# IBM MQ V9.3 機能と構成

## 6. システム管理

2023年09月

日本アイ・ビー・エム　システムズ・エンジニアリング（株）

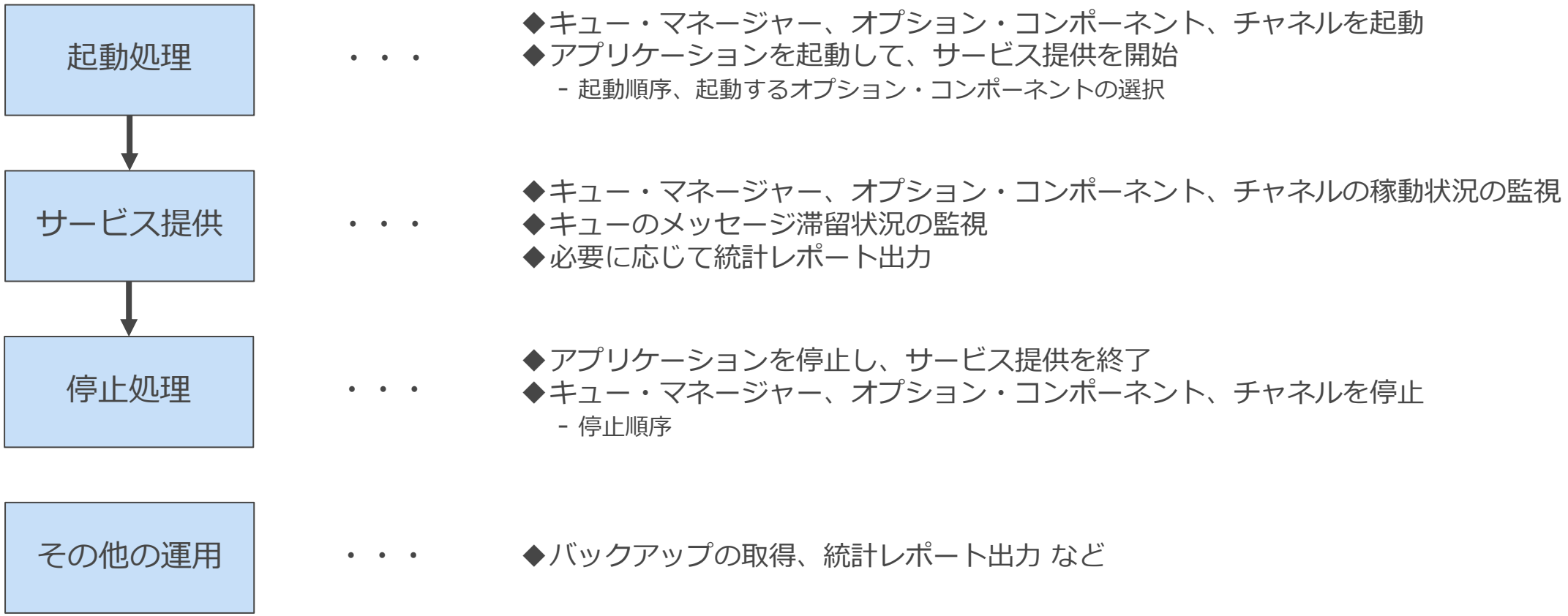
# トピック

- MQ運用概説
  - MQ運用サイクル
- 管理インターフェース
  - 管理インターフェース
  - 管理REST API、MQ Consoleおよびmqwebサーバー
  - 管理インターフェースの提供機能
- 起動・停止
  - 起動処理
  - 停止処理
- 監視
  - 監視概要
  - オンライン・モニタリング
  - イベント・モニタリング
  - システム・トピックを使用した監視
  - エラー・ログ監視
- 会計・統計
  - 会計・統計レポート
  - 会計レポート
  - 統計レポート
- バックアップ・リカバリー
  - リカバリーの種別
  - ログ・タイプ
  - ログ・タイプと回復可能な障害種別
  - メディア・イメージの取得
  - メディア障害とリカバリー
  - キュー・マネージャーのバックアップ
  - MQディレクトリのバックアップ
  - バックアップからのリカバリー
  - ディザスター・リカバリー

# MQ運用概説

# MQ運用サイクル

■ 運用サイクルに合わせて運用方法、監視方法を検討



# 管理インターフェース

# 管理インターフェース

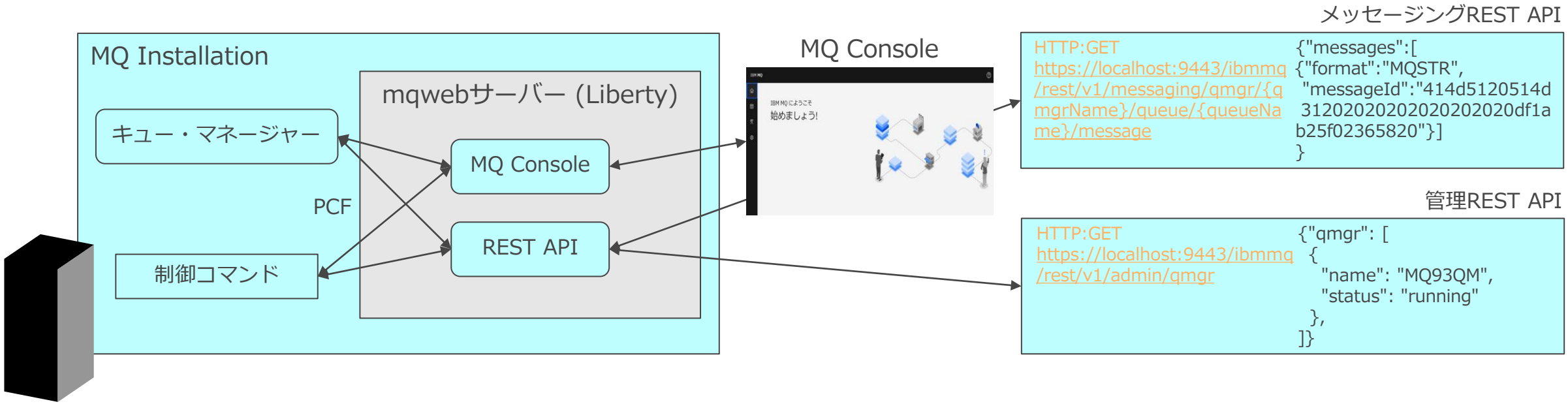
- MQは管理インターフェースとして下記を提供
  - ◆ 制御コマンド
    - キュー・マネージャーの作成、削除、開始、停止
    - MQリスナーの開始、停止
    - トレースの取得開始、停止、フォーマット
    - コマンド・ラインで実行
      - 例）crtmqm、runmqsc、endmqlsr など
  - ◆ MQSCコマンド
    - プラットフォーム共通のコマンド・スクリプト
    - キューやチャネルなどのMQオブジェクトの作成、削除、属性照会、属性変更、開始、停止、ステータスの照会
    - キュー・マネージャーの属性照会、属性変更
    - runmqsc ユーティリティ経由で実行
      - 例）DEFINE CHANNEL、DISPLAY QLOCAL など
  - ◆ PCF (Programable Command Format)
    - MQオブジェクトをプログラムから管理するためのメッセージ・フォーマット
    - MQSCコマンドが提供する機能と同等
    - リモートのキュー・マネージャーの管理が可能
    - コマンド・サーバー経由で実行

# 管理インターフェース

- MQは管理インターフェースとして下記を提供(続き)
  - ◆ 管理REST API
    - キュー・マネージャー、MQオブジェクトを管理するためのAPIを提供
      - REST APIを利用するアプリケーションとの接続が簡易化
    - データ・フォーマットはJSON
    - URLはSwagger文書で確認
    - 以下の機能を提供
      - キューに対する管理操作、インストレーションの情報取得、MQSCコマンドの発行 など
  - ◆ GUIツール
    - MQ Console
      - ブラウザでのキュー・マネージャー、MQオブジェクトの管理機能を提供
        - クライアント側へのコンポーネントのインストール不要
      - 以下の機能を提供
        - キュー・マネージャーおよびMQオブジェクトの管理
        - リソースモニター
          - CPUやディスク使用率のモニタリング、APIコールのモニタリング など
    - MQ Explorer
      - Eclipseベースの管理ツールでのキュー・マネージャー、MQオブジェクトの管理機能を提供
      - ※V9.3以降、MQ Explorerは製品に同梱されなくなったため、別途Fix Centralからのインストールが必要

# 管理REST API、MQ ConsoleおよびmqwebサーバーMQ Console

- 管理REST API、MQ Consoleはmqwebサーバー上で稼動するアプリケーション
  - mqwebサーバーの実体はWebsphere Liberty Profile(WLP)
  - mqwebサーバーはインストレーションごとに構成
    - 同一インストレーション内のキュー・マネージャーの管理が可能
  - セキュリティはmqwebuser.xmlで設定
    - mqwebサーバーを利用する管理REST API、MQ Consoleは共通の設定



# 管理インターフェースの提供機能

## ■ 各管理インターフェースが提供する機能

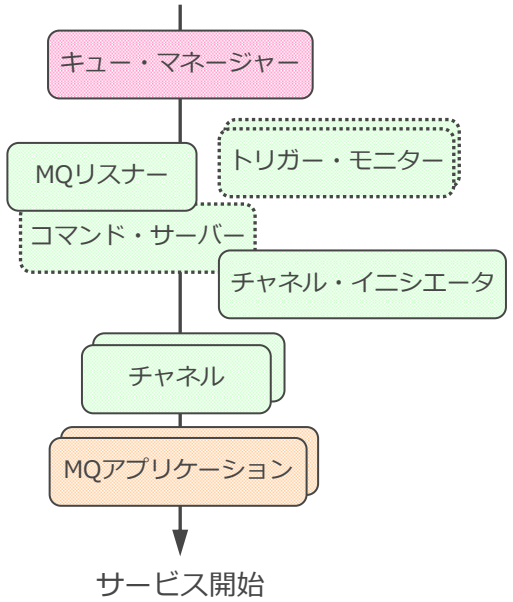
| 管理インターフェース | | リモート管理 | 古いバージョンのMQ管理 | 追加コンポーネント |
|---|---|---|---|---|
| 制御コマンド | | × | N/A | N/A |
| MQSCコマンド | | ○ | × | N/A |
| PCF | | ○ | ○ | N/A |
| 管理REST API | | ○ | × | mqweb (製品同梱) |
| GUIツール | MQ Console | ○ | ○ | mqweb (製品同梱) |
| | MQ Explorer | ○ | ○ | サポート・パックとして別途導入 |

# 起動・停止

# 起動処理

## MQの起動処理

- キュー・マネージャー、オプション・コンポーネント、チャネルを起動
- 通常は、以下の順序で起動
  1. キュー・マネージャー
  2. オプション・コンポーネント
     - MQリスナー、チャネル・イニシエータ、トリガー・モニター、コマンド・サーバー など
  3. チャネル
  4. MQアプリケーション




- 起動順序の制約
  - キュー・マネージャーが稼動していないと、オプション・コンポーネントとチャネルの起動は不可
    - キュー・マネージャーを除いた他のコンポーネントの起動順序は任意で、運用要件に応じて起動順序の入れ替えが可能
  - オプション・コンポーネントはコマンドで起動、もしくはキュー・マネージャー起動時に自動起動させることが可能

# 起動処理

## MQの起動方法

| 対象 | | 制御コマンド | MQSCコマンド | 説明/備考 |
|---|---|---|---|---|
| キュー・マネージャー | | strmqm | N/A | キュー・マネージャーの起動<br>(Windowsのみ、amqmdainコマンドが使用可能) |
| オプション・コンポーネント | MQリスナー | runmqlsr *1 *2 | START LISTENER *3 *5 | リスナーの起動 |
| | チャネル・イニシエータ | runmqchi *1 | START CHINIT *5 | チャネル・イニシエータの起動 |
| | トリガー・モニター | runmqtrm *1 *2 | START SERVICE(name) *4 *5 | サービス定義したトリガー・モニターの起動 |
| | コマンド・サーバー | strmqcsv | START SERVICE(name) *4 *5 | サービス定義したコマンド・サーバーの起動 |
| チャネル | | runmqchl *1 | START CHANNEL | チャネルの起動 |

*1）制御コマンドで起動すると各コンポーネントは終了するまで制御が戻らないため、バックグラウンド・プロセスとして起動する必要がある
*2）通常、Windows版のMQではコマンドでの起動はせず、キュー・マネージャーでの自動起動
*3）リスナー・オブジェクトを作成した場合
*4）サービス・オブジェクトを作成した場合
*5）自動起動させることも可能

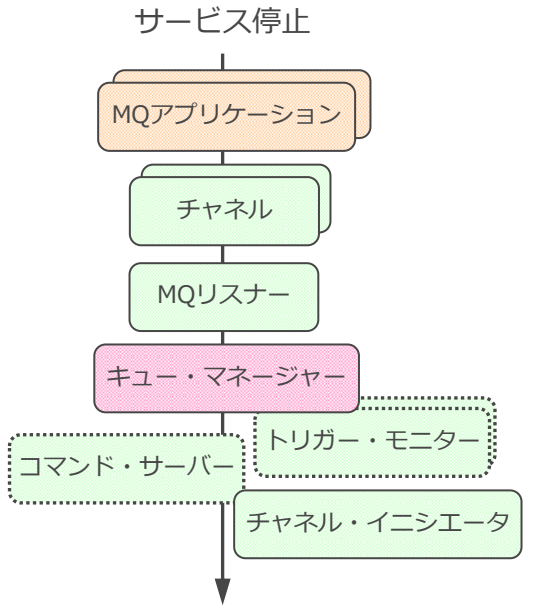
# 起動処理

## ■ 起動成否の確認方法

| 対象 | | 制御コマンドでの開始時 | MQSCコマンドでの開始時 | 備考 (確認対象プロセス例) |
|---|---|---|---|---|
| キュー・マネージャー | | • コマンドのリターン・コード<br>• プロセスの有無<br>• dspmq | N/A | amqzxma0 : 実行コントローラー<br>amqzmuc0 : 重要なプロセス・マネージャー<br>amqzfuma : OAMプロセス<br>amqxssvn (Windowsのみ) : 共用メモリ・サーバー<br>など |
| オプション・コンポーネント | MQリスナー | • コマンドのリターン・コード<br>• プロセスの有無 | DISPLAY LSSTATUS | runmqlsr : MQリスナー |
| | チャネル・イニシエータ | • コマンドのリターン・コード<br>• プロセスの有無 | N/A | runmqchi : チャネル・イニシエータ |
| | トリガー・モニター | • コマンドのリターン・コード<br>• プロセスの有無 | N/A | runmqtrm : トリガー・モニター |
| | コマンド・サーバー | • コマンドのリターン・コード<br>• プロセスの有無 | N/A | amqpcsea : コマンド・サーバー |
| チャネル | | • コマンドのリターン・コード<br>• DISPLAY CHSTATUS | DISPLAY CHSTATUS | - |

