

App Connect Enterprise モニタリング機能概説

- 第2.0版(2024.1.15)

日本アイ・ビー・エム システムズ・エンジニアリング

はじめに

- 本資料は、App Connect Enterprise（ACE）の稼働状況を確認することが可能な製品提供機能について解説します。
- 記載内容は、ACE V11、およびACE V12をベースにしており、今後のFixpackやバージョンで変更される場合があります。

当資料に記載している内容は可能な限り稼働確認を取っておりますが、日本アイ・ビー・エム株式会社、及び日本アイ・ビー・エムシステムズ・エンジニアリング株式会社の正式なレビューを受けておらず、当資料で提供される内容に関して日本アイ・ビー・エム株式会社、日本アイ・ビー・エムシステムズ・エンジニアリング株式会社は何ら保証するものではありません。従って、当資料の利用またはこれらの技法の実施はひとえに使用者の責任において為されるものであり、当資料によって受けたいかなる被害に関しても一切の補償をするものではありません。

内容

- モニタリング機能一覧
- 前提情報
- アクティビティ・ログ
- メッセージフロー・モニタリング
- Record&Replay
- ビジネス・トランザクション・モニタリング
- OpenTelemetry
- ログ
- メッセージフロー統計
- リソース統計
- ユーザー・トレース
- Traceノード
- 管理ログ

モニタリング機能一覧

モニタリング機能一覧

機能	機能説明	出力先/参照方法	用途	備考
アクティビティ・ログ	<ul style="list-style-type: none">メッセージフローの処理状況、外部リソースへのアクセス状況を簡易的に示すログを出力 (業務メッセージの中身は出力されない)デフォルトで出力され、Web UIで参照可能	<ul style="list-style-type: none">Web UI(GUI)/REST API(JSON)ファイル(CSV)ELK stack(GUI)	<ul style="list-style-type: none">メッセージフローの稼働状況のリアルタイム監視	<ul style="list-style-type: none">デフォルトで出力動的なON/OFFは不可
メッセージフロー・モニタリング	<ul style="list-style-type: none">メッセージフローの各種イベント(トランザクション開始/終了、ターミナルの通過など)を出力業務メッセージの内容をイベントに搭載することも可能	<ul style="list-style-type: none">MQトピック(XML/JSON)MQTTトピック(XML/JSON)ファイル(XML/JSON)ELK stack(GUI)	<ul style="list-style-type: none">メッセージフローの稼働状況のリアルタイム監視監査証跡問題判別	<ul style="list-style-type: none">MQ/MQTTブローカーの稼働とイベントを処理するモニターツールが必要 (Record&Replay機能を使用することでWeb UI/REST APIで表示可能)
Record&Replay	<ul style="list-style-type: none">メッセージフローを流れる業務メッセージをキャプチャーし、DBに保管、Web UIで表示、再送できる	<ul style="list-style-type: none">Web UI(GUI)/REST API(JSON)	<ul style="list-style-type: none">メッセージフローの稼働状況のリアルタイム監視監査証跡問題判別	<ul style="list-style-type: none">データベースの構成、MQキューマネージャーの関連付けが必須再送はMQメッセージのみ
ビジネス・トランザクション・モニタリング	<ul style="list-style-type: none">複数のフローをまたがる一連のビジネス・トランザクションをモニタリングする機能	<ul style="list-style-type: none">Web UI(GUI)	<ul style="list-style-type: none">メッセージフローの稼働状況のリアルタイム監視監査証跡問題判別	<ul style="list-style-type: none">データベースの構成、MQキューマネージャーの関連付けが必須
OpenTelemetry	<ul style="list-style-type: none">メッセージフローの処理時間や実行成否をOpenTelemetryの仕様準拠した形で出力	<ul style="list-style-type: none">OpenTelemetry準拠のCollector(モニターツール)	<ul style="list-style-type: none">メッセージフローの稼働状況のリアルタイム監視	<ul style="list-style-type: none">OpenTelemetryに対応したモニターツール等が必要
ログ	<ul style="list-style-type: none">ランタイムのコンポーネントが内部的なアクティビティ情報(BIPメッセージ)をローカル・エラーログに出力Elastic stackに出力することも可能	<ul style="list-style-type: none">ローカル・エラーログ(統合ノード構成)標準出力/ログファイル(独立統合サーバー)ELK stack(GUI)	<ul style="list-style-type: none">ランタイムの障害検知問題判別	<ul style="list-style-type: none">ELK stackに出力する場合は別途ELK stackの構成もしくはサービスが必要
メッセージフロー統計	<ul style="list-style-type: none">メッセージフローの各種統計、会計情報を出力(処理件数、処理時間、CPU時間、メッセージサイズなど)ノード統計、ターミナル統計、スレッド統計も出力可	<ul style="list-style-type: none">MQトピック(XML/JSON)MQTTトピック(XML/JSON)ファイル(CSV)ユーザー・トレースWeb UI(GUI)/REST API(JSON)	<ul style="list-style-type: none">メッセージフローの稼働状況のモニタリング問題判別傾向把握	
リソース統計	<ul style="list-style-type: none">内部コンポーネントのリソース使用状況の統計情報を出力(JVM、パーサー、ODBC、JDBC、FTPなど)	<ul style="list-style-type: none">MQトピック(XML/JSON)MQTTトピック(XML/JSON)ファイル(CSV)Web UI(GUI)/REST API(JSON)	<ul style="list-style-type: none">ランタイムのリソース状況のモニタリング問題判別傾向把握	

モニタリング機能一覧

機能	機能説明	出力先/参照方法	用途	備考
ユーザー・トレース	<ul style="list-style-type: none">メッセージフロー処理の詳細を出力ターミナルの遷移やESQLのステップも確認できる	<ul style="list-style-type: none">ファイル(Text)	<ul style="list-style-type: none">問題判別UT等動作確認/エビデンス	<ul style="list-style-type: none">動的にON/OFF可能
Traceノード	<ul style="list-style-type: none">メッセージフローの任意の箇所でメッセージツリーやExceptionListなど任意のメッセージを出力できる	<ul style="list-style-type: none">ファイル(Text)ユーザー・トレース(Text)ローカル・エラーログ(Text)	<ul style="list-style-type: none">アプリケーションログ問題判別監査証跡	<ul style="list-style-type: none">Traceノードの出力を動的にON/OFF可能
管理ログ	<ul style="list-style-type: none">稼働環境のコンポーネントに対する管理操作のログを出力	<ul style="list-style-type: none">MQトピック(JSON)MQTTトピック(JSON)ファイル(JSON)Web UI(GUI)/REST API(JSON)	<ul style="list-style-type: none">監査証跡	<ul style="list-style-type: none">Web UIではデフォルトで表示可能

前提情報

統合ノード構成と独立統合サーバー構成

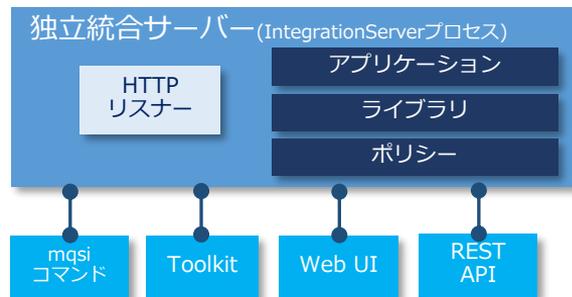
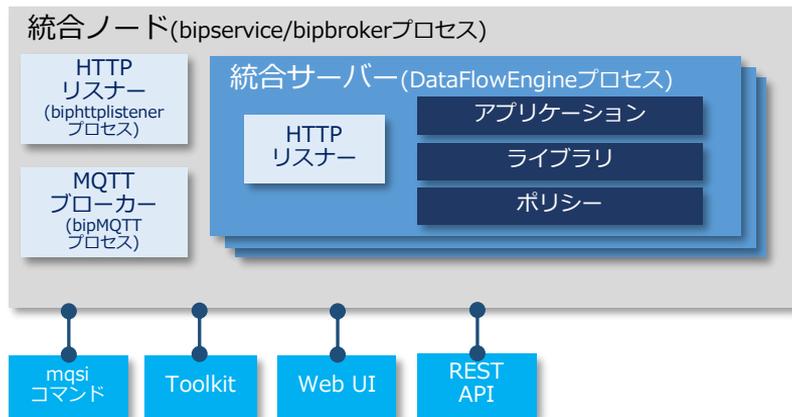
■ ACE V11から、2種類の稼働形態を提供

■ 統合ノード構成

- ◆ 統合ノードと管理下の統合サーバーで稼働
- ◆ V10以前の構成と同等
- ◆ 構成ディレクトリ：/var/mqsi/components/<統合ノード>
- ◆ 統合ノード下のHTTPリスナーと統合サーバー内のHTTPリスナーが稼働
- ◆ 統合ノード下にMQTTブローカーが稼働

■ 独立統合サーバー構成

- ◆ 単一の統合サーバーで稼働
- ◆ V11で新たに提供された構成
- ◆ 主にコンテナ環境での利用を想定
- ◆ 構成ディレクトリ：任意のワーク・ディレクトリ
- ◆ 統合サーバー内のHTTPリスナーのみ
- ◆ MQTTブローカーは稼働しない



構成ファイル(.yamlファイル)

- ACE V11から統合ノード、統合サーバーの構成情報はYAML形式の構成ファイルに設定

統合ノード	/var/mqsi/components/<統合ノード>/ node.conf.yaml
統合サーバー	/var/mqsi/components/<統合ノード>/servers/<統合サーバー>/ server.conf.yaml ※独立統合サーバーの場合はワーク・ディレクトリ直下のserver.conf.yaml

- ◆ コンポーネント作成時にデフォルトの構成ファイルが生成される
 - デフォルトの構成ファイルには設定可能なプロパティが全て記載
 - 大部分はコメントアウトされた状態でデフォルト値が有効となる
- ◆ 構成変更は、構成ファイルを直接変更
 - もしくは、上記ディレクトリ下のoverridesディレクトリに変更するプロパティだけを記載した構成ファイルを配置して上書きすることも可能
 - ✓ overridesディレクトリ下の構成ファイルの設定のほうが優先される
 - 変更した内容はコンポーネントの再起動で反映
- ◆ 統合サーバーに関する一部のプロパティはnode.conf.yaml、server.conf.yamlの両方に設定可能
 - node.conf.yamlの設定は配下の全ての統合サーバーに有効となる
 - ただし、両方に同じプロパティが設定されている場合は、server.conf.yamlの設定のほうが優先される
- ◆ 一部の設定はコマンドやツールキット、Web UI、管理REST APIから動的に(再起動せずに)変更可能
 - メッセージフロー統計/リソース統計やメッセージフロー・モニター、トレースの開始/停止など
 - デフォルトでは、変更した設定はoverridesディレクトリ下の構成ファイルに記載される(永続化される)ため、統合サーバーの再起動やフローの再デプロイ後も有効となる
 - 永続化したくない場合はコマンドの --non-persist オプションを指定

ポリシー

■ 外部のリソースやサービスへの接続情報をポリシーに設定

- ◆ 以前の構成可能サービス（JDBC/TCPIPなど）や運用ポリシー（MQ/MQTTなど）の設定をポリシーに定義

- ◆ ツールキットでポリシーを作成

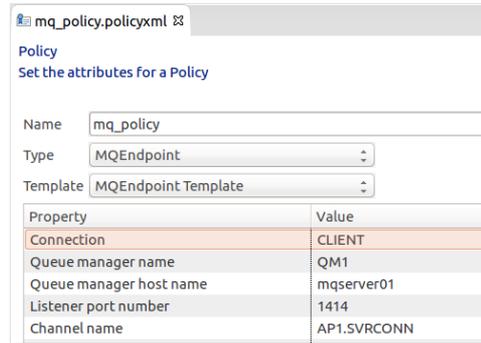
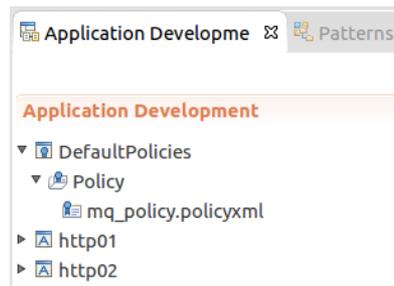
- ポリシー・エディターでタイプと関連するプロパティを指定
 - ✓ タイプ：MQEndpoint、MQTTPublish、JDBCプロバイダーなど
- ポリシー・プロジェクトに格納
 - ✓ アプリケーションやライブラリには格納できない
- フロー（ノード）と紐づくポリシーは、ノード・プロパティでポリシーを指定
 - ✓ 以下のフォーマットで指定

{ポリシー・プロジェクト名}:ポリシー名

- ✓ デフォルトのポリシー・プロジェクトに格納したポリシーはポリシー名だけで指定可能
 - デフォルトのポリシー・プロジェクトはserver.conf.yamlのDefaults: policyProjectプロパティで指定

- ◆ BARファイルに格納してデプロイ

- 統合サーバー稼働中にデプロイする場合、アプリケーションから指定されるポリシーはアプリケーションより前に、もしくは同時にデプロイする必要がある
- デプロイしたポリシーは、<workdir>/runディレクトリ下に配置される



キューマネージャーの関連付け方法

■ 統合ノード／独立統合サーバーにキューマネージャーを関連付ける場合の構成方法

- ◆ 一部の機能はキューマネージャーの関連付けが必須
 - ・ 内部的にキューマネージャーのオブジェクトを利用するため

1. キューマネージャーの作成

- ・ 統合ノード構成の場合はローカルに作成
- ・ 独立統合サーバー構成の場合はリモートでもよい

2. システム・オブジェクトの作成

- ・ 内部的に使用するMQオブジェクトを、製品提供のスクリプトを利用して作成
 - ✓ `<install_dir>/server/sample/wmq/iib_createqueues.sh`
 - ✓ Windows環境ではバッチファイル(.cmd)を提供

3. 関連付け

- ・ 作成したキューマネージャーを構成ファイルの`defaultQueueManager`プロパティに指定
 - ✓ 統合ノード構成は、`node.conf.yaml`、独立統合サーバー構成は、`server.conf.yaml`

```
defaultQueueManager: 'IN01QM'
```

- ・ 独立統合サーバーは`remoteDefaultQueueManager`プロパティにリモートのキューマネージャーを指定することも可能
 - ✓ 事前にリモート・キューマネージャーを指定したMQEndpointポリシーを作成し、ランタイムにデプロイしておく必要あり

```
remoteDefaultQueueManager: '{myPolicyProject}:myMQEndpointPolicy'
```

- ・ 関連付けした後はキューマネージャーが起動し、アクセス可能な状態になっていないとランタイムを起動できない点に注意

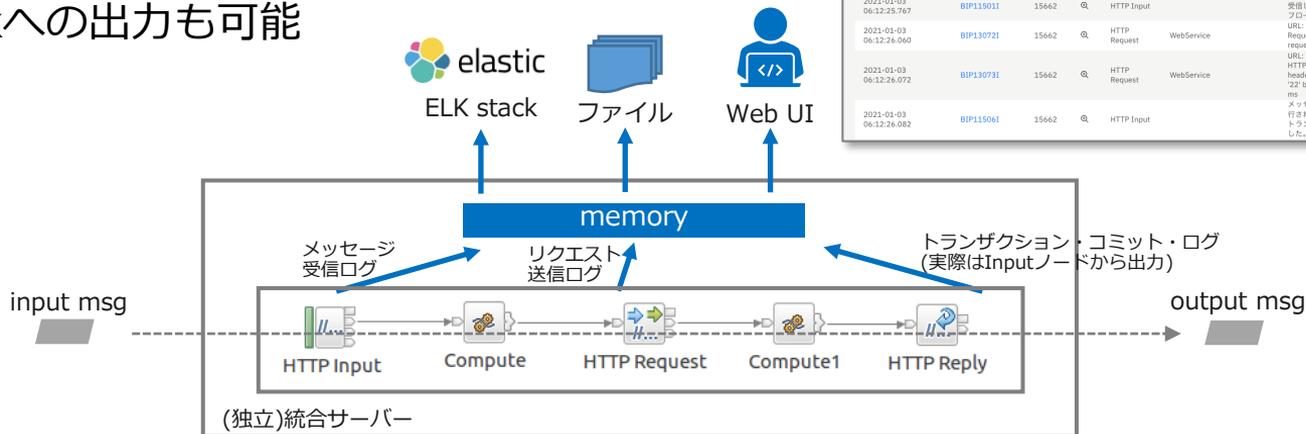
アクティビティ・ログ

機能概要

- メッセージフローの処理状況、外部リソースへのアクセス状況を簡易的に示すログを出力
 - ◆ メッセージの受信/送信、MQやDBなど外部リソースへの接続などを示す簡単なメッセージを、フロー処理1件毎にタイムスタンプ、スレッドID等とともに出力
 - ◆ トランザクションを特定する情報や業務メッセージの中身は出力されない
 - ・ ただし、MQメッセージの場合はメッセージID/相関IDは出力
- デフォルトで収集され、Web UIで参照可能
 - ◆ データは統合サーバー内のメモリ上に保持
 - ・ デフォルトで1000エントリーまで保持(変更可)
 - ◆ 動的な出力ON/OFFやフロー単位のON/OFF設定は不可
- 循環ログ・ファイルに出力することも可能
 - ◆ 任意のパス/ファイルにCSV形式で出力
- ELK stackへの出力も可能

Web UI画面

タイム・スタンプ	メッセージ	スレッド ID	ノード	リソース・マネージャー	メッセージの詳細	タグ
2021-01-03 06:12:25.767	BIP115011	15662	HTTP Input		入力ノード「HTTP Input」がデータを生成し、そのデータをメッセージ・フロー「httpreq01」に転送しました。 URL: 'http://localhost:7800/http01', Request headers: '102' bytes, request body: '0' bytes	NOETYP E-ININPUT
2021-01-03 06:12:26.060	BIP130721	15662	HTTP Request	WebService		
2021-01-03 06:12:26.072	BIP130731	15662	HTTP Request	WebService	URL: 'http://localhost:7800/http01', HTTP status code: '200', Response headers: '150' bytes, response body: '2' bytes, total request time: '12' ms	
2021-01-03 06:12:26.082	BIP115061	15662	HTTP Input		メッセージ・フロー・スレッドで実行された作業に関して、ローカル・トランザクションがコミットされました。	



Web UIによる表示

- Web UIにて対象のメッセージフローを選択し、「アクティビティ・ログ」タブで参照
 - ◆ 初回時は「アクティビティ・ログの取り出し」ボタンを押してサーバーからログの取得が必要
 - ◆ 「リフレッシュ」ボタンを押して更新

ノード: IN01 / サーバー: ex1 / アプリケーション: httpreq01 / メッセージ・フロー: httpreq01

httpreq01

開始済み

プロパティ フロー統計 **アクティビティ・ログ**

Last updated: a few seconds ago

リフレッシュボタン

検索: スレッド ID による表のフィルター エクスポート

タイム・スタンプ	↑	メッセージ	スレッド ID	ノード	リソース・マネージャー	メッセージの詳細	タグ
2021-01-03 06:12:25.767		BIP11501I	15662	HTTP Input		入力ノード 'HTTP Input' がデータを受信し、そのデータをメッセージ・フロー 'httpreq01' に伝搬しました。	NODETYPE=INPUT
2021-01-03 06:12:26.060		BIP13072I	15662	HTTP Request	WebService	URL: 'http://localhost:7800/http01', Request headers: '102' bytes, request body: '0' bytes	
2021-01-03 06:12:26.072		BIP13073I	15662	HTTP Request	WebService	URL: 'http://localhost:7800/http01', HTTP status code: '200', Response headers: '150' bytes, response body: '22' bytes, Total request time: '12' ms	
2021-01-03 06:12:26.082		BIP11506I	15662	HTTP Input		メッセージ・フロー・スレッドで実行された作業に関して、ローカル・トランザクションがコミットされました。	

設定

- 統合サーバーの構成ファイル(server.conf.yaml)の以下のセクションで設定
 - ◆ ResourceManagersセクション/ActivityLogManagerサブセクション
 - ◆ activityLogEnabledプロパティでON/OFFを設定 (true/false)
 - 初期状態のserver.conf.yamlではコメントアウトされた状態だが、デフォルトでON
 - ◆ defaultLogSizePerThreadプロパティでメモリに保持するログ・エントリーの数を指定 (デフォルトは1000)
 - 指定値を超えたログは古いものから削除される
 - ◆ minSeverityLevelプロパティで保持するログの重要度 (INFO<WARN<ERROR) を指定
 - 指定した重要度以上のログが保持される

設定例

```
ResourceManagers:  
  ActivityLogManager:  
    activityLogEnabled: true  
    defaultLogSizePerThread: 1000  
    minSeverityLevel: 'INFO' # choose one of INFO|WARN|ERROR
```

設定方法詳細は以下参照

<https://www.ibm.com/docs/en/app-connect/12.0?topic=logs-viewing-setting-runtime-properties-activity>

ファイル/ELK stack出力設定

- ファイルおよびELK stackへの出力はアクティビティ・ログ・ポリシーで設定
 - ◆ ツールキットでポリシーを作成後、統合サーバーにデプロイして適用

プロパティ	説明
enabled	ポリシーの使用可否 (デフォルト true)
Tag filter string	フィルター名と値のペアでフィルター条件を指定。複数指定の場合はセミコロンで連結 (デフォルト 値なし) 例: RM=JMS;MSGFLOW=myflow
File name	出力先のログ・ファイルのディレクトリ・パスとファイル名 (デフォルト 値なし) ファイル名はアクティビティ・ログ・ポリシーでユニークに指定 複数ファイルを循環ログとして使用する場合、ファイル名にsuffix番号がつく (myLog.1 など)
Number of log files	循環ログとして使用するファイル数 (デフォルト 4)
Maximum log file size (MB)	ファイルサイズによるログのローテーションを実施する場合に、1ファイルあたりの上限値を指定 (デフォルト 25MB) 値0に設定した場合、ファイルサイズによるログのローテーションは行われない
Maximum age of log files (minutes)	時間によるログのローテーションを実施する場合に、分単位で時間を指定 (デフォルト 0) 値0に設定した場合、時間によるログのローテーションは行われない
Format the entries	フォーマットされたログを出力するかどうかを指定 (デフォルト false)
Minimum severity of log entries	ログ出力するメッセージの重要度を指定 (デフォルト INFOMATIONAL) INFOMATIONAL / WARNING / ERROR
Enable logging to ELK	ELK stackへの出力有無 (デフォルト false)
Name of the ELK Connection for logging to ELK	ELK stackへの接続情報をセットしたELK Connectionを指定 (別途、ELK Connectionの設定をserver.conf.yamlに実施)

※ELK Connectionの設定については「ログ」の章を参照

設定方法詳細は以下参照

<https://www.ibm.com/docs/en/app-connect/12.0?topic=properties-activity-log-policy-activitylog>

ファイル出力例

■ ファイル出力例

「Format the entries」プロパティ=false(デフォルト)の場合

```
MessageNumber,Severity,TimeStamp,ThreadID,FormattedMessage,Tags,Inserts
11501,I,"2021-01-03 06:12:25.767559",15662,, "MSGFLOW=httpreq01,NODE=HTTP Input,NODETYPE=INPUT,", "HTTP Input,httpreq01,
13072,I,"2021-01-03 06:12:26.060285",15662,, "MSGFLOW=httpreq01,NODE=HTTP Request,RM=WebService,", "http://localhost:7800/http01,102,0,
13073,I,"2021-01-03 06:12:26.072288",15662,, "MSGFLOW=httpreq01,NODE=HTTP
Request,RM=WebService,", "http://localhost:7800/http01,200,150,22,12,
11506,I,"2021-01-03 06:12:26.082049",15662,, "MSGFLOW=httpreq01,NODE=HTTP Input,",
11302,I,"2021-01-03 06:12:26.082497",15662,, "RM=Parsers,MSGFLOW=httpreq01,NODE=HTTP Input,", "19,
```

