the verification task. You might choose to remove the changes you made, and try the **sudo** command to run the **put** and **get** sample programs.

1. Remove the authorizations you set by running the following commands:

```
sudo -u mqm setmqaut -m QmgrName -t qmgr -p virtuser -connect
sudo -u mqm dspmqaut -m QmgrName -t qmgr -p virtuser
sudo -u mqm setmqaut -m QmgrName -n SYSTEM.DEFAULT.LOCAL.QUEUE -t queue -p virtuser -put
-get -inq
sudo -u mqm dspmqaut -m QmgrName -n SYSTEM.DEFAULT.LOCAL.QUEUE -t queue -p virtuser
```

2. Run the verification again with the **sudo** command:

   a. Put a message onto SYSTEM.DEFAULT.LOCAL.QUEUE

   ```
   sudo -u mqm /opt/mqm/samp/bin/amqsput SYSTEM.DEFAULT.LOCAL.QUEUE QmgrName
   ```

   Type a message followed by two new lines.

   b. Get the message from SYSTEM.DEFAULT.LOCAL.QUEUE

   ```
   sudo -u mqm /opt/mqm/samp/bin/amqsget SYSTEM.DEFAULT.LOCAL.QUEUE QmgrName
   ```

   Wait for 15 seconds for the program to finish.

LTS Linux AIX Managing virtual system instances

When you deploy a virtual system pattern into the cloud, the resulting virtual system instance is a working IBM MQ environment. The virtual system instance contains running virtual machines that are created from software components in the pattern.

You can manage your virtual system by running various virtual system instance management tasks on the virtual system itself.

You can update a deployed pattern instance, that is, the scripts that manage the instance lifecycle. You can also apply fix packs or interim fixes. There are a number of other administration and troubleshooting operations that you can perform from the IBM MQ software component.

LTS Linux AIX Updating a deployed pattern instance

You can apply updates to a deployed virtual system pattern using the Workload console.

Before you begin

You must have the "Workload" resources administration role with full permission, including permission to manage workload resources.

Procedure

1. Open the properties of a virtual system instance that contain an IBM MQ software component.

   In the appliance user interface, click **Instances** > **Virtual Systems**. Click the virtual system instance in the navigator.

   The property window for the virtual system instance opens.
2. In the Pattern Type section, click **Check for updates**.
3. If an update is available, click **Apply** to update the instance.

   View the Status section and verify that the pattern plug-in is in the Available state.

   - If the update is successful, the **Status field** arrow turns green. Review the updated deployment.
     - To complete the update, click **Commit**.

– To discard the changes that were made in the update, click **Revert**.

Your deployed virtual instances are updated. Any virtual machine that contains IBM MQ software components is updated to contain the enabled pattern type version.

- If the update fails, the following message is displayed under **History**: `Error encountered during upgrade, automatically reverting the deployment. No action is required. The backup data is restored and the virtual application is reverted to its previous state.`

LTS Linux AIX Applying fix packs or interim fixes

You can upgrade existing virtual system instances that are running IBM MQ installations to a new fix pack level or install interim fixes.

About this task

The IBM MQ environment stops prior to installing fix packs or interim fixes, and restarts after the fix pack or interim fix is applied.

When the installation repository contains an IBM MQ fix pack or interim fix, you can schedule a service request, if the offering ID is compatible with the IBM MQ product.

Procedure

To apply a fix pack or interim fix to a virtual system instance that contains an IBM MQ Virtual System Pattern Type software component, complete the following steps.

1. In the **Virtual System Instances** window, select a virtual system instance to update.
2. Click **Manage**. The Instance Console window is displayed.
3. On the **Operations** tab, select **Maintenance**.
4. Optional: If you are installing an IBM MQ interim fix, you must stop the messaging service first. On the **Operations tab**, select MQ9_BASE, expand **Stop the messaging service** and click **Submit.**
5. In the **Fundamental** panel, expand **Maintenance fix packs** or **Maintenance fixes**.
6. Select the fix pack or interim fix that you want to apply.
7. Click **Submit** and view the results in the **Operation Execution Results** window.
8. Optional: Remember to restart the messaging service if you stopped it in step “4” on page 36.

LTS Linux AIX Installing an IBM MQ interim fix into a deployed pattern instance

You can install a standard IBM MQ interim fix by using the IBM PureApplication System instance console. You first need to package the interim fix into an archive file with extra metadata and a script that installs the interim fix. You can then add it into the PureApplication System catalog and it is available for use.