# 停止処理

- MQの停止処理
  - ◆ キュー・マネージャー、オプション・コンポーネント、チャネルを停止
  - ◆ 通常は、以下の順序で停止
    1. MQアプリケーション
    2. チャネル
    3. MQリスナー (制御コマンドで起動した場合)
    4. キュー・マネージャー
       - オプション・コンポーネントは自動停止
  - ◆ 停止順序の制約
    - コンポーネントの停止順序は任意で、運用要件に応じて停止順序の入れ替えが可能



# 停止処理

## MQの停止方法

| 対象 | | 制御コマンド | MQSCコマンド | 説明/備考 |
|---|---|---|---|---|
| チャネル | | N/A | STOP CHANNEL | チャネルの停止 |
| キュー・マネージャー | | endmqm | N/A | キュー・マネージャーの停止<br>(Windowsのみ、amqmdainコマンドが使用可能) |
| オプション・コンポーネント | MQリスナー | endmqlsr *1 | STOP LISTENER | MQリスナーの停止 |
| | チャネル・イニシエータ | N/A | N/A | キュー・マネージャー停止時に自動停止 |
| | トリガー・モニター | N/A | N/A | |
| | コマンド・サーバー | endmqcsv | N/A | コマンド・サーバーの停止 *2 |

*1）制御コマンドrunmqlsrで起動した場合
*2）通常は、キュー・マネージャー停止時に自動停止

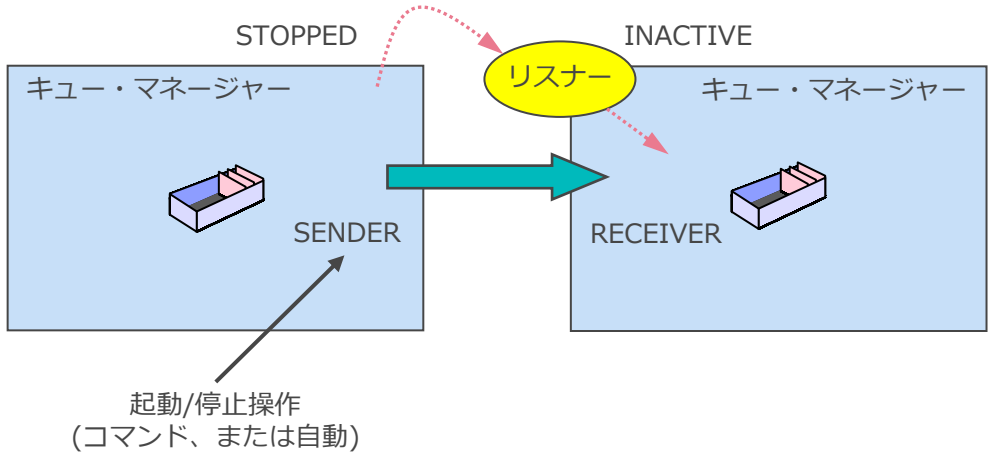
# 停止処理

## ■ 停止成否の確認方法

| 対象 | | 制御コマンドでの停止時 | MQSCコマンドでの停止時 | 備考 |
|---|---|---|---|---|
| チャネル | | N/A | DISPLAY CHSTATUS | |
| キュー・マネージャー | | • コマンドのリターン・コード<br>• プロセスの有無 | N/A | MQ関連の全てのプロセスの停止を確認 |
| オプション・コンポーネント | MQリスナー | • コマンドのリターン・コード<br>• プロセスの有無 | DISPLAY LSSTATUS | |
| | チャネル・イニシエータ | N/A | N/A | |
| | トリガー・モニター | N/A | N/A | |
| | コマンド・サーバー | • コマンドのリターン・コード<br>• プロセスの有無 | N/A | |

# ＜参考＞チャネルの起動・停止

- チャネル運用
  - ◆ チャネルの起動/停止はSENDER側から行う
    - ● SENDERを起動すると、リスナーにチャネル起動要求を送信し、リスナーがRECEIVERを起動
      - RECEIVERはINACTIVEになっている必要がある
    - ● SENDERを停止すると、RECEIVERも自動で停止
      - RECEIVERから停止した場合、送信側チャネルは「RETRYING」
  - ◆ コマンド(手動)での起動/停止、自動起動/自動停止のどちらも可能



# ＜参考＞チャネル・ステータス

## ■ 送信側チャネル (SENDER、SERVER)

| ステータス | 当ステータスへの遷移、考慮事項 | キュー・マネージャー<br>再起動後のステータス |
|---|---|---|
| RUNNING | ● 「RUNNING」への遷移<br>- チャネルの起動が成功（START CHANNELコマンド、トリガー起動、チャネル・イニシエータからの再試行）<br>● 考慮事項<br>- チャネルの起動が失敗するとRETRYINGになる | INACTIVE |
| INACTIVE | ● 「INACTIVE」への遷移<br>- チャネル稼動中(RUNNING)にキュー・マネージャーを再起動<br>- STOP CHANNEL STATUS(INACTIVE)コマンドでチャネルを停止<br>- 満了時間(DISCINT)の経過でチャネルが自動停止 | INACTIVE |
| STOPPED | ● 「STOPPED」への遷移<br>- STOP CHANNEL STATUS(STOPPED)コマンドでチャネルを停止<br>- チャネル・イニシエータからの再試行の満了<br>● 考慮事項<br>- トリガーによるチャネルの自動起動は不可、START CHANNELコマンドが必要<br>- チャネル起動要求の拒否（受信側がREQUESTERのとき） | STOPPED |
| RETRYING | ● 「RETRYING」への遷移<br>- チャネルの起動が失敗<br>- 稼動中(RUNNING)に以下の事象が発生<br>・RECEIVERが異常終了 *1（次ページを参照)<br>・ネットワーク障害<br>・受信側チャネルを停止、受信側キュー・マネージャーを停止<br>● 考慮事項<br>- チャネル・イニシエータが稼動していないと、RETRYINGステータスにならない<br>・RETRYINGの代わりにINACTIVEになる<br>- 接続先情報を含まないSERVERチャネルでは、RETRYINGステータスはない<br>・RETRYINGの代わりにINACTIVEになる | RETRYING |

# ＜参考＞チャネル・ステータス

## ■ 受信側チャネル (RECEIVER、REQUESTER)

| ステータス | 当ステータスへの遷移、考慮事項 | キュー・マネージャー<br>再起動後のステータス |
|---|---|---|
| RUNNING | ● 「RUNNING」への遷移<br>- チャネルの起動が成功<br>● 考慮事項<br>- REQUESTERでは受信側からチャネル起動が可能 | INACTIVE |
| INACTIVE | ● 「INACTIVE」への遷移<br>- RECEIVERではSTART CHANNELコマンドでチャネルを開始（STOPPEDの解除）<br>- チャネル稼動中(RUNNING)にキュー・マネージャーを再起動<br>- STOP CHANNEL STATUS(INACTIVE)コマンドでチャネルを停止<br>・送信側チャネルはRETRYING、チャネル・イニシエータからの再試行で再接続<br>- 稼動中(RUNNING)に以下の事象が発生<br>・異常終了 *1<br>・ネットワーク障害<br>・送信側チャネルを停止、送信側キュー・マネージャーを停止<br>● 考慮事項<br>- 起動可能（送信側チャネルからの起動要求によって起動可能） | INACTIVE |
| STOPPED | ● 「STOPPED」への遷移<br>- STOP CHANNEL STATUS(STOPPED)コマンドでチャネルを停止<br>● 考慮事項<br>- 起動要求の拒否、送信側チャネルはRETRYING<br>- RECEIVERではSTART CHANNELコマンドでINACTIVEになる<br>- REQUESTERでは起動要求を送信、接続が成功するとRUNNINGになる | STOPPED |

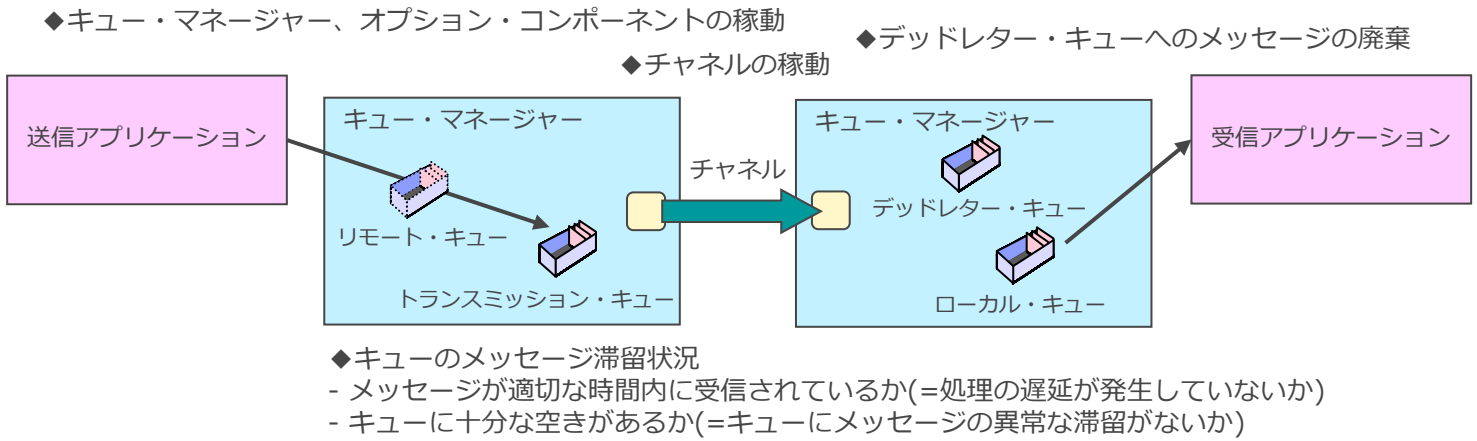
*1）宛先キューへのメッセージのPUTが失敗したときの動作

1. デッドレター・キューにPUT（受信側キュー・マネージャーにデッドレター・キューがある場合）
2. デッドレター・キューがない、デッドレター・キューへのPUTが失敗した時の動作
   2-1. チャネルのタイプがFASTでノンパーシステント・メッセージを転送 → メッセージ消失
   2-2. チャネルのタイプがNORMAL、またはパーシステント・メッセージを転送 → 異常終了

# 監視

# 監視概要

- メッセージ送受信のための基本的な監視項目
  - ◆ キュー・マネージャー、オプション・コンポーネントの稼動
  - ◆ チャネルの稼動
  - ◆ キューのメッセージ滞留状況
    - 適切な時間内におけるメッセージの受信、キューの十分な空き など
  - ◆ デッドレター・キュー(DLQ)へのメッセージの破棄
    - 破棄された旨のイベントやエラー・メッセージはないため、デッドレター・キュー・ハンドラーでの処理
- 主なオプションの監視項目
  - ◆ アクセス権限違反 (不正ユーザーからのアクセスなど)
  - ◆ キュー・アクセス・エラー (PUT/GET禁止など)
  - ◆ パフォーマンス　など
- 監視方法
  - ◆ オンライン・モニタリング
  - ◆ イベント・モニタリング
  - ◆ システム・トピックを使用した監視
  - ◆ エラー・ログ監視



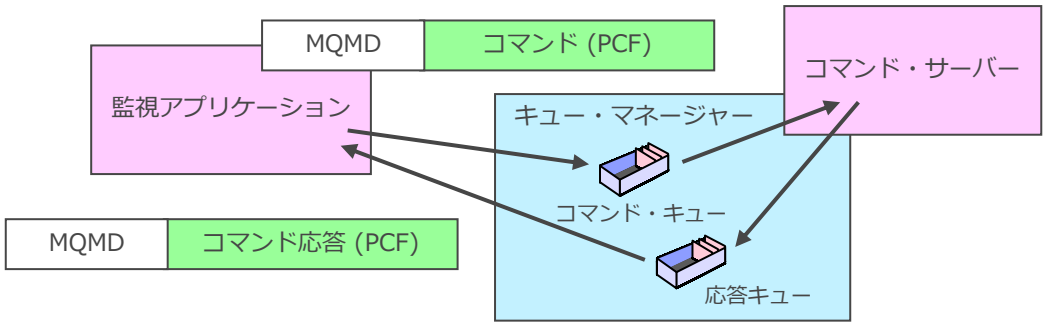
# オンライン・モニタリング

- 監視アプリケーションにて、チャネルの稼動状況やキューの滞留状況などの各種ステータスを定期的に照会
  - ◆ チャネル・ステータスとキュー・ステータスには、スイッチをONにしないと照会できない項目がある
- 実装方法
  - ◆ MQSCコマンド
    - 監視アプリケーションはシェル・スクリプトで実装
    - キュー・マネージャーとの接続・切断が都度発生するため、監視間隔が短いと負荷が上がる
    - 照会結果のパラメータ・キーワードの位置は、MQのバージョンアップなどにより変動する可能性がある