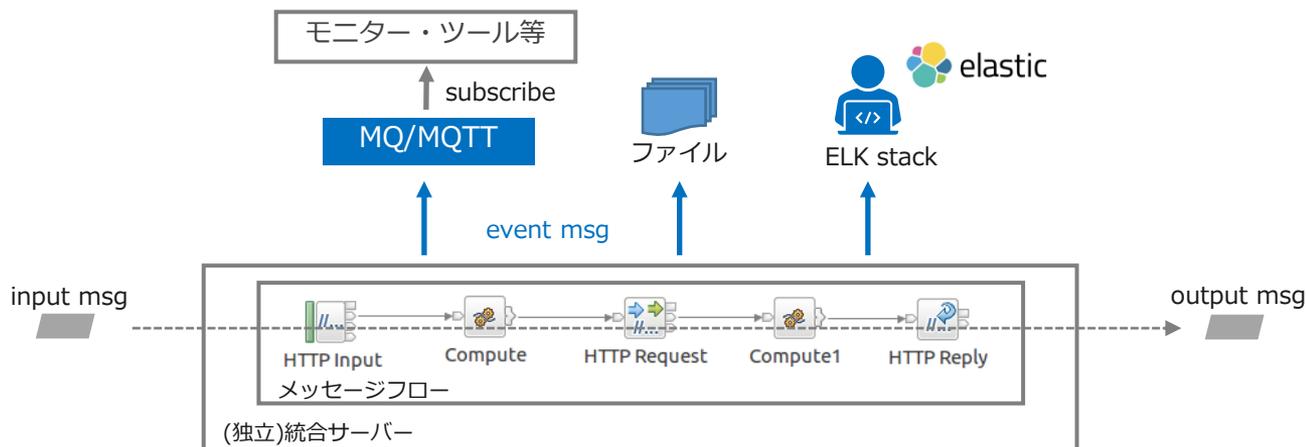
「Format the entries」プロパティ=trueの場合

```
MessageNumber,Severity,TimeStamp,ThreadID,FormattedMessage,Tags,Inserts
11501,I,"2021-01-03 06:22:04.581317",15662,"BIP11501I: 入力ノード 'HTTP Input' からデータを受信しました。", "MSGFLOW=httpreq01,NODE=HTTP
Input,NODETYPE=INPUT,", "HTTP Input,httpreq01,
13072,I,"2021-01-03 06:22:04.590154",15662,"BIP13072I: Invoking a Web Service.", "MSGFLOW=httpreq01,NODE=HTTP
Request,RM=WebService,", "http://localhost:7800/http01,102,0,
13073,I,"2021-01-03 06:22:04.597632",15662,"BIP13073I: Received reply from Web Server.", "MSGFLOW=httpreq01,NODE=HTTP
Request,RM=WebService,", "http://localhost:7800/http01,200,150,22,7,
11506,I,"2021-01-03 06:22:04.598764",15662,"BIP11506I: ローカル・トランザクションをコミットしました。", "MSGFLOW=httpreq01,NODE=HTTP Input,",
```

メッセージフロー・モニタリング

機能概要

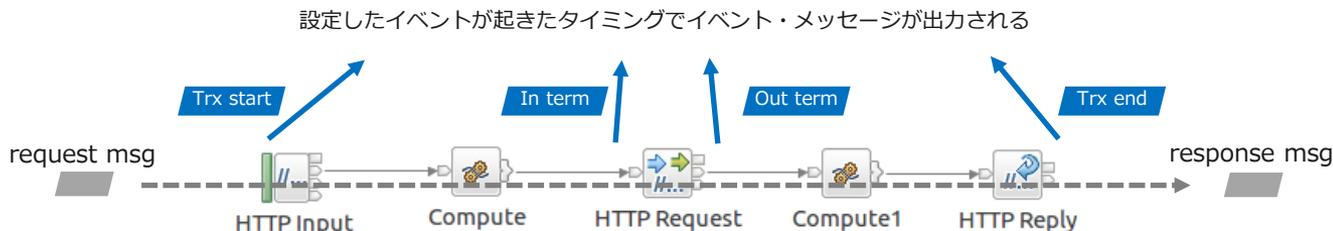
- メッセージフロー処理の各種イベントを出力する機能
 - ◆ トランザクションの開始/終了/ロールバックや特定のターミナルの入出力などをイベント・メッセージとして出力
 - ◆ ユーザーが出力したいイベントを選択してフローに設定
 - ◆ 業務メッセージの本体(Bit stream)や特定のデータ項目をイベント・メッセージに載せることも可能
- フロー処理1件毎に指定したイベントが起きるとイベント・メッセージを出力
 - ◆ イベント・メッセージはMQトピック/MQTTトピックに出力
 - ・ イベント・メッセージを収集(Subscribe)し、画面等にアウトプットする機能は別途実装が必要
 - ◆ ファイルやELK stackに出力することも可能
- フローのリアルタイム・モニタリングや監査証跡に利用



イベントの種類

■ イベントのタイプとイベント出力のトリガーとなるイベント・ソース

イベント・タイプ	イベント・ソース	説明	備考
トランザクション・イベント	Transaction start	フロー処理が開始したタイミングでイベントを出力	Inputノードにのみ設定可
	Transaction end	フロー処理が完了したタイミングでイベントを出力	Inputノードにのみ設定可
	Transaction rollback	フロー処理がロールバックしたタイミングでイベントを出力 (処理途中で例外が出てもフロー内でハンドリングしている場合はこのイベントは出力されない)	Inputノードにのみ設定可
ターミナル・イベント	任意のターミナル	指定したターミナルをメッセージが通過したタイミングでイベントを出力	全ノードに設定可



※便宜上、Transaction endイベントがHTTP Replyノードから出力しているように図示していますが、実際にはフロー処理完了時にInputノードから出力されます。

イベント・メッセージの出力先

■ イベント・メッセージの出力先は以下から選択

- ◆ MQトピック
 - デフォルト・キューマネージャー（統合ノード、独立統合サーバーに関連付けられたキューマネージャー）
 - 外部のキューマネージャー（MQEndpointポリシーで指定したキューマネージャー）
- ◆ MQTTトピック
 - 組み込みMQTTブローカー（統合ノード構成に標準で組み込まれているMQTTブローカー。独立統合サーバーでは利用不可）
 - 外部のMQTTブローカー（MQTTPublishポリシーで指定したMQTTブローカー）
- ◆ 循環ログ・ファイル
 - /var/mqsi/common/monitoringEvents下のログ・ファイル
 - ✓ JSON形式 : MonitoringEvents_<統合サーバ>.json
 - ✓ XML形式 : MonitoringEvents_<統合サーバ>.xml
- ◆ ELK stack
 - 外部のELK stack(Elasticsearch、Kibana、Beats、Logstash)

トピック

■ MQ/MQTTの出力先トピック

◆ 統合ノード構成

• MQトピック

- ✓ XML : \$SYS/Broker/*integrationNodeName*/Monitoring/*integrationServerName*/*application_name*/*flow_name*
- ✓ JSON : \$SYS/Broker/*integrationNodeName*/MonitoringEvents/JSON/*integrationServerName*/*flow_name*

• MQTTトピック

- ✓ XML : IBM/IntegrationBus/*integrationNodeName*/Monitoring/*integrationServerName*/*application_name*/*flow_name*
- ✓ JSON : IBM/IntegrationBus/*integrationNodeName*/MonitoringEvents/JSON/*integrationServerName*/*flow_name*

◆ 独立統合サーバー構成

- MQトピック : \$SYS/Broker/integration_server/Monitoring/*integrationServerName*/*application_name*/*flow_name*
- MQTTトピック : IBM/IntegrationBus/integration_server/Monitoring/*integrationServerName*/*application_name*/*flow_name*

◆ サブスクライブする際は、ワイルドカード (+、#) を使用してまとめて取得可

• 例

```
$SYS/Broker/+/Monitoring/#
```

構成概要

■ メッセージフロー・モニタリング機能を有効にするには以下の構成、設定を行う

1. イベント・ソースの構成

- どのイベントを出力させるか、以下のいずれかの方法で構成
 - ① ノードのモニター・プロパティで構成する方法
 - ② モニタリング・プロファイルで構成する方法
 - ✓ モニタリング・プロファイルではフロー自身を変更することなく、イベントの設定変更が可能
 - ✓ モニタリング・プロファイルの設定のほうがノードのモニター・プロパティの設定より優先される

2. キューマネージャー/MQTTの構成 (MQ/MQTTトピックに出力する場合のみ)

- イベントの出力先とするキューマネージャー、もしくはMQTTブローカーを構成する

3. イベント出力先の設定

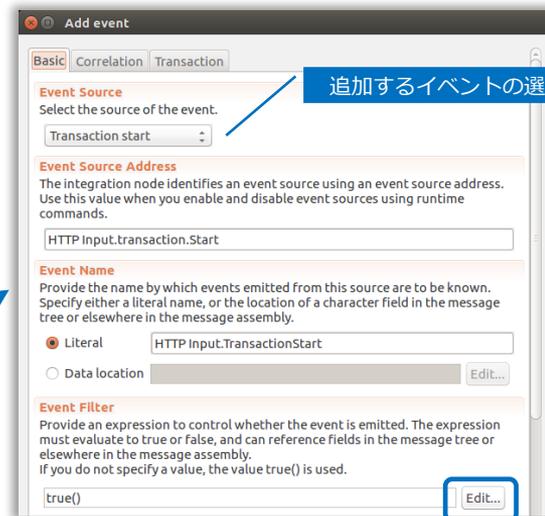
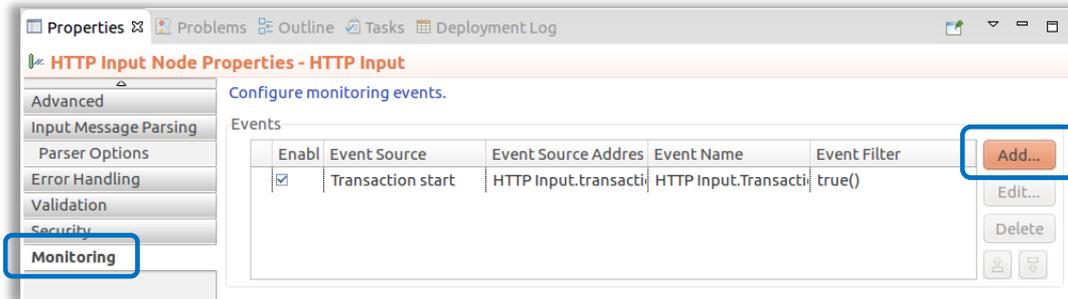
- イベントの出力先を構成ファイル(.yaml)のEventsセクション/BusinessEventsサブセクションで設定
- V10以前と同様に、mqsischangepropertiesで設定することも可能

4. モニタリングのアクティブ化

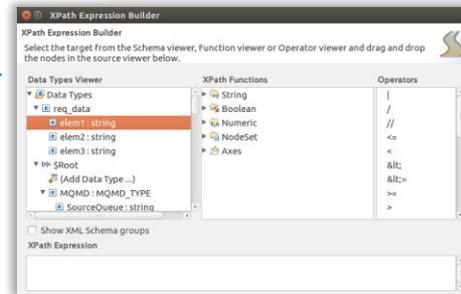
- メッセージフロー・モニタリングのアクティブ化を構成ファイル(.yaml)のMonitoringセクション/MessageFlowサブセクションで設定
- V10以前と同様に、mqsischangeflowmonitoringで設定することも可能

1. イベント・ソースの構成

- ① ノードのモニター・プロパティで構成する方法
- ◆ ツールキットにて各ノードのモニタリング・プロパティに設定



イベントの出力条件をフィルターで設定可能
XPath式で指定
デフォルトは「true()」（常に出力)
XPathでメッセージツリーの特定の項目の値を条件に
出力の有無を指定することも可
(XPathビルダーでメッセージ定義を表示させるには
事前にメッセージ定義を作成しておく必要がある)

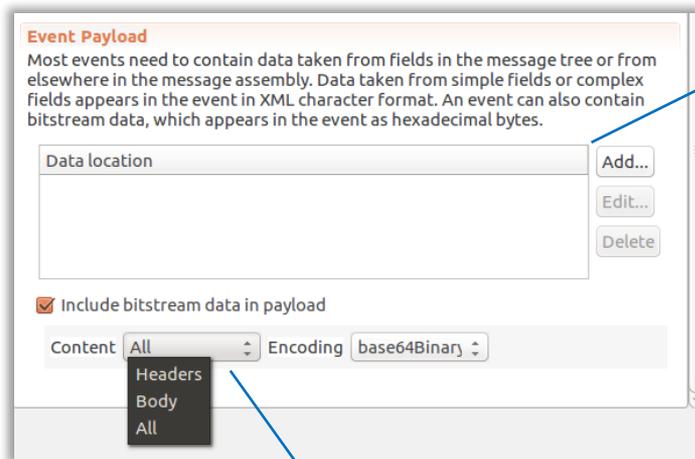


設定方法詳細は以下参照

<https://www.ibm.com/docs/en/app-connect/12.0?topic=cmmf-configuring-monitoring-event-sources-by-using-monitoring-properties>

1. イベント・ソースの構成

- ◆ イベント・メッセージに業務メッセージの特定の項目やメッセージ全体のビット・ストリームを含めることも可能

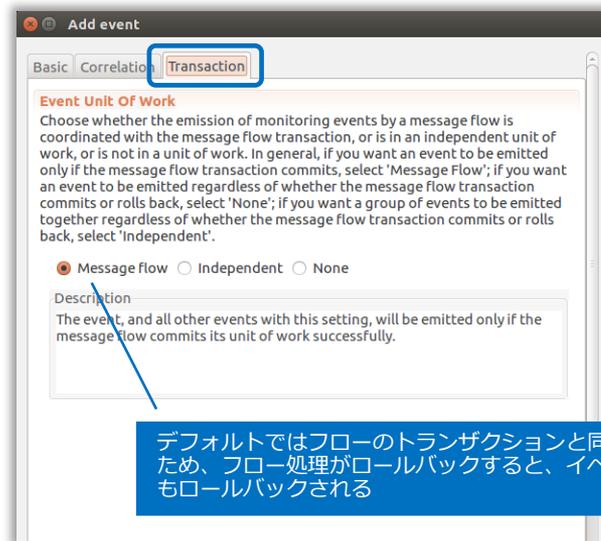
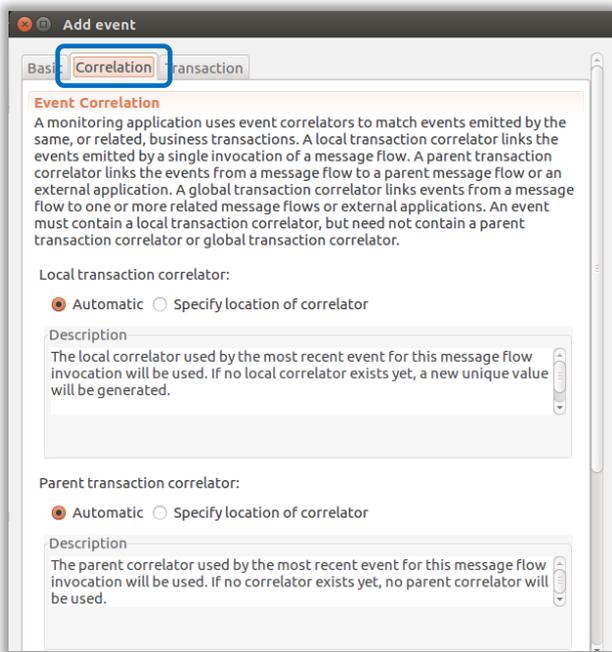


メッセージツリーの任意の項目を指定することが可能

業務メッセージのビット・ストリームも含めることができる
Headers、Body、All(Headers+Body)から選択

1. イベント・ソースの構成

- ◆ 相関(Correlation)タブで複数のイベントの関連付け、トランザクション・タブでフロー・トランザクションに合わせてイベントの出力を制御することも可能



1. イベント・ソースの構成

② モニタリング・プロファイルで構成する方法

1. モニタリング・プロファイルにイベント・ソースを設定

- 任意のXMLファイル(拡張子は「.monprofile.xml」とする)に、以下のXMLスキーマに従ってイベント・ソースの設定を記述
 - ✓ `Install_root/server/sample/Monitoring/MonitoringProfile.xsd`
- モニタリング・プロファイルはポリシー・プロジェクトに格納
- デプロイされているフローのモニター・プロパティ設定（前述）からプロファイルを生成することも可能
 - ✓ XMLスキーマを参考に1からプロファイルを作成するより、開発環境等にデプロイしたモニター設定済みフローからプロファイルを生成し、そのプロファイルをベースに作成したほうが容易
 - ✓ `mqsiextractflowmonitoring`コマンドもしくは管理REST APIを利用
 - `mqsiextractflowmonitoring`コマンドについては下記リンク参照
<https://www.ibm.com/docs/en/app-connect/12.0?topic=commands-mqsiextractflowmonitoring-command>
 - 管理REST APIのパス
`/apiv2/servers/{server}/applications/{application}/messageflows/{messageflow}/extract-monitoring-profile`
 - ✓ 生成されたプロファイルをポリシー・プロジェクトのインポートする

2. メッセージフローとモニタリング・プロファイルの関連付け

- BARファイルのメッセージフローの「モニタリング・プロファイル名」プロパティにプロファイル名(拡張子は除く)を指定
 - ✓ 「DefaultPolicies」プロジェクトに格納した場合はプロファイル名だけで指定可能
 - ✓ 他の名前プロジェクトに格納した場合は以下のフォーマットで指定
 - {ポリシー・プロジェクト名}:プロファイル名

3. デプロイ

- ポリシー・プロジェクトはメッセージフローのプロジェクトより先にデプロイ、もしくは同じBARファイルに格納してデプロイする

設定方法詳細は以下参照

<https://www.ibm.com/docs/en/app-connect/12.0?topic=cirrbmf-configuring-monitoring-event-sources-by-using-monitoring-profile>

1. イベント・ソースの構成

■ 参考

モニタリング・プロファイル例

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<profile:monitoringProfile xmlns:profile="http://www.ibm.com/xmlns/monitoring/11/profile" profile:version="2.0">
  <profile:eventSource profile:eventSourceAddress="HTTP Input.transaction.Start" profile:enabled="true">
    <profile:eventPointDataQuery>
      <profile:eventIdentity>
        <profile:eventName profile:literal="HTTP Input.TransactionStart"></profile:eventName>
      </profile:eventIdentity>
      <profile:eventCorrelation>
        <profile:localTransactionId profile:sourceOfId="automatic"></profile:localTransactionId>
        <profile:parentTransactionId profile:sourceOfId="automatic"></profile:parentTransactionId>
        <profile:globalTransactionId profile:sourceOfId="automatic"></profile:globalTransactionId>
      </profile:eventCorrelation>
      <profile:eventFilter profile:queryText="true()"></profile:eventFilter>
    </profile:eventPointDataQuery>
    <profile:applicationDataQuery></profile:applicationDataQuery>
    <profile:bitstreamDataQuery profile:bitstreamContent="all" profile:encoding="base64Binary"></profile:bitstreamDataQuery>
  </profile:eventSource>
  <profile:eventSource profile:eventSourceAddress="Compute.terminal.out" profile:enabled="true">
    <profile:eventPointDataQuery>
      <profile:eventIdentity>
        <profile:eventName profile:literal="Compute.OutTerminal"></profile:eventName>
      </profile:eventIdentity>
      <profile:eventCorrelation>
        <profile:localTransactionId profile:sourceOfId="automatic"></profile:localTransactionId>
        <profile:parentTransactionId profile:sourceOfId="automatic"></profile:parentTransactionId>
        <profile:globalTransactionId profile:sourceOfId="automatic"></profile:globalTransactionId>
      </profile:eventCorrelation>
      <profile:eventFilter profile:queryText="true()"></profile:eventFilter>
    </profile:eventPointDataQuery>
    <profile:applicationDataQuery></profile:applicationDataQuery>
    <profile:bitstreamDataQuery profile:bitstreamContent="none" profile:encoding="none"></profile:bitstreamDataQuery>
  </profile:eventSource>
</profile:monitoringProfile>
```

HTTP Inputノードの
transaction.Startイベント
の指定

Computeノードの
terminal.outイベント
の指定

2. キューマネージャー／MQTTの構成

■ キューマネージャーに出力する場合

- ◆ 任意のキューマネージャーをローカル、もしくはリモートに作成する
- ◆ 当機能にキューマネージャーの関連付けは必須ではない
- ◆ 関連付けを行わない場合
 - キューマネージャーへの接続情報を設定したMQEndpointポリシーを作成し、ランタイムにデプロイしておく
- ◆ 関連付けを行う場合
 - 前提情報の「キューマネージャーの関連付け方法」を参照

■ MQTTブローカーに出力する場合

- ◆ 組み込みのMQTTブローカーを利用する場合
 - 統合ノード構成でのみ利用可能
 - デフォルトで利用可能となっているため、明示的な設定は不要
 - ✓ node.conf.yamlの以下のプロパティで設定されている

```
MQTTServer:  
#enabled: true          # Enables the MQTT server  
#port: 11883           # Sets the port for the MQTT server
```

- ◆ 外部のMQTTブローカーを利用する場合
 - 任意のMQTTブローカーをローカル、もしくはリモートに構成する
 - MQTTブローカーへの接続情報を設定したMQTTPublishポリシーを作成し、ランタイムにデプロイしておく

3. イベント出力先の設定

■ 構成ファイル(.yaml)の以下のセクションで設定

◆ Eventsセクション/BusinessEventsサブセクション

- 出力先のセクションのenabledプロパティをtrueに設定
- 外部のMQ/MQTTに出力する場合は、事前に作成/デプロイしたMQEndpointポリシー、またはMQTTPublishポリシーをpolicyプロパティに設定する
- ELK stackに出力する場合、ELK connectionを別途設定（次ページ）

設定例

```
Events:
  BusinessEvents: # Monitoring events
    MQ:
      #policy: "          # Specify a {policy project}:policy if not using 'defaultQueueManager'
      enabled: true      # Set true or false, default false
      #format: "         # Set string or none
      #outputFormat: 'xml' # Set comma separated list of one or more of : json,xml. Defaults to 'xml'
      #publishRetryInterval: 0 # Set the retry interval (in milliseconds), to pause all publications and retry, when publication failures are causing serious delay to the transaction.
    MQTT:
      #policy: "          # Specify a {policy project}:policy if not using 'default MQTTServer'
      enabled: true      # Set true or false, default false
      #outputFormat: 'xml' # Set comma separated list of one or more of : json,xml. Defaults to 'xml'
      #publishRetryInterval: 0 # Set the retry interval (in milliseconds), to pause all publications and retry, when publication failures are causing serious delay to the transaction.
    ELK:
      #enabled: false     # Set true or false, default false
      #outputFormat: 'json' # Set json
      #elkConnections: "  # Name of the ELK connection to use, for example 'elkConnection1', must be defined in the ELKConnections section below.
  File:
    enabled: true        # Set true or false, default false
    #outputFormat: 'json' # Set comma separated list of one or more of : json,xml. Defaults to 'json'
```

設定方法詳細は以下参照

<https://www.ibm.com/docs/en/app-connect/12.0?topic=events-configuring-publication-event-messages>

3. イベント出力先の設定

- ELK stackに出力する場合、ELK connectionを設定
 - ◆ Logstashのinputプロトコルとして以下を選択可能
 - beats、beatsTls、http、https
 - ◆ 接続情報や必要に応じてその他のプロパティを設定

設定例

```
ELKConnections:
# Description for ELK Connections.
elkConnection1:
  elkProtocol: 'beats'           # Logstash input protocol. Valid values are: 'beats', 'beatsTls', 'http', or 'https'.
  hostname: 'localhost'        # Hostname for the elkProtocol endpoint.
  port: 5044                    # Port for the elkProtocol endpoint.
  uploadIntervalMilliSecs: 60000 # Interval between uploading cached data, set in milliseconds.
# elkCredential: ""           # Set an 'elk' credential alias name to enable basic authentication, if it is required by the Logstash input protocol.
# keystoreFile: '/path/to/keystore.jks' # Set the path to the keystore to be used, if it is required by the Logstash input protocol.
# keystorePass: 'P4s5w0rd'         # Set the password, or 'keystore' credential alias to the password, of the keystore.
# keyAlias: ""                     # Set the alias name of the private key, if mutual authentication is required by the Logstash input protocol.
# keyPass: ""                       # Set the password, or 'keystorekey' credential alias to the password, for accessing the private mutual authentication key.
# truststoreFile: '/path/tp/truststore.jks' # Set the path to the truststore to be used, if it is required by the Logstash input protocol.
# truststorePass: 'P4s5w0rd'       # Set the password, or 'truststore' credential alias to the password, for accessing the truststore.
```

設定方法詳細は以下参照

<https://www.ibm.com/docs/en/app-connect/12.0?topic=pmwm-configuring-integration-servers-send-logs-events-logstash-in-elk-stack>

3. イベント出力先の設定

■ mqsichangepropertiesコマンドで設定する場合

- ◆ 旧バージョンとの互換性のため、mqsichangepropertiesコマンドで出力先を設定することも可能

統合ノード構成の組み込みMQTTブローカーへ出力する場合の設定例

```
mqsichangeproperties integrationNode -b pubsub -o BusinessEvents/MQTT -n enabled -v true
```

- overridesフォルダの構成ファイル(.yaml)に設定されるため、ランタイムを再起動したタイミングで反映される

設定方法詳細は以下参照

<https://www.ibm.com/docs/en/app-connect/12.0?topic=events-configuring-publication-event-messages>

4. モニタリングのアクティブ化

■ 構成ファイル(.yaml)で設定する場合

- ◆ Monitoringセクション/MessageFlowサブセクションの「publicationOn」プロパティをactiveに設定

```
Monitoring:
MessageFlow:
  publicationOn: active          # choose 1 of : active|inactive
  eventFormat: MonitoringEventV2 # choose 1 of : MonitoringEventV2|WMB
```

- WebUI/管理REST APIから動的にactive/inactiveに設定することも可

設定方法詳細は以下参照

<https://www.ibm.com/docs/en/app-connect/12.0?topic=flows-activating-monitoring>

■ mqsichangeflowmonitoringコマンドで設定する場合

- ◆ 旧バージョンとの互換性のため、mqsichangeflowmonitoringコマンドでアクティブ化することも可能
- ◆ フロー単位にアクティブ/インアクティブを切り替えることが可
 - ただし、V10までできていたイベント・ソース単位の使用可能化/不可化はできない(-sオプションは使えない)
 - ✓ イベント・ソース単位の設定はノード・プロパティやモニタリング・プロファイルで行う

app01アプリケーションのflow01メッセージフローのモニタリングをアクティブに設定する場合