About this task

You can package a standard IBM MQ interim fix and install it in following steps:

Procedure

1. Download the IBM MQ interim fix from Fix Central or from the link that is provided by IBM Support.
   An interim fix is a tar archive file, for example: `9.0.0.2-WS-MQ-LinuxX64-LAIT13465.tar.gz`.
2. Extract the file contents.
3. Open the `readme.txt` file in the archive and make a note of the fix name at the top of the file.
   You can use it in step “6” on page 37.
4. Create a file, save it as `service.xml`, and paste in the following code:

```
<?xml version="1.0" encoding="UTF-8" ?>
  <rmsd:Service xmlns="https://www.w3.org/2001/XMLSchema"
   xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
   xmlns:rmsd="https://www.ibm.com/websphere/rainmaker/service/servicedescription"
   xsi:schemaLocation="https://www.ibm.com/websphere/rainmaker/service/servicedescription ./
Service.xsd">
  <rmsd:PluginPrereqs>
    <rmsd:prereq pluginvrmf="21.0.0.x" pluginname="plugin.com.ibm.vsys.mq"/>
    <rmsd:prereq pluginvrmf="21.0.0.x" pluginname="plugin.com.ibm.vsys.mq"/>
  </rmsd:PluginPrereqs>
  <rmsd:Packages>
    <rmsd:Package name="FIX-NAME" type="ifix" target="APPLICATION">
      <rmsd:Command name="/bin/sh /tmp/FIX-NAME/install.sh">
        <rmsd:Location>/tmp/FIX-NAME</rmsd:Location>
      </rmsd:Command>
    </rmsd:Package>
  </rmsd:Packages>
</rmsd:Service>
```

5. Check that the <rmsd:PluginPrereqs> parent element includes a list of child elements with your versions of the IBM MQ Virtual System Pattern Type plugin for your deployed patterns.

   To check your installed plug-in versions in the **Workload Console**, click **Catalog** -> **System Plug-ins** and then enter `plugin.com.ibm.vsys.mq` in the **Filter**. The version is shown next to the plug-in name. Edit the `service.xml` file from step "4" on page 36 with the version information of your pattern:

   ```
     <rmsd:PluginPrereqs>
       <rmsd:prereq pluginvrmf="21.0.0.0" pluginname="plugin.com.ibm.vsys.mq"/>
       <rmsd:prereq pluginvrmf="21.0.0.2" pluginname="plugin.com.ibm.vsys.mq"/>
     </rmsd:PluginPrereqs>
   ```

6. Edit the attributes in the <rmsd:Packages> element of the `service.xml` file to replace the value of the name attribute **FIX-NAME** to the value that you found at the top of the `readme.txt` file from step "3" on page 36.

   For example, if the interim fix name is `9.0.0.2-WS-MQ-LinuxX64-LAIT13465-157376`, change the line 11 from:

   ```
       <rmsd:Package name="FIX-NAME" type="ifix" target="APPLICATION">
   ```

   to

   ```
   <rmsd:Package name="9.0.0.2-WS-MQ-LinuxX64-LAIT13465-157376" type="ifix"
   target="APPLICATION">
   ```

   On line 12, change the path to the script file from:

   ```
         <rmsd:Command name="/bin/sh /tmp/FIX-NAME/install.sh">
   ```

   to

   ```
         <rmsd:Command name="/bin/sh /tmp/9.0.0.2-WS-MQ-LinuxX64-LAIT13465-157376/install.sh">
   ```

   On line 13, change the location path from:

   ```
            <rmsd:Location>/tmp/FIX-NAME</rmsd:Location>
   ```

   to

   ```
            <rmsd:Location>/tmp/9.0.0.2-WS-MQ-LinuxX64-LAIT13465-157376</rmsd:Location>
   ```

   Save the `service.xml` file.

7. Create a file, save it as `install.sh`, and paste in the following code:

   ```
   #!/bin/sh
   set -x
   MQInstallDir=MQ_INST_DIR
   FixName=MQ_FIX_NAME
   FixPackage=MQ_FIX_ARCHIVE
   ```

```
echo "Unpack MQ interim fix"
cd /tmp/${FixName}
tar -xvf ${FixPackage}
echo "== Run MQ fix installer =="
cd ${FixName}
./mqfixinst.sh ${MQInstallDir} install ${FixName}
```

8. Change the value of *MQInstallDir* to match the directory where IBM MQ is installed.

   The default location on Linux systems is `/opt/mqm` and `/usr/mqm` on AIX systems. For example, replace *MQ_INST_DIR* with `/opt/mqm` on line 3.
9. Change the value of *FixName* to match the name of the IBM MQ interim fix you noted in step "3" on page 36.