<DISPLAY QSTATUSコマンドの実行例>

```
DISPLAY QSTATUS(Q1) ALL
AMQ8450I: キュー状況の詳細を表示します。
QUEUE(Q1)                       TYPE(QUEUE)
CURDEPTH(0)                     CURFSIZE(1)
CURMAXFS(2088960)               IPPROCS(0)
LGETDATE(2023-08-01)            LGETTIME(14.11.26)
LPUTDATE(2023-08-01)            LPUTTIME(13.44.01)
MEDIALOG( )                     MONQ(HIGH)
MSGAGE(0)                       OPPROCS(0)
QTIME(116609155, 116609155)     UNCOM(NO)
```

  - ◆ PCFコマンド
    - 監視アプリケーションはプログラムで実装
    - 接続・切断を繰り返さない常駐型で作成することができる



# オンライン・モニタリング

## ■ ステータス照会のMQSCコマンド

| MQSCコマンド | 主な項目 |
|---|---|
| DISPLAY QMSTATUS | チャネル・イニシエータの稼動状況、コマンド・サーバーの稼動状況 |
| DISPLAY LSSTATUS | リスナーの稼動状況 |
| DISPLAY SVSTATUS | サービス定義したアプリケーション、オプション・コンポーネントの稼動状況 |
| DISPLAY CONN | 接続状況 (接続しているアプリケーションの一覧) |
| DISPLAY CONN TYPE(HANDLE) | オープン状況 (アプリケーションがオープンしているキューの一覧) |
| DISPLAY QSTATUS | メッセージの滞留状況 |
| DISPLAY QSTATUS TYPE(HANDLE) | オープン状況 (キューをオープンしているアプリケーションの一覧) |
| DISPLAY CHSTATUS | チャネルの稼動状況 |
| DISPLAY APSTATUS | アプリケーションの稼動状況 |
| DISPLAY PUBSUB | PUB/SUBの稼動状況 |
| DISPLAY SBSTATUS | サブスクリプションの状況 |
| DISPLAY TPSTATUS | トピックの状況 |

# オンライン・モニタリング

- ステータス照会のPCFコマンド
  - ステータス照会系コマンドは、コマンド・サーバー経由でも実行可能

| PCFコマンド | 説明 |
|---|---|
| MQCMD_INQUIRE_Q_MGR_STATUS | DISPLAY QMSTATUSに相当 |
| MQCMD_INQUIRE_LISTENER_STATUS | DISPLAY LSSTATUSに相当 |
| MQCMD_INQUIRE_SERVICE_STATUS | DISPLAY SVSTATUSに相当 |
| MQCMD_INQUIRE_CONNECTION | DISPLAY CONN、DISPLAY CONN TYPE(HANDLE)に相当 |
| MQCMD_INQUIRE_Q_STATUS | DISPLAY QSTATUS、DISPLAY QSTATUS TYPE(HANDLE)に相当 |
| MQCMD_INQUIRE_CHANNEL_STATUS | DISPLAY CHSTATUSに相当 |
| MQCMD_INQUIRE_APPL_STATUS | DISPLAY APSTATUSに相当 |
| MQCMD_INQUIRE_PUBSUB_STATUS | DISPLAY PUBSUBに相当 |
| MQCMD_INQUIRE_SUB_STATUS | DISPLAY SBSTATUSに相当 |
| MQCMD_INQUIRE_TOPIC_STATUS | DISPLAY TPSTATUSに相当 |

# オンライン・モニタリング (モニター・スイッチ)

- モニター・スイッチの ON/OFF をオブジェクト属性で設定
  - ◆ キュー・マネージャー属性
    - MONQ (キュー・ステータス)、MONCHL (チャネル・ステータス)、MONACLS (自動定義クラスター送信チャネル・ステータス)
  - ◆ キュー属性
    - MONQ (キュー・ステータス)
  - ◆ チャネル属性
    - MONCHL (チャネル・ステータス)
- モニター・スイッチの ON/OFF で取得情報を制御
  - ◆ モニター・スイッチの ON
    - 上記のオブジェクト属性に LOW/MEDIUM/HIGH のいずれかを指定 (取得頻度)
      - ※ただし、キューのMONQ属性に関しては、LOW/MEDIUM/HIGH のどれを指定しても取得頻度は変わらない (実質的にはON/OFF)
  - ◆ モニター・スイッチの OFF
    - 上記のオブジェクト属性に OFF/NONE のいずれかを指定 (取得しない)
      - OFF：情報の取得を無効化
      - NONE：キューのMONQ属性、またはチャネルのMONCHL属性での設定に関わらず、強制的に情報の取得を無効化
        - ※キュー・マネージャーのMONQ属性とMONCHL属性でのみ指定可能な設定値
        - 例）チャネルのMONCHL属性に MEDIUM が設定されている場合でも、キュー・マネージャーのMONCHL属性に NONE が設定されていれば、チャネルのオンライン・モニタリングは無効

# オンライン・モニタリング (モニター・スイッチ)

## ■ スイッチの ON により取得できるステータス情報

### ◆ キュー・ステータス

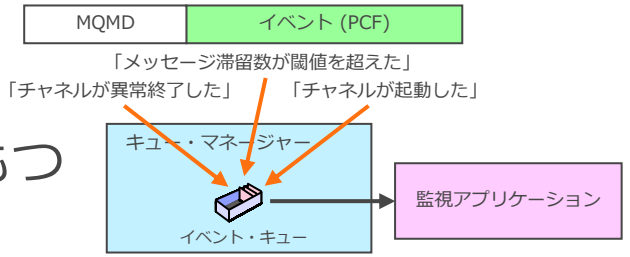
| キュー・ステータス属性 | 内容 |
|---|---|
| LPUTDATE | 最後にメッセージがPUTされた年月日 (YYYY-MM-DD) |
| LPUTTIME | 最後にメッセージがPUTされた時刻 (HH.MM.SS) |
| LGETDATE | 最後にメッセージがGETされた年月日 (YYYY-MM-DD) |
| LGETTIME | 最後にメッセージがGETされた時刻 (HH.MM.SS) |
| MONQ | 現在のモニター・レベル |
| MSGAGE | 最も滞留しているメッセージの滞留時間 (秒) |
| QTIME | メッセージの平均滞留時間 (マイクロ秒) |

### ◆ チャネル・ステータス

| チャネル・ステータス属性 | 内容 |
|---|---|
| COMPRATE | メッセージの圧縮率 |
| COMPTIME | メッセージの圧縮または展開にかかった時間 (マイクロ秒) |
| EXITTIME | EXITの実行にかかった時間 (マイクロ秒) |
| MONCHL | 現在のモニター・レベル |
| NETTIME | ネットワークの平均転送時間 (マイクロ秒) |
| XBATCHSZ | 1つのバッチ送信に含まれる平均メッセージサイズ |
| XQMSGSA | 転送待ちのメッセージ数 |
| XQTIME | トランス・ミッション・キューの平均滞留時間 (マイクロ秒) |

# イベント・モニタリング

- エラー、警告、その他のイベント事象をMQメッセージで報告
  - ◆ 監視アプリケーション(PCF)は各イベント・キューからのメッセージを受信し、異常があれば通知
    - ● ※複数の監視アプリケーションがある場合は、Pub/Sub機能によってイベント・メッセージをコピーさせることが可能
      - https://www.ibm.com/docs/ja/ibm-mq/9.3?topic=monitoring-publishing-your-mq-event-messages
    - ● 照会コマンドの定期的な実行よりも、即座にエラー事象を検知可能
    - ● 監視対象オブジェクトが増えてもキュー・マネージャーへの負担が少ない
- イベントは以下の6つのカテゴリに分類され、個別にイベント・キューをもつ
  - ◆ イベント・タイプによってイベントの ON/OFF が切り替え可能




| イベント・カテゴリ | イベント・キュー | 事象内容によるイベント・タイプ (イベント・スイッチの単位) |
|---|---|---|
| キュー・マネージャー・イベント | SYSTEM.ADMIN.QMGR.EVENT | 開始/停止イベント、禁止イベント、権限イベント、ローカル・イベント、リモート・イベント<br>(キュー・マネージャーの起動/停止、PUT/GET禁止キューへのPUT/GETが実行されたときなどにイベント発生) |
| チャネル・イベント | SYSTEM.ADMIN.CHANNEL.EVENT | チャネル・イベント、SSLイベント、チャネル自動定義イベント<br>(チャネルの起動/停止、SSL/TLSセッションの確立が失敗したときなどにイベント発生) |
| パフォーマンス・イベント | SYSTEM.ADMIN.PERFM.EVENT | キュー・サイズ・イベント、キュー・サービス間隔イベント<br>(キューに異常なメッセージ滞留があるとき、メッセージが定期的に受信されていないときなどにイベント発生) |
| 構成イベント | SYSTEM.ADMIN.CONFIG.EVENT | オブジェクト作成イベント、オブジェクト変更イベント、オブジェクト削除イベント、オブジェクト・リフレッシュ・イベント<br>(MQオブジェクトの作成、変更、削除、リフレッシュをしたときにイベント発生)<br>※キュー・マネージャーの作成/削除ではイベントは発生しない |
| コマンド・イベント | SYSTEM.ADMIN.COMMAND.EVENT | コマンド・イベント<br>(MQSC および PCFコマンドが正常に実行されたときなどにイベント発生) |
| ロガー・イベント | SYSTEM.ADMIN.LOGGER.EVENT | ロガー・イベント<br>(キュー・マネージャーが新しいログ・ファイルへの書き込みを開始したときなどにイベント発生)<br>※リニア・ログの場合のみ設定可能 |

# <参考> イベント・モニタリング (イベント・スイッチ)

- イベント・スイッチ
  - ◆ 以下のイベント・カテゴリは、キュー・マネージャー属性で DISABLED/ENABLED を設定
    - キュー・マネージャー・イベント
    - チャネル・イベント
    - 構成イベント
    - コマンド・イベント
    - ロガー・イベント
  - ◆ パフォーマンス・イベントは、キュー・マネージャー属性とキュー属性の両方で設定が必要
  - ◆ チャネル・イベントは、キュー・マネージャーのCHLEV属性で EXCEPTION が設定可能
    - EXCEPTION
      - エラー事象のみを報告させる
  - ◆ コマンド・イベントは、キュー・マネージャーのCMDEV属性で NODISPLAY が設定可能
    - NODISPLAY
      - DISPLAYコマンド(MQSC)、および、照会コマンド(PCF)を除くコマンドが正常に実行されたときにイベントを生成

# <参考> イベント・モニタリング (イベント・スイッチ)

■ キュー・マネージャー属性の設定により、イベントの ON/OFF を切り替え

**太字**：デフォルト値

| イベント・カテゴリ | イベント・タイプ | キュー・マネージャー属性 |
|---|---|---|
| キュー・マネージャー・イベント | 開始・停止イベント | STRSTPEV (DISABLED \| **ENABLED**) |
| | 禁止イベント | INHIBTEV (**DISABLED** \| ENABLED) |
| | 権限イベント | AUTHOREV (**DISABLED** \| ENABLED) |
| | ローカル・イベント | LOCALEV (**DISABLED** \| ENABLED) |
| | リモート・イベント | REMOTEEV (**DISABLED** \| ENABLED) |
| チャネル・イベント | チャネル・イベント | CHLEV (**DISABLED** \| ENABLED \| EXCEPTION) |
| | SSLイベント | SSLEV (**DISABLED** \| ENABLED) |
| | チャネル自動定義イベント | CHADEV (**DISABLED** \| ENABLED) |
| パフォーマンス・イベント | キュー・サイズ・イベント、<br>キュー・サービス間隔イベント | PERFMEV (**DISABLED** \| ENABLED) |
| 構成イベント | オブジェクト作成イベント | CONFIGEV (**DISABLED** \| ENABLED) |
| | オブジェクト変更イベント | |
| | オブジェクト削除イベント | |
| | オブジェクト・リフレッシュ・イベント | |
| コマンド・イベント | コマンド・イベント | CMDEV (**DISABLED** \| ENABLED \| NODISPLAY) |
| ロガー・イベント | ロガー・イベント | LOGGEREV (**DISABLED** \| ENABLED) |

# <参考> イベント・モニタリング (イベント・スイッチ)

■ キュー属性の設定により、イベントの ON/OFF を切り替え

**太字**：デフォルト値

| イベント・カテゴリ | イベント・タイプ | キュー属性 |
|---|---|---|
| パフォーマンス・イベント | キュー・サイズ・イベント | QDPLOEV (**DISABLED** \| ENABLED) |
| | | QDPHIEV (**DISABLED** \| ENABLED) |
| | | QDPMAXEV (DISABLED \| **ENABLED**) |
| | | QDEPTHLO (**20** \| n) |
| | | QDEPTHHI (**80** \| n) |
| | | MAXDEPTH (**5000** \| n) |
| | キュー・サービス間隔イベント | QSVCIEV (**NONE** \| HIGH \| OK) |
| | | QSVCINT (**999999999** \| n) |