```
mqsichangeflowmonitoring integrationNode -e integrationServer -c active -k app01 -f flow01
```

- ◆ このコマンドで設定した内容はランタイムの再起動やフローの再デプロイ後も保持される
 - 再起動や再デプロイでリセットさせるには、「--non-persist」オプションを指定
 - ✓ この場合、再起動や再デプロイ後は構成ファイルの設定が有効となる

設定方法詳細は以下参照

<https://www.ibm.com/docs/en/app-connect/12.0?topic=command-mqsichangeflowmonitoring-windows-linux-aix-systems>

ファイル出力例

■ 循環ログ・ファイルへの出力例

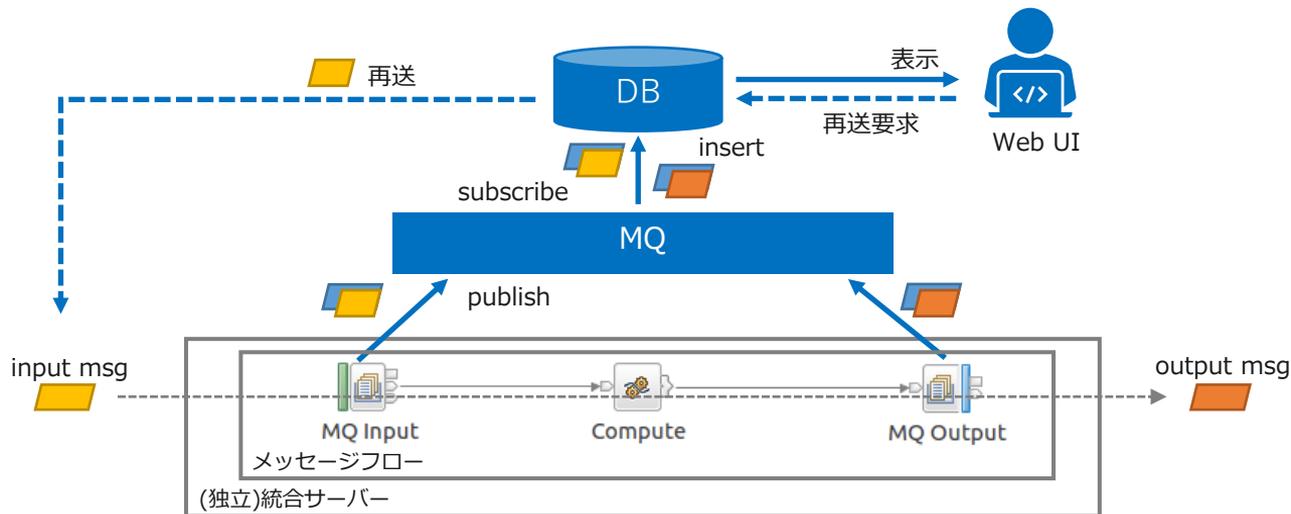
```
{
  "type": "ace_monitoring_event",
  "ibm_product": "IBM App Connect Enterprise",
  "ibm_recordtype": "monitor",
  "host": "ISEP20220119-1458-mysterious-01.jp-ise.com",
  "module": "SBK02.is01",
  "ibm_serverName": "is01",
  "productVersion": "120100",
  "profileVersion": "11",
  "eventSourceAddress": "HTTP Input.transaction.Start",
  "eventName": "HTTP Input.TransactionStart",
  "ibm_datetime": "2023-12-28T08:28:45.013253Z",
  "localTransactionId": "f0ca1c7e-a55a-11ee-afd1-7f0000010000-1",
  "parentTransactionId": "",
  "globalTransactionId": "",
  "applicationName": "HTTPRequest01",
  "messageFlowUniqueName": "SBK02.is01.HTTPRequest01.httprequest01",
  "messageFlowName": "httprequest01",
  "messageFlowNodeName": "HTTP Input",
  "messageFlowNodeType": "ComIbmWSInputNode",
  "bitstreamData": {
    "bitstream":
      "R0VUIGh0dHA6Ly9sb2NhbGhvc3Q6NzgwMS9odHRwcmVxdwVzdDAXIEhUVFAvMS4xDQpIb3N00iBsb2NhbGhvc3Q6NzgwMQ0KVXN1ci1BZ2VudDogY3VybyC83LjYxLjENCkFjY2VwdDogKi8qDQpYLU9yaWdpbmFsLUhUVFAtQ29tbWZudDogR0VUIGh0dHA6Ly9sb2NhbGhvc3Q6NzgwMS9odHRwcmVxdwVzdDAXIEhUVFAvMS4xDQpYLVN1cnZlc10YW110iBsb2NhbGhvc3Q6NzgwMS9odHRwcmVxdwVzdDogaHR0cA0KDQo=",
    "serializationEncoding": "base64Binary",
    "dataCCSID": "1208",
    "dataEncoding": "546"
  },
  "ibm_sequence": "1"
}
```

※JSONを整形して表示（実際には上記内容が1行で出力される）

Record & Replay

機能概要

- メッセージフローを通過する業務メッセージを保管、表示、再送する機能
 - ◆ メッセージフロー・モニタリングの機能 (MQ Pub/Sub) を利用して業務メッセージをキャプチャー
 - ◆ イベント・メッセージとともに業務メッセージをデータベースに格納
- データベースに格納したイベント・メッセージ/業務メッセージはWeb UIに表示できる
 - ◆ バイナリデータを含むメッセージは表示できない場合あり
- 業務メッセージをWeb UIから再送できる
 - ◆ 再送できるのはMQメッセージのみ
- 監査証跡や問題判別目的で利用



Web UIによる表示、再送

■ 画面イメージ

IBM App Connect

ノード: IN01 / サーバー: ex1 / 記録および再生ストア: MBRECORD_store

MBRECORD_store

メッセージ 再生

メッセージ

DBに格納されたメッセージの一覧

<input type="checkbox"/>	Event time UTC	Local transaction ID	Parent transaction ID	Global transaction ID	Data	Errors	Event name
<input type="checkbox"/>	2020-09-03 13:07:42.380	631e255c-ed66-11ea-96fb-7f0001010000-1	-	-	↓		MQ Input.TransactionStart
<input type="checkbox"/>	2020-09-03 13:35:44.790	4def9aea-edea-11ea-beca-7f0001010000-1	-	-	↓		MQ Input.TransactionStart
<input type="checkbox"/>	2020-09-03 22:50:49.159	4def9aea-edea-11ea-beca-7f0001010000-2	-	-	↓		MQ Input.TransactionStart
<input type="checkbox"/>	2020-09-03 22:51:13.772	4def9aea-edea-11ea-beca-7f0001010000-3	-	-	↓		MQ Input.TransactionStart
<input type="checkbox"/>	2020-09-03 22:53:04.399	4def9aea-edea-11ea-beca-7f0001010000-4	-	-	↓		MQ Input.TransactionStart
<input type="checkbox"/>	2020-09-17 01:54:34.331	7e115274-f888-11ea-962b-7f0001010000-1	-	-	↓		MQ Input.TransactionStart
<input type="checkbox"/>	2020-09-17 02:08:31.189	2dccc9b8-f88a-11ea-adb1-7f0001010000-1	-	-	↓		MQ Input.TransactionStart
<input type="checkbox"/>	2020-09-17 02:16:48.340	ca9d30f6-f88b-11ea-adb1-7f0001010000-1	-	-	↓		MQ Input.TransactionStart

IBM App Connect

ノード: IN01 / サーバー: ex1 / 記録および再生ストア: MBRECORD_store / メッセージ: 414d51204953514d202...

2020-09-03 13:07:42.380 (UTC)

プロパティ データ

このビット・ストリームをダウンロード

業務メッセージの内容表示
(バイナリデータを含む場合は表示できない場合あり。ファイルにDL可能)

MD MQSTR AMQ ISQM 9MB 9% ISQM shoheim 1000 amqspst

2020090313074237 aaaaa

IBM App Connect

ノード: IN01 / サーバー: ex1 / 記録および再生ストア: MBRECORD_store

MBRECORD_store

メッセージ 再生

再生先: MQDestination

1 item selected 再生 ● 削除 ● | Cancel

<input type="checkbox"/>	Event time UTC	Replay status	Local transaction ID	Parent transaction ID	Global transaction ID	Data	Errors	Event name
<input type="checkbox"/>	2020-09-03 13:07:42.380	成功	631e255c-ed66-11ea-96fb-7f0001010000-1	-	-	↓		MQ Input.TransactionStart
<input type="checkbox"/>	2020-09-03 13:35:44.790	成功	4def9aea-edea-11ea-beca-7f0001010000-1	-	-	↓		MQ Input.TransactionStart
<input checked="" type="checkbox"/>	2020-09-03 22:50:49.159	再生準備完了	4def9aea-edea-11ea-beca-7f0001010000-2	-	-	↓		MQ Input.TransactionStart

任意のメッセージを選択し、再送できる
(MQメッセージのみ)

考慮点

- 機能を利用する上での考慮点
 - ◆ データベースに保管したメッセージを定期的に削除する運用が必要
 - Web UIから削除することはできない

構成概要

■ Record&Replay機能を有効にするには以下の構成、設定を行う

1. メッセージフロー・モニタリングの構成

- Record&Replay機能は内部的にメッセージフロー・モニタリングの機能を利用するため、「メッセージフロー・モニタリング」の章の構成、設定を行う
- イベント・メッセージには、**業務メッセージの本体を搭載**するよう設定する
- **イベントの出力先はMQとし、キューマネージャーの関連付けも行う**こと

2. データベースの構成

- メッセージ保管用のデータベースを構成する
 - ✓ IBM Db2、Microsoft SQL Server、Oracleをサポート
 - ✓ データベース作成のためのスクリプトを提供
- 統合サーバーから接続するためのODBC設定やクレデンシャル情報の設定も行う

3. メッセージ保管のための設定

- 構成ファイル(.yaml)のRecordReplayセクションにデータベースの情報やキャプチャー対象のイベントの情報等を設定

4. メッセージ再送のための設定（オプション）

- MQメッセージの再送を行う場合は、構成ファイル(.yaml)のRecordReplayセクションに再送先とするMQの情報を設定

5. Web UIによる操作

1. メッセージフロー・モニタリングの構成

- メッセージのキャプチャーを行うフローにメッセージフロー・モニタリングの設定を行う
 - ◆ イベントに、業務メッセージ全体のビット・ストリームを搭載するよう設定する
 - MQメッセージの再送を行う場合、ContentはAll(MQMD+body)を選択
 - ✓ MQMDが含まれていると、再送されるメッセージにはそのMQMDの内容がセットされる
 - ✓ MQMDが含まれていない(Content=Body)場合、再送されるメッセージのMQMDは統合サーバーがデフォルト値を設定
- イベントの出力先は、MQとし、キューマネージャーの関連付けを行う
 - ◆ メッセージフロー・モニタリング機能としてはキューマネージャーの関連付けは必須ではないが、Record&Replayでは内部的にSYSTEMキューを利用するため関連付けしておく必要がある
- 具体的な設定は前章「メッセージフロー・モニタリング」参照
 - ◆ キューマネージャーの関連付けは「前提情報」の章の「キューマネージャーの関連付け方法」を参照

2. データベースの構成

■ メッセージ保管用のデータベースを作成

- ◆ Record&Replay機能に必要なデータベースやテーブルを作成するためのスクリプトを提供
 - `<install_dir>/server/ddl/db2/DataCaptureSchema.sql`
 - 作成されるデータベース定義
 - ✓ データベース : MBRECORD
 - ✓ スキーマ : デフォルト・スキーマ
 - ✓ テーブル : WMB_MSGS、WMB_BINARY_DATA、WMB_EVENT_TYPES、WMB_EVENT_FIELDS
 - ✓ スクリプトを修正すれば、データベース名とスキーマの変更は可能
 - ただし修正した場合は後述の構成ファイルの設定も変更が必要
 - ✓ 保管できる業務メッセージのサイズはデフォルトでは5MBが上限
 - サイズを拡張する場合は、スクリプトのWMB_BINARY_DATAテーブルのDATA項目のサイズを変更
- ◆ スクリプトを実行し、データベース、テーブルを作成する

```
db2 -tvf DataCaptureSchema.sql
```

- ◆ 作成したデータベースのODBC接続定義の作成
 - 使用するデータベース環境に応じて、ODBC接続定義を作成

設定方法詳細は以下参照

<https://www.ibm.com/docs/en/app-connect/12.0?topic=data-creating-configuring-database-recording>

2. データベースの構成

- ◆ データベース接続のクレデンシャル情報を設定
 - mqsisetdbparmsコマンドでクレデンシャル情報を設定

データソースMBRECORDの設定例

```
mqsisetdbparms IN01 -n odbc::MBRECORD -u db2inst1 -p 1qazxsw2
```

- 事前にVaultを構成し、暗号化してクレデンシャル情報を設定することも可
 - ✓ 統合ノード構成はFixpack11.0.0.7から、独立統合サーバー構成はFixpack 11.0.0.6からVaultを利用可能

統合ノード起動時のデータソースMBRECORDの設定例

```
mqsicredentials IN01 --create --integration-server ex1 --credential-type odbc --credential-name MBRECORD --username db2inst1 --password 1qazxsw2
```

Vaultの詳細は以下参照

<https://www.ibm.com/docs/en/app-connect/12.0?topic=enterprise-configuring-encrypted-security-credentials>

3. メッセージ保管のための設定

- 統合サーバーの構成ファイル(server.conf.yaml)にメッセージを保管するための設定を行う
 - ◆ 統合ノード構成であっても各統合サーバーのserver.conf.yamlに設定する
 - ◆ RecordReplayセクションのrecordReplayEnabledプロパティをtrueに設定

```
RecordReplay:  
  recordReplayEnabled: true # Set to true to enable all Record and Replay functionality. Default is true.
```

- ◆ StoresサブセクションにあるテンプレートStoreTemplateをコピーして、任意の名前でStoreを定義
 - ここではメッセージを格納するために用意したデータベースやキューの情報を設定する
 - 個々のプロパティの意味は、テンプレートに記載されている説明を確認

設定例

```
Stores:  
  MBRECORD_store:  
    dataSource: 'MBRECORD'  
    schema: 'db2inst1'  
    storeMode: 'all'  
    queue: 'SYSTEM.BROKER.DC.RECORD'  
    backoutQueue: 'SYSTEM.BROKER.DC.BACKOUT'  
    useCoordinatedTransaction: false  
    commitCount: 10  
    threadPoolSize: 10  
    commitIntervalSecs: 5
```

3. メッセージ保管のための設定

- ◆ SourcesサブセクションにあるテンプレートSourceTemplateをコピーして、任意の名前でSourceを定義
 - ここではRecord&Replay機能でキャプチャーするイベントのトピック情報と保管先のStore情報(前述)を設定する
 - 個々のプロパティの意味は、テンプレートに記載されている説明を確認

設定例

```
Sources:  
MBRECORD_source:  
  #topic: '$SYS/Broker/[nodename]/Monitoring/[servername]/[applicationname]/[flowname]'  
  topic: '$SYS/Broker/IN01/Monitoring/IS01/#'  
  store: 'MBRECORD_store'
```

- ✓ ここで指定したトピック・ストリングでメッセージフロー・モニタリング機能でPublishされるイベントをSubscribeする
- ✓ ワイルドカードを使用して、複数のイベントをまとめてキャプチャー可能

4. メッセージ再送のための設定(オプション)

- MQメッセージの再送を行う場合は以下の設定を行う
 - ◆ RecordReplayセクションのDestinationsサブセクションにあるテンプレートMQDestinationTemplateをコピーして、任意の名前でDestinationを定義する
 - ここで再送先のキュー名、キューマネージャー名を指定する
 - 個々のプロパティの意味は、テンプレートに記載されている説明を確認

設定例

```
Destinations:  
  MQ_destination:  
    endpointType: 'WMQDestination'  
    endpoint: 'wmq:/msg/queue/INPUTQ@IN01QM'
```

5. Web UIによる操作

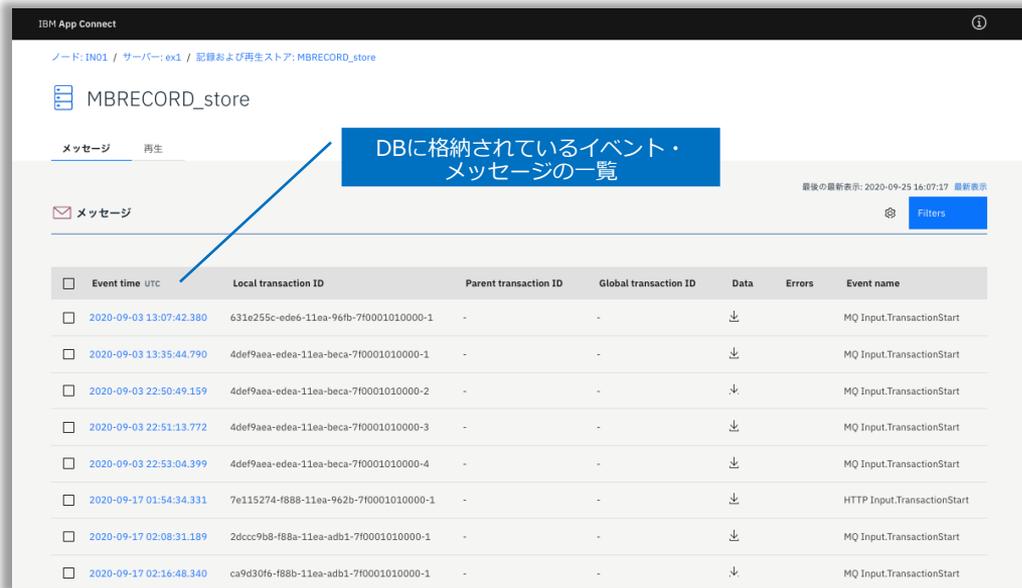
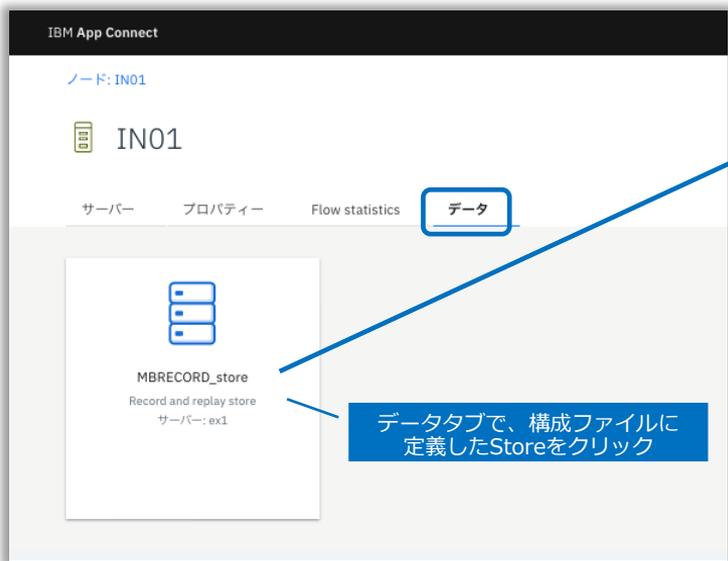
■ Web UIにアクセスし、保管されたメッセージが表示されることを確認

- ◆ ここまでの構成を実施後、対象のメッセージフローの処理を実行し、メッセージが保管された状態にしておくこと
- ◆ 任意のブラウザからWeb UIにアクセス

URL例

http://localhost:4414/

※Web UIの構成に応じて、プロトコル(http or https)やポートは異なる



5. Web UIによる操作

■ メッセージ内容の確認

前の画面で、個別のメッセージのEvent timeの箇所をクリックすると詳細が確認できる

IBM App Connect

ノード: IN01 / サーバー: ex1 / 記録および再生ストア: MBRECORD_store / メッセージ: 414d51204953514d202...

✉ 2020-09-03 13:07:42.380 (UTC)

プロパティ データ

プロパティ	Persisted value
ID	414d51204953514d202020202020202020202020203919505f06113925:414d5120495351...
Application name	MQ01
Application UUID	-
Data	↓
Detail	Q1
Errors	-
Event name	MQ Input.TransactionStart
Event source	MQ Input.transaction.Start
Event time (local)	2020-09-03 22:07:42.380
Event time (UTC)	2020-09-03 13:07:42.380
Event type	MQ Input.TransactionStart
Flow name	mq01
Flow node name	MQ Input

IBM App Connect

ノード: IN01 / サーバー: ex1 / 記録および再生ストア: MBRECORD_store / メッセージ: 414d51204953514d202...

✉ 2020-09-03 13:07:42.380 (UTC)

プロパティ データ

このビット・ストリームをダウンロード