   For example, replace *MQ_FIX_NAME* with `9.0.0.2-WS-MQ-LinuxX64-LAIT13465-157376` on line 4.
10. Change the value of *FixPackage* to match the name of the interim fix archive you downloaded in step 1.

    For example, replace the *MQ_FIX_ARCHIVE* with `9.0.0.2-WS-MQ-LinuxX64-LAIT13465.tar.gz` on line 5.
11. Save the `install.sh` file.
12. Create a compressed archive file that contains the interim fix archive that you downloaded in step 1, the edited `install.sh`, and `service.xml` files you created.

    The files must be in the root directory of the compressed file.
13. You can now upload the interim fix archive file that contains the amended `service.xml` file and the `install.sh` script to the PureApplication System catalog, as described in Adding emergency fixes to the catalog.
14. You can then install the interim fix, as described in "Applying fix packs or interim fixes" on page 36.

    Note: You must stop the messaging service through the **Instance Console** → **Operations panel** before you install an IBM MQ interim fix.

Related tasks

"Applying fix packs or interim fixes" on page 36

LTS Linux AIX

Running IBM MQ operations

Manage your virtual suystems instances. Start or stop a messaging service or a queue manager. Set tracing and logging. Use diagnostics operations and information. Enable failover.

About this task

The IBM MQ software component comes with a set of administration and troubleshooting operations that are available in the **Instance Console**, on the **Operations** tab. Click **Instances** > **Virtual Systems** > **Select an instance** > **Manage**.

Start or restart the messaging service
Starts the messaging service that is currently stopped. Select the checkbox, **Perform this operation if the messaging service is running** if you want to restart a messaging service that is already running.

Stop the messaging service
Stops the messaging service.

Start or restart the queue manager
Starts the queue manager that is currently stopped. Select the checkbox, **Perform this operation if the queue manager is running** if you want to restart a queue manager that is already running.

Note: Each time your IBM MQ Virtual System Pattern Type instance is restarted, any IBM MQ queue managers that are a part of that deployment will start automatically, even if they were stopped during the previous session. This follows the normal operation logic for IBM MQ environments on Windows platforms but is not the usual behavior for IBM MQ environments on Linux platforms.

Stop the queue manager
Stops the queue manager.

Logs and first failure data capture
Returns the servers' runtime and error logs, including First Failure Data Capture (`.fdc`) files. The IBM MQ error directory for a specified queue manager is compressed into an archive file. This is a convenient way to obtain any error logs and first failure error files that might be present.

Trace
Provides an extra level of diagnostic information. Your IBM support representative might ask you to apply this setting if they require additional information. Trace information is obtained for individual queue managers, but using the **Collect trace** control obtains trace output for all the queue managers installed on a particular virtual machine. After trace has been stopped, you can collect the trace files either in the native form or in a formatted form. Clearing trace removes all trace files from the system.

Diagnostics
Operations that are available for problem determination. These operations might affect the operation of the system and should only be used under the guidance of your IBM support representative. This operation runs the IBM MQ runmqras command with certain predefined options.

Version information
Displays the current version of IBM MQ messaging service.

Show HA mode
Shows the high availability mode of this queue manager. Returns whether the queue manager is an active instance or a standby instance.

- Returns `Queue manager: queue_manager_name> is not in HA mode` if the queue manager is not an HA instance or there is no standby instance.
- Returns `Queue manager: queue_manager_name> is not started` if the queue manager is stopped.

Failover the queue manager
Starts the failover of this queue manager if it is an active instance.

- Returns `Queue manager: queue_manager_name> is not started` if the queue manager is stopped.
- Returns `Cannot perform failover on a standby queue manager: queue_manager_name>` if the queue manager is a standby instance.
- Returns `Cannot perform failover: Queue manager queue_manager_name> does not have a standby instance to take over` if the queue manager does not have a running standby instance to take over from it.
- Returns `Cannot perform failover: Queue manager queue_manager_name> is not an HA instance or there is no standby instance` if the queue manager is a single instance.