# <参考> イベント・モニタリング (パフォーマンス・イベント)

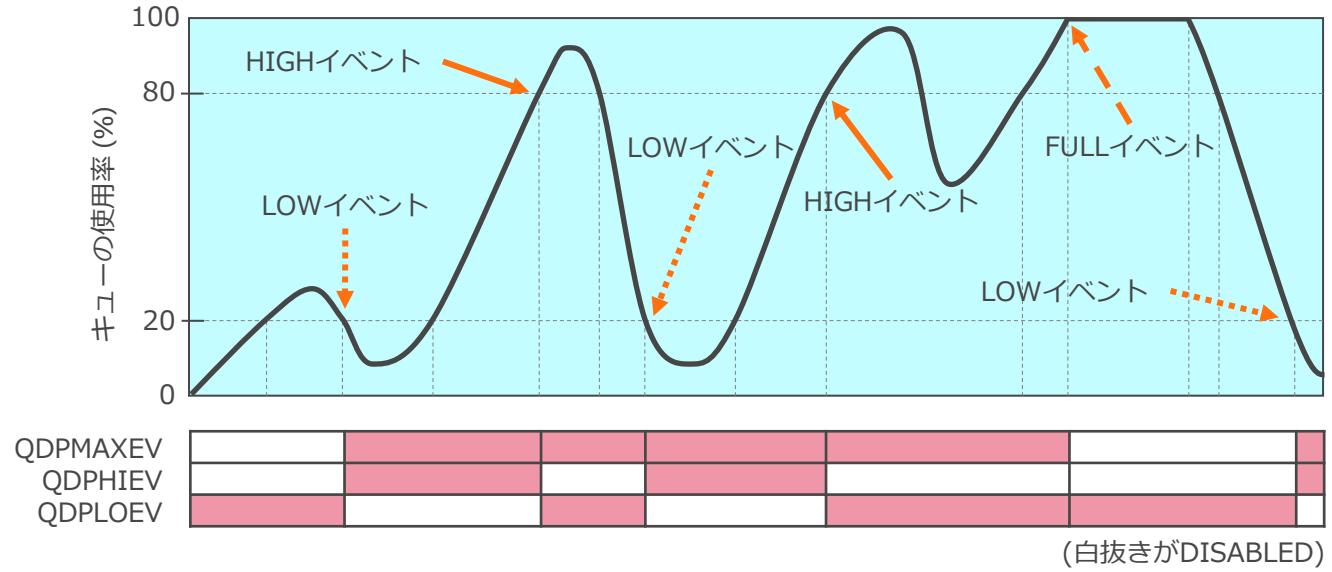
## ■ キュー・サイズ・イベント

- ◆ キューに異常なメッセージ滞留がないか、十分な空きがあるかを監視するイベント
- ◆ 最大滞留数(MAXDEPTH)と閾値(QDEPTHHI、QDEPTHLO)をもとに、3種のイベントを生成
  - ● キュー・サイズ上限イベント(HIGHイベント)：上限閾値を超えてメッセージが滞留
  - ● キュー満杯イベント(FULLイベント)：キューが満杯
  - ● キュー・サイズ下限イベント(LOWイベント)：メッセージの滞留が収束

- イベント生成の閾値となるQDEPTHHIおよびQDEPTHLOには、MAXDEPTHに対する割合(%)を設定
- 各々のイベントがスイッチをもち、イベント生成時にスイッチを自動的に切り替え
  (短時間における同イベントの繰り返し生成を回避)

<初期設定の例>

・QDPHIEV(DISABLED)
・QDPLOEV(ENABLED)
・QDPMAXEV(DISABLED)
・QDEPTHHI(80)
・QDEPTHLO(20)
・MAXDEPTH(1000)



# <参考> イベント・モニタリング (パフォーマンス・イベント)

## ■ キュー・サービス間隔イベント

◆ メッセージが定期的に受信されているか、受信処理の遅延はないかを監視するイベント

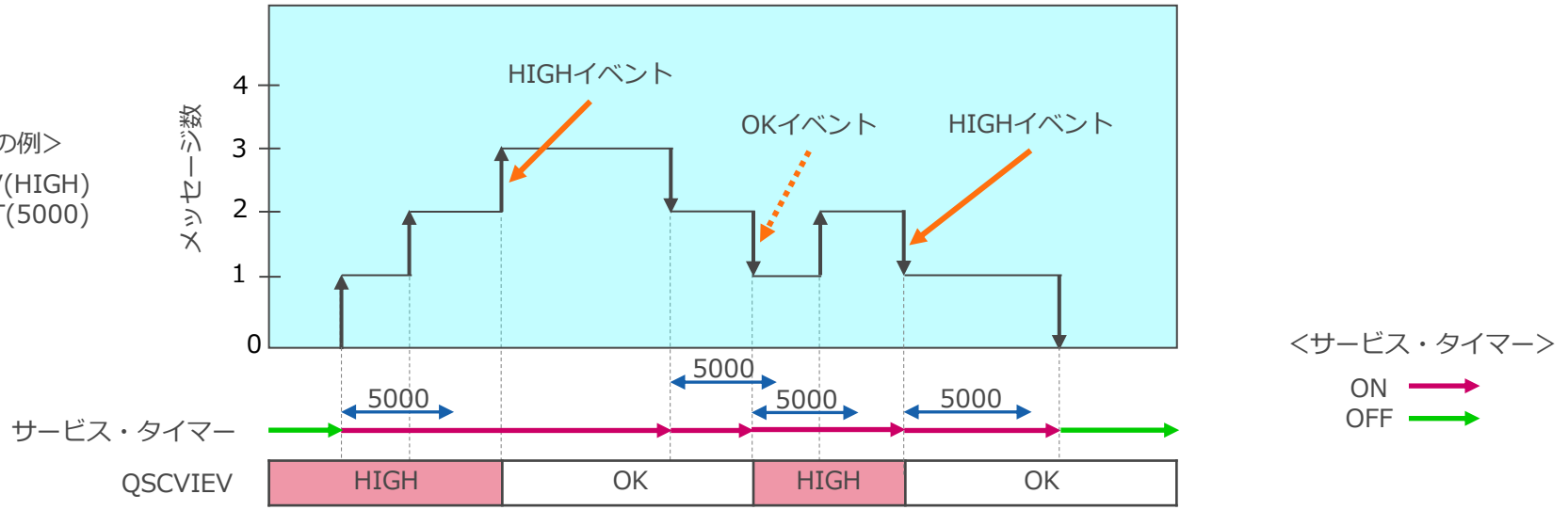
◆ インターバル閾値(QSVCINT)をもとに2種のイベントを生成

- キュー・サービス間隔上限イベント(HIGHイベント)：インターバル内にメッセージが受信されなかった
- キュー・サービス間隔OKイベント(OKイベント)：インターバル内にメッセージが受信された

- イベント生成の閾値となるQSVCINTには、時間(ミリ秒)を設定
- 1件目のMQPUTでサービス・タイマーがONになり、MQGETの度にサービス・タイマーが初期化される
  (サービス・タイマーは、1件以上のメッセージがキューにあるときMQGET～MQGETの時間を計測)
- サービス・タイマー ＜ QSVCINT でMQGETが行われると、OKイベントを生成
- サービス・タイマー ＞ QSVCINT でMQGET、またはMQPUTが行われると、HIGHイベントを生成
  (MQGETが行われないときでもHIGHイベントが生成されるように、MQPUTでもイベントを生成)
- イベント生成時にスイッチを自動的に切り替え (短時間における同イベントの繰り返し生成を回避)

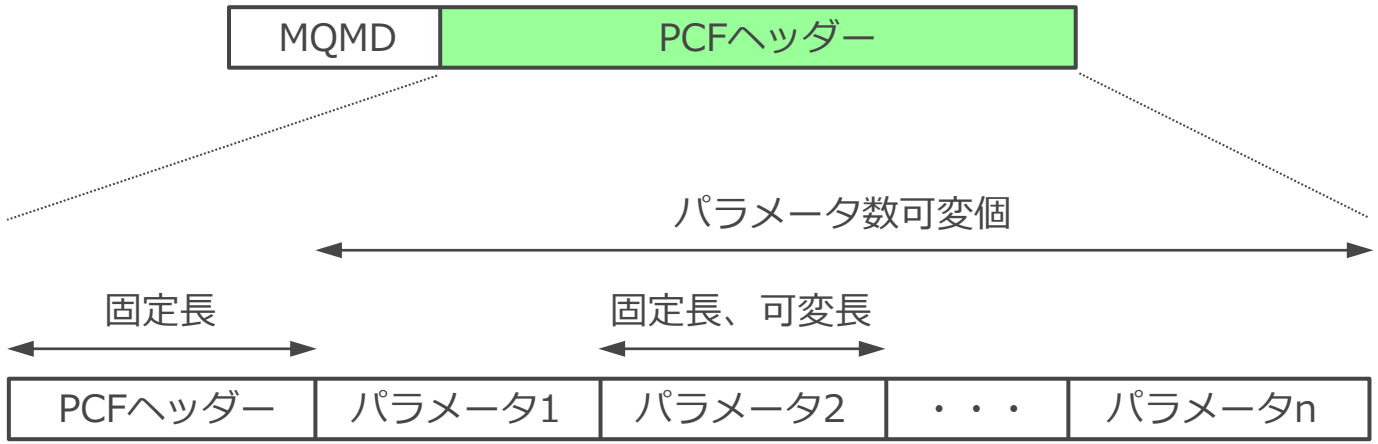
<初期設定の例>

・QSVCIEV(HIGH)
・QSVCINT(5000)



# ＜参考＞PCFメッセージ

- PCF(Programable Command Format)メッセージ
  - ◆ コマンド・サーバー経由のオンライン・モニタリング、イベント・モニタリングを行う監視アプリケーションは、PCFメッセージの組み立て/分解が必要
  - ◆ PCFメッセージは、可変長データ
    - PCFメッセージを扱うためのAPIを提供




  - ◆ PCFを使用してMQ管理アプリケーションを作成する場合
    - コマンド入力キューにPCFメッセージを書き込む
      - MQPUTの宛先キュー：SYSTEM.ADMIN.COMMAND.QUEUE
    - MQGETでReply-toキューに戻ったPCFメッセージを解析する

# システム・トピックを使用した監視

- キュー・マネージャーがシステム・トピックに対してパフォーマンス情報をパブリッシュ
  - ユーザーがシステムトピックに対してサブスクライブし、情報を受け取ることが可能
  - これまでのキューイング型ではなく、パブリッシュ・サブスクライブ(Pub/Sub)型の情報発信
  - メッセージ・フォーマットは、従来通りPCF
- 使用するトピック
  - トピック・ストリング：$SYS/MQ
  - 管理オブジェクト：SYSTEM.ADMIN.TOPIC
- キュー・マネージャーがパブリッシュする情報
  - リソース・モニター
    - キュー・マネージャー稼動によるシステム・リソースの使用状況
    - MQ会計・統計と同等の情報
    - サンプル・アプリケーション
      - amqsruaコマンドを使用してサブスクライブ可能
  - アクティビティー・トレース
    - キュー・マネージャー上で稼動するアプリケーションのアクティビティ情報 (発行したMQIやオプションなど)
    - サンプル・アプリケーション
      - amqsactコマンドを使用してサブスクライブ可能

# システム・トピック ($SYS/MQ)

- $SYS/MQは完全に独立したトピック・ツリー
  - 上位からのセキュリティおよび属性は引き継がない
- 上位のトピック・ツリー(/#)に対するワイルドカードでのサブスクリプションでは、サブスクライブ不可
  - ワイルドカードでのサブスクリプションでは、＄SYS/MQを含むトピック・ツリーにマッチしない
  - SYSTEM.ADMIN.TOPICのWILDCARD属性はBLOCKに設定され、変更不可
  - runmqscコマンドでのワイルドカード使用にも該当するため、DISPLAYコマンド使用時には注意
    - $SYS/MQ以下のトピック・ストリングをDISPLAYするには、$SYS/MQ/# を使用
- $SYS/MQの制限事項
  - キュー・マネージャーのみパブリッシュ可能
    - ユーザー・アプリケーションは、$SYS/MQトピック・ストリングにパブリッシュ不可
  - $SYS/MQ以下のトピックをクラスターに公開することは不可
  - トピック・オブジェクトのPROXYSUB属性をFORCEに設定することは不可
  - パブリッシュおよびサブスクライブのスコープはLOCALに設定

# リソース・モニター情報

- リソース・モニター情報のパブリッシュ
  - キュー・マネージャーは起動時からリソース・モニター情報をパブリッシュ
    - パブリッシュ先のトピック

      $SYS/MQ/INFO/QMGR/*qmgr_name*/Monitor/*class*[/*instance*]/*type*
  - パブリッシャーの起動/停止は不可、パブリッシュ間隔は約10秒
- パブリケーションの受信方法
  - ユーザーによる管理アプリケーションを開発
  - サンプル提供のamqsruaコマンドを使用
- amqsruaコマンドの仕様
  - コマンドのパラメータで指定したトピックに非継続サブスクリプションを作成
    - 動的キューを経由してサブスクライブ
  - トピックに対してMQGETし、メッセージを標準出力に送信
  - 指定した数のパブリケーションを受信、またはエラーが発生(MQCC_FAILED)したら停止
  - エラー時にはReasonCodeを出力 (MQRC_NONEを除く)
  - サンプル・ソース・コード: amqsruaa.c
    - UNIX/Linux : *MQ_INSTALLATION_PATH*/samp/
    - Windows : *MQ_INSTALLATION_PATH*¥tools¥c¥Samples¥

<$SYS/MQブランチへのアクセスに必要なOAM権限 (amqsruaの場合)>
- キュー・マネージャーへの connect権限、inq権限
- SYSTEM.DEFAULT.MODEL.QUEUEへの get権限、dsp権限
- SYSTEM.ADMIN.TOPICへの sub権限

補足）チャネル認証、接続認証を使用している場合は必要な権限を付与

# アクティビティー・トレース機能

- アクティビティー・トレース機能
  - アプリケーション内のMQI詳細をPCFメッセージとして生成し、SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUEに書き込む
  - トレース取得の詳細は mqat.ini に指定
- V9.0以降、キュー・マネージャーによるアクティビティー・トレース情報のパブリッシュも可能
  - サブスクライバーが稼動している間、トレース情報をサブスクライバー・キューに送信
  - サブスクライバーが停止/終了したら、パブリッシュを停止
  - パブリッシュ先のトピック

    $SYS/MQ/INFO/QMGR/*qmgr_name*/ActivityTrace/*resource_type*/*resource_identifier*