```
MD 000000000000MQSTR AMQ ISQM 9P 000000000000 ISQM shoheim 100000000000 amqsput
2020090313074237 00000000aaaaa
```

「データ」タブでメッセージの内容の確認およびダウンロードが可能

MQメッセージの場合、バイナリデータのMQMDも含まれるため、一部文字化けしているように表示される

5. Web UIによる操作

■ MQメッセージの再送

The screenshot shows the IBM App Connect interface for the MBRECORD_store. A blue callout box contains the instruction: 対象メッセージをチェックし、「再生対象としてマーク」をクリック (複数まとめてチェック可). The interface shows a table of messages with a '再生対象としてマーク' button highlighted by a red box. An arrow points from this button to the text '(次ページへ)'.

IBM App Connect

ノード: IN01 / サーバー: ex1 / 記録および再生ストア: MBRECORD_store

MBRECORD_store

メッセージ 再生

最後の最新表示: 2020-09-25 16:48:09 最新表示

メッセージ

1 item selected

再生対象としてマーク | Cancel

<input type="checkbox"/>	Event time UTC	Local transaction ID	Parent transaction ID	Global transaction ID	Data	Errors	Event name
<input type="checkbox"/>	2020-09-03 13:07:42.380	631e255c-ed6-11ea-96fb-7f0001010000-1	-	-	↓		MQ Input.TransactionStart
<input type="checkbox"/>	2020-09-03 13:35:44.790	4def9aea-edea-11ea-beca-7f0001010000-1	-	-	↓		MQ Input.TransactionStart
<input checked="" type="checkbox"/>	2020-09-03 22:50:49.159	4def9aea-edea-11ea-beca-7f0001010000-2	-	-	↓		MQ Input.TransactionStart
<input type="checkbox"/>	2020-09-03 22:51:13.772	4def9aea-edea-11ea-beca-7f0001010000-3	-	-	↓		MQ Input.TransactionStart

(次ページへ)

5. Web UIによる操作

■ MQメッセージの再送

IBM App Connect

ノード: IN01 / サーバー: ex1 / 記録および再生ストア: MBRECORD_store

MBRECORD_store

メッセージ 再生

再生先 MQDestination

構成ファイルに定義したDestinationを選択

「再生」をクリックし、メッセージの再送を実行
再送が成功するとReplay statusの列に「成功」と表示される

1 item selected

再生 除去 Cancel

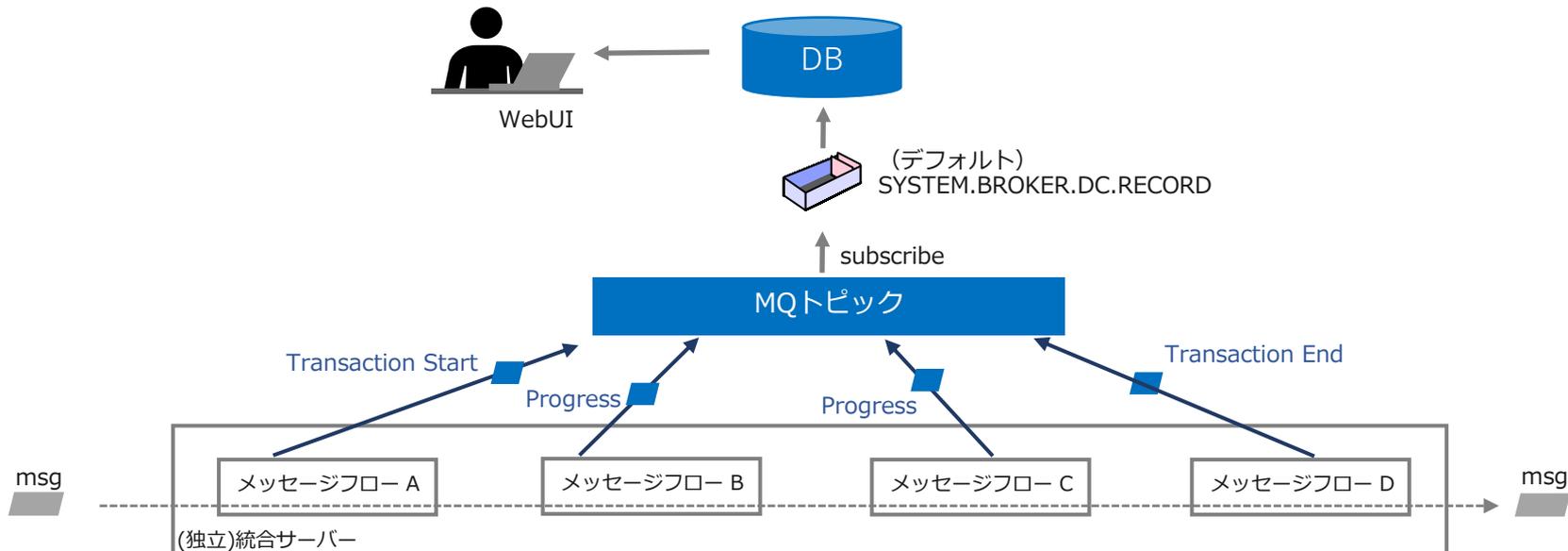
<input type="checkbox"/>	Event time UTC	Replay status	Local transaction ID	Parent transaction ID	Global transaction ID	Data	Errors	Event name
<input type="checkbox"/>	2020-09-03 13:07:42.380	成功	631e255c-ede6-11ea-96fb-7f0001010000-1	-	-	↓		MQ Input.Transaction
<input type="checkbox"/>	2020-09-03 13:35:44.790	成功	4def9aea-edea-11ea-beca-7f0001010000-1	-	-	↓		MQ Input.Transaction
<input checked="" type="checkbox"/>	2020-09-03 22:50:49.159	再生準備完了	4def9aea-edea-11ea-beca-7f0001010000-2	-	-	↓		MQ Input.Transaction

対象メッセージをチェック
(複数チェック可)

ビジネス・トランザクション・モニタリング

機能概要

- 複数のフローをまたがる一連のビジネス・トランザクションをモニタリングする機能
 - ◆ 複数のフローをまたがる一連のビジネス・トランザクションがどの処理まで完了しているか
 - ◆ 途中でエラーが発生していないか など



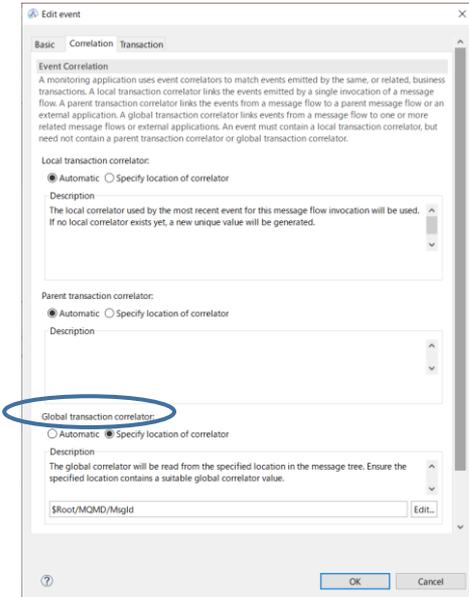
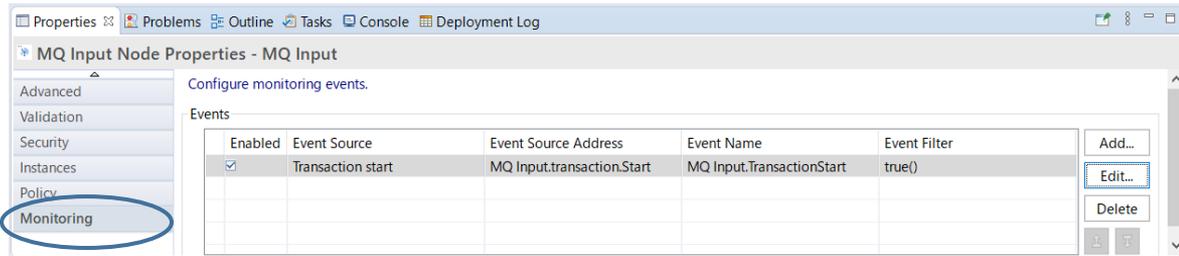
機能概要

- ポリシーとserver.conf.yamlで必要な情報を定義
 - ◆ ビジネス・トランザクションに含まれる各フローでモニタリングの設定を有効化
 - ◆ ビジネス・トランザクション・ポリシーでビジネス・トランザクションとしてモニタリングするイベントを定義
 - ・ 主なイベントとしては開始、終了、進行中、失敗など
 - ・ WebUIでポリシー定義を作成することも可能
 - ◆ server.conf.yamlでポリシー名や使用するデータソース名などの構成情報を定義
- グローバル・トランザクションIDを使用して、1つのビジネス・トランザクション内の各イベントを関連付け、処理の遷移を管理
 - ◆ メッセージ・ツリー内の任意のフィールドをグローバル・トランザクションIDとして指定可能
- WebUIからビジネス・トランザクションの結果を確認可能
 - ◆ Configureタブ ・・・モニタリングの定義を作成
 - ◆ Monitorタブ ・・・ビジネス・トランザクションのステータスをモニター

1. 各フローでモニタリング設定を有効化

■ 各フローでモニタリングの設定を有効化

- ◆ いずれかの方法で有効化が可能
 - ・ メッセージ・フロー・ノードの「Monitoring」プロパティ
 - ・ モニタリング・プロファイル
- ◆ グローバル・トランザクション・IDを利用し、関連情報がセットされている必要がある



2. モニタリング機能を有効化

■ 統合サーバでモニタリング機能を有効化

- ◆ server.conf.yamlで以下の設定を有効にする

server.conf.yaml例

```
Monitoring:
  MessageFlow:
    publicationOn: 'active'           # choose 1 of : active|inactive, default is inactive
                                       # Ensure Events.BusinessEvents.MQ|MQTT is set
    eventFormat: 'MonitoringEventV2' # choose 1 of : MonitoringEventV2|WMB
```

■ 統合ノードまたは独立統合サーバでビジネス・モニタリング・イベントを有効化

- ◆ node.conf.yamlまたはserver.conf.yamlで以下の設定を有効にする
 - イベントをパブリッシュするキュー・マネージャーをdefaultQueueManagerプロパティまたはポリシーで指定
 - イベント情報を保管するキュー・マネージャーと同一のキュー・マネージャーを指定

node.conf.yaml例

```
Events:
  BusinessEvents: # Monitoring events
  MQ:
    policy: "           # Specify a {policy project}:policy if not using 'defaultQueueManager'
    enabled: true      # Set true or false, default false
    format: "          # Set string or none
    outputFormat: 'xml' # Set comma separated list of one or more of : json,xml. Defaults to 'xml'
    publishRetryInterval: 0 # Set the retry interval (in milliseconds), to pause all publications and retry,
                           when publication failures are causing serious delay to the transaction.
```

3. データベースやMQの準備

- モニタリング・イベントやビジネス・トランザクション・データを保管するためのデータベースを用意
 - ◆ レコード&リプレイ機能と同一のテーブルを使用可能
 - ◆ DataCaptureSchema.sqlを実行して、必要なテーブルを作成
 - ◆ BusinessCaptureSchema.sqlを実行して、必要なテーブルを作成
 - ・ いずれも<製品導入ディレクトリ>/server/ddl/配下にddlが格納

- 統合ノードまたは統合サーバでデフォルト・キュー・マネージャーを設定
 - ◆ node.conf.yaml または server.conf.yaml でデフォルト・キュー・マネージャーを設定
 - ◆ 独立統合サーバの場合は、リモートのデフォルト・キュー・マネージャーの指定も可能

node.conf.yaml例

```
defaultQueueManager: 'BTMQM'
```

- 必要なシステム・キューが作成されていることを確認
 - ◆ ビジネス・トランザクション・モニタリングでは、SYSTEM.BROKER.DC.RECORDキューとSYSTEM.BROKER.DC.BACKOUTキューを使用
 - ◆ <製品導入ディレクトリ>/server/sample/wmq/iib_createqueues スクリプトを実行して作成可能

3. データベースやMQの準備

- ビジネス・トランザクション・モニタリングに使用するリソース情報を設定
 - ◆ server.conf.yaml のレコード&リプレイのStoresセクションでデータソースやキューの情報を設定

server.conf.yaml例

```
RecordReplay:
  Stores:
    BTMDataSource:
      dataSource: 'MBRECORD'
      schema: ''
      storeMode: 'all'
      queue: 'SYSTEM.BROKER.DC.RECORD'
      backoutQueue: 'SYSTEM.BROKER.DC.BACKOUT'
      useCoordinatedTransaction: false
      commitCount: 10
      threadPoolSize: 10
      commitIntervalSecs: 5
```

- レコード&リプレイ機能を有効化
 - ◆ server.conf.yaml でレコード&リプレイを有効化する

server.conf.yaml例

```
RecordReplay:
  recordReplayEnabled: true
```

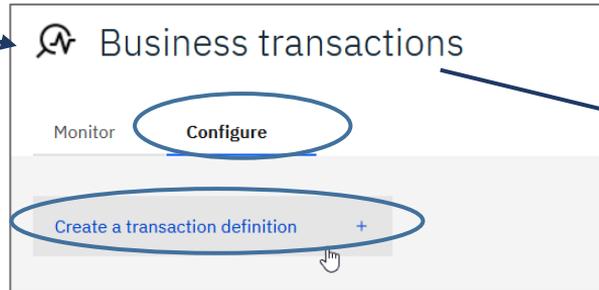
- 管理セキュリティを有効化している場合は、必要な権限を必要なユーザーに付与

4. ポリシーの定義

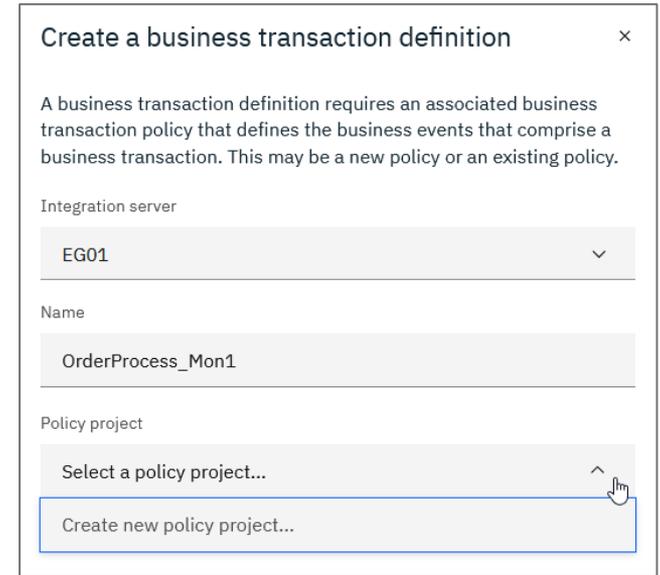
- Web UIを使用してビジネス・トランザクション・ポリシー定義を作成



ビジネス・トランザクション
アイコンをクリック



「Configure」タブで「Create a
transaction definition」をクリック



ビジネス・トランザクション・ポリシー
を作成

4. ポリシーの定義

Create a business event

OrderProc1

Type
Start

Select a monitoring event

Integration server
EG01

Integration
BMT_TEST

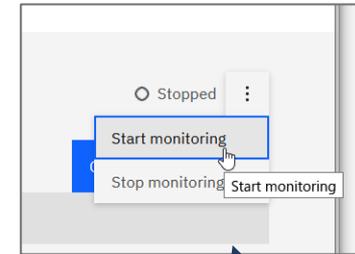
Library
None

Message flow
RECEIVE_ORDER

Monitoring event source
MQ Input.transaction.Start

Cancel Create

定義を作成後、「Start monitoring」で
モニタリングを開始



Business transactions

Monitor Configure

Create a transaction definition +

Business events for OrderProcess_Mon1

EG01

OrderProcess_Mon1

Create an event +

Event name	Event type	Message flow	Event source		
OrderProc1	Start	RECEIVE_ORDER	MQ Input.transaction.Start		
OrderProc2	Progress	PROC_ORDER	MQ Input.transaction.Start		
OrderProc3	End	COMPLETE_ORDER	MQ Input.transaction.End		

ビジネス・トランザクションのStartイベント、
Endイベントなどの定義を作成

5. モニター

「Monitor」タブで実行状況を確認

Business transactions

Monitor Configure

EG01

OrderProcess_Mon1

Transactions for OrderProcess_Mon1

Started | 最終更新: 数秒前 ⓘ ↺

▼ Last 30 minutes ▼

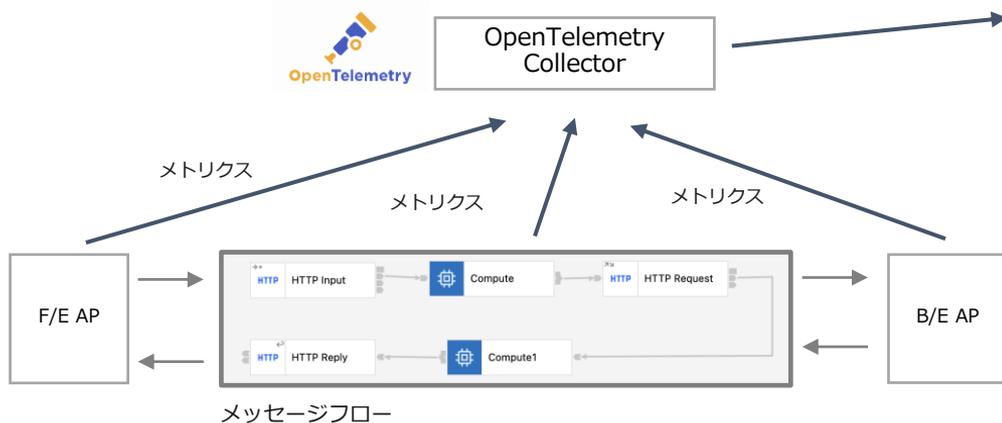
Transaction ID	Transaction status	Start time local	Last updated local	Duration
414d5120544553543031202020202020aa86836124772d03	In progress	2021-11-05 13:29:22.112	2021-11-05 13:29:22.131	27 min 1 sec
414d5120544553543031202020202020aa86836124773202	In progress	2021-11-05 13:29:45.151	2021-11-05 13:29:45.159	26 min 38 sec
414d5120544553543031202020202020aa86836124773203	In progress	2021-11-05 13:29:46.029	2021-11-05 13:29:46.032	26 min 37 sec
414d5120544553543031202020202020aa86836124773204	In progress	2021-11-05 13:29:46.799	2021-11-05 13:29:46.805	26 min 36 sec
414d5120544553543031202020202020aa86836124773205	In progress	2021-11-05 13:29:47.523	2021-11-05 13:29:47.525	26 min 36 sec
414d5120544553543031202020202020aa86836124773206	In progress	2021-11-05 13:29:48.240	2021-11-05 13:29:48.244	26 min 35 sec

OpenTelemetry

機能概説

■ OpenTelemetryの仕様に準じてメッセージフローのメトリクスを出力可能

- ◆ メッセージフローからOpenTelemetryのCollectorにメトリクスを出力
 - フロー処理の経過時間や処理の成否、処理ノードの各種属性など
 - MQ/HTTP/REST/SOAP/Callableノードで利用可能
- ◆ ACEに限らず、OpenTelemetryをサポートしているシステムをまたがって、トランザクションの一連のメトリクスを参照可能
- ◆ 以下のプラットフォームでサポート
 - Linux x86-64、Linux zSeries、Windows
- ◆ OpenTelemetryは仕様であるため、CollectorやモニターはOpenTelemetryに対応した製品・ツールを利用する必要がある

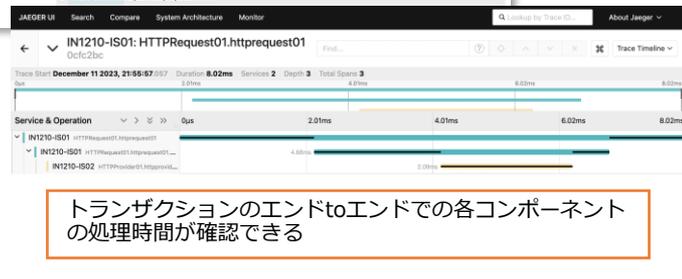
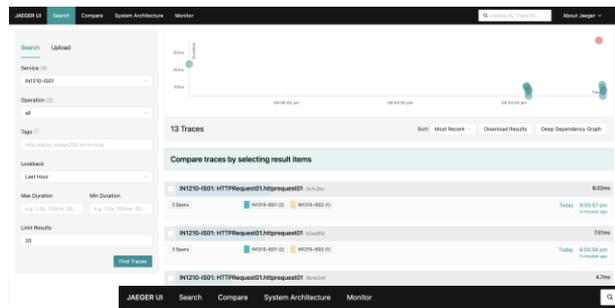


OpenTelemetryとは

<https://opentelemetry.io/>

CNCF Incubatingプロジェクトの1つ
オープンソースのオブザーバビリティ・フレームワーク
テレメトリ・データを収集してバックエンド・プラットフォームに転送するための方法を標準化

モニターツール

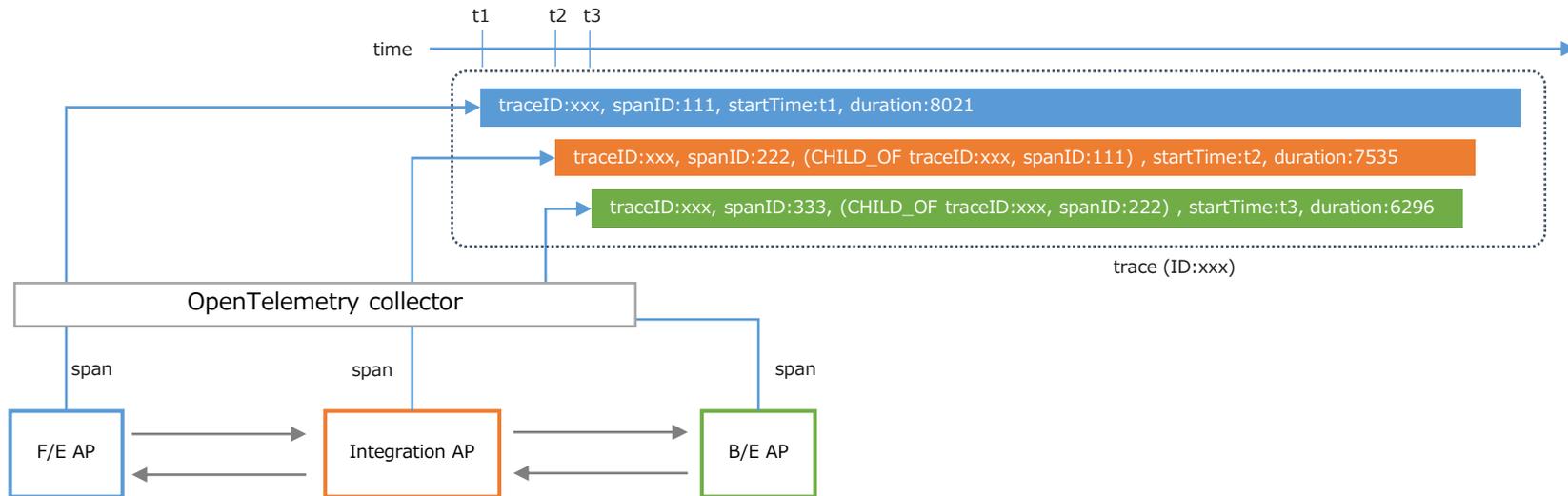


トランザクションのエンドtoエンドでの各コンポーネントの処理時間が確認できる

[参考] OpenTelemetryのコンセプト

- テレメトリ・データを収集してバックエンド・プラットフォームに転送するための方法を標準化し、ベンダー非依存のオブザーバビリティを実現
- OpenTelemetryでは3種類のテレメトリ・データを規定
 - ◆ トレース：トランザクションの一連の処理全体を指す。トランザクション内の個々の処理をスパンと呼ぶ。
 - ◆ メトリクス：システムやアプリケーションの動作に関する測定値（CPUやメモリの使用率、アプリケーションが受信したデータ・サイズなど）
 - ◆ ログ：システムやアプリケーションが出力するログ

OpenTelemetryにおけるトレースとスパンのイメージ



ACEにおけるOpenTelemetryの実装、制約事項

- OpenTelemetryのトレース/スパンの出力が可能
- 入力ノードの動き
 - ◆ Inputノードは処理開始とともにスパンを生成し、処理完了もしくはロールバックするまで実行
 - ◆ 入力データのヘッダーにトレース/スパンの情報(Identifier)が存在する場合は、そのスパンの子スパンとして生成
- 要求/出力ノードの動き
 - ◆ 新たにスパンを生成し、リクエストの送信/レスポンスの受信を実行
 - ◆ HTTPRequestノードの場合、リクエストのHTTPヘッダーのTraceparentにトレースIDとスパンIDをセットして送信
 - ◆ MQOutputノードの場合、MQRFH2ヘッダーのusrフォルダにtrace parent idをセット

設定方法

- 統合サーバーに対してOpenTelemetry traceの出力設定を行う
- 構成ファイル(server.conf.yaml)の以下のセクションで設定
 - ◆ ResourceManagersセクション/OpenTelemetryManagerサブセクション
 - ◆ 主なプロパティ
 - openTelemetryEnabledプロパティ : OpenTelemetryへの出力有無を指定(true | false)
 - exporterOtlpGrpcEndpointプロパティ : OpenTelemetry collectorのGRPCエンドポイントを指定
 - exporterOtlpHttpUrlプロパティ : OpenTelemetry collectorのHTTP URLを指定
※GRPCエンドポイントかHTTP URLのいずれか1つを指定

設定例

```
ResourceManagers:
  OpenTelemetryManager:
    openTelemetryEnabled: true # Enable or disable OpenTelemetry tracing. OpenTelemetry is only available on the Linux/x86-64 platform
    #openTelemetryServiceName: " # Override the Service Name attribute for telemetry spans. Defaults to integration server name.
    #openTelemetryHostName: " # Override the value of the Hostname attribute for telemetry spans.
    exporterOtlpGrpcEndpoint: 'localhost:4317' # Specify a GRPC endpoint to which OpenTelemetry span data is sent. For example, '<hostname>:4317'.
    #exporterOtlpHttpUrl: " # Specify a HTTP Url to which OpenTelemetry span data (json) is sent. For example, 'http://<hostname>:<port>/v1/traces'.
    #openTelemetryTruststoreType: " # The type of the open telemetry truststore [PKCS12, PEM or JKS]. Only used for the GRPC exporter.
    #openTelemetryTruststoreFile: " # The location of the open telemetry truststore. Only used for the GRPC exporter.
    #openTelemetryTruststorePass: " # Credential or resource alias containing the truststore password. Only used for the GRPC exporter.
    #openTelemetryTrustAlias: " # The alias of the trust certificate in the PKCS12 or JKS truststore.
    #openTelemetrySpanProcessor: 'batch' # Specify the OpenTelemetry span processor ['batch' or 'simple'].
    #openTelemetryBatchSpanOptions: " # Specify the OpenTelemetry batch span processor options. For example,
    # 'max_queue_size=4096:schedule_delay_millis=5000:max_export_batch_size=4096'
```

設定方法詳細は以下参照

<https://www.ibm.com/docs/en/app-connect/12.0?topic=management-configuring-opentelemetry-trace-integration-server>

設定方法

- メッセージフロー単位にOpenTelemetry traceの出力を無効にすることができる
 - ◆ 管理REST APIを利用して、対象のフローに対してopenTelemetryEnabledプロパティをfalseに設定する

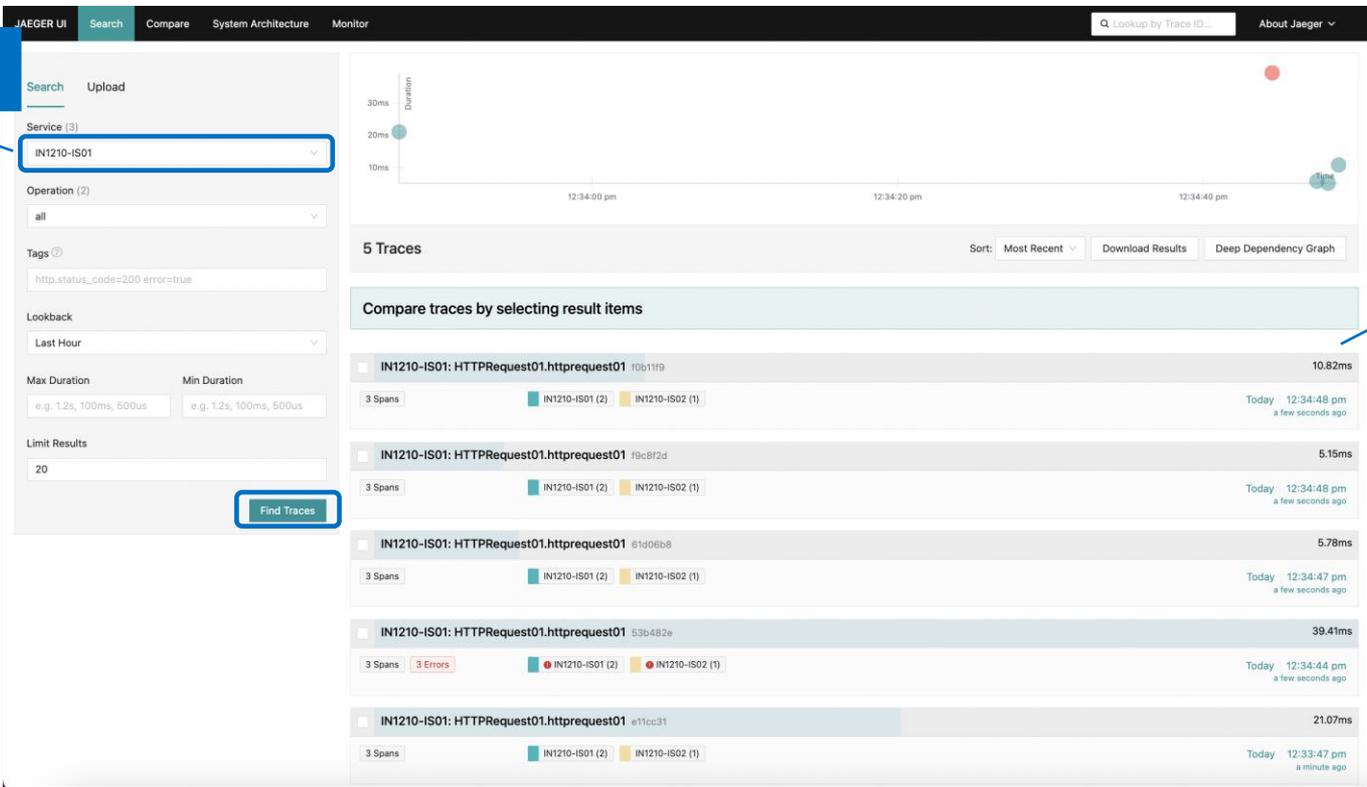
統合サーバーIS01のアプリケーションApp1のメッセージフローFlow1に対して無効にする場合の設定例