Failover the messaging service
Starts the failover of all the queue managers that are active instances.

Related information
IBM MQ runmqras command

LTS Linux AIX Migrating queue manager data between fix packs and installing interim fixes

You can install a fix pack or interim fix for the IBM MQ Virtual System Pattern Type that contains an updated IBM MQ fix pack. When you want to migrate your deployed pattern instance to a new fix pack level of IBM MQ, you can deploy new instances that use existing data.

Before you begin

Check the restrictions for the IBM MQ software component that is running in your operating system environment. For more information, see “Restrictions for IBM MQ Virtual System Pattern Type” on page 9.

About this task

This task assumes that you have an existing, deployed pattern type instance with a working IBM MQ installation. This task takes you through an example of installing a fix pack or an interim fix for the IBM MQ Virtual System Pattern Type that also contains a new fix pack or interim fix for IBM MQ. You update the pattern and upgrade the IBM MQ fix pack for a running instance of the pattern type. In the final step, the existing IBM MQ queue manager data is migrated to the new fix pack version of IBM MQ that is running in a newly updated, deployed IBM MQ Virtual System Pattern Type instance.

Procedure

- Install and enable the new IBM MQ Virtual System Pattern Type. For detailed steps, see “Installing IBM MQ Virtual System Pattern Type on IBM PureApplication” on page 11.
- Apply the update to your deployed pattern instance. For detailed steps, see “Updating a deployed pattern instance” on page 35.
- Upgrade the IBM MQ installation that is running in the deployed IBM MQ Virtual System Pattern Type instance, to a new version or fix pack level. For detailed steps, see “Applying fix packs or interim fixes” on page 36.

  As a part of this step, your existing IBM MQ installation, including any IBM MQ queue managers and applications, is stopped, migrated, and restarted in the upgraded installation of IBM MQ.

Results

You installed and enabled a new version of the IBM MQ pattern type in your IBM PureApplication System environment. You applied an update to your deployed pattern instance and upgraded the IBM MQ version to a new fix pack level.

Note: At IBM MQ Virtual System Pattern Type version 2.0.0.2, you have three optional parameters for creating IBM MQ PureApplication System patterns. You can manually manage data to prevent data deletion when the deployed IBM MQ Virtual System Pattern Type instance is deleted, resume a queue manager from an existing data directory in a subsequently deployed instance, and you can choose not to install the advanced IBM MQ software components.

The options are made available in the IBM MQ Virtual System Pattern but are not selected as a default. Your existing pattern instance and IBM MQ installation are not affected but you can choose these options when you deploy future instances from the updated IBM MQ Virtual System Pattern Type.

Related tasks

“Running IBM MQ operations” on page 38

LTS Linux AIX Migrating from IBM MQ Virtual System Pattern Type version 1.0.0.x to 2.0.0.x

You can install a new version of the IBM MQ Virtual System Pattern Type that contains a new version of IBM MQ. To keep using a pattern that you defined in version 1.0.0.x, you must manually re-create this

pattern in version 2.0.0.x. You can then deploy a version 2.0.0.x pattern instance that uses the queue manager data from a version 1.0.0.x pattern instance.

Before you begin

Check the restrictions for the IBM MQ software component that is running in your operating system environment. For more information, see “Restrictions for IBM MQ Virtual System Pattern Type” on page 9.

This task assumes that you had a deployed, version 1.0.0.x IBM PureApplication System pattern instance with a working IBM MQ installation and a running queue manager.

About this task

You have queue manager data in a pattern instance that was deployed from a version 1.0.0.x pattern. You want to reuse this data in a pattern instance that is deployed from a version 2.0.0.x pattern. If you leave your data inside a deployed pattern instance file space, the data is deleted when the instance is deleted. Therefore you must make sure that you back up your queue manager data, or manage your original queue manager data manually. Managing your queue manager data manually means specifying a location for your queue manager data away from the deployed pattern instance file space. See “Choosing a lifecycle option for a pattern instance” on page 10. You can then upgrade your pattern type to the new version, re-create your pattern, and use existing data from the previously deployed version 1.0.0.x instance to deploy the version 2.0.0.x pattern instance.

Procedure

- Make sure that you have access to the data from the previously deployed version 1.0.0.x pattern instance.