- サブスクライバーは、リソース・タイプとリソースIDを指定して受信
  - リソース・タイプ：以下のいずれかを指定
    - アプリケーション名 (ApplName)
    - チャネル名 (ChannelName)
    - 接続ID (ConnectionId)
  - リソースID
    - 指定したリソース・タイプに対して、個別名または総称名を指定

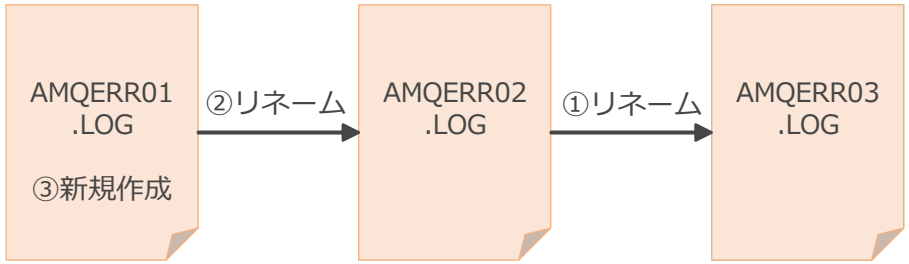
# アクティビティー・トレース機能

- トレース情報のサブスクライブ方法
  - ユーザーによる管理アプリケーションを開発
  - サンプル提供のamqsactコマンドを使用
    - 動的モード(ダイナミック・モード)
      - これまでのamqsactコマンドを拡張し、サブスクライバーとして使用可能
    - 照会モード
      - SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUEに書き込まれたメッセージを表示
  - サンプル・ソース・コード：amqsact0.c
    - UNIX/Linux : *MQ_INSTALLATION_PATH*/samp/
    - Windows : *MQ_INSTALLATION_PATH*¥tools¥c¥Samples¥

# エラー・ログ監視

- エラー、警告、その他のイベント事象をログ・ファイルにテキスト・メッセージで出力
  - ◆ メッセージは固有のID(AMQnnnnS)で識別可能
- エラー・ログを2箇所で保持
  - ◆ キュー・マネージャー名を特定できる事象の書き出し先
    - UNIX系：/var/mqm/qmgrs/<*QMNAME*>/errors
    - Windows：<*data_dir*>¥qmgrs¥<*QMNAME*>¥errors
  - ◆ その他の事象 (キュー・マネージャーの内部エラー、MQクライアント環境でのエラーを出力)
    - UNIX系：/var/mqm/errors
    - Windows：<*data_dir*>¥errors
    - 内部エラー発生時は、FDCファイルが上記ディレクトリ配下に出力される
- ログ・ファイル
  - ◆ 各ディレクトリにある3つのテキスト・ファイルを循環して利用
    - AMQERR01.LOG、AMQERR02.LOG、AMQERR03.LOG
    - 最新のメッセージは常にAMQERR01.LOGに出力
    - ファイルが一杯になるとリネーム
      - デフォルトのエラー・ログ・サイズは32MB (qm.iniファイルのQMErrorLogスタンザで変更可能)
  - ◆ Windowsではアプリケーション・ログへもメッセージを出力
    - OS提供のイベント・ビューアーで参照可能

① AMQERR02.LOGをAMQERR03.LOGにリネーム
② AMQERR01.LOGをAMQERR02.LOGにリネーム
③ AMQERR01.LOGを作成



# エラー・ログの重大度

- V9.1以降、エラー・ログやMQSCコマンドの出力等、すべての診断メッセージに重大度が付与される
  - メッセージIDに情報(I)、警告(W)、エラー(E)のような識別子が含まれる
    - AMQメッセージのみで、重大度の判断が可能

<MQSCコマンドの出力例>

```
DEF QL(Sample)
    1 : DEF QL(Sample)
AMQ8006I: IBM MQ queue created.
```

<エラー・ログの出力例>

```
----- amqzmgr0.c : 2996 --------------------------------------------------------
09/05/2023 08:23:10 AM - Process(9621.1) User(mqm) Program(strmqm)
                         Host(mqm4) Installation(Installation1)
                         VRMF(9.3.0.6) QMgr(QM1)
                         Time(2023-09-05T08:23:10.500Z)

AMQ9722W: Plain text communication is enabled.
```

| 重大度 | suffix |
|---|---|
| Information | I |
| Warning | W |
| Error | E |
| Stop Error | S |
| System Error | T |

- エラー・ログの考慮点
  - すべてのエラー事象がエラー・ログに記録されるわけではない
    - 「キューが満杯になりそう」 「キューが満杯になった」といった事象はエラー・ログには記録されない
      - エラー・ログだけでは十分な監視にはならない
  - エラー・ログが一杯になるとファイルが再作成されるため、監視アプリケーションは再作成のタイミングでファイルの再オープンが必要
  - エラー・メッセージには複数行で出力されるものがある
    - 80カラムで折り返し

# エラー・ログの出力

- エラー・ログをAMQERR0*.LOG以外に出力可能
  - 出力先(Service)は、file または syslog を選択可能
  - 出力フォーマット(Format)は、text または json を選択可能
  - 出力対象のログを重大度(Severities)で絞り込み可能
  - qm.iniファイルで設定

| Severitiesに指定可能な値 | | |
|---|---|---|
| Information | I | 0 |
| Warning | W | 10 |
| Error | E | 20<br>30 |
| Stop Error | S | 40 |
| System Error | T | 50 |

※プラス(+)を付与することで、指定した重大度レベルとそれより上位のレベルを表示することも可能

<qm.iniのDiagnosticMessagesスタンザサンプル>

```
DiagnosticMessages:
        Name=ErrorsToFile
        Service=file
        Severities=E
        FilePrefix=Errors
        Format=text
        FilePath=/var/mqm/qmgrs/QM1/errors
DiagnosticMessages:
        Name=NonErrorsToFile
        Service=file
        Severities=0
        FilePrefix=Information
DiagnosticMessages:
        Name=syslog
        Service=syslog
        Severities=I+
```

重大度がエラーの出力のみを、テキスト形式で/var/mqm/gmgrs/QM1/errorsにErrors0*.LOGとして出力

重大度がインフォメーションの出力のみを、JSON形式でInformation0*.jsonとして出力

重大度がインフォメーション以上の出力のみを、syslogに出力

# 会計・統計

# 会計・統計レポート

## ■ 会計・統計レポート概要

### ◆ キャパシティ・プランニングに利用可能な5種類の情報の出力が可能

| 情報 | 種別 | 説明 |
|---|---|---|
| 会計情報 | MQI会計 | MQCONN～MQDISCでアプリケーションが発行したMQIコールの種類・回数 など |
| | キュー会計 | MQCONN～MQDISCでアプリケーションが個々のキューにPUT/GETした回数・総バイト数 など |
| 統計情報 | MQI統計 | 計測インターバル内にキュー・マネージャー全体で発行されたMQIコールの種類・回数 など |
| | キュー統計 | 計測インターバル内に個々のキューで発生したメッセージの滞留数(最大/最小)・滞留時間の平均 など |
| | チャネル統計 | 計測インターバル内に個々のチャネルが転送したメッセージ数・総バイト数、1バッチの平均メッセージ数 など |

### ◆ 会計・統計情報は、キュー・マネージャーがPCFメッセージでキューに出力

| キュー名 | 種別 |
|---|---|
| SYSTEM.ADMIN.ACCOUNTING.QUEUE | MQI会計、キュー会計 |
| SYSTEM.ADMIN.STATISTICS.QUEUE | MQI統計、キュー統計、チャネル統計 |

### ◆ メッセージを出力するタイミング

- 会計メッセージ：MQDISC発行時に出力 (長時間発行がない場合、一定時間間隔で出力)
- 統計メッセージ：一定時間間隔で出力

### ◆ PCFメッセージを受信して解析し、レポートにまとめるプログラムをユーザーが作成する必要がある

- サンプル・ソース・コード amqsmon を提供 (テキストにダンプするだけ)

# 会計レポート

- MQI会計
  - MQアプリケーションごとのMQ API(MQI)の発行回数などの情報
  - 1つの会計メッセージに1つのMQI情報
  - MQI会計メッセージには、MQアプリケーション単位に以下の情報が含まれる
    - 接続情報 (アプリケーション名、プロセスID、接続ID、接続(切断)時間、… )
    - 発行したAPIの数 (MQOPEN、MQCLOSE、MQPUT、MQGET、MQCMIT、MQINQ、… )
    - PUT/GETしたメッセージ数・バイト数 (パーシステント/ノン・パーシステント)　など
  - MQI会計情報の取得を制御
    - キュー・マネージャーのACCTMQI属性で ON/OFF を設定
    - MQCONNXのコネクト・オプションでも設定が可能
      - アプリケーション単位でMQI会計の設定が可能
      - 優先順位：アプリケーションの設定 > キュー・マネージャーの設定
      - キュー・マネージャーのACCTCONO属性が ENABLED に設定されているときにのみ有効

# 会計レポート

- キュー会計
  - MQアプリケーションが使用したキューごとのPUT/GETされたメッセージ数などの情報
  - 1つの会計メッセージに最大100個のキュー情報
  - キュー会計メッセージには、MQアプリケーション単位で使用したキューごとに以下の情報が含まれる
    - キュー情報 (キュー名、キューの作成日時、キュー・タイプ、… )
    - 発行したAPIの数 (MQOPEN、MQCLOSE、MQPUT、MQGET、BROWSE、MQCB)
    - PUT/GETしたメッセージ数・バイト数 (パーシステント/ノン・パーシステント)・メッセージ・サイズ (最大/最小)
    - メッセージの滞留時間 (最大/平均/最小)　など
  - キュー会計情報の取得を制御
    - キューおよびキュー・マネージャーのACCTQ属性で設定
      - キューのACCTQ属性で ON/OFF/QMGR を設定
      - キュー・マネージャーのACCTQ属性で ON/OFF/NONE を設定
      - キューのACCTQ属性の設定が優先される
        - ※キューのACCTQ属性に QMGR が設定されている場合は、キュー・マネージャーのACCTQ属性の設定が適用される
        - ※キュー・マネージャーのACCTQ属性に NONE が設定されている場合は、キューのACCTQ属性での設定に関わらずOFF
    - MQCONNXのコネクト・オプションでも設定が可能
      - アプリケーション単位でキュー会計の設定が可能
      - 優先順位：キューの設定 > アプリケーションの設定 > キュー・マネージャーの設定
      - キュー・マネージャーのACCTCONO属性が ENABLED に設定されているときにのみ有効

# 統計レポート

- MQI統計
  - MQ API(MQI)の発行回数などの情報
  - 1つの統計メッセージに1つのMQI情報
  - MQI統計メッセージには、全てのアプリケーションに関する以下の情報が含まれる
    - 発行したAPIの数 (MQOPEN、MQCLOSE、MQPUT、MQGET、MQCMIT、MQINQ、… )
    - PUT/GETしたメッセージ数・バイト数 (パーシステント/ノン・パーシステント)　など
  - MQI統計情報の取得を制御
    - キュー・マネージャーのSTATMQI属性で ON/OFF を設定

# 統計レポート

- キュー統計
  - ◆ キューごとのPUT/GETされたメッセージ数、総バイト数などの情報
  - ◆ 1つの統計メッセージに最大100個のキュー情報
  - ◆ キュー統計メッセージには、キューごとに以下の情報が含まれる
    - キュー情報 (キュー名、キューの作成日時、キュー・タイプ、… )
    - PUT/GETしたメッセージ数・バイト数 (パーシステント/ノン・パーシステント)
    - 滞留メッセージ数 (最大/最小)
    - メッセージの平均滞留時間　など
  - ◆ キュー統計情報の取得を制御
    - キューおよびキュー・マネージャーのSTATQ属性で設定
      - キューのSTATQ属性で ON/OFF/QMGR を設定
      - キュー・マネージャーのSTATQ属性で ON/OFF/NONE を設定
      - キューのACCTQ属性の設定が優先される
        - ※キューのSTATQ属性に QMGR が設定されている場合は、キュー・マネージャーのSTATQ属性の設定が適用される
        - ※キュー・マネージャーのSTATQ属性に NONE が設定されている場合は、キューのSTATQ属性での設定に関わらずOFF

# 統計レポート

- チャネル統計
  - チャネルごとの送受信したメッセージ数、総バイト数などの情報
  - 1つの統計メッセージに最大100個のチャネル情報
  - チャネル統計メッセージには、チャネルごとに以下の情報が含まれる
    - チャネル情報 (チャネル名、チャネル・タイプ)
    - 転送したメッセージ数・バイト数 (パーシステント、ノン・パーシステント)
    - ハートビートのラウンド・トリップ・タイム (最大/平均/最小)　など
  - チャネル統計情報の取得を制御
    - チャネルおよびキュー・マネージャーのSTATCHL属性で設定
      - チャネルのSTATCHL属性で LOW/MEDIUM/HIGH/OFF/QMGR を設定
      - キュー・マネージャーのSTATCHL属性で LOW/MEDIUM/HIGH/OFF/NONE を設定
      - チャネルのSTATCHL属性の設定が優先される
        - ※チャネルのSTATCHL属性に QMGR が設定されている場合は、キュー・マネージャーのSTATCHL属性の設定が適用される
        - ※キュー・マネージャーのSTATCHL属性に NONE が設定されている場合は、チャネルのSTATCHL属性での設定に関わらずOFF
    - 自動定義クラスター送信チャネルは、キュー・マネージャーのSTATACLS属性で LOW/MEDIUM/HIGH/OFF/QMGR を設定
      - キュー・マネージャーのSTATACLS属性に QMGR が設定されている場合は、キュー・マネージャーのSTATCHL属性の設定が適用される
      - キュー・マネージャーのSTATCHL属性に NONE が設定されている場合は、キュー・マネージャーのSTATACLS属性での設定に関わらずOFF
    - 設定を適用するにはチャネルの再起動が必要

※属性の設定値 LOW/MEDIUM/HIGH は、
チャネル統計のサンプリング・レートの度合い

# ＜参考＞MQI会計で取得可能な情報

## ■ MQCONN~MQDISCでのアプリケーションの活動

### ◆ MQDISCのタイミングで下記メッセージを生成

| フィールド | 説明 |
|---|---|
| QueueManager | キュー・マネージャー名 |
| IntervalStartDate | 会計の取得を開始した日付 |
| IntervalStartTime | 会計の取得を開始した時刻 |
| IntervalEndDate | 会計の取得を終了した日付 |
| IntervalEndTime | 会計の取得を終了した時刻 |
| CommandLevel | コマンド・レベル |
| ConnectionId | 接続ID |
| SeqNumber | シーケンス番号 |
| ApplicationName | アプリケーション名 |
| ApplicationPid | アプリケーション・プロセスID |
| ApplicationTid | アプリケーション・スレッドID |
| UserId | ユーザーID |
| ConnDate | 接続日付 |
| ConnTime | 接続時刻 |
| ConnName | クライアント接続の接続名 |
| ChannelName | クライアント接続のチャネル名 |
| RemoteProduct | クライアント接続のリモート製品ID |
| RemoteVersion | クライアント接続のリモート製品バージョン |
| DiscDate | 切断日付 |
| DiscTime | 切断時刻 |
| DiscType | 切断タイプ |

| フィールド | 説明 |
|---|---|
| OpenCount | MQOPENが成功した回数 |
| OpenFailCount | MQOPENが失敗した回数 |
| CloseCount | MQCLOSEが成功した回数 |
| CloseFailCount | MQCLOSEが失敗した回数 |
| PutCount | MQPUTが成功した回数 |
| PutFailCount | MQPUTが失敗した回数 |
| Put1Count | MQPUT1が成功した回数 |
| Put1FailCount | MQPUT1が失敗した回数 |
| PutBytes | Putしたバイト数 |
| GetCount | MQGETが成功した回数 |
| GetFailCount | MQGETが失敗した回数 |
| GetBytes | Getしたバイト数 |
| BrowseCount | ブラウズが成功した回数 |
| BrowseFailCount | ブラウズが失敗した回数 |
| BrowseBytes | ブラウズしたバイト数 |
| CommitCount | MQCMITに成功した回数 |
| CommitFailCount | MQCMITに失敗した回数 |
| BackCount | MQBACKに成功した回数 |
| InqCount | MQINQが成功した回数 |
| InqFailCount | MQINQが失敗した回数 |
| SetCount | MQSETが成功した回数 |
| SetFailCount | MQSETが失敗した回数 |

| フィールド | 説明 |
|---|---|
| SubCountDur | 継続サブスクリプションに対する<br>サブスクライブが成功した回数 |
| SubCountNDur | 非継続サブスクリプションに対する<br>サブスクライブが成功した回数 |
| SubFailCount | サブスクライブが失敗した回数 |
| UnsubCountDur | 継続サブスクリプションに対する<br>アンサブスクライブが成功した回数 |
| UnsubCountNDur | 非継続サブスクリプションに対する<br>アンサブスクライブが成功した回数 |
| UnsubFailCount | アンサブスクライブが失敗した回数 |
| SubRqCount | MQSUBRQが成功した回数 |
| SubRqFailCount | MQSUBRQが失敗した回数 |
| CBCount | MQCBが成功した回数 |
| CBFailCount | MQCBが失敗した回数 |
| CtlCount | MQCTLが成功した回数 |
| CtlFailCount | MQCTLが失敗した回数 |
| StatCount | MQSTATが成功した回数 |
| StatFailCount | MQSTATが失敗した回数 |
| PutTopicCount | トピックに対するMQPUTが成功した回数 |
| PutTopicFailCount | トピックに対するMQPUTが失敗した回数 |
| Put1TopicCount | トピックに対するMQPUT1が成功した回数 |
| Put1TopicFailCount | トピックに対するMQPUT1が失敗した回数 |
| PutTopicBytes | トピックに対してPutしたバイト数 |

# ＜参考＞キュー会計で取得可能な情報

- MQCONN~MQDISCのアプリケーションの活動
  - ◆ MQDISCのタイミングで下記メッセージを生成
    - 会計を取得しているキューが複数あると、個々のキューに対して QAccountingData を作成
    - 最大100個のキューまで1つのメッセージで報告

| フィールド | 説明 |
|---|---|
| QueueManager | キュー・マネージャー名 |
| IntervalStartDate | 会計の取得を開始した日付 |
| IntervalStartTime | 会計の取得を開始した時刻 |
| IntervalEndDate | 会計の取得を終了した日付 |
| IntervalEndTime | 会計の取得を終了した時刻 |
| CommandLevel | コマンド・レベル |
| ConnectionId | 接続ID |
| SeqNumber | シーケンス番号 |
| ApplicationName | アプリケーション名 |
| ApplicationPid | アプリケーション・プロセスID |
| ApplicationTid | アプリケーション・スレッドID |
| UserId | ユーザーID |
| ChannelName | クライアント接続のチャネル名 |
| ConnName | 接続名 |
| ObjectCount | アクセスしたキューの数 |

| フィールド | 説明 |
|---|---|
| QAccountingData | (PCFグループ) |
| QNname | キュー名 |
| CreateDate | キューが作成された日付 |
| CreateTime | キューが作成された時刻 |
| QType | キュー・タイプ |
| QDefinitionType | キューの定義タイプ |
| OpenCount | MQOPENが成功した回数 |
| OpenDate | MQOPENした日付 |
| OpenTime | MQOPENした時刻 |
| CloseDate | MQCLOSEした日付 |
| CloseTime | MQCLOSEした時刻 |
| PutCount | MQPUTが成功した回数 |
| PutFailCount | MQPUTが失敗した回数 |
| Put1Count | MQPUT1が成功した回数 |
| Put1FailCount | MQPUT1が失敗した回数 |
| PutBytes | Putしたバイト数 |
| PutMinBytes | Putした最小バイト数 |
| PutMaxBytes | Putした最大バイト数 |

| フィールド | 説明 |
|---|---|
| GetCount | MQGETが成功した回数 |
| GetFailCount | MQGETが失敗した回数 |
| GetBytes | Getしたバイト数 |
| GetMinBytes | Getした最小バイト数 |
| GetMaxBytes | Getした最大バイト数 |
| BrowseCount | ブラウズが成功した回数 |
| BrowseFailCount | ブラウズが失敗した回数 |
| BrowseBytes | ブラウズしたバイト数 |
| BrowseMinBytes | ブラウズした最小バイト数 |
| BrowseMaxBytes | ブラウズした最大バイト数 |
| TimeOnQMin | メッセージの最小滞留時間 |
| TimeOnQAvg | メッセージの平均滞留時間 |
| TimeOnQMax | メッセージの最大滞留時間 |
| CBCount | MQCBが成功した回数 |
| CBFailCount | MQCBが失敗した回数 |

# ＜参考＞MQI統計で取得可能な情報

## ■ インターバル間でのキュー・マネージャーの活動

◆ インターバルごとに下記メッセージを生成

| フィールド | 説明 |
|---|---|
| QueueManager | キュー・マネージャー名 |
| IntervalStartDate | 統計の取得を開始した日付 |
| IntervalStartTime | 統計の取得を開始した時刻 |
| IntervalEndDate | 統計の取得を終了した日付 |
| IntervalEndTime | 統計の取得を終了した時刻 |
| CommandLevel | コマンド・レベル |
| ConnCount | MQCONN(X)が成功した回数 |
| ConnFailCount | MQCONN(X)が失敗した回数 |
| ConnsMax | 一時点での最大接続数 |
| DiscCount | MQDISCが成功した回数 |
| OpenCount | MQOPENが成功した回数 |
| OpenFailCount | MQOPENが失敗した回数 |
| CloseCount | MQCLOSEが成功した回数 |
| CloseFailCount | MQCLOSEが失敗した回数 |
| InqCount | MQINQが成功した回数 |
| InqFailCount | MQINQが失敗した回数 |
| SetCount | MQSETが成功した回数 |
| SetFailCount | MQSETが失敗した回数 |
| PutCount | MQPUTが成功した回数 |
| PutFailCount | MQPUTが失敗した回数 |
| Put1Count | MQPUT1が成功した回数 |
| Put1FailCount | MQPUT1が失敗した回数 |

| フィールド | 説明 |
|---|---|
| PutBytes | Putしたバイト数 |
| GetCount | MQGETが成功した回数 |
| GetFailCount | MQGETが失敗した回数 |
| GetBytes | Getしたバイト数 |
| BrowseCount | ブラウズが成功した回数 |
| BrowseFailCount | ブラウズが失敗した回数 |
| BrowseBytes | ブラウズしたバイト数 |
| CommitCount | MQCMITが成功した回数 |
| CommitFailCount | MQCMITが失敗した回数 |
| BackCount | MQBACKが成功した回数 |
| ExpiredMsgCount | 有効期限が切れたメッセージ数 |
| PurgeCount | パージされたメッセージ数 |
| SubCountDur | 継続サブスクリプションに対するサブスクライブが成功した回数 |
| SubCountNDur | 非継続サブスクリプションに対するサブスクライブが成功した回数 |
| SubFailCount | サブスクライブが失敗した回数 |
| UnsubCountDur | 継続サブスクリプションに対するアンサブスクライブが成功した回数 |
| UnsubCountNDur | 非継続サブスクリプションに対するアンサブスクライブが成功した回数 |
| UnsubFailCount | アンサブスクライブが失敗した回数 |

| フィールド | 説明 |
|---|---|
| SubRqCount | MQSUBRQが成功した回数 |
| SubRqFailCount | MQSUBRQが失敗した回数 |
| CBCount | MQCBが成功した回数 |
| CBFailCount | MQCBが失敗した回数 |
| CtlCount | MQCTLが成功した回数 |
| CtlFailCount | MQCTLが失敗した回数 |
| StatCount | MQSTATが成功した回数 |
| StatFailCount | MQSTATが失敗した回数 |
| SubCountDurHighWater | 継続サブスクリプション数の最高水準点 |
| SubCountDurLowWater | 継続サブスクリプション数の最低水準点 |
| SubCountNDurHighWater | 非継続サブスクリプション数の最高水準点 |
| SubCountNDurLowWater | 非継続サブスクリプション数の最低水準点 |
| PutTopicCount | トピックに対するMQPUTが成功した回数 |
| PutTopicFailCount | トピックに対するMQPUTが失敗した回数 |
| Put1TopicCount | トピックに対するMQPUT1が成功した回数 |
| Put1TopicFailCount | トピックに対するMQPUT1が失敗した回数 |
| PutTopicBytes | トピックにPutしたバイト数 |
| PublishMsgCount | サブスクリプションに配信されたメッセージ数 |
| PublishMsgBytes | サブスクリプションに配信されたバイト数 |

# ＜参考＞キュー統計で取得可能な情報

- インターバル間でのキューの活動
  - ◆ インターバルごとに下記メッセージを生成
    - 統計を取得しているキューが複数あると、個々のキューに対して QStatisticsData を作成
    - 最大100個のキューまで1つのメッセージで報告

| フィールド | 説明 |
|---|---|
| QueueManager | キュー・マネージャー名 |
| IntervalStartDate | 統計の取得を開始した日付 |
| IntervalStartTime | 統計の取得を開始した時刻 |
| IntervalEndDate | 統計の取得を終了した日付 |
| IntervalEndTime | 統計の取得を終了した時刻 |
| CommandLevel | コマンド・レベル |
| ObjectCount | キュー・オブジェクト数 |
| QStatisticsData | (PCFグループ) |
| QName | キュー名 |
| CreateDate | キューが作成された日付 |
| CreateTime | キューが作成された時刻 |
| QType | キュー・タイプ |
| QDefinitionType | キューの定義タイプ |
| QMinDepth | 滞留メッセージの最小数 |
| QMaxDepth | 滞留メッセージの最大数 |
| AvgTimeOnQ | メッセージの平均滞留時間 |

| フィールド | 説明 |
|---|---|
| PutCount | MQPUTが成功した回数 |
| PutFailCount | MQPUTが失敗した回数 |
| Put1Count | MQPUT1が成功した回数 |
| Put1FailCount | MQPUT1が失敗した回数 |
| PutBytes | Putしたバイト数 |
| GetCount | MQGETが成功した回数 |
| GetFailCount | MQGETが失敗した回数 |
| GetBytes | Getしたバイト数 |
| BrowseCount | ブラウズが成功した回数 |
| BrowseFailCount | ブラウズが失敗した回数 |
| BrowseBytes | ブラウズしたバイト数 |
| NonQueuedMsgCount | Putした直後にGetされたメッセージ数 |
| ExpiredMsgCount | 有効期限が切れたメッセージ数 |
| PurgeCount | パージされたメッセージ数 |

# ＜参考＞チャネル統計で取得可能な情報

- インターバル間でのチャネルの活動
  - ◆ インターバルごとに下記メッセージを生成
    - 統計を取得しているキューが複数あると、個々のチャネルに対して ChlStatisticsData を作成
    - 最大100個のチャネルまで1つのメッセージで報告

| フィールド | 説明 |
|---|---|
| QueueManager | キュー・マネージャー名 |
| IntervalStartDate | 統計の取得を開始した日付 |
| IntervalStartTime | 統計の取得を開始した時刻 |
| IntervalEndDate | 統計の取得を終了した日付 |
| IntervalEndTime | 統計の取得を終了した時刻 |
| CommandLevel | コマンド・レベル |
| ObjectCount | チャネル・オブジェクト数 |
| ChlStatisticsData | (PCFグループ) |
| ChannelName | チャネル名 |
| ChannelType | チャネル・タイプ |
| RemoteQmgr | リモート・キュー・マネージャー名 |
| ConnectionName | 接続名 |
| MsgCount | 転送したメッセージ数 |
| TotalBytes | 転送したメッセージ・バイト数 |
| NetTimeMin | ラウンド・トリップに要した最小時間 |
| NetTimeAvg | ラウンド・トリップに要した平均時間 |
| NetTimeMax | ラウンド・トリップに要した最大時間 |

| フィールド | 説明 |
|---|---|
| ExitTimeMin | EXITで要した最小時間 |
| ExitTimeAvg | EXITで要した平均時間 |
| ExitTimeMax | EXITで要した最大時間 |
| FullBatchCount | BATCHSZ/BATCHLIMに達して転送されたバッチの回数 |
| IncmplBatchCount | BATCHSZに達せずに転送されたバッチの回数 |
| AverageBatchSize | 平均バッチ・サイズ |
| PutRetryCount | リトライ回数 |

# バックアップ・リカバリー

# リカバリーの種別

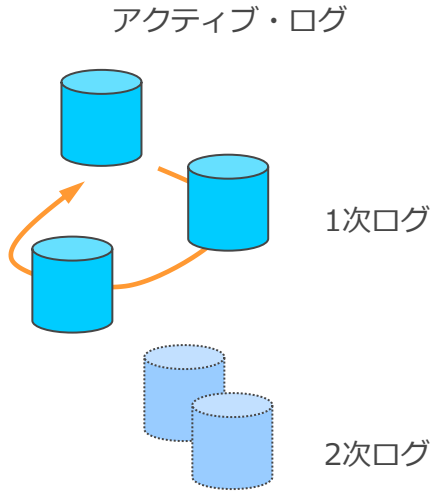
- MQは2種類のリカバリーをサポート
  - ◆ リスタート・リカバリー、クラッシュ・リカバリー
    - 正常終了、または予期せぬ障害によって異常終了したキュー・マネージャーを再起動した際のリカバリー
    - キュー・マネージャーの起動、チャネル接続時の再同期処理により以下を回復
      - トランザクションの整合性
      - キューに滞留していたパーシステント・メッセージ
  - ◆ メディア・リカバリー
    - キューなどのオブジェクト定義が損傷を受けた際のリカバリー
    - コマンドまたは自動で、破損した以下のオブジェクト定義を回復可能
      - キュー (キューに滞留していたパーシステント・メッセージを含む)
      - キュー・マネージャー
      - チャネル
      - チャネル同期ファイル など
    - 事前にメディア・イメージの取得が必要
- MQはいずれのリカバリーもログに基づいて行う
  - ◆ ログに記録される主な内容
    - パーシステント・メッセージのPUT/GET
    - トランザクションの開始/コミット/バックアウト
    - オブジェクト定義の作成/削除/属性変更

# ログ・タイプ

## 循環ログ

◆ ログ・ファイルを循環しながらログを記録

- リスタート・リカバリー/クラッシュ・リカバリーに不要となったログ・ファイルを再利用
- ログ・スペースが足りなくなると、一時的に2次ログをアロケーション
  - 2次ログは不要になれば自動的に削除される
- 1次ログ、2次ログ以上のディスクを消費しない



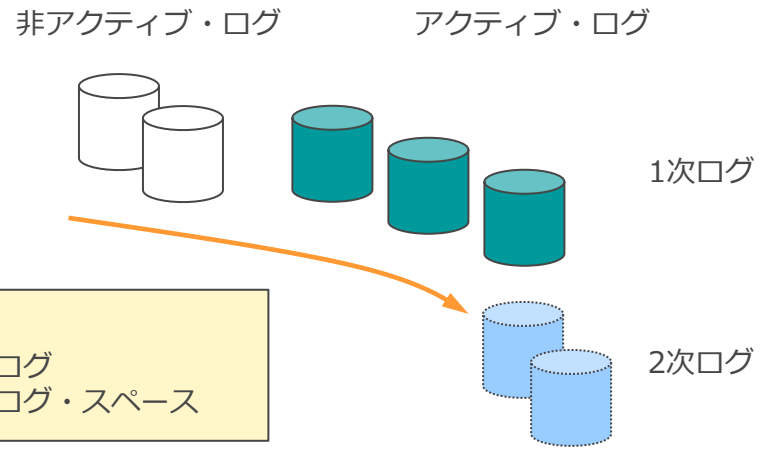

## リニア・ログ

◆ ログ・ファイルを随時アロケーションしながらログを記録

- リスタート・リカバリー/クラッシュ・リカバリーに不要となったログ・ファイルは非アクティブ・ログになる
- メディア・リカバリーにも、不要な非アクティブ・ログの削除、またはテープなどへの移動といった運用が必要

◆ V9.1以降、リニア・ログの自動管理が可能

- 不要ログの自動削除および再利用
- qm.ini の Logスタンザ で設定
  - 自動管理に設定する場合、LogManagement=Automatic を指定




<アクティブ・ログ>
- キュー・マネージャーのリスタートに必要なログ
- 未コミットのトランザクションが存在可能なログ・スペース

# ログ・タイプと回復可能な障害種別

- ログ・タイプと可能なリカバリー
  - ◆ ログ・タイプは、循環ログとリニア・ログの2種類
  - ◆ メディア・リカバリーはリニア・ログでのみ可能
    - 循環ログでは、リスタート・リカバリー/クラッシュ・リカバリーのみ可能
    - リニア・ログでは、リスタート・リカバリー/クラッシュ・リカバリーとメディア・リカバリーが可能
- 回復可能な障害種別
  - ◆ キュー・マネージャーの異常終了を伴う障害は、いずれのログ・タイプでも回復可能
    - クラッシュ・リカバリーで回復
  - ◆ ログが損傷した際は、いずれのログ・タイプでも回復不可能
    - キュー・マネージャーの再作成、またはバックアップの戻しが必要
    - ログは、RAIDなどの二重化ディスクでの保護が必要
  - ◆ オブジェクト定義が損傷した際は、リニア・ログのみ回復可能
    - メディア・リカバリーで回復
    - 循環ログでは、オブジェクト定義の再作成 (滞留しているメッセージはパーシステントでも消失)
- V9.1以降、キュー・マネージャー作成後でもmigmqlogコマンドでログ・タイプの変更が可能

| | キュー・マネージャー異常終了 | オブジェクト定義の損傷 | ログの損傷 |
|---|---|---|---|
| 循環ログ | ○ | × | × |
| リニア・ログ | ○ | ○ | × |

# メディア・イメージの取得

- メディア・イメージを手動で取得
  - キュー・マネージャー稼動中に、rcdmqimgコマンドを使用してメディア・イメージを取得
    - コマンド実行時にワイルド・カードを使用し、一括で取得することも可能
- V9.1以降、メディア・イメージの自動作成が可能に
  - キュー・マネージャー属性の設定で、メディア・イメージの自動取得、取得間隔などを指定
    - 自動取得の設定は、キュー・マネージャーの稼動中でも変更可能
    - IMGINTVL、IMGLOGLNのどちらかが満たされたタイミングでメディア・イメージが作成される

| 属性 | 値 | デフォルト | 説明 |
|---|---|---|---|
| IMGSCHED | AUTO \| MANUAL | MANUAL | メディア・イメージを自動で作成するか否か |
| IMGINTVL | 数値(分) \| OFF | 60 | メディア・イメージを書き込む間隔 (分) |
| IMGLOGLN | 数値(メガバイト) \| OFF | OFF | オブジェクトの前回のメディア・イメージ取得以降に書き込まれたログのサイズ (メガバイト) |

- チャネルやキューなどのすべてのオブジェクトのメディア・イメージ取得が必要
- メディア・イメージにはキューに滞留しているパーシステント・メッセージも記録されるため、メッセージ滞留の少ないとき(業務終了後など)に取得
  - 小さいサイズで高速に取得可能
  - リカバリーの際に、パーシステント・メッセージもリストアされてしまうため注意が必要

# ＜参考＞メディア・リカバリーの対象制御

- キュー・マネージャー属性とキュー属性の設定により、メディア・リカバリーの対象を制御可能
  - ◆ キュー・マネージャー属性

| 属性 | 値 | デフォルト | 説明 |
|---|---|---|---|
| IMGRCOVO | YES \| NO | YES | 以下のオブジェクトを、メディア・イメージからリカバリー可能にするかどうかを指定<br>• 認証情報<br>• チャネル<br>• クライアント接続<br>• リスナー<br>• 名前リスト<br>• プロセス<br>• 別名キュー<br>• リモート・キュー<br>• サービス・オブジェクト |
| IMGRCOVQ | YES \| NO | YES | 以下のオブジェクトを、メディア・イメージからリカバリー可能にするかどうかを指定<br>• ローカル・キュー<br>• 永続動的キュー |

  - ● キュー属性 (ローカル・キュー、モデル・キューのみ指定可能)

| 属性 | 値 | デフォルト | 説明 |
|---|---|---|---|
| IMGRCOVQ | YES \| NO \| QMGR | QMGR | メディア・イメージからリカバリー可能にするかどうかを指定<br>※QMGRを指定した場合は、キュー・マネージャー属性の設定に準ずる |

# メディア障害とリカバリー

- メディア(ディスク)障害時のリカバリー方法は損傷度合いにより異なる
  - ◆ ディスクの部分損傷
    - 損傷箇所と使用しているログ・タイプにより、必要なリカバリー作業が異なる

| | リカバリー方法 | |
|---|---|---|
| **ログ・タイプ** | **オブジェクト定義の損傷** | **ログの損傷** |
| 循環ログ | ・オブジェクト定義の再作成<br>・キュー・マネージャーの再作成、またはバックアップの戻し | ・キュー・マネージャーの再作成、またはバックアップの戻し |
| リニア・ログ | ・メディア・リカバリーの実行<br>・データ・ディレクトリを手動、またはバックアップで回復<br>＋ メディア・リカバリーの実行 | |

  - ◆ ディスク全体の損傷 (ボリューム障害)
    - MQディレクトリ全体をバックアップから回復
- メディア・リカバリーでも救えないディスク全体障害に備え、リニア・ログの場合でもバックアップの取得が必要
  - ◆ RAIDなどの二重化ディスクでの保護も要検討

# キュー・マネージャーのバックアップ

## 2種類のキュー・マネージャーのバックアップ方法

### ◆ キュー・マネージャー再作成に必要な情報をバックアップ

- キュー・マネージャーの作成/変更をすべてシェル・スクリプト＆ファイル経由で行うようにし、これらのファイルをバックアップしておく

| バックアップ対象ファイル | 説明 |
|---|---|
| キュー・マネージャー作成時のオプション | crtmqmコマンドをシェル・スクリプトから実行、シェル・スクリプトのバックアップ |
| キュー、チャネルなどのオブジェクトを定義するMQSCコマンド | MQSCコマンドをファイルに記述し、ファイルをrunmqscにリダイレクト、もしくは runmqsc –f オプションでファイルを指定して実行、ファイルのバックアップ *1 |
| OAM権限の設定コマンド (setmqaut) | setmqautコマンドをシェル・スクリプトから実行、シェル・スクリプトのバックアップ |
| qm.iniファイル | キュー・マネージャー作成後に内容を変更した場合、ファイルのバックアップ |

- *1）dmpmqcfgコマンドで、キュー・マネージャーの構成をMQSCファイルとしてバックアップすることも可能

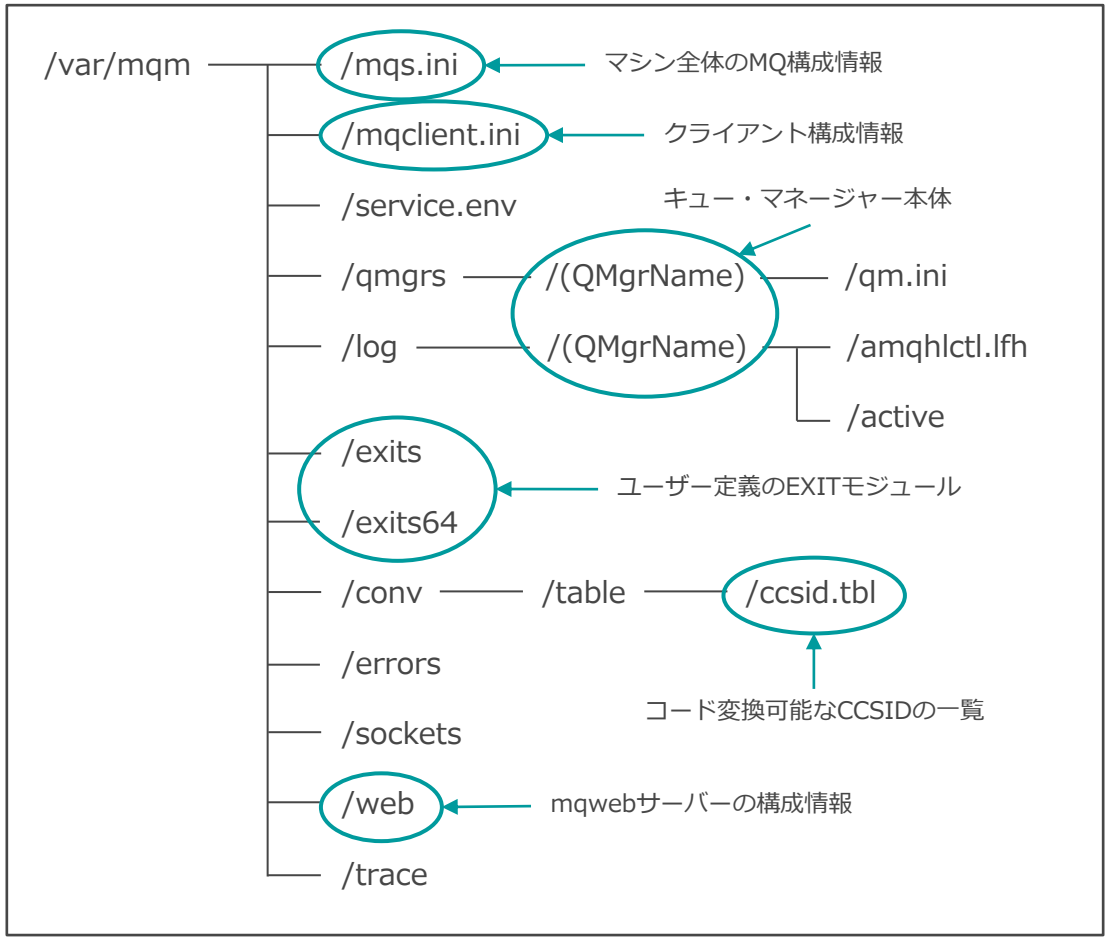
### ◆ キュー・マネージャーそのものをバックアップ

- キュー・マネージャーを停止した状態で、下記の2つのディレクトリをバックアップ
  - /var/mqm/qmgrs/(QMgrName)以下：キューなどのオブジェクト定義のファイルが保存されているディレクトリ
  - /var/mqm/log/(QMgrName)以下：ログ・ファイルが保存されているディレクトリ
- キュー・マネージャーをバックアップから回復する際は、同時に取得した上記ディレクトリを同時に戻す

# MQディレクトリのバックアップ

- MQ for UNIXのデータ・ディレクトリ構造
  - ◆ MQは /var/mqm にデータを保存
    - mqs.ini
      - マシン全体のキュー・マネージャーの構成情報
    - mqclient.ini
      - クライアントの構成情報
    - qmgrs、log
      - キュー・マネージャー本体
        - キュー・マネージャーごとにディレクトリが作成される
    - exits、exits64
      - EXITモジュールを置くディレクトリ
    - conv
      - コード変換テーブル など
        - ccsid.tbl はコード変換可能なCCSIDの一覧
    - web
      - mqwebサーバー・ディレクトリ
    - 他
  - ◆ MQの初期構成後にバックアップを取得
    - さらに、mqs.iniファイルの変更、キュー・マネージャーの追加、キュー構成の変更、EXITモジュールの追加、ccsid.tblファイルの変更などのタイミングでもバックアップを取得

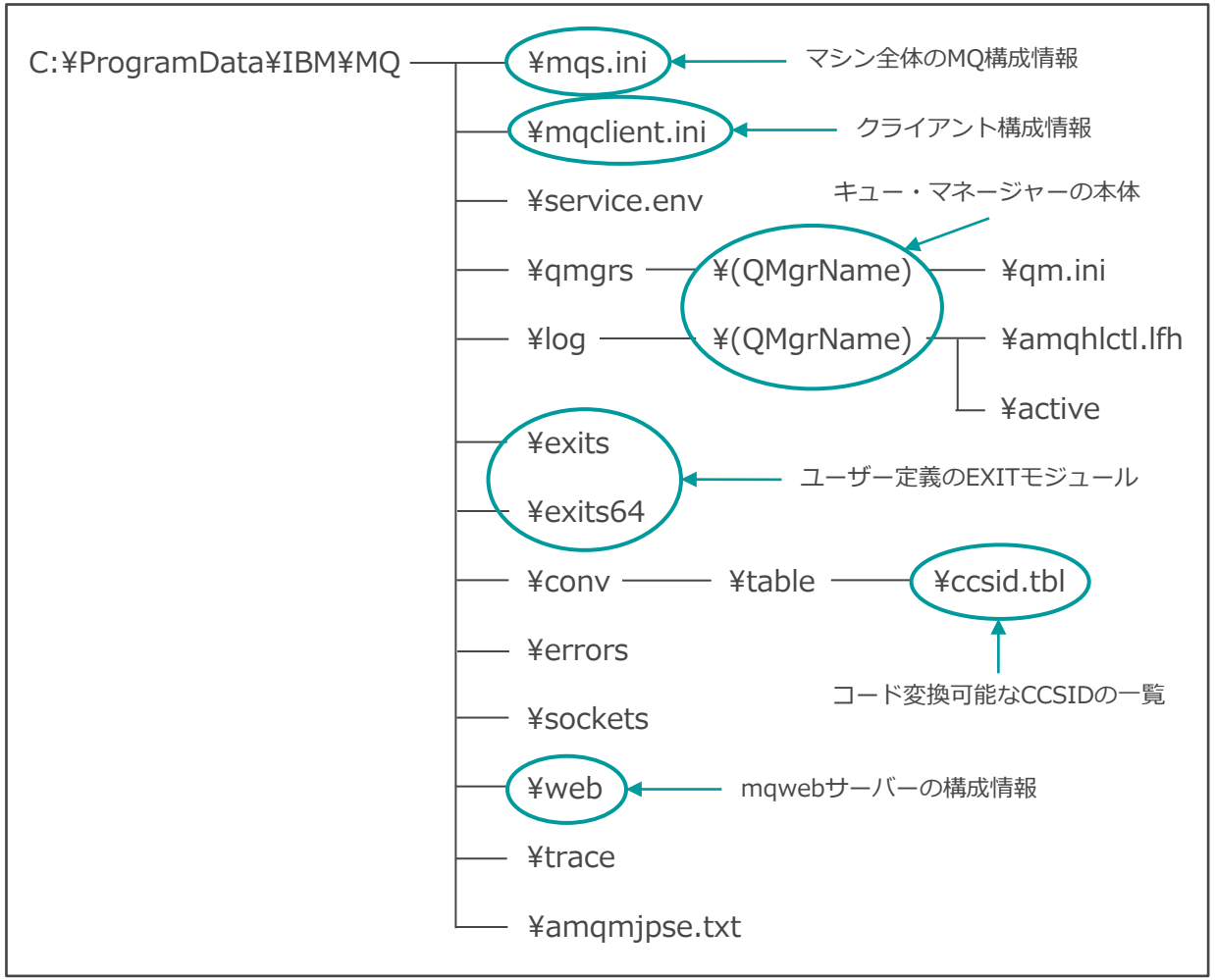
<データ・ディレクトリ構造の例 (MQ for UNIX)>



# MQディレクトリのバックアップ

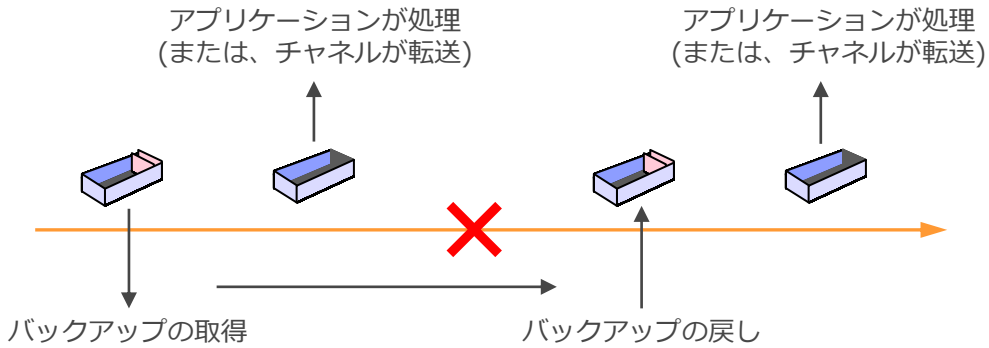
- MQ for Windowsのデータ・ディレクトリ構造
  - MQは C:¥ProgramData¥IBM¥MQ にデータを保存
  - ディレクトリの構造は MQ for UNIX と同等

<データ・ディレクトリ構造の例 (MQ for Windows)>



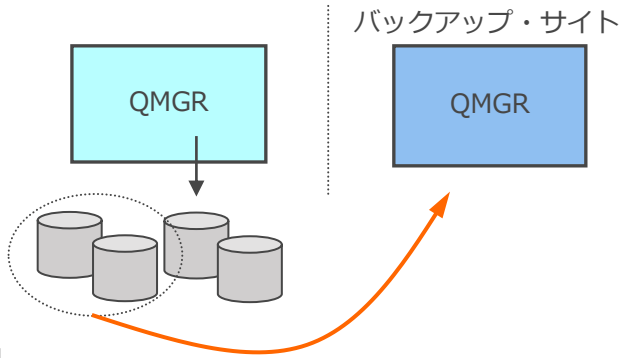
# バックアップからのリカバリー

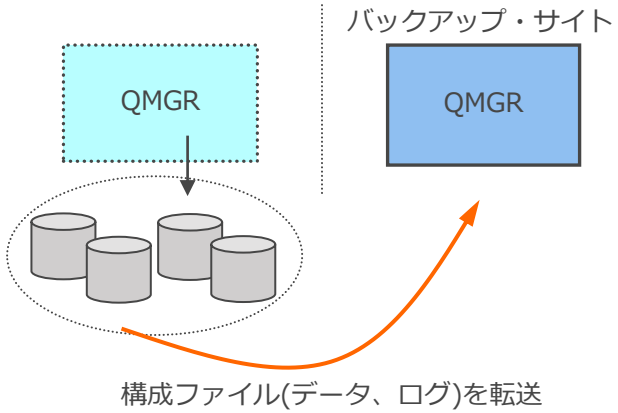
- リカバリー方法によってバックアップからの回復状態が異なる
  - ◆ いずれの回復状態がアプリケーションに合うかを検討し、バックアップ方法を選択
  - ◆ キュー・マネージャーを再作成して回復
    - ● 回復したキュー・マネージャーのキューはすべて空
      - キューに滞留していたパーシステント・メッセージはすべて消失
    - ● アプリケーションからの再処理が必要
  - ◆ バックアップを戻してキュー・マネージャーを回復
    - ● バックアップを取得した時点の状態に回復
      - キューに滞留していたパーシステント・メッセージはすべて消失
        - ただし、バックアップ取得時点で滞留していたパーシステント・メッセージはキューに回復
    - ● アプリケーションからの再処理が必要、さらに、アプリケーションを二重処理可能に設計



# ディザスター・リカバリー

- 災害などにより通常サイトでサービスを継続できないとき、バックアップ・サイトでサービスを提供するために直近のキュー・マネージャーのコピーを構築
  - ◆ 通常サイトのログ・ファイルをバックアップ・サイトへ転送/反映することで、直近状態のMQ環境を作成
- 下記4つの機能を提供
  - ◆ ログ・ファイルの強制スイッチ
    - RESET QMGRコマンド
  - ◆ ログ・ファイル・スイッチ発生の通知
    - ロガー・イベント
  - ◆ 書き込み済み(転送可能な)ログ・ファイルの判別
    - DISPLAY QMSTATUSコマンド
  - ◆ コピー・キュー・マネージャーへのログ・ファイルの適用

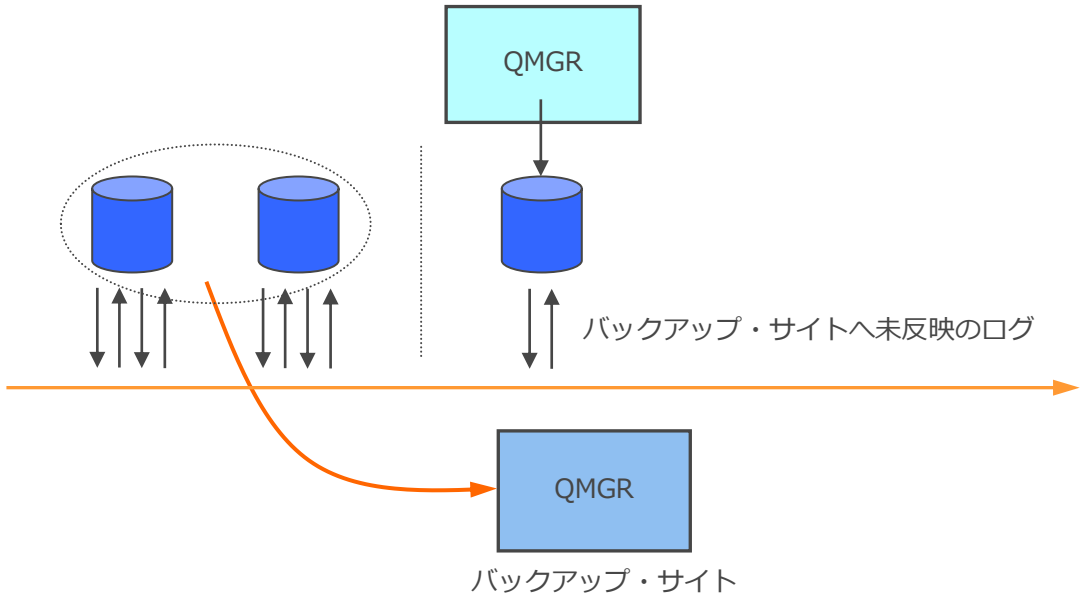



- リニア・ログでのみサポート
  - ◆ ディザスター・リカバリー機能がない場合は、キュー・マネージャーを停止して取得した構成ファイル(データ、ログ)のバックアップをバックアップ・サイトで復元
- RDQM構成
  - ◆ 本番サイトと災対サイトで稼動する1ペアのサーバーで構成される、DR RDQM(災対RDQM)の利用も可能
    - 2章「構成」p.61~ 参照

# ディザスター・リカバリー

- 未反映ログとバックアップ・サイトでの不整合
  - ◆ 転送可能なログ・ファイルは、キュー・マネージャーが書き込みを終えたもののみ
  - ◆ バックアップ・サイトでサービスを開始した際、未反映ログによる不整合が発生
    - MQPUTのログ・レコードが未反映：メッセージ消失
    - MQGETのログ・レコードが未反映：メッセージ重複




  - ◆ 未反映ログを最少にするためには、ログ・スイッチの発生を監視してログ・ファイルを転送
    - MQはログ・ファイルのスイッチをイベント・メッセージ(ロガー・イベント)で通知
    - ログ・ファイルのサイズも小さく設定