```
curl -X PATCH --header 'content-type: application/json' ¥  
--data "{\"properties¥\": { ¥\"openTelemetryEnabled¥\": ¥\"false¥\" } }" ¥  
localhost:4414/apiv2/servers/IS01/applications/App1/messageflows/Flow1
```

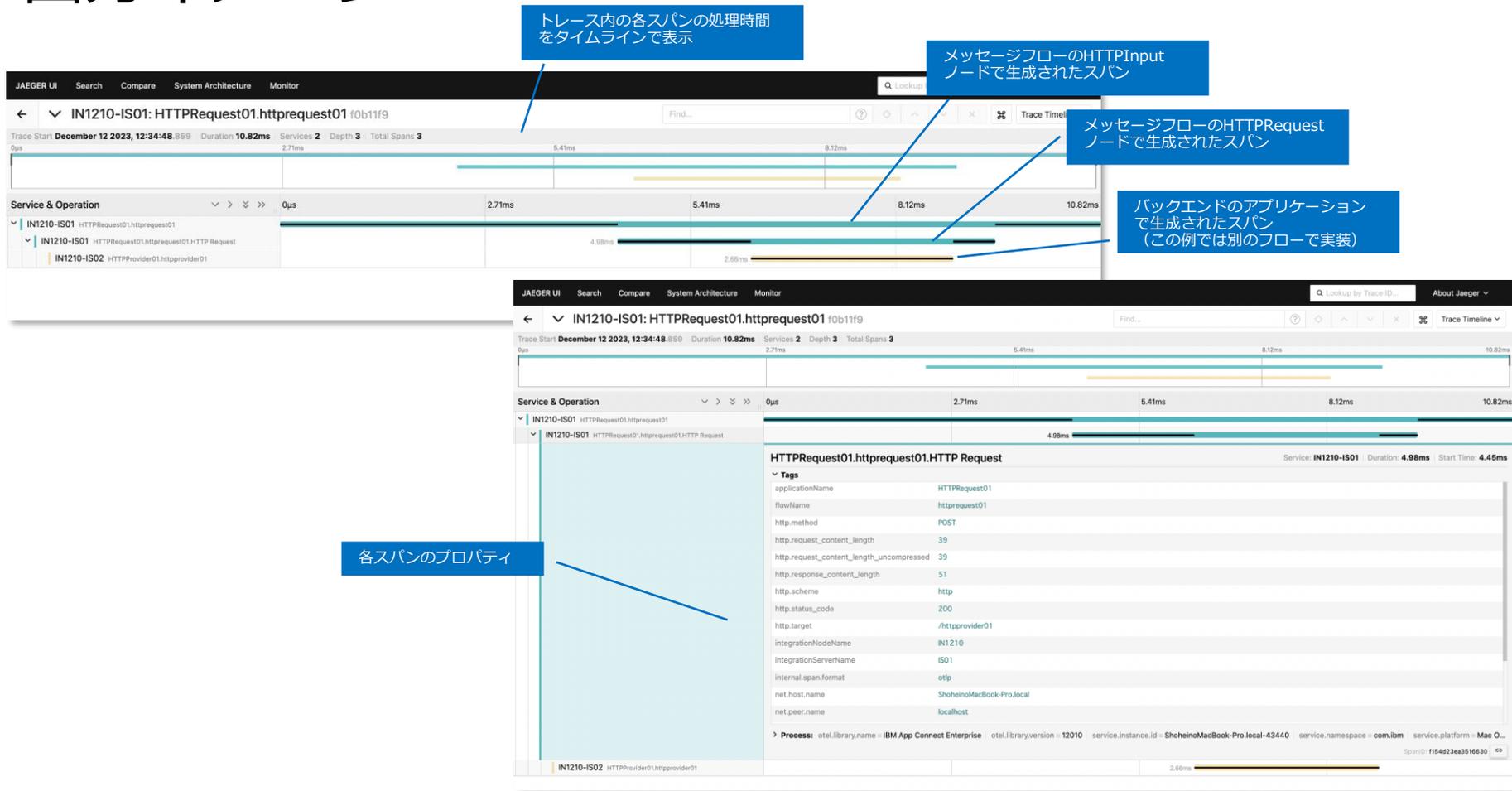
- 動的に反映される
- ◆ 有効に戻す場合は、inheritを設定する (trueではない)

出カイメージ

- OpenTelemetryに対応したモニターツールJaegerを利用した場合の出カイメージ



出カイメージ



ログ

機能概要

- ランタイムのコンポーネントが内部的なアクティビティ情報(BIPメッセージ)をログに出力
 - ◆ エラー時だけでなく正常時（起動、停止やデプロイ処理など）のアクティビティ情報も出力
 - ◆ タイムスタンプやコンポーネント名、スレッド番号、アクティビティを表すメッセージなど
 - ◆ TRACEノードやESQL LOGステートメントから任意のメッセージを出力することも可能
- 各アクティビティ情報にはBIPコードが割り当てられている
 - ◆ BIPコードは、頭文字「BIP」と4桁の番号、重大度を示す1文字で構成
 - ◆ BIPnnnnX
 - ・ 重大度を示すXは、I(Information)、W(Warning)、E(Error)、S(Severe)のいずれか
 - ◆ BIPコードをもとにログ監視機能によって障害検知が可能
- 監視や障害発生時の問題判別に利用
 - ◆ ログ監視機能と組み合わせてリアルタイム監視、障害時の通報等に利用可能

統合ノード起動時のログ

```
Sep 17 20:58:55 shoheim-vb ACE[22476]: IBM App Connect Enterprise v11009 (IN01) [Thread 22476] (Msg 1/1) BIP2001I: IBM App Connect Enterprise サービスは、バージョン 11009;、プロセス ID 22481 で開始されました。
Sep 17 20:58:59 shoheim-vb ACE[22481]: IBM App Connect Enterprise v11009 (IN01) [Thread 22481] (Msg 1/1) BIP2866I: IBM App Connect Enterprise 管理セキュリティは inactive です。
Sep 17 20:58:59 shoheim-vb ACE[22481]: IBM App Connect Enterprise v11009 (IN01) [Thread 22489] (Msg 1/1) BIP3132I: HTTP リスナーがポート '4414' で 'RestAdmin http' 接続の listen を開始しました。
Sep 17 20:59:02 shoheim-vb ACE[22500]: IBM App Connect Enterprise v11009 (IN01.ex1) [Thread 22500] (Msg 1/1) BIP2208I: 統合サーバー (64) が開始されました: プロセス '22500'、スレッド '22500'、追加情報: integrationNodeName 'IN01' (動作モード 'advanced')、integrationServerUUID '00000000-0000-0000-0000-000000000000'、integrationServerLabel 'ex1'、 queueManagerName 'IN01QM'、trusted 'false'、userId 'shoheim'、migrationNeeded 'false'、 integrationNodeUUID 'c2100d84-f4a1-11ea-9724-7f0001010000'、filePath '/opt/IBM/ace-11.0.0.9/server'、workPath '/var/mqsi'、ICU Converter Path ''、 ordinality '1'。
Sep 17 20:59:22 shoheim-vb ACE[22500]: IBM App Connect Enterprise v11009 (IN01.ex1) [Thread 22500] (Msg 1/1) BIP2152I: 構成メッセージが受信されました。
Sep 17 20:59:22 shoheim-vb ACE[22500]: IBM App Connect Enterprise v11009 (IN01.ex1) [Thread 22500] (Msg 1/1) BIP2153I: 統合サーバーをまもなく 'Start' します。
Sep 17 20:59:22 shoheim-vb ACE[22500]: IBM App Connect Enterprise v11009 (IN01.ex1) [Thread 22500] (Msg 1/1) BIP2155I: タイプ 'Application' のデプロイ済みリソース 'HTTP01' をまもなく 'Initialize' します。
(以下、省略)
```

BIPメッセージの出力先

■ 統合ノード構成の場合

- ◆ ローカルのシステム・ログ
 - Linux/UNIX : syslog
 - Windows : Windows event log

■ 独立統合サーバー構成の場合

- ◆ 標準出力
 - BIPコードやスレッド番号などは出力されない
- ◆ ログ・ファイル
 - `<workdir>/log/integration_server.<統合サーバー名>.events.txt`
 - 起動のたびに新しいログ・ファイルを生成
 - ✓ 古いログ・ファイルはサフィックス(1~9)をつけて保存される (`integration_server.<統合サーバー名>.events.txt.n`)

■ Elastic Stackに出力することも可

- ◆ 統合サーバー（統合ノード構成の統合サーバー、および独立統合サーバー）のログのみ
 - 統合ノード自体のログは出力されない
- ◆ 複数環境のログを一元的に管理することが可能

補足

Elastic Stackとは、Elastic社が提供する、ログの収集、分析、可視化を実現する製品群の総称。分析・検索エンジンであるElasticsearch、データ収集パイプラインであるLogstash、データの可視化を行うKibanaの頭文字を取って以前はELK Stackと呼んでいたが、新たにデータ送信専用のBeatsが加わり、現在はElastic Stackを呼ぶ。
詳細はElastic社のサイト参照。

<https://www.elastic.co/jp/what-is/elk-stack>

syslogへの出力設定

- Linux/UNIX環境の統合ノード構成はBIPメッセージをsyslogサブシステムに送信
 - ◆ BIPメッセージの属性
 - facility : user
 - priority : err、warn、info (内容に応じていずれかがセットされる)
- syslogデーモンの構成ファイルに出力先を指定
 - ◆ 具体的な設定は使用されるOSのガイドを参照
 - ◆ 出力先ファイルのパーミッションは-rw-rw-rw(666)に設定

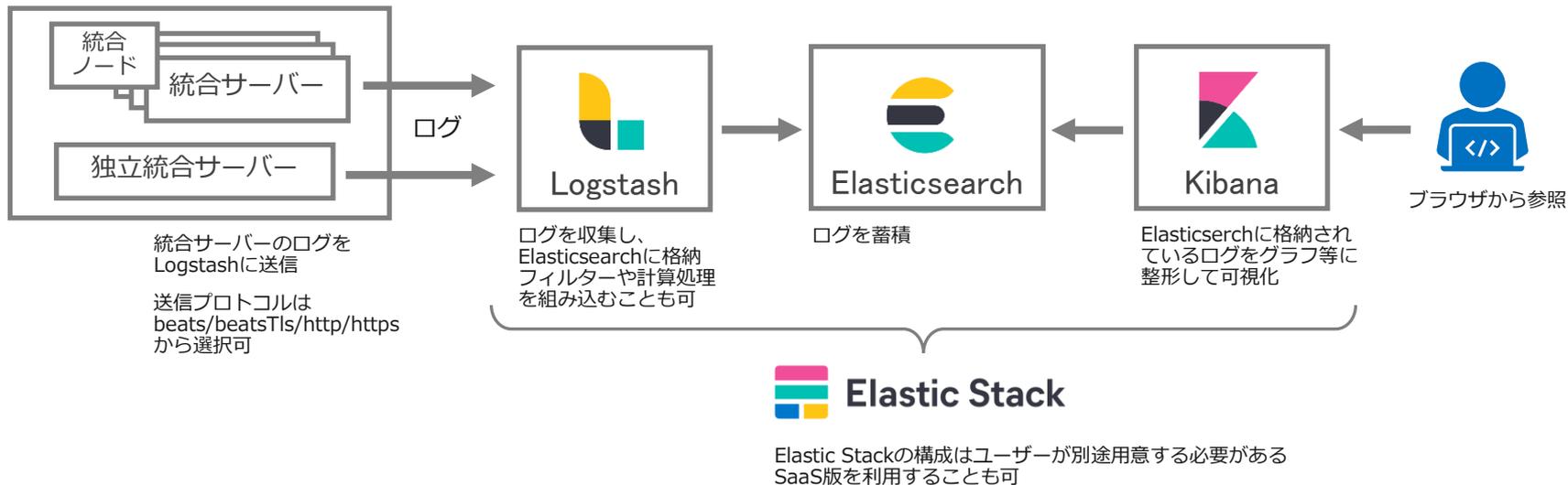
syslog.conf 設定例

```
user.info /var/log/user.log
```

この設定でuserファシリティのinfo以上(err,warn含む)のメッセージが指定したファイルに出力される

Elastic Stackへの出力

■ 構成イメージ



Elastic Stackへの出力設定

■ 構成ファイル(.yaml)の以下のセクションに設定

- ◆ 統合ノード構成の場合、統合ノードのログはElastic Stackへは出力されないが、設定はnode.conf.yamlに実施可
- ◆ ELKConnectionsセクション
 - 構成ファイルの例を参考に、任意の名前でサブセクションを追加し、必要なプロパティを設定

設定例

```
ELKConnections:
# Description for ELK Connections.
# elkConnection1:
# elkProtocol: 'beats' # Logstash input protocol. Valid values are: 'beats', 'beatsTls', 'http', or 'https'.
# hostname: 'myhost.domain.com' # Hostname for the elkProtocol endpoint.
# port: 0 # Port for the elkProtocol endpoint.
# uploadIntervalMillisecs: 60000 # Interval between uploading cached data, set in milliseconds.
# elkCredential: "" # Set an 'elk' credential alias name to enable basic authentication, if it is required by the Logstash input protocol.
# keystoreFile: '/path/to/keystore.jks' # Set the path to the keystore to be used, if it is required by the Logstash input protocol.
# keystorePass: 'P4s5w0rd' # Set the password, or 'keystore' credential alias to the password, of the keystore.
# keyAlias: "" # Set the alias name of the private key, if mutual authentication is required by the Logstash input protocol.
# keyPass: "" # Set the password, or 'keystorekey' credential alias to the password, for accessing the private mutual authentication key.
# truststoreFile: '/path/tp/truststore.jks' # Set the path to the truststore to be used, if it is required by the Logstash input protocol.
# truststorePass: 'P4s5w0rd' # Set the password, or 'truststore' credential alias to the password, for accessing the truststore.

elkConnection1:
elkProtocol: 'beats'
hostname: 'logstashsvr'
port: 5044
```

Logstash側の構成に合わせてプロトコルやポートを指定

設定方法詳細は以下参照

<https://www.ibm.com/docs/en/app-connect/12.0?topic=pmwm-configuring-integration-servers-send-logs-events-logstash-in-elk-stack>

Elastic Stackへの出力設定

- ◆ Logセクション
 - elkLogプロパティをtrueにセットし、ELKConnectionsセクションに定義したサブセクション名を指定

設定例

```
Log:
#elkLog: false # Control the publication of BIP messages to an ELK (Elasticsearch, Logstash, Kibana) stack. Set to true or false, default is false.
#elkConnections: '' # Name of the ELK connection to use, for example 'elkConnection1'
# Each named ELK Connection must be defined in the ELKConnections section below.

elkLog: true
elkConnections: 'elkConnection1'
```

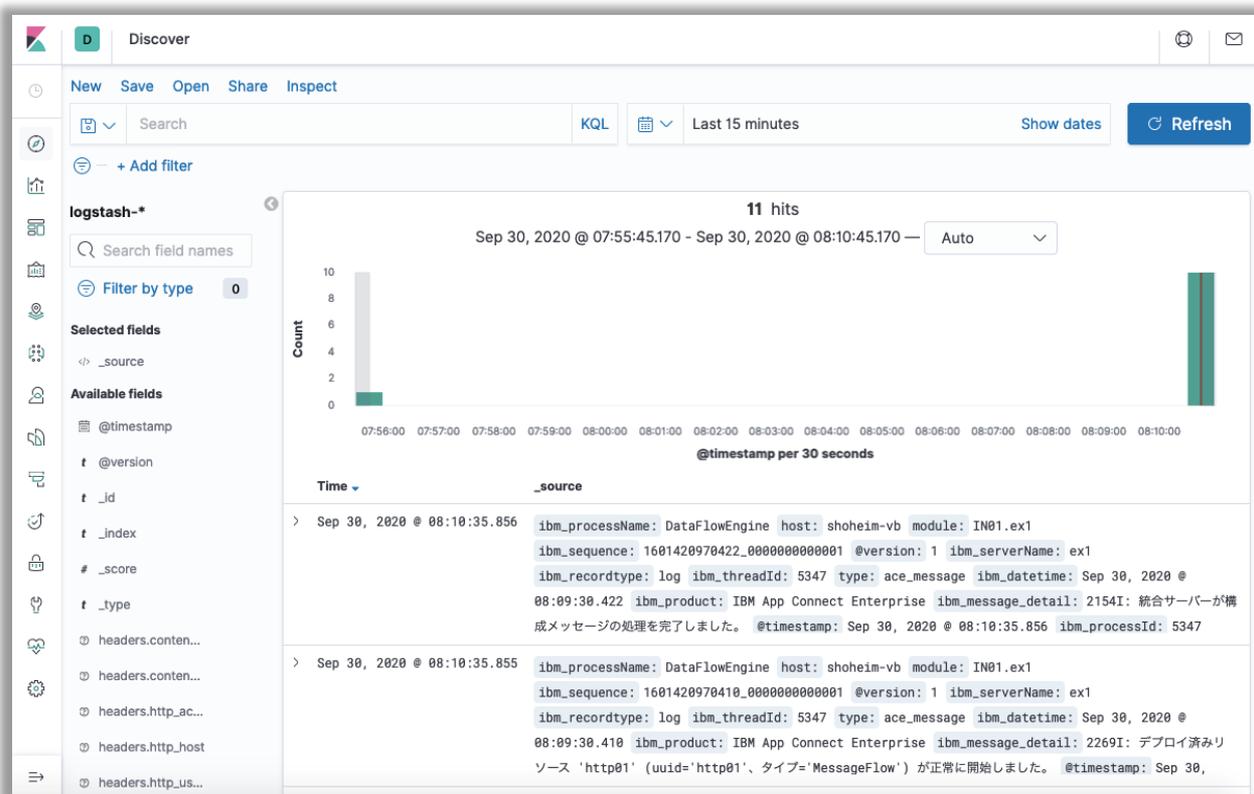
■ 設定後、ランタイムを起動

- ◆ Elastic Stackへの出力が成功すると、以下のメッセージがローカルのログに出力される

```
BIP6503I: (IN01.ex1) The integration server successfully sent data to ELK connection "elkConnection1" using elkProtocol "beats", hostname "logstashsvr" and port "5044".
```

Kibanaによる可視化

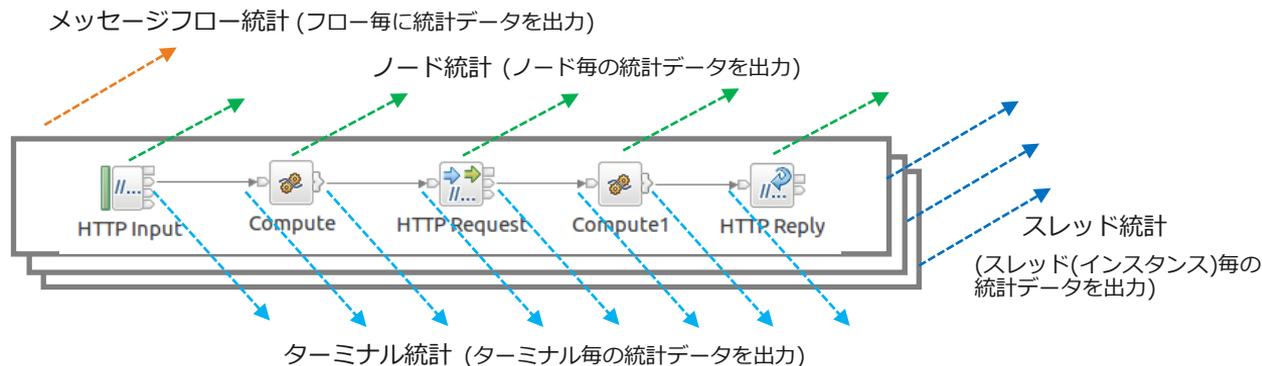
- Kibanaのダッシュボードにアクセスし、ログを確認



メッセージフロー統計

機能概要

- メッセージフロー処理の各種指標を統計レコードとして出力する機能
 - ◆ メッセージフローが単位時間あたりに処理したCPU時間、経過時間、待ち時間、処理件数、メッセージサイズなど
 - ◆ スレッド/ノード/ターミナル単位の統計データを出力することも可能
- スナップショット統計とアーカイブ統計
 - ◆ 統計レコードを出力するインターバルが異なる2種類の統計があり、それぞれに出力有無を設定（両方出力も可）
 - ・ スナップショット統計 : 20秒インターバル（変更不可）
 - ・ アーカイブ統計 : デフォルトは60分インターバル（変更可）
 - ◆ 出力される指標は同じ
 - ◆ スナップショット統計のメッセージフロー統計とノード統計はデフォルトでON
- 出力先とフォーマットは以下から選択可能
 - ◆ MQトピック(XML/JSON)
 - ◆ MQTTトピック(XML/JSON)
 - ◆ ファイル(CSV)
 - ◆ ユーザー・トレース
- Web UI/REST APIにて表示、ON/OFF設定可



統計指標

■ メッセージフロー統計

指標
メッセージフロー名
統合サーバー名
データ収集の開始、終了時刻
統計タイプ (スナップショット、アーカイブ)
メッセージ処理に費やしたCPU時間、経過時間
メッセージの入力待ちに費やしたCPU時間、経過時間
処理したメッセージの件数
処理したメッセージの最小サイズ、最大サイズ、平均サイズ
アサインされたスレッド数
コミット、バックアウトしたメッセージ数
統計起点

■ ターミナル統計

指標
ターミナル名
ターミナルタイプ
通過したメッセージの件数

■ スレッド統計

指標
スレッド番号
メッセージ処理に費やしたCPU時間、経過時間
メッセージの入力待ちに費やしたCPU時間、経過時間
処理したメッセージの件数
処理したメッセージの最小サイズ、最大サイズ、平均サイズ

■ ノード統計

指標
ノード名
ノードタイプ
メッセージ処理に費やしたCPU時間、経過時間
メッセージの入力待ちに費やしたCPU時間、経過時間
ノードが実行された回数
処理したメッセージの件数
処理したメッセージの最小サイズ、最大サイズ、平均サイズ

統計データの出力先

■ 統計データの出力先は以下から選択可能（複数指定可）

◆ MQ(XML/JSON)

- デフォルト・キューマネージャー（統合ノード、独立統合サーバーに関連付けられたキューマネージャー）
- 外部のキューマネージャー（MQEndpointポリシーで指定したキューマネージャー）

◆ MQTT(XML/JSON)

- 組み込みMQTTブローカー（統合ノード構成に標準で組み込まれているMQTTブローカー。独立統合サーバーでは利用不可）
- 外部のMQTTブローカー（MQTTPublishポリシーで指定したMQTTブローカー）

◆ ファイル(CSV)

- 統合ノード構成

ディレクトリ	/var/mqsi/common/stats
ファイル名	<統合ノード>_<統合サーバー>__flowStats.csv（フロー統計） <統合ノード>_<統合サーバー>__nodeStats.csv（ノード統計） <統合ノード>_<統合サーバー>__threadStats.csv（スレッド統計）

- 独立統合サーバー構成

ディレクトリ	/<ワークディレクトリ>/config/common/stats
ファイル名	integration_server_<統合サーバー>__flowStats.csv（フロー統計） integration_server_<統合サーバー>__nodeStats.csv（ノード統計） integration_server_<統合サーバー>__threadStats.csv（スレッド統計）

◆ ユーザー・トレース

- 以下のBIPメッセージで出力
 - ✓ BIP2380I : メッセージフロー統計
 - ✓ BIP2381I : スレッド統計
 - ✓ BIP2382I : ノード統計
 - ✓ BIP2383I : ターミナル統計

トピック

■ MQおよびMQTTの出力先トピック

◆ MQトピック

- JSON

```
$SYS/Broker/integrationNodeName/Statistics/JSON/statsType/integrationServerName/applications/applicationName/messageflows/messageflowName
```

- XML

```
$SYS/Broker/integrationNodeName/StatisticsAccounting/statsType/integrationServerName/messageflowName
```

◆ MQTTトピック

- JSON

```
IBM/IntegrationBus/integrationNodeName/Statistics/JSON/statsType/integrationServerName/applications/applicationName/messageflows/messageflowName
```

- XML

```
IBM/IntegrationBus/integrationNodeName/StatisticsAccounting/statsType/integrationServerName/messageflowName
```

※*integrationNodeName* : 独立統合サーバー構成の場合は、“integration_server”(固定値)となる
statsType : スナップショット統合の場合 “Snapshot”、アーカイブ統計の場合 “Archive”
XML形式の場合、アプリケーション名はトピックに入らない

構成概要

■ キューマネージャーの準備

- ◆ 統計データをMQトピックに出力する場合はキューマネージャーをローカル、もしくはリモートに用意
 - Pub/Subエンジンを有効にしておく（キューマネージャーのPSMODE属性をCOMPAT、もしくはENABLEDに設定）
- ◆ ランタイムとの関連付けを行う、もしくはMQEndpointポリシーを指定
 - MQEndpointポリシーで指定する場合はポリシー作成後、統合サーバーにデプロイ

■ MQTTブローカーの準備

- ◆ 統計データをMQTTトピックに出力する場合はMQTTブローカーをローカル、もしくはリモートに用意
 - 統合ノード構成の場合はデフォルトで有効となっている組み込みMQTTブローカーを利用可能
 - ✓ 独立統合サーバーの場合は独自にMQTTブローカーを用意する必要あり
 - 組み込みMQTTブローカー以外を利用する場合は、MQTTPublishポリシーを作成し、統合サーバーにデプロイ

■ 出力先設定

- ◆ MQトピック/MQTTトピックに出力する場合は、構成ファイルのEventsセクション/OperationalEventsサブセクションで設定

■ メッセージフロー統計のON/OFFを統合ノード、もしくは統合サーバーに設定

- ◆ 以下のいずれかの方法で設定
 - 構成ファイル（Statisticsセクション/SnapShot|Archiveサブセクション）（起動/再起動で反映）
 - Web UI（動的反映、統合サーバー起動時のみ設定可）
 - REST API（動的反映、統合サーバー起動時のみ設定可）
 - mqsischangeresourcesstats（動的反映、統合サーバー起動時のみ設定可）
- ◆ アプリケーション/フロー単位のON/OFF設定はできない

出力先設定

- MQトピック/MQTTトピックに出力する場合は構成ファイルの以下のセクションで設定
 - ◆ Eventsセクション/OperationalEventsサブセクション
 - 出力先として使用するMQもしくはMQTTのenabledプロパティをtrueに設定
 - ✓ ランタイムの構成によっては、デフォルトで出力設定がONとなっている場合もあり
 - 関連付けたキューマネージャーもしくは組み込みMQTTブローカーへ出力する以外は、事前に作成/デプロイしたMQEndpointポリシー、またはMQTTPublishポリシーをpolicyプロパティに設定する

ランタイムに関連付けたMQ（デフォルト・キューマネージャー）へ出力する場合の設定例

```
Events:
OperationalEvents: # Message flow and Resource statistics plus Workload management
MQ:
  #policy: " # Specify a {policy project}:policy if not using 'defaultQueueManager'
  enabled: true # Set true or false, default false
  #format: " # Set string or none
  #publishRetryInterval: 0 #Set the retry interval (in milliseconds), to pause all publications and retry,
    when publication failures are causing serious delay to the transaction.
MQTT:
  #policy: " # Specify a {policy project}:policy if not using 'default MQTTServer'
  enabled: true # Set true or false, default false
  #format: " # Set string or none
  #publishRetryInterval: 0 #Set the retry interval (in milliseconds), to pause all publications and retry,
    when publication failures are causing serious delay to the transaction.
```

設定方法詳細は以下参照

<https://www.ibm.com/docs/en/app-connect/12.0?topic=events-configuring-publication-event-messages>

メッセージフロー統計の設定

■ 構成ファイル(.yaml)の以下のセクションで設定

◆ Statisticsセクション/Snapshot|Archiveサブセクション

- publicationOnプロパティでメッセージフロー統計の出力有無を設定(active | inactive)
- nodeDataLevelプロパティでノード統計とターミナル統計の出力有無を設定
 - ✓ none : 出力なし、basic : ノード統計を出力、advanced : ノード統計とターミナル統計を出力
- threadDataLevelプロパティでスレッド統計の出力有無を設定(none | basic)
- outputFormatプロパティで出力先およびフォーマットを設定
 - ✓ xml/jsonを指定した場合はMQおよびMQTTに出力、csvの場合はファイルに出力、usertraceの場合はユーザートレースに出力
 - ✓ カンマ区切りで複数指定可能

スナップショット統計のメッセージフロー統計とノード統計をJSON形式でMQおよびMQTTに出力する設定例

```
Statistics:
  Snapshot:
    publicationOn: 'active' # choose 1 of : active|inactive, explicitly set 'active'. If unset will default to 'inactive'
    #accountingOrigin: 'none' # choose 1 of : none|basic, default none
    nodeDataLevel: 'basic' # choose 1 of : none|basic|advanced, explicitly set 'basic'. If unset will default to 'none'
    outputFormat: 'json' # choose comma separated list of one or more of : csv,json,xml,usertrace. Explicitly set to 'json' for WebUI. If unset will default to ''
    #threadDataLevel: 'none' # choose 1 of : none|basic. If unset will default to 'none'
  Archive:
    #archivalOn: 'inactive' # choose 1 of : active|inactive, default inactive
    # Also set Events.OperationalEvents.MQ|MQTT for outputFormat xml to be published to MQ/MQTT
    #accountingOrigin: 'none' # choose 1 of : none|basic
    #majorInterval: 60 # Sets the interval in minutes at which archive statistics are published
    #nodeDataLevel: 'none' # choose 1 of : none|basic|advanced
    #outputFormat: 'usertrace' # comma separated list of : csv,xml,usertrace
    #threadDataLevel: 'none' # choose 1 of : none|basic
```

設定方法詳細は以下参照

<https://www.ibm.com/docs/en/app-connect/12.0?topic=ccmfasd-configuring-collection-message-flow-statistics-by-using-yaml-configuration-file>

Web UIによる表示

メッセージフロー統計とノード統計（一部）を表示可能

ノード: IN01 / サーバー: ex1 / アプリケーション: HTTP01 / メッセージ・フロー: http01

http01

プロパティ フロー統計 アクティビティ・ログ

メッセージ速度 (メッセージ数/秒)

最新 0 平均 1 最高 2.36 最低 0

合計入力メッセージ

最新 0 平均 11 最高 46 最低 0

最大経過時間 (ミリ秒)

最新 0 平均 3874 最高 28833 最低 0

表示する指標を選択できる

- MQ エラーの数
- エラーを含んだメッセージの数
- コミットの数
- スレッドの最大数に到達した回数
- バックアウトの数

バックアウトの数

ノード統計の一部を表示

```
graph LR; A[HTTP Input] --> B[Compute]; B --> C[HTTP Reply]
```

フロー・ノード名	平均経過時間 (ms)	平均 CPU 時間 (ms)	合計入力メッセージ
Compute	605	581	107
HTTP Input	230	172	107
HTTP Reply	574	241	107

出力例(JSON)

メッセージフロー統計、スレッド統計、ノード統計、ターミナル統計の全てを
スナップショット統計でJSON形式で出力した場合
(1フロー分の全ての統計指標が1つのMQ/MQTTメッセージとして出力される)

メッセージフロー統計

```
{
  "WMQIStatisticsAccounting": {
    "RecordType": "SnapShot",
    "RecordCode": "SnapShot",
    "MessageFlow": {
      "BrokerLabel": "IN01",
      "BrokerUUID": "5eb081f6-ac4e-11ea-8860-7f0001010000",
      "ExecutionGroupName": "ex1",
      "ExecutionGroupUUID": "00000000-0000-0000-0000-000000000000",
      "MessageFlowName": "http01",
      "MessageFlowURI": "/applications/HTTP01/messageflows/http01",
      "ApplicationName": "HTTP01",
      "StartDate": "2021-02-21",
      "StartTime": "15:15:37.182469",
      "GMTStartTime": "2021-02-21T06:15:37.182+00:00",
      "EndDate": "2021-02-21",
      "EndTime": "15:15:56.624939",
      "GMTEndTime": "2021-02-21T06:15:56.624+00:00",
      "TotalElapsedTime": 28969,
      "MaximumElapsedTime": 4562,
      "MinimumElapsedTime": 478,
      "TotalCPUTime": 27027,
      "MaximumCPUTime": 3957,
      "MinimumCPUTime": 476,
      "CPUTimeWaitingForInputMessage": 18974,
      "ElapsedTimeWaitingForInputMessage": 19397958,
      "TotalInputMessages": 33,
      "TotalSizeOfInputMessages": 13464,
      "MaximumSizeOfInputMessages": 408,
      "MinimumSizeOfInputMessages": 408,
      "NumberOfThreadsInPool": 1,
      "TimesMaximumNumberOfThreadsReached": 33,
      "TotalNumberOfMQErrors": 0,
      "TotalNumberOfMessagesWithErrors": 0,
      "TotalNumberOfErrorsProcessingMessages": 0,
      "TotalNumberOfTimeOutsWaitingForRepliesToAggregateMessages": 0,
      "TotalNumberOfCommits": 0,
      "TotalNumberOfBackouts": 0,
      "AccountingOrigin": "Anonymous"
    }
  },
  "NumberOfThreads": 1,
  "ThreadStatistics": [
    {
      "Number": 26076,
      "TotalNumberOfInputMessages": 33,
      "TotalElapsedTime": 28969,
      "TotalCPUTime": 27027,
      "CPUTimeWaitingForInputMessage": 18974,
      "ElapsedTimeWaitingForInputMessage": 19397958,
      "TotalSizeOfInputMessages": 13464,
      "MaximumSizeOfInputMessages": 408,
      "MinimumSizeOfInputMessages": 408
    }
  ]
}
```

メッセージフロー統計

```
"NumberOfNodes": 3,
"Nodes": [
  {
    "Label": "Compute",
    "Type": "ComputeNode",
    "TotalElapsedTime": 18349,
    "MaximumElapsedTime": 2914,
    "MinimumElapsedTime": 308,
    "TotalCPUTime": 17879,
    "MaximumCPUTime": 2643,
    "MinimumCPUTime": 290,
    "CountOfInvocations": 33,
    "NumberOfInputTerminals": 1,
    "NumberOfOutputTerminals": 6,
    "TerminalStatistics": [
      {
        "Label": "failure",
        "Type": "Output",
        "CountOfInvocations": 0
      },
      {
        "Label": "in",
        "Type": "Input",
        "CountOfInvocations": 33
      },
      {
        "Label": "out",
        "Type": "Output",
        "CountOfInvocations": 33
      },
      {
        "Label": "out1",
        "Type": "Output",
        "CountOfInvocations": 0
      },
      {
        "Label": "out2",
        "Type": "Output",
        "CountOfInvocations": 0
      },
      {
        "Label": "out3",
        "Type": "Output",
        "CountOfInvocations": 0
      },
      {
        "Label": "out4",
        "Type": "Output",
        "CountOfInvocations": 0
      }
    ]
  }
]
```

ノード統計

```
{
  "Label": "HTTP Input",
  "Type": "WSInputNode",
  "TotalElapsedTime": 2710,
  "MaximumElapsedTime": 682,
  "MinimumElapsedTime": 43,
  "TotalCPUTime": 2697,
  "MaximumCPUTime": 679,
  "MinimumCPUTime": 43,
  "CountOfInvocations": 33,
  "NumberOfInputTerminals": 0,
  "NumberOfOutputTerminals": 4,
  "TerminalStatistics": [
    {
      "Label": "catch",
      "Type": "Output",
      "CountOfInvocations": 0
    },
    {
      "Label": "failure",
      "Type": "Output",
      "CountOfInvocations": 0
    },
    {
      "Label": "out",
      "Type": "Output",
      "CountOfInvocations": 33
    },
    {
      "Label": "timeout",
      "Type": "Output",
      "CountOfInvocations": 0
    }
  ]
}
```

ターミナル統計

```
{
  "Label": "HTTP Reply",
  "Type": "WSReplyNode",
  "TotalElapsedTime": 7910,
  "MaximumElapsedTime": 966,
  "MinimumElapsedTime": 124,
  "TotalCPUTime": 6451,
  "MaximumCPUTime": 635,
  "MinimumCPUTime": 124,
  "CountOfInvocations": 33,
  "NumberOfInputTerminals": 1,
  "NumberOfOutputTerminals": 2,
  "TerminalStatistics": [
    {
      "Label": "failure",
      "Type": "Output",
      "CountOfInvocations": 0
    },
    {
      "Label": "in",
      "Type": "Input",
      "CountOfInvocations": 33
    },
    {
      "Label": "out",
      "Type": "Output",
      "CountOfInvocations": 0
    }
  ]
}
```

出力例(XML)

メッセージフロー統計、スレッド統計、ノード統計、ターミナル統計の全てを
スナップショット統計でXML形式で出力した場合
(1フロー分の全ての統計指標が1つのMQ/MQTTメッセージとして出力される)

メッセージフロー統計

```
<WMQIStatisticsAccounting RecordType="SnapShot" RecordCode="SnapShot">
  <MessageFlow
    BrokerLabel="IN01"
    BrokerUUID="5eb081f6-ac4e-11ea-8860-7f0001010000"
    ExecutionGroupName="ex1"
    ExecutionGroupUUID="00000000-0000-0000-0000-000000000000"
    MessageFlowName="http01"
    ApplicationName="HTTP01"
    StartDate="2021-02-21"
    StartTime="17:34:51.965018"
    GMTStartTime="2021-02-21T08:34:51.965+00:00"
    EndDate="2021-02-21"
    EndTime="17:35:09.022215"
    GMTEndTime="2021-02-21T08:35:09.022+00:00"
    TotalElapsedTime="5626"
    MaximumElapsedTime="2927"
    MinimumElapsedTime="509"
    TotalCPUTime="4848"
    MaximumCPUTime="2333"
    MinimumCPUTime="508"
    CPUTimeWaitingForInputMessage="3219"
    ElapsedTimeWaitingForInputMessage="17051001"
    TotalInputMessages="4"
    TotalSizeOfInputMessages="1632"
    MaximumSizeOfInputMessages="408"
    MinimumSizeOfInputMessages="408"
    NumberOfThreadsInPool="1"
    TimesMaximumNumberOfThreadsReached="4"
    TotalNumberOfMQErrors="0"
    TotalNumberOfMessagesWithErrors="0"
    TotalNumberOfErrorsProcessingMessages="0"
    TotalNumberOfTimeOutsWaitingForRepliesToAggregateMessages="0"
    TotalNumberOfCommits="0"
    TotalNumberOfBackouts="0"
    AccountingOrigin="Anonymous" />
  <Threads Number="1">
    <ThreadStatistics Number="26076"
      TotalNumberOfInputMessages="4"
      TotalElapsedTime="5626"
      TotalCPUTime="4848"
      CPUTimeWaitingForInputMessage="3219"
      ElapsedTimeWaitingForInputMessage="17051001"
      TotalSizeOfInputMessages="1632"
      MaximumSizeOfInputMessages="408"
      MinimumSizeOfInputMessages="408" />
  </Threads>
```

メッセージフロー
統計

スレッド統計

```
<Nodes Number="3">
  <NodeStatistics Label="Compute" Type="ComputeNode"
    TotalElapsedTime="2913" MaximumElapsedTime="1188"
    MinimumElapsedTime="331" TotalCPUTime="2891"
    MaximumCPUTime="1170" MinimumCPUTime="331"
    CountOfInvocations="4" NumberOfInputTerminals="1"
    NumberOfOutputTerminals="6">
    <TerminalStatistics Label="failure" Type="Output" CountOfInvocations="0" />
    <TerminalStatistics Label="in" Type="Input" CountOfInvocations="4" />
    <TerminalStatistics Label="out" Type="Output" CountOfInvocations="4" />
    <TerminalStatistics Label="out1" Type="Output" CountOfInvocations="0" />
    <TerminalStatistics Label="out2" Type="Output" CountOfInvocations="0" />
    <TerminalStatistics Label="out3" Type="Output" CountOfInvocations="0" />
    <TerminalStatistics Label="out4" Type="Output" CountOfInvocations="0" />
  </NodeStatistics>
  <NodeStatistics Label="HTTP Input" Type="WSInputNode"
    TotalElapsedTime="1045" MaximumElapsedTime="867"
    MinimumElapsedTime="48" TotalCPUTime="895"
    MaximumCPUTime="719" MinimumCPUTime="47"
    CountOfInvocations="4" NumberOfInputTerminals="0"
    NumberOfOutputTerminals="4">
    <TerminalStatistics Label="catch" Type="Output" CountOfInvocations="0" />
    <TerminalStatistics Label="failure" Type="Output" CountOfInvocations="0" />
    <TerminalStatistics Label="out" Type="Output" CountOfInvocations="4" />
    <TerminalStatistics Label="timeout" Type="Output" CountOfInvocations="0" />
  </NodeStatistics>
  <NodeStatistics Label="HTTP Reply" Type="WSReplyNode"
    TotalElapsedTime="1668" MaximumElapsedTime="1010"
    MinimumElapsedTime="130" TotalCPUTime="1062"
    MaximumCPUTime="567" MinimumCPUTime="130"
    CountOfInvocations="4" NumberOfInputTerminals="1"
    NumberOfOutputTerminals="2">
    <TerminalStatistics Label="failure" Type="Output" CountOfInvocations="0" />
    <TerminalStatistics Label="in" Type="Input" CountOfInvocations="4" />
    <TerminalStatistics Label="out" Type="Output" CountOfInvocations="0" />
  </NodeStatistics>
</Nodes>
</WMQIStatisticsAccounting>
```

ノード統計

ターミナル統計

出力例(CSV)

スレッド統計 (IN01_ex1__threadStats.csv) ※ファイルの1行目はヘッダー

```
"Record Type","Record Code","Broker Name","EG Name","Message Flow Name","Application Name","Library Name","Record Start Date","Record Start Time","Record GMT Start  
Timestamp","Record End Date","Record End Time","Record GMT End Timestamp","Thread Number (ID)","Total Number of Input Messages","Total Elapsed Time","Total CPU  
Time","CPU Time Waiting for Input Messages","Elapsed Time Waiting for Input Messages","Total Size of Input Messages","Maximum Size of Input Messages","Minimum Size of  
Input Messages"  
"SnapShot","SnapShot","IN01","ex1","http01","HTTP01","","2021-02-21","21:45:19.852747","2021-02-21 12:45:19.852747","2021-02-21","21:45:46.290653","2021-02-21  
12:45:46.290653","28220","38","42990","36173","23852","26391894","15504","408","408"  
"SnapShot","SnapShot","IN01","ex1","http01","HTTP01","","2021-02-21","21:45:46.290718","2021-02-21 12:45:46.290718","2021-02-21","21:46:06.292006","2021-02-21  
12:46:06.292006","28220","0","0","0","429","20000694","0","0","0"  
"SnapShot","SnapShot","IN01","ex1","http01","HTTP01","","2021-02-21","21:46:06.292114","2021-02-21 12:46:06.292114","2021-02-21","21:46:26.294804","2021-02-21  
12:46:26.294804","28220","0","0","0","374","20001668","0","0","0"  
"SnapShot","SnapShot","IN01","ex1","http01","HTTP01","","2021-02-21","21:46:26.294901","2021-02-21 12:46:26.294901","2021-02-21","21:46:45.264000","2021-02-21  
12:46:45.264000","28220","27","39979","32647","18540","18921904","11016","408","408"
```

出力例(UserTrace)

BIP2380I : メッセージフロー統計

BIP2381I : スレッド統計

BIP2382I : ノード統計

BIP2383I : ターミナル統計

```
2021-02-21 23:14:18.263324 28646 UserTrace BIP2380I: ACE message flow statistics. ProcessID='28385', Key='0', Type='Snapshot', Reason='Snapshot', integrationNodeLabel='IN01', integrationNodeUUID='5eb081f6-ac4e-11ea-8860-7f0001010000', integrationServerName='ex1', integrationServerUUID='00000000-0000-0000-0000-000000000000', MessageFlowName='http01', StartDate='2021-02-21', StartTime='23:13:56.566137', GMTStartTime='2021-02-21T14:13:56.566+00:00', EndDate='2021-02-21', GMTEndTime='2021-02-21T14:14:17.851+00:00', EndTime='23:14:17.851122', TotalElapsedTime='26367', MaximumElapsedTime='14173', MinimumElapsedTime='12194', TotalCPUTime='18907', MaximumCPUTime='13735', MinimumCPUTime='5172', CPUTimeWaitingForInputMessage='2374', ElapsedTimeWaitingForInputMessage='21257791', TotalInputMessages='2', TotalSizeOfInputMessages='816', MaximumSizeOfInputMessages='408', MinimumSizeOfInputMessages='408', NumberOfThreadsInPool='1', TimesMaximumNumberOfThreadsReached='2', TotalNumberOfMQErrors='0', TotalNumberOfMessagesWithErrors='0', TotalNumberOfErrorsProcessingMessages='0', TotalNumberOfTimeOuts='0', TotalNumberOfCommits='0', TotalNumberOfBackouts='0', AccountingOrigin='Anonymous'.
2021-02-21 23:14:18.263516 28646 UserTrace BIP2381I: ACE thread statistics. ProcessID='28385', Key='0', Number='28692', TotalNumberOfInputMessages='2', TotalElapsedTime='26367', TotalCPUTime='18907', CPUTimeWaitingForInputMessage='2374', ElapsedTimeWaitingForInputMessage='21257791', TotalSizeOfInputMessages='816', MaximumSizeOfInputMessages='408', MinimumSizeOfInputMessages='408'.
2021-02-21 23:14:18.263598 28646 UserTrace BIP2382I: ACE node statistics. ProcessID='28385', Key='0', Label='Compute', Type='ComputeNode', TotalElapsedTime='12558', MaximumElapsedTime='9238', MinimumElapsedTime='3320', TotalCPUTime='5983', MaximumCPUTime='3154', MinimumCPUTime='2829', CountOfInvocations='2', NumberOfInputTerminals='1', NumberOfOutputTerminals='6'.
2021-02-21 23:14:18.263664 28646 UserTrace BIP2383I: ACE terminal statistics. ProcessID='28385', Key='0', Label='failure', Type='Output', CountOfInvocations='0',
2021-02-21 23:14:18.263706 28646 UserTrace BIP2383I: ACE terminal statistics. ProcessID='28385', Key='0', Label='in', Type='Input', CountOfInvocations='2',
2021-02-21 23:14:18.298760 28646 UserTrace BIP2383I: ACE terminal statistics. ProcessID='28385', Key='0', Label='out', Type='Output', CountOfInvocations='2',
2021-02-21 23:14:18.298820 28646 UserTrace BIP2383I: ACE terminal statistics. ProcessID='28385', Key='0', Label='out1', Type='Output', CountOfInvocations='0',
2021-02-21 23:14:18.298858 28646 UserTrace BIP2383I: ACE terminal statistics. ProcessID='28385', Key='0', Label='out2', Type='Output', CountOfInvocations='0',
2021-02-21 23:14:18.298894 28646 UserTrace BIP2383I: ACE terminal statistics. ProcessID='28385', Key='0', Label='out3', Type='Output', CountOfInvocations='0',
2021-02-21 23:14:18.298928 28646 UserTrace BIP2383I: ACE terminal statistics. ProcessID='28385', Key='0', Label='out4', Type='Output', CountOfInvocations='0',
2021-02-21 23:14:18.300334 28646 UserTrace BIP2382I: ACE node statistics. ProcessID='28385', Key='0', Label='HTTP Input', Type='WSInputNode', TotalElapsedTime='11021', MaximumElapsedTime='10309', MinimumElapsedTime='712', TotalCPUTime='10876', MaximumCPUTime='10165', MinimumCPUTime='711', CountOfInvocations='2', NumberOfInputTerminals='0', NumberOfOutputTerminals='4'.
2021-02-21 23:14:18.300418 28646 UserTrace BIP2383I: ACE terminal statistics. ProcessID='28385', Key='0', Label='catch', Type='Output', CountOfInvocations='0',
2021-02-21 23:14:18.300458 28646 UserTrace BIP2383I: ACE terminal statistics. ProcessID='28385', Key='0', Label='failure', Type='Output', CountOfInvocations='0',
2021-02-21 23:14:18.300492 28646 UserTrace BIP2383I: ACE terminal statistics. ProcessID='28385', Key='0', Label='out', Type='Output', CountOfInvocations='2',
2021-02-21 23:14:18.300526 28646 UserTrace BIP2383I: ACE terminal statistics. ProcessID='28385', Key='0', Label='timeout', Type='Output', CountOfInvocations='0',
2021-02-21 23:14:18.300556 28646 UserTrace BIP2382I: ACE node statistics. ProcessID='28385', Key='0', Label='HTTP Reply', Type='WSReplyNode', TotalElapsedTime='2788', MaximumElapsedTime='2244', MinimumElapsedTime='544', TotalCPUTime='2048', MaximumCPUTime='1632', MinimumCPUTime='416', CountOfInvocations='2', NumberOfInputTerminals='1', NumberOfOutputTerminals='2'.
2021-02-21 23:14:18.300600 28646 UserTrace BIP2383I: ACE terminal statistics. ProcessID='28385', Key='0', Label='failure', Type='Output', CountOfInvocations='0',
2021-02-21 23:14:18.300636 28646 UserTrace BIP2383I: ACE terminal statistics. ProcessID='28385', Key='0', Label='in', Type='Input', CountOfInvocations='2',
2021-02-21 23:14:18.300668 28646 UserTrace BIP2383I: ACE terminal statistics. ProcessID='28385', Key='0', Label='out', Type='Output', CountOfInvocations='0',
```

リソース統計

機能概要

- 統合サーバーが使用するシステム・リソースの利用状況や内部コンポーネントの処理状況を統計レコードとして出力する機能
 - ◆ JVM内のメモリ使用量やGC発生頻度
 - ◆ パーサーのリソース消費やパース(read)/シリアライズ(write)件数
 - ◆ SOAPメッセージの送受信数、成功/エラー回数 など
- 出力インターバルは20秒（固定）
- 出力先とフォーマットは以下から選択可
 - ◆ MQトピック(XML/JSON)
 - ◆ MQTTトピック(XML/JSON)
 - ◆ ファイル(CSV)
- Web UIでモニターすることも可能
 - ◆ REST APIからも取得可
- デフォルトでON



機能詳細は以下参照

<https://www.ibm.com/docs/en/app-connect/12.0?topic=performance-analyzing-resource>

リソース・タイプ

■ リソース統計で取得できるリソース一覧

リソース・タイプ	内容
.NET アプリケーション・ドメイン	.NETアプリケーション・ドメインが前回のガーベッジ・コレクション時に使用していたメモリ・サイズ、起動以降アサインされたメモリの合計サイズ
.NET ガーベッジ・コレクション	外部要求によって強制的に実行されたガーベッジ・コレクションの回数、ジェネレーション0/1/2で実行されたガーベッジ・コレクション回数、コミットされたトータルのヒープサイズ、予約されたトータルのヒープサイズ、など
Connect Direct	Connect:Directサーバーへの接続情報、受信した処理回数/データサイズ、受信したが処理されなかった回数、出力した処理回数/データサイズ
CICS	CICS に対して実行した呼び出しのうち、成功した呼び出しと失敗した呼び出しの数、失敗した要求のうち、認証障害などのセキュリティー問題に関連して失敗した要求の数、接続試行が失敗した頻度
CORBA	CORBA サーバーを呼び出す回数、およびそれらの呼び出しが成功する回数と CORBA 例外になる回数
ファイル	あらゆるタイプのファイル・ノードによって実行されたファイル・アクションのローカル・ファイル・システムに関する要約を表示。正常に読み取られたファイル数/レコード数/バイト数、正常に作成されたファイル数/レコード数/バイト数
FTP	使用されるプロトコル: FTP または SFTP、リモート・サーバーからブローカーのファイル・システムへの転送の数/バイト数、ブローカーのファイル・システムからリモート・サーバーへの転送の数/バイト数
FTEエージェント	FTE エージェントが送受信した転送数
グローバル・キャッシュ	キャッシュへの接続成功/失敗数、マップの使用/READ/WRITE/REMOVE/失敗数、など
Java仮想マシン	JVM のメモリー使用量、実行グループ内でのガーベッジ・コレクションの発生頻度
JDBC接続プール	接続プール内の接続数、最大接続数、接続要求数、遅延された要求数、遅延した最大時間、タイムアウト(15s)した要求数
JMS	現在オープンな接続/セッション数、クローズされた接続/セッション数、JMSInput/JMSReceiveが受信したメッセージ数、JMSOutputが送信したメッセージ数、JMSReceiveがブラウズしたメッセージ数、接続エラー数

リソース・タイプ

■ リソース統計で取得できるリソース一覧（続き）

リソース・タイプ	内容
MQTT	現在オープンな接続数、統合サーバー起動以降クローズされた接続の合計数、受信/送信したメッセージ数/メッセージサイズ、接続失敗数
ODM	成功/失敗した実行数、ルールに合致した件数
アウトバウンド・ソケット	SOAP、SCA、または HTTP のいずれかの要求ノードを経由してメッセージを送信するときに実行グループが使用するソケットを流れるメッセージ・データのサイズとボリューム
パーサー	パーサーが備えるメッセージ・ツリーとビット・ストリームによるリソース消費量
SOAP	SOAPInput ノードが受信しているインバウンド・メッセージの数、SOAPReply ノードが送信している応答の数、それらの呼び出しが成功する回数と SOAP Faultになる回数
セキュリティ	セキュリティー承認要求数、そのうちの成功した数、およびセキュリティー・キャッシュを利用した数
TCPIPClientNodes	TCPIPClientノードにおけるオープン接続数、再始動時以降にクローズされた接続数、受信/送信メッセージ数、受信/送信データ数、失敗した接続試行の総数
TCPIPServerNodes	TCPIPServerノードにおけるオープン接続数、再始動時以降にクローズされた接続数、受信/送信メッセージ数、受信/送信データ数、失敗した（拒否された）SSL接続試行の総数
ODBC	DSN に対して何らかのステートメントが実行された数/失敗した数、現在オープン/クローズされている接続数

トピック

■ MQ/MQTTに出力する場合の出力先となるトピック

◆ 統合ノード構成

- MQトピック
 - ✓ XMLフォーマット

```
$SYS/Broker/integrationNodeName/ResourceStatistics/integrationServerName
```

- ✓ JSONフォーマット

```
$SYS/Broker/integrationNodeName/Statistics/JSON/Resource/integrationServerName
```

• MQTTトピック

- ✓ XMLフォーマット

```
IBM/IntegrationBus/integrationNodeName/ResourceStatistics/integrationServerName
```

- ✓ JSONフォーマット

```
IBM/IntegrationBus/integrationNodeName/Statistics/JSON/Resource/integrationServerName
```

※斜体青字の箇所は可変

- ◆ 独立統合サーバー構成の場合は、上記「*integrationNodeName*」の箇所が固定値「integration_server」となる

- ◆ サブスクライブする際は、ワイルドカード (+、#) を使用してまとめて取得可

- 例

```
$SYS/Broker/+/ResourceStatistics/#
```

統計レコード

- 統合サーバー単位でインターバル毎に統計レコードを出力先トピックに出力
 - ◆ 1レコードにすべてのリソース・タイプの統計情報が含まれる

リソース統計レコード例 (XMLフォーマットの場合)

```
<ResourceStatistics brokerLabel="IN01" brokerUUID="5eb081f6-ac4e-11ea-8860-7f0001010000" executionGroupName="ex1" executionGroupUUID="00000000-0000-0000-0000-000000000000" collectionStartDate="2020-10-01" collectionStartTime="22:42:12" startDate="2020-10-08" startTime="13:30:50" endDate="2020-10-08" endTime="13:31:10" timezone="Asia/Tokyo">
  <ResourceType name="Security">
    <resourceIdentifier name="summary" TotalCacheEntries="0" TotalOperations="0" TotalSuccessfulOperations="0" TotalOperationsServicedByCache="0" />
  </ResourceType>
  <ResourceType name="Kafka">
    <resourceIdentifier name="summary" />
  </ResourceType>
  <ResourceType name="MQTT">
    <resourceIdentifier name="summary" BytesReceived="0" BytesSent="57981888" ClosedConnections="0" FailedConnections="0" MessagesReceived="0" MessagesSent="29047" OpenConnections="1" />
  </ResourceType>
  <ResourceType name="GlobalCache">
    <resourceIdentifier name="summary" TotalMapActions="0" MapReads="0" MapWrites="0" MapRemoves="0" FailedActions="0" MapsUsed="0" ConnectionFailures="0" Connects="0" />
  </ResourceType>
  <ResourceType name="JMS">
    <resourceIdentifier name="summary" NumberOfOpenJMSConnections="0" NumberOfClosedJMSConnections="0" NumberOfOpenJMSSessions="0" NumberOfClosedJMSSessions="0" NumberOfMessagesReceived="0" NumberOfMessagesSent="0" NumberOfMessagesBrowsed="0" NumberOfJMSConnectionFailures="0" />
  </ResourceType>
  :
  :
</ResourceStatistics>
```

統計レコード

リソース統計レコード例 (JSONフォーマットの場合)

```
{
  "ResourceStatistics": {
    "brokerLabel": "IN01",
    "brokerUUID": "5eb081f6-ac4e-11ea-8860-7f0001010000",
    "executionGroupName": "ex1",
    "executionGroupUUID": "00000000-0000-0000-0000-000000000000",
    "collectionStartDate": "2020-10-08",
    "collectionStartTime": "16:19:50",
    "startDate": "2020-10-08",
    "startTime": "16:20:35",
    "endDate": "2020-10-08",
    "endTime": "16:20:55",
    "timezone": "Asia/Tokyo",
    "ResourceType": [
      {
        "name": "Security",
        "resourceIdentifier": [
          {
            "name": "summary",
            "TotalCacheEntries": 0,
            "TotalOperations": 0,
            "TotalSuccessfulOperations": 0,
            "TotalOperationsServicedByCache": 0
          }
        ]
      },
      {
        "name": "FTP",
        "resourceIdentifier": [
          {
            "name": "summary",
            "FTPGets": 0,
            "BytesReceived": 0,
            "FTPPuts": 0,
            "BytesSent": 0,
            "Protocol": "All"
          }
        ]
      }
    ]
  },
  :
  :
```

```
:
{
  "name": "JVM",
  "resourceIdentifier": [
    {
      "name": "summary",
      "InitialMemoryInMB": 305,
      "UsedMemoryInMB": 30,
      "CommittedMemoryInMB": 311,
      "MaxMemoryInMB": -1,
      "CumulativeGCTimeInSeconds": 0,
      "CumulativeNumberOfGCCollections": 14
    },
    {
      "name": "Heap Memory",
      "InitialMemoryInMB": 32,
      "UsedMemoryInMB": 8,
      "CommittedMemoryInMB": 32,
      "MaxMemoryInMB": 256
    },
    {
      "name": "Non-Heap Memory",
      "InitialMemoryInMB": 273,
      "UsedMemoryInMB": 22,
      "CommittedMemoryInMB": 279,
      "MaxMemoryInMB": -1
    },
    {
      "name": "Garbage Collection - scavenge",
      "CumulativeGCTimeInSeconds": 0,
      "CumulativeNumberOfGCCollections": 13
    },
    {
      "name": "Garbage Collection - global",
      "CumulativeGCTimeInSeconds": 0,
      "CumulativeNumberOfGCCollections": 1
    }
  ]
},
:
```

構成概要

- MQ/MQTTに出力する場合の出力先の準備、設定はメッセージフロー統計と同じ
 - ◆ 前章「メッセージフロー統計」の構成概要参照
- リソース統計のON/OFFを統合ノード、もしくは統合サーバーに設定
 - ◆ リソース・タイプ単位の設定はできない
 - ◆ 以下のいずれかの方法で設定
 - ◆ 構成ファイル（Statisticsセクション/Resourceサブセクション）（起動/再起動で反映）
 - ・ デフォルトでON（Fixpack11.0.0.8以降）
 - ・ ファイルに出力する場合はResourceサブセクションのプロパティで設定
 - ✓ 構成ファイルでのみファイルへの出力設定が可能
 - ◆ Web UI（動的反映、統合サーバー起動時のみ設定可）
 - ◆ REST API（動的反映、統合サーバー起動時のみ設定可）
 - ◆ mqsichangeresourcstats（動的反映、統合サーバー起動時のみ設定可）

リソース統計設定

■ 構成ファイル(.yaml)に設定する方法

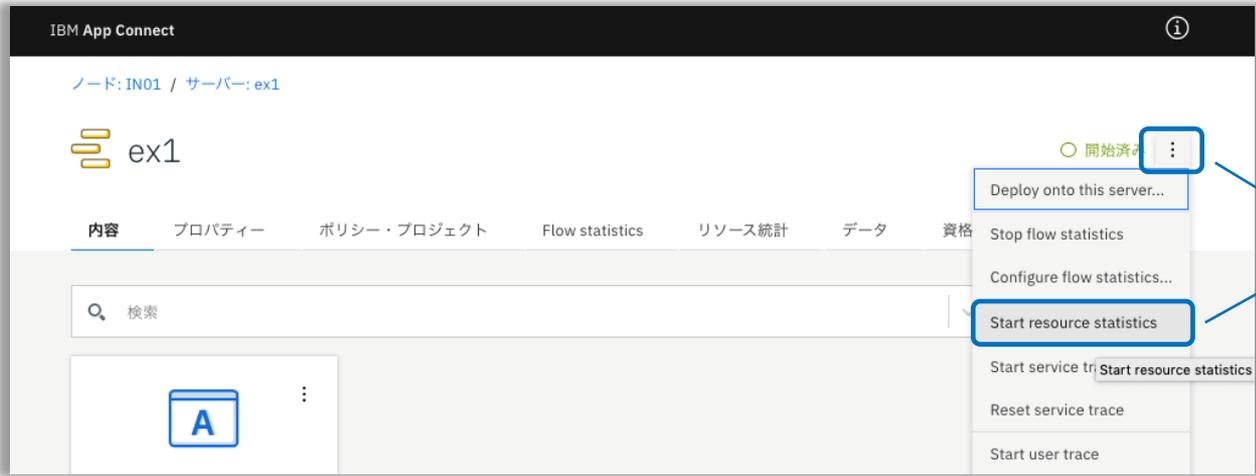
- ◆ Statisticsセクション/ResourceサブセクションのreportingOnプロパティ(true/false)で設定
 - trueに設定すると、Eventsセクション/OperationalEventsサブセクションの設定に基づいてMQ/MQTTトピックに出力
 - Fixpack 11.0.0.8以降はデフォルトでtrue
- ◆ ファイルに出力する場合は、outputFormatプロパティにcsvFileを設定

設定例

```
Statistics:
  Resource:
    reportingOn: true      # choose 1 of : true|false, Set to 'true'. If unset will default to 'false'
    outputFormat: 'csvFile' # choose 'csvFile', or 'file' for IIB v10 compatibility. if unset will default to ''
```

ON/OFF設定

■ Web UIで設定する方法



統合サーバーの画面で設定ボタンを押し、「Start resource statistics」を押す（動的に反映）

ファイル出力例

- 以下のフォルダにリソース毎にファイルが生成され、統計レコードが出力される

ディレクトリ	/var/mqsi/common/resourceStats
ファイル名	ResourceStats_<統合ノード>_<統合サーバー>_<リソース名>.txt

JVMのリソース統計出力例 (ResourceStats_IN01_ex1_JVM.txt)

1行目はヘッダー

```
ResourceName,IntegrationNodeName,IntegrationServerName,CollectionTimestamp,StartTimestamp,EndTimeStam,InitialMemoryInMB,UsedMemoryInM
B,CommittedMemoryInMB,MaxMemoryInMB,CumulativeGCTimeInSeconds,CumulativeNumberOfGCCollections
summary,IN01,ex1,2021-03-07 11:15:34.000,2021-03-07 11:15:34.000,2021-03-07 11:15:34.000,305,29,324,-1,0,12
Heap Memory,IN01,ex1,2021-03-07 11:15:34.000,2021-03-07 11:15:34.000,2021-03-07 11:15:34.000,32,9,32,256
Non-Heap Memory,IN01,ex1,2021-03-07 11:15:34.000,2021-03-07 11:15:34.000,2021-03-07 11:15:34.000,273,20,292,-1
Garbage Collection - scavenge,IN01,ex1,2021-03-07 11:15:34.000,2021-03-07 11:15:34.000,2021-03-07 11:15:34.000,0,11
Garbage Collection - global,IN01,ex1,2021-03-07 11:15:34.000,2021-03-07 11:15:34.000,2021-03-07 11:15:34.000,0,1
summary,IN01,ex1,2021-03-07 11:15:34.000,2021-03-07 11:15:34.000,2021-03-07 11:15:56.000,305,30,325,-1,0,13
Heap Memory,IN01,ex1,2021-03-07 11:15:34.000,2021-03-07 11:15:34.000,2021-03-07 11:15:56.000,32,10,32,256
Non-Heap Memory,IN01,ex1,2021-03-07 11:15:34.000,2021-03-07 11:15:34.000,2021-03-07 11:15:56.000,273,20,293,-1
Garbage Collection - scavenge,IN01,ex1,2021-03-07 11:15:34.000,2021-03-07 11:15:34.000,2021-03-07 11:15:56.000,0,12
Garbage Collection - global,IN01,ex1,2021-03-07 11:15:34.000,2021-03-07 11:15:34.000,2021-03-07 11:15:56.000,0,1
summary,IN01,ex1,2021-03-07 11:15:34.000,2021-03-07 11:15:56.000,2021-03-07 11:16:16.000,305,30,309,-1,0,13
Heap Memory,IN01,ex1,2021-03-07 11:15:34.000,2021-03-07 11:15:56.000,2021-03-07 11:16:16.000,32,10,32,256
Non-Heap Memory,IN01,ex1,2021-03-07 11:15:34.000,2021-03-07 11:15:56.000,2021-03-07 11:16:16.000,273,20,277,-1
Garbage Collection - scavenge,IN01,ex1,2021-03-07 11:15:34.000,2021-03-07 11:15:56.000,2021-03-07 11:16:16.000,0,12
Garbage Collection - global,IN01,ex1,2021-03-07 11:15:34.000,2021-03-07 11:15:56.000,2021-03-07 11:16:16.000,0,1
```

インターバル毎にこの分のレコードが出力される

ユーザー・トレース

機能概要

■ 製品が提供するトレース機能の1つ

- ◆ ACEでは、ユーザー・トレースとサービス・トレースの2種類のトレース機能を提供
- ◆ ユーザー・トレース
 - ・ 主にユーザー実装部分の動作を出力
- ◆ サービス・トレース
 - ・ ユーザー・トレースの出力に加え、より詳細な内部動作を出力
 - ・ 基本的に問題判別時に製品サポート部門から依頼された場合に取得

■ メッセージフローやESQLのデバッグやエビデンスに利用可能

- ◆ メッセージフローのターミナルの遷移やESQLのステップの動きを確認できる
- ◆ UT時の動作確認、エビデンスに利用することも可能

■ ランタイム稼働時に一時的に取得

- ◆ トレースの取得は性能へのインパクトがあるため、基本的にはデバック時など一時的に取得
- ◆ デフォルトはInactive
- ◆ コマンド、Web UI、REST APIで動的に設定

■ テキストファイルに出力

- ◆ V10まではバイナリ・ファイルに出力され、別途コマンドで抽出、整形する必要があったが、V11から直接テキストファイルに出力される

設定 (mqsischangetrace)

■ mqsischangetraceコマンドの構文

```
mqsischangetrace <統合ノード> -e <統合サーバー> -u -l <レベル> [-r] [-c <サイズ>]
```

- -u : ユーザー・トレースの指定
- -l <レベル> : debug = 開始、none = 停止
- -r : リセット (指定すると既存トレースファイルの内容は削除される)
- -c <サイズ> : トレースのサイズ(KB)を指定 (デフォルトは102400KB ※4ファイルの合計サイズ)

- その他のオプションについては下記リンク参照

<https://www.ibm.com/docs/en/app-connect/12.0?topic=commands-mqsischangetrace-command>

■ 実行例

◆ トレース開始

```
mqsischangetrace IN01 -e ex1 -u -l debug -r
```

- -r 指定により、その時点のトレース・ファイルの内容は削除され、以降のトレースが出力される

◆ トレース停止

```
mqsischangetrace IN01 -e ex1 -u -l none
```

設定 (Web UI)

■ Web UIによるトレースの開始、停止

The image shows a sequence of three screenshots from the IBM App Connect Web UI, illustrating the steps to start and stop user tracing for a specific server.

Left Screenshot: Shows the main interface for a node named 'IN01'. A blue box highlights the three-line menu icon for the 'ex1' server. A callout box points to this icon with the text: **対象の統合サーバーの右上の設定マークをクリック** (Click the setting mark in the top right of the target integrated server).

Middle Screenshot: Shows the context menu that appears after clicking the menu icon. The 'ユーザー・トレースの開始' (Start User Tracing) option is highlighted with a blue box. A callout box points to this option with the text: **「ユーザー・トレースの開始」をクリックして開始 (debugレベル)** (Click 'Start User Tracing' to start (debug level)).

Right Screenshot: Shows the context menu again, but now the 'ユーザー・トレースの停止' (Stop User Tracing) option is highlighted with a blue box. A callout box points to this option with the text: **「ユーザー・トレースの停止」をクリックして停止** (Click 'Stop User Tracing' to stop).

Bottom Callout: A separate blue box at the bottom states: **トレース・ファイルをリセット (内容削除) する場合は、「ユーザー・トレースのリセット」をクリック** (If you want to reset (delete content) the trace file, click 'Reset User Tracing').

出力先

■ 以下のディレクトリ、ファイルに出力

ディレクトリ	/var/mqsi/common/log ※独立統合サーバーの場合は<ワーク・ディレクトリ>/common/log
ファイル名	<統合ノード>.<統合サーバー>.userTrace.n.txt n=0~3 ※4つのファイルをローテーションして使用

■ サイズは指定可能

- ◆ mqsi changetrace コマンドの -c オプションで指定
- ◆ デフォルトは 102400KB
- ◆ 4つのファイルの合計サイズ
 - 1つのファイルは指定サイズの 1/4

出力例

```
2021-03-28 09:25:57.967736 9508 UserTrace BIP11501I: Received data from input node 'HTTP Input'.
The input node 'HTTP Input' has received data and has propagated it to the message flow 'http01'.
2021-03-28 09:25:57.967900 9508 UserTrace BIP6060I: Node 'http01.HTTP Input' used parser type 'Properties' to process a portion of the incoming message of length '0' bytes beginning at offset '0'.
2021-03-28 09:25:57.968056 9508 UserTrace BIP6061I: Node 'http01.HTTP Input' used parser type 'HTTPHeader' to process a portion of the incoming message of length '397' bytes beginning at offset '0'. The parser type was selected based on value 'WSINPHDR' from the previous parser.
2021-03-28 09:25:57.968144 9508 UserTrace BIP3122I: Message received and propagated to 'out' terminal of HTTP input node 'http01.HTTP Input'.
2021-03-28 09:25:57.968308 9508 UserTrace BIP6061I: Node 'http01.HTTP Input' used parser type 'BLOB' to process a portion of the incoming message of length '3' bytes beginning at offset '397'. The parser type was selected based on value 'NONE' from the previous parser.
2021-03-28 09:25:57.968456 9508 UserTrace BIP2537I: Node 'http01.Compute': Executing statement 'BEGIN ... END;' at (.http01_Compute.Main, 2.2).
2021-03-28 09:25:57.970584 9508 UserTrace BIP2537I: Node 'http01.Compute': Executing statement 'CALL CopyMessageHeaders();' at (.http01_Compute.Main, 3.3).
2021-03-28 09:25:57.970656 9508 UserTrace BIP2538I: Node 'http01.Compute': Evaluating expression 'CopyMessageHeaders()' at (.http01_Compute.Main, 3.8).
2021-03-28 09:25:57.970808 9508 UserTrace BIP2537I: Node 'http01.Compute': Executing statement 'BEGIN ... END;' at (.http01_Compute.CopyMessageHeaders, 1.40).
2021-03-28 09:25:57.970888 9508 UserTrace BIP2537I: Node 'http01.Compute': Executing statement 'DECLARE I INTEGER 1;' at (.http01_Compute.CopyMessageHeaders, 2.3).
2021-03-28 09:25:57.970932 9508 UserTrace BIP2537I: Node 'http01.Compute': Executing statement 'DECLARE J INTEGER;' at (.http01_Compute.CopyMessageHeaders, 3.3).
2021-03-28 09:25:57.970960 9508 UserTrace BIP2537I: Node 'http01.Compute': Executing statement 'SET J = CARDINALITY(InputRoot.*:[I]);' at (.http01_Compute.CopyMessageHeaders, 4.3).
2021-03-28 09:25:57.971368 9508 UserTrace BIP2539I: Node ':': Evaluating expression 'InputRoot.*:[I]' at (.http01_Compute.CopyMessageHeaders, 4.23). This resolved to 'InputRoot.*:[I]'. The result was 'LIST... First Element Type=16777216 NameSpace=' Name='Properties' Value=NULL'.
2021-03-28 09:25:57.971456 9508 UserTrace BIP2540I: Node ':': Finished evaluating expression 'CARDINALITY(InputRoot.*:[I])' at (.http01_Compute.CopyMessageHeaders, 4.11). The result was '3'.
2021-03-28 09:25:57.971504 9508 UserTrace BIP2537I: Node 'http01.Compute': Executing statement 'WHILE I < J DO ... END WHILE;' at (.http01_Compute.CopyMessageHeaders, 5.3).
2021-03-28 09:25:57.971540 9508 UserTrace BIP2539I: Node ':': Evaluating expression 'I' at (.http01_Compute.CopyMessageHeaders, 5.9). This resolved to 'I'. The result was '1'.
2021-03-28 09:25:57.971560 9508 UserTrace BIP2539I: Node ':': Evaluating expression 'J' at (.http01_Compute.CopyMessageHeaders, 5.13). This resolved to 'J'. The result was '3'.
2021-03-28 09:25:57.971604 9508 UserTrace BIP2539I: Node ':': Evaluating expression 'I < J' at (.http01_Compute.CopyMessageHeaders, 5.9). This resolved to '1 < 3'. The result was 'TRUE'.
2021-03-28 09:25:57.971652 9508 UserTrace BIP2537I: Node 'http01.Compute': Executing statement 'SET OutputRoot.*:[I] = InputRoot.*:[I];' at (.http01_Compute.CopyMessageHeaders, 6.4).
2021-03-28 09:25:57.971688 9508 UserTrace BIP2539I: Node ':': Evaluating expression 'I' at (.http01_Compute.CopyMessageHeaders, 6.38). This resolved to 'I'. The result was '1'.
2021-03-28 09:25:57.971724 9508 UserTrace BIP2539I: Node ':': Evaluating expression 'I' at (.http01_Compute.CopyMessageHeaders, 6.38). This resolved to 'I'. The result was '1'.
2021-03-28 09:25:57.971748 9508 UserTrace BIP2539I: Node ':': Evaluating expression 'InputRoot.*:[I]' at (.http01_Compute.CopyMessageHeaders, 6.26). This resolved to 'InputRoot.*:[I]'. The result was 'ROW... Root Element Type=16777216 NameSpace=' Name='Properties' Value=NULL'.
2021-03-28 09:25:57.971784 9508 UserTrace BIP2539I: Node ':': Evaluating expression 'I' at (.http01_Compute.CopyMessageHeaders, 6.21). This resolved to 'I'. The result was '1'.
2021-03-28 09:25:57.971904 9508 UserTrace BIP2568I: Node 'http01.Compute': Copying sub-tree from 'InputRoot.*:[I]' to 'OutputRoot.*:[I]'.
2021-03-28 09:25:57.972028 9508 UserTrace BIP2537I: Node 'http01.Compute': Executing statement 'SET I = I + 1;' at (.http01_Compute.CopyMessageHeaders, 7.4).
2021-03-28 09:25:57.972068 9508 UserTrace BIP2539I: Node ':': Evaluating expression 'I' at (.http01_Compute.CopyMessageHeaders, 7.12). This resolved to 'I'. The result was '1'.
2021-03-28 09:25:57.972100 9508 UserTrace BIP2539I: Node ':': Evaluating expression 'I + 1' at (.http01_Compute.CopyMessageHeaders, 7.14). This resolved to '1 + 1'. The result was '2'.
2021-03-28 09:25:57.972116 9508 UserTrace BIP2539I: Node ':': Evaluating expression 'I' at (.http01_Compute.CopyMessageHeaders, 5.9). This resolved to 'I'. The result was '2'.
2021-03-28 09:25:57.972128 9508 UserTrace BIP2539I: Node ':': Evaluating expression 'J' at (.http01_Compute.CopyMessageHeaders, 5.13). This resolved to 'J'. The result was '3'.
(省略)
2021-03-28 09:25:57.973420 9508 UserTrace BIP2539I: Node ':': Evaluating expression 'I < J' at (.http01_Compute.CopyMessageHeaders, 5.9). This resolved to '3 < 3'. The result was 'FALSE'.
2021-03-28 09:25:57.973532 9508 UserTrace BIP2537I: Node 'http01.Compute': Executing statement 'SET OutputRoot.JSON.Data.res.data1 = 'ok';' at (.http01_Compute.Main, 6.3).
2021-03-28 09:25:57.973632 9508 UserTrace BIP2566I: Node 'http01.Compute': Assigning value 'ok' to field / variable 'OutputRoot.JSON.Data.res.data1'.
2021-03-28 09:25:57.973688 9508 UserTrace BIP2537I: Node 'http01.Compute': Executing statement 'RETURN TRUE;' at (.http01_Compute.Main, 8.3).
2021-03-28 09:25:57.973828 9508 UserTrace BIP4015I: Message propagated to the 'out' terminal of node 'http01.Compute' with the following message trees: InputLocalEnvironment, OutputRoot, InputExceptionList.
2021-03-28 09:25:57.974488 9508 UserTrace BIP11305I: The Parser of type 'MQROOT' at address '0x7f65a0166aa0' has been reset ready for re-use. This thread has '16' cached parsers.
(省略)
2021-03-28 09:25:57.975836 9508 UserTrace BIP11506I: Committed a local transaction.
A local transaction has been committed for work done on the message flow thread.
```

リクエスト受信

HTTP Inputノードのout
ターミナルへ遷移

ESQLの各ステップ
が出力される

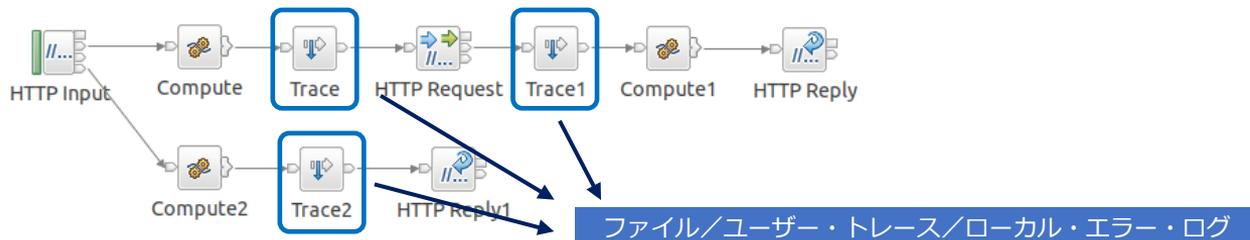
フロー処理終了

Computeノードのout
ターミナルへ遷移

Trace ノード

機能概要

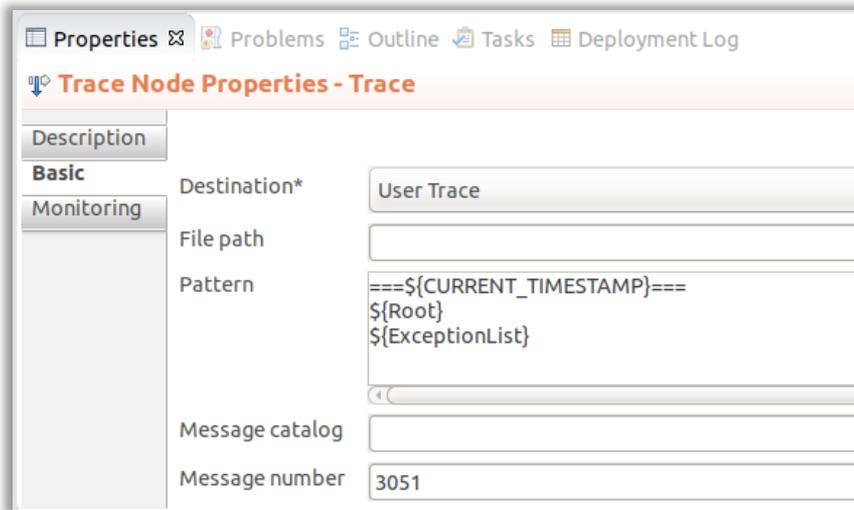
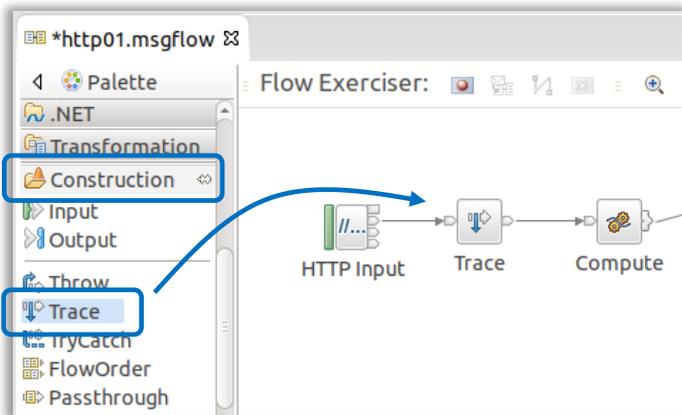
- 製品が提供する組み込みノードの1つ
- フローの任意の箇所に配置して、任意の内容を出力
 - ◆ タイムスタンプやメッセージツリー、ExceptionListなど任意の内容を出力できる
 - ◆ ESQLの関数を利用することも可能
- 出力先は以下から選択
 - ◆ 任意のディレクトリ、ファイル
 - ◆ ユーザー・トレース
 - ◆ ローカル・エラー・ログ
 - ・ UNIX:syslog、Window:イベントログ
- 動的に出力有無を変更可能
 - ◆ mqsichange命令の -n on|off
 - ◆ Web UIからは変更不可
- 監査証跡としてのアプリケーションログや問題判別用のエラーログとして利用



設定

■ ツールキットにて対象のフローに設定

- ◆ パレットのConstructionタブを開き、Traceノードをフローの任意の箇所に配置
- ◆ 前後のノードのターミナルと接続
- ◆ ノード・プロパティで出力先や出力内容を指定



設定

■ 主なノード・プロパティ

プロパティ	値、説明
Destination	出力先として以下のいずれかを指定 User Trace : ユーザー・トレースに出力 File : 任意のファイルに出力 Local Error Log : ローカル・エラー・ログ(syslog/イベントログ)
File Path	DestinationでFileを指定した場合、出力先のディレクトリ、ファイル名を指定
Pattern	出力内容を任意の文字列で指定 \${パス}の形式で動的に論理ツリーの値をセットすることも可能 主なパス ・Root : メッセージツリー ・Body : メッセージツリーのBody部 ・LocalEnvironment : LocalEnvironmentツリー ・Environment : Environmentツリー ・ExceptionList : ExceptionListツリー 特定のフィールドを指定することもできる 例 : \${Root.JSON.Data.elem01} ESQLの関数を指定することもできる 例 : \${CURRENT_TIMESTAMP} その他、ランタイム情報の変数を指定することも可能 下記リンク参照 https://www.ibm.com/docs/en/app-connect/12.0?topic=er-integration-properties-that-are-accessible-from-esql-mapping-node-java

ノード・プロパティの詳細は以下参照

<https://www.ibm.com/docs/en/app-connect/12.0?topic=nodes-trace-node>

出力例

TraceノードのPattern設定

```
===${CURRENT_TIMESTAMP}===
${Root}
```

出力例

```
===2021-03-28 14:59:33.569562===
( ['WSRoot' : 0x7f658cc0d5e0]
  (0x01000000:Name):Properties = ( ['WSPROPERTYPARSER' : 0x7f658c6b2890]
    (0x03000000:NameValue):MessageSet = " (CHARACTER)
    (0x03000000:NameValue):MessageType = " (CHARACTER)
    (0x03000000:NameValue):MessageFormat = " (CHARACTER)
    (0x03000000:NameValue):Encoding = 546 (INTEGER)
    (0x03000000:NameValue):CodedCharSetId = 1208 (INTEGER)
    (0x03000000:NameValue):Transactional = FALSE (BOOLEAN)
    (0x03000000:NameValue):Persistence = FALSE (BOOLEAN)
    (0x03000000:NameValue):CreationTime = GMTTIMESTAMP '2021-03-28 05:59:33.568911' (GMTTIMESTAMP)
    (0x03000000:NameValue):ExpirationTime = -1 (INTEGER)
    (0x03000000:NameValue):Priority = 0 (INTEGER)
    (0x03000000:NameValue):ReplyIdentifier = X'0000000000000000000000000000000000000000000000000000000000000000' (BLOB)
    (0x03000000:NameValue):ReplyProtocol = 'SOAP-HTTP' (CHARACTER)
    (0x03000000:NameValue):Topic = NULL
    (0x03000000:NameValue):ContentType = 'application/x-www-form-urlencoded' (CHARACTER)
    (0x03000000:NameValue):IdentitySourceType = " (CHARACTER)
    (0x03000000:NameValue):IdentitySourceToken = " (CHARACTER)
    (0x03000000:NameValue):IdentitySourcePassword = " (CHARACTER)
    (0x03000000:NameValue):IdentitySourceIssuedBy = " (CHARACTER)
    (0x03000000:NameValue):IdentityMappedType = " (CHARACTER)
    (0x03000000:NameValue):IdentityMappedToken = " (CHARACTER)
    (0x03000000:NameValue):IdentityMappedPassword = " (CHARACTER)
    (0x03000000:NameValue):IdentityMappedIssuedBy = " (CHARACTER)
  )
  (0x01000000:Name):HTTPHeader = ( ['WSINPHDR' : 0x7f658c2e65c0]
    (0x03000000:NameValue):Host = '192.168.58.4:7800' (CHARACTER)
    (0x03000000:NameValue):User-Agent = 'curl/7.64.1' (CHARACTER)
    (0x03000000:NameValue):Accept = '*/*' (CHARACTER)
    (0x03000000:NameValue):Content-Length = '3' (CHARACTER)
    (0x03000000:NameValue):Content-Type = 'application/x-www-form-urlencoded' (CHARACTER)
    (0x03000000:NameValue):X-Original-HTTP-Command = 'POST http://192.168.58.4:7800/http01 HTTP/1.1' (CHARACTER)
    (0x03000000:NameValue):X-Query-String = 'aaa' (CHARACTER)
    (0x03000000:NameValue):X-Server-Name = '192.168.58.4' (CHARACTER)
    (0x03000000:NameValue):X-Server-Port = '7800' (CHARACTER)
    (0x03000000:NameValue):X-Remote-Addr = '192.168.58.1' (CHARACTER)
    (0x03000000:NameValue):X-Remote-Host = '192.168.58.1' (CHARACTER)
    (0x03000000:NameValue):X-Scheme = 'http' (CHARACTER)
  )
  (0x01000000:Name):BLOB = ( ['none' : 0x7f658cc0d250]
    (0x03000000:NameValue):UnknownParserName = " (CHARACTER)
    (0x03000000:NameValue):BLOB = X'616161' (BLOB)
  )
)
```

考慮点

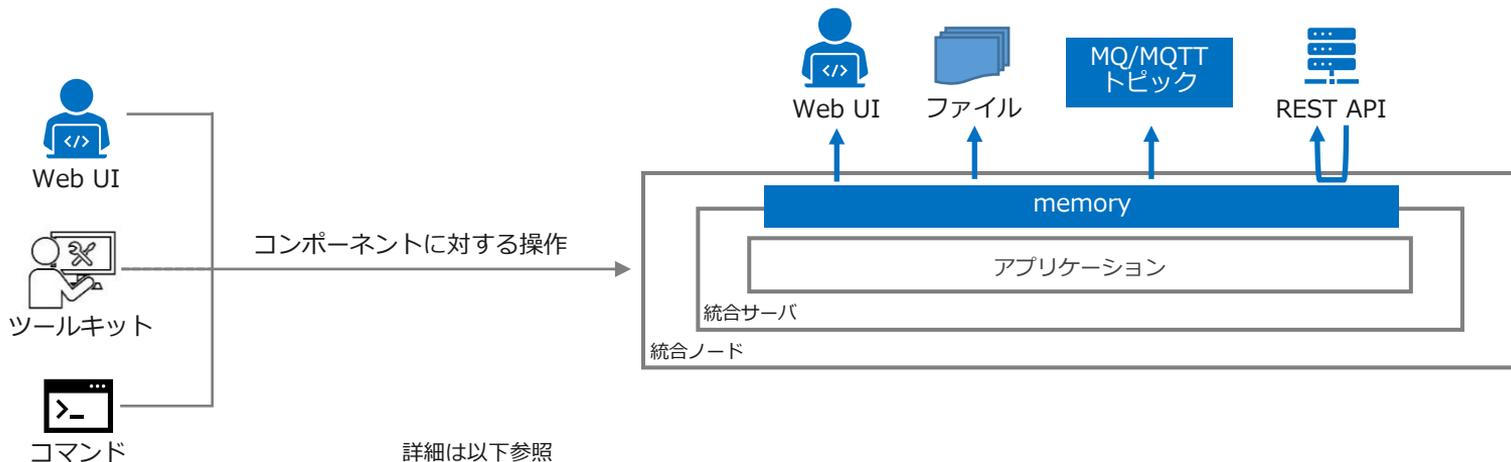
■ 任意のファイルに出力する場合(Destination=File)の考慮点

- ◆ 複数行の内容を複数のフローやインスタンスから同一のファイルに出力すると出力内容が入れ子になるケースがある
 - ・ 出力内容の途中で別のフロー/インスタンスの内容が出力される
- ◆ ファイルのサイズや数、ローテーションを指定できない
 - ・ 指定したファイルに出力し続けるため、別途ファイルの退避、クリーンナップ等の運用が必要

管理ログ

機能概要

- 稼働環境のコンポーネントに対する各種操作のログを出力
 - ◆ 統合ノード/統合サーバの起動/停止やアプリケーションのデプロイ/起動/停止、WebUIのログイン/操作など
 - ◆ 操作によって生成されるBIPメッセージ単位でログに出力
 - ◆ ログには、BIPメッセージやタイムスタンプ、実行ユーザの情報等が含まれる
- Web UIのAdmin logタブで表示
 - ◆ データは、統合ノードまたは統合サーバ内のメモリ上に保持
 - ◆ 管理REST APIで参照することも可能
- ファイルやMQ/MQTTトピックへの出力も可能
 - ◆ JSON形式で出力



詳細は以下参照

<https://www.ibm.com/docs/en/app-connect/12.0?topic=enterprise-viewing-administration-activity-in-admin-log>

管理ログとして出力される操作

■ 管理ログが出力される主な操作(*1)

コンポーネント	操作
統合ノード	作成(*2)、起動、停止、削除(*3) プロパティ設定(mqsichangeproperties)
統合サーバ	作成、起動、停止、削除 プロパティ設定(mqsichangeproperties)
アプリケーション/メッセージフロー	デプロイ、起動、停止、削除 ※権限エラーによる操作失敗も出力される
Vault(*4)	作成(mqsivault) クレデンシャルの作成、変更、削除(mqsicredentials) ※不正なVaultkey指定による操作失敗は出力されない
資格情報(*4)	作成、変更、削除(mqsisetdbparms)
WebUI	ログイン(成功、失敗)、ログアウト
Webユーザー(*4)	作成、変更、削除(mqsiwebuseradminコマンド)

*1 実機環境で確認できたものを記載

*2 統合ノード作成時のログはファイルにのみ出力される（管理ログのファイル出力はデフォルトでOFFになっているが、この操作は出力される）

*3 統合ノードの停止、削除に関するログはWeb UIでは確認できない（停止・削除後はWeb UIでアクセスできないため）

*4 統合ノード/統合サーバ停止時に実行可能な操作（Vault/資格情報/Webユーザーに関する操作）はファイルにのみ出力される

出力情報

■ 管理ログに含まれる情報

情報	説明
BIPメッセージ番号	BIPメッセージの番号、Severity
BIPメッセージ内容	BIPメッセージの内容
タイムスタンプ	操作実行時のタイムスタンプ
実行ユーザー	操作を実行したユーザー
権限ロール	実行ユーザーに紐づけられたロール (ロールが紐づけられている場合のみ)
タグ	操作に関する各種情報 (操作によって内容は異なる)

管理ログの出力設定

■ node.conf.yamlまたはserver.conf.yamlで設定

- ◆ デフォルトで有効
 - 無効化する場合は、enabledプロパティをfalseに設定

統合ノード構成で管理ログを有効化する場合のnode.conf.yaml設定例

```
AdminLog:
  enabled: true
# Control logging admin log messages. Set to true or false, default is true.
# When enabled the maximum amount of disk space required for admin log files is
# fileLogRetentionPeriod * fileLogCountDaily * fileLogSize * (1 + number of integration servers
# managed by the integration node)

#logServerStartStopDetail: true           # Control logging of detailed application and message flow messages during
integration server startup and shutdown, default is true.

#addNodeAndServerNameTags: false         # Control addition of integration node and integration server name tags to all admin
log messages, default is false.
```

詳細は以下参照

<https://www.ibm.com/docs/en/app-connect/12.0?topic=log-configuring-administration-logging>

管理ログのファイル出力設定

■ 出力設定

- ◆ node.conf.yamlまたはserver.conf.yamlで設定
 - デフォルトは無効
 - 有効化する場合は、fileLogプロパティをtrueに設定

統合ノード構成でファイル出力を有効化する場合のnode.conf.yaml設定例

```
AdminLog:
  fileLog: true           # Control writing admin log messages to file. Set to true or false, default is false.
  #fileLogRetentionPeriod: 30 # Sets the number of days to record admin log.
                              # After this, old files are deleted as new ones are created. Default is 30 days.
  #fileLogCountDaily: 10   # Maximum number of admin log files to write per day, default is 10 per day.
  #fileLogSize: 100        # Maximum size in MB for each admin log file. Maximum size is 2000MB, default size is 100MB.
```

■ 出力先

- ◆ 以下のディレクトリ/ファイルに出力
 - 統合ノード構成の場合
 - ✓ 統合ノード・レベル
 - workDir/common/adminLog/<統合ノード>/<統合ノード>.adminLog.log.yyyymmdd
 - ✓ 統合サーバ・レベル
 - workDir/common/adminLog/<統合ノード>/<統合サーバ>/<統合ノード>.<統合サーバ>.adminLog.log.yyyymmdd
 - 独立統合サーバ構成の場合
 - ✓ workDir/adminLog
 - ✓ コンソールに出力することも可能

詳細は以下参照

<https://www.ibm.com/docs/en/app-connect/12.0?topic=log-writing-admin-entries-file>

管理ログのMQ/MQTT出力設定

■ MQ/MQTTの出力設定

- ◆ node.conf.yamlまたはserver.conf.yamlのEventsセクション/AdminEventsサブセクションで設定
 - 出力先として使用するMQもしくはMQTTのenabledプロパティをtrueに設定
 - 関連付けたキューマネージャーもしくは組み込みMQTTブローカー以外に出力する場合は、事前に作成/デプロイしたMQEndpointポリシーまたはMQTTPublishポリシーをpolicyプロパティに設定

ランタイムに関連付けたMQ(デフォルト・キューマネージャー)へ出力する場合のnode.conf.yaml設定例

```
Events:
  AdminEvents: # Administration events - AdminLog
    MQ:
      enabled: true           # Set true or false, default false
      #policy: ""            # Specify a {policy project};policy if not using 'defaultQueueManager'
      #format: ""           # Set string or none
      #publishRetryInterval: 0 # Set the retry interval (in milliseconds), to pause all publications and retry,
                              # when publication failures are causing serious delay to the transaction.
    MQTT:
      enabled: true         # Set true or false, default false
      #policy: ""          # Specify a {policy project};policy if not using 'defaultQueueManager'
      #publishRetryInterval: 0 # Set the retry interval (in milliseconds), to pause all publications and retry,
                              # when publication failures are causing serious delay to the transaction.
```

■ MQ/MQTTの出力先トピック

- MQ : `$SYS/Broker/<nodeName>/AdminLog/JSON/<serverName>`
- MQTT : `IBM/IntegrationBus/<nodeName>/AdminLog/JSON/<serverName>`

詳細は以下参照

<https://www.ibm.com/docs/en/app-connect/12.0?topic=log-publishing-admin-logs-over-mqtt-mq>

Web UIによる管理ログの表示

- Web UIのAdmin logタブで表示
 - ◆ Web UIからはデフォルトで表示可能
 - ◆ 統合ノード・レベルと統合サーバー・レベルで表示
 - ◆ Web UIでの管理ログは、統合ノードの停止によりリセットされる

統合ノード・レベルの管理ログの表示例

IBM App Connect

ACE1210NODE

Servers Properties Flow statistics Data **Admin log**

Last updated: a minute ago ⓘ ↻

Timestamp <small>local</small>	Message	Message text	User	Authorized role	Tags
2023-12-22 15:21:39.332	BIP20010I	ユーザー 'aceadm' はログインしました。	aceadm		REQUEST_ID:20231222062139331000- adminui-0 ROLES:<no-auth-role>
2023-12-22 15:20:35.153	BIP20002I	統合ノード 'ACE1210NODE' がバージョン '12.0.10.0' で開始されました。			VERSION:12.0.10.0

Entries per page 25 ▼ 1-2 of 2 items 1 ▼ of 1 page ◀ ▶

ファイル/MQ/MQTTの出力内容

■ ファイル、MQ/MQTTトピックへはJSON形式で出力

統合サーバ停止時の出力例（複数のログが出力されるが、1件のみを表示）

```
{
  "message": {
    "number": 20020,
    "severityCode": "I",
    "source": "BIPmsgs",
    "timestamp": "2023-12-25T04:26:51.616304Z",
    "localTimestamp": "2023-12-25T13:26:51.616304+09:00"
  },
  "text": "メソッド 'POST' とパス '/apiv2/servers/ACE1210SERVER/stop' で REST 要求を受け取りました。 ",
  "detailedText": "",
  "inserts": [
    {"text": "POST"},
    {"text": "/apiv2/servers/ACE1210SERVER/stop"}
  ],
  "tags": [
    {"name": "REQUEST_ID",
     "value": "20231225042651611241-toolkit-0"},
    {"name": "USER",
     "value": "aceadmin"},
    {"name": "ROLES",
     "value": "adminRole"},
    {"name": "AUTHORIZED_ROLE",
     "value": "adminRole"},
    {"name": "HTTP_METHOD",
     "value": "POST"},
    {"name": "PATH",
     "value": "/apiv2/servers/ACE1210SERVER/stop"}
  ]
}
```

※JSONを整形して表示（実際には上記内容が1行/1メッセージとして出力される）