  You might have used the option to **Manage IBM MQ data manually** for your old pattern instance, or you backed up the queue manager data before you deleted the old pattern instance.
- Install and enable the new IBM MQ Virtual System Pattern Type. For detailed steps, see “Installing IBM MQ Virtual System Pattern Type on IBM PureApplication” on page 11.

  After a new pattern type is installed, you can no longer see the older version of the pattern type in the pattern manager console.
- Re-create your pattern at pattern type version 2.0.0.x.
- Deploy a pattern instance from the version 2.0.0.x re-created pattern and specify the following options:
  - Specify **Use existing IBM MQ queue manager data directory** to add the location for your old queue manager data.
  - Specify **Manage IBM MQ data manually** to add the path for the location, away from the deployed pattern instance file space, where you want to save the queue manager data.

  Your queue manager is restarted and is using data from the previously deployed instance.

Results

You installed a new version of the IBM MQ Virtual System Pattern Type in your IBM PureApplication System environment. You re-created your pattern and deployed a pattern instance. Your queue manager is running at a new version of IBM MQ.

LTS Linux AIX Troubleshooting and support

Learn how to troubleshoot a problem with your IBM software.

If your deployment fails

If your deployment fails, you can search for errors in the log files:

1. See the topics “Collecting IBM MQ error logs from a virtual system instance” on page 42 and “Viewing IBM MQ error logs from a virtual system instance” on page 44 to find out how to access error logs.
2. Click **Logging** and expand the **Operating system** node and then the **IWD Agent** node.
3. Search for the logs for the IBM MQ patterns part.

   Note: The name of the IBM MQ patterns part matches the name of the software component.
4. Review the contents of the `trace.log` file.

If the General Parallel File System (GPFS) server gets out of sync

If you delete a pattern instance and then create a new one, reusing the virtual machine IP address in another pattern instance, the GPFS server can get out of sync. As a result, the volume will no longer mount. You can avoid this happening by shutting down the virtual machine instances cleanly before deletion. For more information, see GPFS Troubleshooting and tuning.

If the pattern you created fails to deploy and the `trace.log` file shows the IBM MQ error (893)

You created an IBM MQ Virtual System Pattern Type pattern with multiple queue managers but the GPFS storage limitation value is set too low and the deployment fails. You can avoid this happening by changing the GPFS storage maximum to 1G, to allow enough space for different scenarios, for example, when configuring two IBM MQ components into an IBM MQ Multi-Instance pattern with active and standby queue managers.

Note: Each queue manager requires just over 100M of space when created but this storage requirement will increase with further configuration and runtime processing of messages. The same GPFS partition might be used in multiple deploys, so you need to allow enough storage for the workload you require.

If you have problems that relate to IBM MQ Multi-Instance

For errors that relate to IBM MQ, see IBM MQ troubleshooting and support.

If the `/usr` file system is not large enough to install IBM MQ and an upgrade fix pack

AIX

On AIX, the storage for the `/usr` file system might not be large enough for you to be able to install the IBM MQ product and an upgrade fix pack. There are two options that you can use to fix this issue:

- Create an AIX OS image that gives enough space for `/usr`, which is the default installation and upgrade path for IBM MQ. The amount of space that you create must be more than 10G.
- Use the OS add-on component to add additional disk and mount to `/usr` for IBM MQ installation and upgrade.

LTS Linux AIX Collecting IBM MQ error logs from a virtual system instance

You can download the IBM MQ error logs from an IBM MQ software component in a virtual system instance. Run the **`Must gather`** command to collect the logs from the IBM PureApplication System.

You can also collect log and trace files by using the operations provided in the software component. See “Running IBM MQ operations” on page 38.

Before you begin

Ensure that the virtual system instance that contains IBM MQ software components is running.

About this task

Obtain the IBM MQ error logs from the PureApplication System. Run the **`Execute now`** command of the `Must Gather Logs` script on the PureApplication System to copy the logs to the PureApplication System, and then transfer the logs to your workstation for analysis. The script gathers the IBM MQ error logs from the IBM MQ software component in a virtual machine in a virtual system instance.

Procedure

1. Open the properties of a virtual system instance that contain an IBM MQ software component.
   a) In the appliance user interface, click **Instances** > **Virtual Systems**. Click the virtual system instance in the navigator.

      The property window for the virtual system instance opens.

   b) In the property window, click the plus sign next to **Virtual machines** to expand the list of virtual machines.

      A list of the virtual machines in the system instance opens. Each virtual machine has a plus sign and an icon to show that it is running: .

   c) Click the plus sign next to the virtual machine you want to connect to.

      The properties of the virtual machine are listed.
2. Capture the current logs.
   a) In the Script packages section, find the **Must Gather Logs** script package and click **Execute now**, .

      The appliance responds with a prompt for an administrator user ID and password.

   b) Click **OK**

      No user ID or password is required.

      The appliance creates a **Must Gather logs** list item in the **Script Packages** properties. The list entry has an hour glass icon while the logs are created. When the logs are available the hour glass changes to a check mark icon .
3. Transfer the logs to your workstation.
   a) In the **Must Gather logs** list item in the **Script Packages**, click the compressed file; for example, `cloudburst_collect1340982954021.zip`.

   b) Follow your browser instructions to save the file on your workstation.
4. Expand the compressed file to retrieve the log files.

   The compressed file contains a tape archive `OS_node_*MQ.tar` (where `*` represents a unique timestamp). The archive contains the IBM MQ logs.

   You must have a compressed file utility on your workstation, such as 7-Zip on Windows, which expands both `.zip` and `.tar` files. Expand both the Must Gather compressed folder, and the `mq.tar` that it contains.

   Expanding the `mq.tar` archive creates the directories `var/mqm/qmgrs` and `var/mqm/errors`, and the IBM MQ log files contained in those directories, from the IBM MQ software component in the virtual machine.

LTS Linux AIX Viewing IBM MQ error logs from a virtual system instance

You can view the IBM MQ error logs created by an IBM MQ software component in a virtual system instance and monitor their activity from the instance console.

Before you begin

Ensure that the virtual system instance that contains IBM MQ software components is running.

Procedure

1. Open the **Virtual Systems Instances** window.
   - In PureApplication 2.0, click **Instances** > **Virtual Systems**.
   - In PureApplication 2.1, click **Patterns** > **Virtual System Instances**.
2. Select a virtual system instance and click **Manage** to open the instance console.
3. Click **Logging** and expand the virtual machine that you want to see the logs for.

   You can click the logs you are interested in to view individual log files and monitor activity in those logs in "real-time".

LTS Linux AIX Reference

IBM MQ Virtual System Pattern Type reference information.

For IBM MQ reference information see IBM MQ reference.

LTS Linux AIX IBM MQ Virtual System Pattern Type script packages

IBM MQ Virtual System Pattern Type script packages contain command scripts and queue manager commands to configure the basic part.

With the tools in the appliances, you can create additional script packages.

You can deploy an IBM MQ Virtual System Pattern Type script package by dragging it onto an IBM MQ basic part.

Table 10. IBM MQ Virtual System Pattern Type script packages

Script package name	Description
`MQExecuteMQSC`	`Run MQSC scripts`
`MQExecuteMQSCCommand`	`Run a single MQSC command`

Table 11. Properties and default values of the MQExecuteMQSC script package

Property	Description
`QMGR_NAME`	Required: The name of the queue manager that MQSC commands must run on.
`MQSC_DIRECTORY`	Optional: The name of the directory that contains the `.mqsc` files to be run. All files in the directory are run. You can specify only a single directory and subdirectories are not included.

Table 12. Properties and default values of the MQExecuteMQSCCommand script package	
Property	**Description**
QMGR_NAME	Required: The name of the queue manager that MQSC commands must run on.
COMMAND	Required: The single MQSC command that is to be run.

Related tasks

“Adding MQSC commands to a pattern” on page 29

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Software Interoperability Coordinator, Department 49XA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

© Copyright IBM Corp. 2007, 2025

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming interface information

Programming interface information, if provided, is intended to help you create application software for use with this program.

This book contains information on intended programming interfaces that allow the customer to write programs to obtain the services of WebSphere® MQ.

However, this information may also contain diagnosis, modification, and tuning information. Diagnosis, modification and tuning information is provided to help you debug your application software.

Important: Do not use this diagnosis, modification, and tuning information as a programming interface because it is subject to change.

Trademarks

IBM, the IBM logo, ibm.com®, are trademarks of IBM Corporation, registered in many jurisdictions worldwide. A current list of IBM trademarks is available on the Web at "Copyright and trademark information"www.ibm.com/legal/copytrade.shtml. Other product and service names might be trademarks of IBM or other companies.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

This product includes software developed by the Eclipse Project (http://www.eclipse.org/).

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Part Number:

(1P) P/N: