

バージョン 6.1.0



WebSphere Process Server のモニター



バージョン 6.1.0



**WebSphere Process Server のモニター**

お願い

本書に記載されている情報をご使用になる前に、本書末尾の『特記事項』セクションに記載されている情報をお読みください。

本書は、WebSphere Process Server for Multiplatforms (製品番号 5724-L01) バージョン 6、リリース 1、モディフィケーション 0、および新しい版で明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

IBM 発行のマニュアルに関する情報のページ

<http://www.ibm.com/jp/manuals/>

こちらから、日本語版および英語版のオンライン・ライブラリーをご利用いただけます。また、マニュアルに関するご意見やご感想を、上記ページよりお送りください。今後の参考にさせていただきます。

(URL は、変更になる場合があります)

お客様の環境によっては、資料中の円記号がバックスラッシュと表示されたり、バックスラッシュが円記号と表示されたりする場合があります。

原 典： WebSphere® Process Server for Multiplatforms  
Version 6.1.0  
Monitoring WebSphere Process Server

発 行： 日本アイ・ビー・エム株式会社

担 当： ナショナル・ランゲージ・サポート

# 目次

<b>第 1 章 サービス・コンポーネント・モニターの概説</b> . . . . .	<b>1</b>
Common Event Infrastructure . . . . .	1
Common Base Event モデル . . . . .	3
モニターを使用する目的 . . . . .	4
モニター対象 . . . . .	5
モニターを使用可能にする方法 . . . . .	6
<b>第 2 章 サービス・コンポーネント・モニターの有効設定と構成</b> . . . . .	<b>9</b>
パフォーマンスのモニター . . . . .	9
パフォーマンス・モニター・インフラストラクチャー統計 . . . . .	10
Service Component Architecture のアプリケーション応答測定統計 . . . . .	15
サービス・コンポーネント・イベントのモニター . . . . .	29
ビジネス・プロセス・イベントとヒューマン・タスク・イベントのモニターの使用可能化 . . . . .	30
サービス・コンポーネント・イベントのログギングの構成 . . . . .	31
Common Event Infrastructure サーバーでのサービス・コンポーネントのモニター . . . . .	38
セッション・モニター . . . . .	42
<b>第 3 章 モニター対象イベントの表示</b> . . . . .	<b>45</b>
Tivoli Performance Viewer によるパフォーマンス・メトリックの表示 . . . . .	45

サービス・コンポーネント・イベント・ログ・ファイルの表示と解釈 . . . . .	47
Common Base Event ブラウザーでのイベントの表示	49
表示するイベントの指定 . . . . .	49
イベント・ブラウザーから戻されるイベントの処理 . . . . .	50
Common Base Event ブラウザーのトラブルシューティング . . . . .	51
<b>第 4 章 イベント・カタログ</b> . . . . .	<b>53</b>
Common Base Event の標準エレメント . . . . .	53
イベント内のビジネス・オブジェクト . . . . .	54
Business Process Choreographer イベント . . . . .	55
ビジネス・プロセス・イベントのモニター . . . . .	55
ヒューマン・タスク・イベントのモニター . . . . .	79
プロセス・サーバーのイベント . . . . .	88
リソース・アダプター・イベント . . . . .	88
ビジネス・ルール・イベント . . . . .	90
ビジネス状態マシン・イベント . . . . .	90
マップ・イベント . . . . .	92
メディエーション・イベント . . . . .	92
リカバリー・イベント . . . . .	93
Service Component Architecture イベント . . . . .	94
セレクター・イベント . . . . .	95
<b>特記事項</b> . . . . .	<b>97</b>



---

## 第 1 章 サービス・コンポーネント・モニターの概説

プロセス・サーバーでサービス・コンポーネントをモニターする理由、モニターするサービス・コンポーネント内のイベント・ポイントの選択、およびシステムのモニターの構成方法などの概念的な概説。

WebSphere® Process Server には、サービス・コンポーネントをモニターする機能があり、パフォーマンスの調整や問題判別などのシステム管理機能を支援します。さらに、こうした従来の機能だけでなく、必ずしも情報技術の専門家ではないユーザーに対して、システムにデプロイされたアプリケーション内のサービス・コンポーネントの処理を継続的にモニターする機能も提供します。相互接続されたコンポーネントの全体的な処理フローを監視することにより、システムで本来行われるべき処理が行われていることを確認することができます。

WebSphere Process Server は、WebSphere Application Server インストール済み環境の上部で稼働します。その結果、システム・パフォーマンスのモニターとトラブルシューティングで、アプリケーション・サーバー・インフラストラクチャーの機能の多くを使用します。また、プロセス・サーバーのサービス・コンポーネントをモニターするために特別に設計された追加の機能も組み込まれています。WebSphere Process Server インフォメーション・センターのこのセクションでは、プロセス・サーバー固有のサービス・コンポーネントをモニターする方法を中心に説明します。このセクションは、WebSphere Application Server バージョン 6.1 インフォメーション・センターに記されたモニターおよびトラブルシューティングについてのトピックを補足することが目的です。そのため、製品を組み合わせる場合のほかのモニター機能の詳細については、このインフォメーション・センターを参照してください。

---

### Common Event Infrastructure

Common Event Infrastructure は、基本的なイベント管理サービスを必要とするアプリケーションにそれらのサービスを提供することを目的とした組み込み可能なテクノロジーです。

このイベント・インフラストラクチャーは、複数の異機種混合ソースからのロー・イベントを統合および永続化して、それらのイベントをイベント・コンシューマーに配布するための統合ポイントとして機能します。イベントは、Common Base Event モデル (イベントの構造を定義する標準の XML ベースの形式) を使用して表現されます。詳しくは、Common Base Event モデルについてのサブトピックを参照してください。

この共通インフラストラクチャーを使用すると、互いに密結合していないさまざまな製品がそれぞれのイベント管理を統合できます。これにより、エンタープライズ・リソースのエンドツーエンドでの表示、およびドメインの境界を越えるイベントの相関関係の設定ができます。例えば、ネットワーク・モニター・アプリケーションが生成したイベントを、セキュリティー・アプリケーションが生成したイベン

トに相関させることができます。各製品が独自の方法でイベントを管理している場合、こうした相関を行うことが難しくなります。

Common Event Infrastructure は、イベントを生成、伝搬、永続化、および利用するための機能を備えています。イベント自体を定義するわけではありません。イベント・タイプ、イベント・グループ、フィルター操作、および相関は、アプリケーションの開発者と管理者が定義します。

## Common Event Infrastructure コンポーネント

Common Event Infrastructure は、主に以下のコンポーネントから構成されます。

### Common Base Event

Common Base Event コンポーネントは、イベントの作成とそのプロパティ・データへのアクセスをサポートします。イベント・ソースは、Common Base Event API を使用して、Common Base Event モデルに準拠した新しいイベントを作成します。イベント・コンシューマーは、これらの API を使用して、受け取ったイベントからプロパティ・データを読み取ります。また、アプリケーションは、イベントの XML テキスト形式への変換、およびその逆ができるため、その他のツールとの間で情報を交換することができます。Common Base Event コンポーネントは、Eclipse Test and Performance Tools Platform (TPTP) の一部です。

### エミッター

エミッター・コンポーネントは、イベントの送信をサポートします。イベント・ソースでイベントが作成され、そのイベントにデータが設定されると、イベント・ソースは、エミッターにイベントを送信します。エミッターは、必要に応じて自動コンテンツ完了を実行してからイベントを検証し、イベントが Common Base Event 仕様に準拠していることを確認します。また、そのイベントと構成可能なフィルター基準との比較も行います。イベントが有効であり、フィルター基準を通過する場合、エミッターはそのイベントをイベント・サービスに送信します。エミッターは、イベントをイベント・サービスに同期的にでも (Enterprise JavaBeans™ 呼び出しを使用) 非同期でも (Java™ Message Service キューを使用) 送信できます。

### イベント・サービス

イベント・サービスは、イベント・ソースとイベント・コンシューマーとの間のコンジットです。イベント・サービスは、イベント・ソースがエミッターに送信したイベントを受け取ります。イベント・サービスは、イベントを永続データ・ストアに保管したあと、サブスクライブしているイベント・コンシューマーにそれらのイベントを非同期的に配布します。また、イベント・サービスでは、履歴イベントを永続ストアから同期的に照会することができます。

### イベント・カタログ

イベント・カタログは、イベント・メタデータのリポジトリです。アプリケーションは、イベントのクラスとそれに許可される内容に関する情報を取得するときにイベント・カタログを使用します。

また、Common Event Infrastructure を使用するアプリケーションまたはソリューションには、以下のコンポーネントも含まれる場合があります (これらは、インフラストラクチャー自体に含まれるわけではありません)。



### イベント・ソース

イベント・ソースは、エミッターを使用してイベントをイベント・サービスに送信するすべてのアプリケーションです。

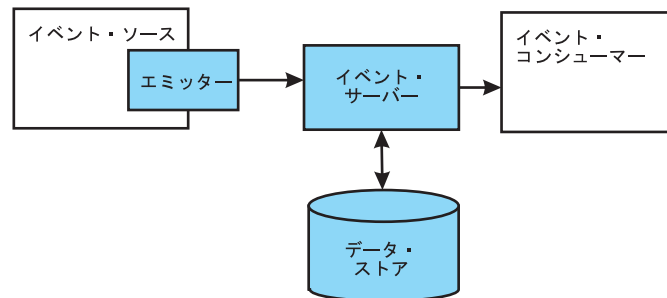
### イベント・コンシューマー

イベント・コンシューマーは、イベント・サービスからイベントを受け取るすべてのアプリケーションです。

### イベント・カタログ・アプリケーション

イベント・カタログ・アプリケーションは、イベント・カタログのイベント・メタデータを保管または取得するすべてのアプリケーションです。このアプリケーションは、管理ツールまたは開発ツールである場合があります。イベント・ソースまたはイベント・コンシューマーである場合もあります。

Common Event Infrastructure を使用した、イベント・ソースからイベント・コンシューマーへの一般的なイベント・フローを以下の図に示します。



## Common Base Event モデル

Common Base Event モデルは、イベントの共通の表現を定義する規格であり、企業の管理およびビジネス・アプリケーションで使用することを目的としています。この規格は、IBM® Autonomic Computing Architecture Board が開発したものであり、ロギング、トレース、管理、およびビジネス・イベントを、共通の XML ベースの形式を使用してエンコードすることができます。これにより、さまざまなアプリケーションから出される各種のイベントを相互に関連付けることが可能です。Common Base Event モデルは、IBM Autonomic Computing Toolkit の一部です。詳しくは、<http://www-06.ibm.com/jp/autonomic/> を参照してください。

現在、Common Event Infrastructure は、バージョン 1.0.1 の仕様をサポートしています。

Common Base Event モデルの背後にある基本的な概念は、シチュエーションです。シチュエーションは、サーバーのシャットダウン、ディスク・ドライブの障害、ユーザー・ログインの失敗など、コンピューターのインフラストラクチャーのあらゆる場所で発生するすべての事象を表します。Common Base Event モデルは、発生する可能性のあるシチュエーション (StartSituation や CreateSituation など) の大部分に対応する、一連の標準的なシチュエーション・タイプを定義します。

イベント は、シチュエーションに関する情報を報告する、構造化された通知です。イベントは、以下の 3 種類の情報を報告します。

- シチュエーション自体 (発生した内容)

- 影響を受けるコンポーネント (シャットダウンしたサーバーなど) の ID
- シチュエーションを報告しているコンポーネントの ID (影響を受けるコンポーネントと同じ場合がある)

Common Base Event 仕様では、イベントが XML エlementとして定義されます。この XML Elementには、上記の 3 種類の情報をすべて提供するプロパティが含まれます。これらのプロパティは、ルート・Element `CommonBaseEvent` の属性およびサブElementとしてエンコードされます。

Common Base Event 形式は、拡張できます。イベントには、標準のイベント・プロパティのほかにも、拡張データ・Elementを含めることができます。これは、アプリケーション固有のElementであり、シチュエーションに関するあらゆる種類の情報を含めることができます。`extensionName` 属性は、イベントにオプションの分類名 (イベント・クラス) のラベルを付けます。これにより、どのような種類の拡張データ・Elementが予期されるのかをアプリケーションが知ることができます。イベント・カタログは、これらのイベント・クラスとそれに許可される内容を記述したイベント定義を保管します。

Common Base Event 形式について詳しくは、IBM Autonomic Computing Toolkit に含まれる仕様の文書と XSD スキーマを参照してください。

---

## モニターを使用する目的

WebSphere Process Server 内でサービス・コンポーネントをモニターする目的は、パフォーマンスを評価し、問題をトラブルシューティングして、システムにデプロイされているアプリケーションを構成するサービス・コンポーネントの全体的な処理の進捗を評価することです。

サービス・コンポーネントは、WebSphere Process Server に組み込まれた不可欠な機能です。この機能を使用して、企業で使用されるプロセスをミラーリングするアプリケーションを、システム上に容易に作成およびデプロイすることができます。したがって、プロセス・サーバーによって完了しようとするタスクを管理するには、それらのサービス・コンポーネントを効果的にモニターすることが重要です。プロセス・サーバーでサービス・コンポーネントをモニターしなければならない理由は主に 3 つあります。

### 問題判別

WebSphere Process Server の基盤となる WebSphere Application Server のロギング機能とトレース機能を使用して、特定のエラーを診断することができます。例えば、特定のアプリケーションが期待される結果を生み出していない場合は、アプリケーションを構成するサービス・コンポーネントの処理をモニターするようにロガーをセットアップできます。ログ出力をファイルにパブリッシュし、そのファイルを調べて問題の原因を特定することができます。トラブルシューティングは、システムのハードウェアとソフトウェアの保守に携わる管理者やその他の担当者にとって重要なタスクです。

### パフォーマンス調整

大半のプロセス・サーバー固有サービス・コンポーネントが作成する特定のパフォーマンス統計をモニターすることができます。この情報を使用して、システムの正常性を維持および調整し、アプリケーションが最適かつ効率的であるように調整されていることを確認します。また、1 つ以上のサービス

が低レベルで実行されている状態を発見できます。この場合、システムに別の問題が存在する可能性があります。問題判別と同様、一般的にパフォーマンス調整は IT スペシャリストが実行する作業です。

### サービス・コンポーネントの処理の評価

問題判別とパフォーマンス調整は、特定の問題を解決するために短期的に実行するタスクです。システムにデプロイされているアプリケーションに組み込まれたサービス・コンポーネントを継続的にモニターするようにプロセス・サーバーをセットアップすることもできます。このタイプのサービス・コンポーネント・モニターは、設計、インプリメント、およびプロセスが設計目標を達成するようにする責任がある担当者にとって重要です。そうした担当者は、必ずしも IT 分野の熟練したスペシャリストであるとは限りません。

---

## モニター対象

処理中にサービス・コンポーネント・イベントが到達する特定のポイントを選択することによって、WebSphere Process Server のサービス・コンポーネント・イベントをモニターできます。各サービス・コンポーネントは、それらのイベント・ポイントを定義します。イベント・ポイントは、その所定のポイントでアプリケーションが処理を行ったときに、イベントを「送出」します。また、サービス・コンポーネント・イベントのパフォーマンス統計をモニターすることもできます。

サービス・コンポーネントで実行するモニターのタイプ (問題判別、パフォーマンス調整、またはプロセス・モニター) に関係なく、実際にはこれらのコンポーネントの処理中に到達する特定のポイントをモニターしています。このポイントは、イベント・ポイント と呼ばれ、これらのポイントをモニター対象として選択します。各イベント・ポイントは、サービス・コンポーネントの種類タグ、オプションの要素種類 (サービス・コンポーネント・タイプの固有機能)、およびイベントの性質 をカプセル化します。これらのすべての因子により、モニターによって送出されるイベントのタイプが決定します。

イベント性質は、サービス・コンポーネントの処理中にイベントを生成 (または「送出」) するために必要な状態について説明します。これらの性質は基本的に、モニターの対象として選択するサービス・コンポーネントの論理構造におけるキーポイントです。サービス・コンポーネント・イベントの最も一般的な性質は ENTRY、EXIT、および FAILURE ですが、特定のコンポーネントやエレメントによっては他にも多くの性質があります。指定されたサービス・コンポーネントを持つアプリケーションが続いて呼び出される場合には常に、サービス・コンポーネントの処理がイベント性質に対応するポイントを経由するたびにイベントが送出されます。

サービス・コンポーネント種類のイベントの定義方法の例として、MAP サービス・コンポーネント種類を挙げます。MAP サービス・コンポーネント種類は ENTRY、EXIT、および FAILURE の性質を持つイベントを直接送出することができます。また、MAP コンポーネント種類内の特定タイプの機能を定義する「変換」という要素種類も含まれています。この要素種類も、ENTRY、EXIT、および FAILURE の性質を持つイベントを送出します。その結果、MAP サービス・コンポーネント種類は、指定する要素と性質の組み合わせに応じて、異なるイベン

トを 6 つまで送出することができます。イベント・カタログには、すべてのサービス・コンポーネントとそのエレメント、およびそのイベント性質のリストがあります。

モニターはアプリケーション処理の最上位にある機能の分離されたレイヤーであるため、サービス・コンポーネントの処理に干渉することはありません。モニターは、指定されたイベント・ポイントでアクティビティを検出した場合のみサービス・コンポーネント処理に関係します。この状態が発生すると、モニターによってイベントが送出されます。モニターは実行するモニターのタイプに基づいて、イベントの送付先およびイベントに含まれるデータ (以下参照) を判断します。

#### パフォーマンス・メトリック

パフォーマンス・メトリックを収集するためにサービス・コンポーネントをモニターしている場合は、軽負荷イベントがパフォーマンス・モニター・インフラストラクチャーに送出されます。プロセス・サーバー固有のサーバー・コンポーネント用に生成される 3 種類のパフォーマンス統計の中から、1 つ以上のパフォーマンス統計をモニター対象として選択できます。

- 各 EXIT イベント性質のカウンター: 成功した計算をカウント。
- 各 FAILURE イベント性質のカウンター: 失敗した計算をカウント。
- 対応する ENTRY イベントと EXIT イベント (同期計算のみ) の間の処理所要時間。

また、アプリケーション応答測定 (ARM) 統計を使用することによって、Service Component Architecture (SCA) レベルでアプリケーションのパフォーマンスをモニターすることもできます。これらの指標を使用すると、アプリケーション内で、他のサービス・コンポーネント・イベントで使用可能な方法よりも詳細なレベルで、アプリケーションをモニターすることができます。この統計で SCA が使用されている場合、この統計を使用して、初期アプリケーション呼び出しとサービス応答との間の多数の異なるポイントをモニターできます。

#### ビジネス・オブジェクトを伴うサービス・コンポーネント・イベント

サービス・コンポーネントの指定されたイベント・ポイントでモニターによって送出されるイベントからデータを収集する場合、プロセス・サーバーがイベントを生成し、そのデータが Common Base Event 形式でエンコードされるように構成します。各サービス・コンポーネント・イベント内で収集するビジネス・オブジェクト・データの詳細レベルを指定できます。これらのイベントは、ロガーか Common Event Infrastructure (CEI) バスのどちらかにパブリッシュできます。このパブリッシュでは、出力は特別に構成された CEI サーバー・データベースに送信されます。

---

## モニターを使用可能にする方法

プロセス・サーバー上で行うモニターのタイプに応じて、モニター用サービス・コンポーネント・イベント・ポイントを指定する方法はいくつかあります。

#### パフォーマンス統計

Performance Monitoring Infrastructure (PMI) 統計の場合、管理コンソールを使用して、モニターする特定のイベント・ポイントとそれに関連するパフォーマンス測定を指定します。モニター・サービス・コンポーネント・パフォ

パフォーマンスを開始した後、生成された統計は一定の間隔で Tivoli® Performance Viewer にパブリッシュされます。このビューアーを使用すると、システムで結果が生成されてから見ることができます。また、オプションで、その結果をファイルにログ記録して、あとで同じビューアー内で表示および分析することもできます。

アプリケーション応答測定 (ARM) 統計の場合、管理コンソールの「要求メトリック」セクションを使用して、モニターする統計を指定します。要求メトリックを使用した作業の詳細については、WebSphere Application Server バージョン 6.1 のインフォメーション・センターを参照してください。

### 問題判別およびビジネス・プロセス・モニターを行う Common Base Event

アプリケーションの作成時、稼働中のサーバーにアプリケーションをデプロイした後にサービス・コンポーネント・イベント・ポイントが継続的にモニターされるように指定できます。また、イベントの詳細レベルも同時に指定できます。また、アプリケーションがデプロイされた後にモニターするイベント・ポイントと、プロセス・サーバーで少なくとも1度は呼び出されるイベントを選択することもできます。どちらの場合も、モニターによって生成されるイベントは、Common Event Infrastructure (CEI) バスを通して送出されます。これらのイベントは、ログ・ファイルにパブリッシュすることも、構成済み CEI サーバーのデータベースにパブリッシュすることもできます。WebSphere Process Server では、問題判別およびビジネス・プロセス・モニターで、次の 2 つのタイプを指定して Common Base Event を使用可能化することができます。

**静的** WebSphere Integration Developer ツールを使用して、アプリケーション内の特定のイベント・ポイントとその詳細レベルに、モニター用のタグを付けることができます。この選択内容は、継続的にモニターするイベント・ポイント、およびプロセス・サーバー・アプリケーションと共に配布およびデプロイされる .mon 拡張子を持つファイルに保管するイベント・ポイントを示します。プロセス・サーバーで CEI サーバーを使用するように構成すると、指定されたサービスが起動されるときにはいつでも、モニター機能が CEI サーバーへのサービス・コンポーネント・イベントの送出を開始します。プロセス・サーバーにアプリケーションがデプロイされている限り、.mon ファイルで指定されているサービス・コンポーネント・イベント・ポイントは、アプリケーションが停止するまで常にモニターされます。実行中のアプリケーションでモニターされる追加イベントを指定したり、すでにモニターされているイベント・ポイントの詳細レベルを上げたりすることもできます。ただし、アプリケーションがアクティブである限り、デプロイ済みアプリケーションの .mon ファイルで指定されているモニター対象イベント・ポイントを停止したり、その詳細レベルを低くしたりすることはできません。

**動的** アプリケーションの処理中に、サーバーをシャットダウンしないで追加イベント・ポイントをモニターする必要がある場合は、動的モニターを使用できます。管理コンソールを使用してモニターするサービス・コンポーネント・イベント・ポイントを指定し、Common Base Event に含まれる有効搭載量の詳細レベルを設定できます。プロセス・サーバーの始動後に、処理済みサービス・コンポーネント



が到達したイベント・ポイントのリストがコンパイルされます。このリストから、モニターする個別のイベント・ポイントまたはイベント・ポイントのグループをロガーまたは CEI サーバー・データベースに送信されるサービス・コンポーネント・イベントと共に選択します。

動的使用可能化の主な目的は、ログにパブリッシュされる関連サービス・コンポーネント・イベントを作成し、サービスでの問題判別を実行できるようにすることです。サービス・コンポーネント・イベントのサイズは大きくなる可能性があります (要求されるデータ量に応じて異なります)。また、イベントを CEI サーバーに送信する場合、データベース・リソースに負荷がかかることがあります。そのため、イベントのビジネス・データを読み取る必要がある場合、または読み取る必要はないがイベントのデータベース・レコードを保持する必要がある場合のみ、モニター済みイベントを CEI サーバーに動的にパブリッシュしてください。ただし、特定のセッションをモニターしている場合、CEI サーバー・データベースを使用してそのセッションに関連するサービス・コンポーネント・イベントにアクセスする必要があります。

---

## 第 2 章 サービス・コンポーネント・モニターの有効設定と構成

プロセス・サーバーでサービス・コンポーネントをモニターするには、まずモニター機能を使用できるように設定する必要があります。次に、モニターするイベント、イベントから収集する情報、および結果のパブリッシュで使用する方法を指定する必要があります。

---

### パフォーマンスのモニター

サービス・コンポーネント・イベント・ポイントに対してパフォーマンス測定を実行できます。パフォーマンス測定は、パフォーマンス・モニター・インフラストラクチャーを使用して処理します。プロセス・サーバーでサービス・コンポーネント・イベント・ポイントからパフォーマンス・メトリックを収集するように構成します。また、Service Component Architecture 固有のパフォーマンス統計をアプリケーションのサーバー呼び出しから直接収集することもできます。

WebSphere Process Server サービス・コンポーネントを調整して効率を最適化する場合も、ローパフォーマンスを診断する場合も、パフォーマンスの観点から、さまざまなランタイム・リソースおよびアプリケーション・リソースの動作を理解しておくことは重要です。パフォーマンス・モニター・インフラストラクチャー (PMI) では、ランタイム・リソースとアプリケーション・リソースの動作を説明する包括的なデータを提供します。PMI データを使用して、アプリケーション・サーバーにおけるパフォーマンスのボトルネックを識別し、修正できます。また、アプリケーション・サーバーの正常性をモニターする目的で PMI データを使用することもできます。

PMI は、ベースとなっている WebSphere Application Server インストールに組み込まれています。このセクションでは、WebSphere Process Server に固有のサービス・コンポーネントに関連するパフォーマンス・モニターについての補足的な情報のみを提供します。そのため、製品全体のほかの部分で PMI を使用方法については、WebSphere Application Server の資料の情報を参照してください。

PMI によってモニター可能な WebSphere Process Server に固有のサービス・コンポーネント・イベント・ポイントは、通常 ENTRY、EXIT、および FAILURE というイベント性質を持っています。このパターンに従って定義されていないイベント・リソースはサポートされません。サポートされるイベントには、測定可能なパフォーマンス統計の 3 つのタイプがあります。

- 成功した呼び出し。
- 失敗した呼び出し。
- イベント完了までの経過時間。

また、Application Response Measurement (ARM) 統計を使用することによって、アプリケーションのサービス呼び出しから取得したパフォーマンス統計をモニターすることもできます。これらの統計では、エンタープライズ・アプリケーションを構成するプロセス・サーバーのサービス・コンポーネント・イベントの基礎となる実

際のランタイム・プロセスを測定します。これらの統計を使用して、アプリケーション処理のさまざまなパフォーマンス測定を取得することができます。

## パフォーマンス・モニター・インフラストラクチャー統計

Performance Monitoring Infrastructure を使用して、3 つのタイプのパフォーマンス統計をモニターできます。つまり、呼び出しの成功回数、失敗回数、およびイベント完了までの経過時間です。これらの統計は、ENTRY、EXIT、および FAILURE のタイプの性質を持つイベントにのみ使用可能です。

### 管理コンソールを使用した PMI の使用可能化

パフォーマンス・データをモニターするには、まずサーバーでパフォーマンス・モニター・インフラストラクチャーを使用可能にする必要があります。

#### このタスクについて

パフォーマンス・モニター・インフラストラクチャー (PMI) は、管理コンソールを使用して使用可能にすることができます。

#### プロシージャ

1. 管理コンソールを開きます。
2. コンソールのナビゲーション・ツリーで、「サーバー」>「アプリケーション・サーバー」をクリックします。
3. *server\_name* をクリックします。

注: 管理コンソールから、「モニターおよびチューニング」>「パフォーマンス・モニター・インフラストラクチャー (PMI)」>「*server\_name*」をクリックして同じパネルを開きます。

4. 「構成」タブをクリックします。
5. 「パフォーマンス・モニター・インフラストラクチャー (PMI) を使用可能にする (Enable Performance Monitoring Infrastructure (PMI))」チェック・ボックスを選択します。
6. オプション: 「順次カウンター更新を使用」チェック・ボックスを選択して、正確な統計の更新を使用可能に設定します。
7. サーバー名のリンクをクリックして、「サーバー PMI 構成 (server PMI configuration)」ページに戻ります。
8. 「適用」または「OK」をクリックします。
9. 「保管」をクリックします。
10. サーバーを再始動します。

#### 次のタスク

変更内容は、サーバーを再始動するまで有効になりません。

### イベント・パフォーマンス統計

パフォーマンス・モニター統計は、大部分のサーバー・イベントについて使用可能です。呼び出し要求 (良および不良) のカウントとイベントの完了までに要した時間をモニターできます。



Performance Monitoring Infrastructure (PMI) を使用して、特定のサーバー・イベントにより生成される、以下の表に示す 3 つのパフォーマンス統計をモニターできます。

表 1. イベントの PMI 統計

統計名	タイプ	説明
<b>BadRequests</b>	カウンター	イベントの呼び出しが失敗した数
<b>GoodRequests</b>	カウンター	イベントの呼び出しが成功した数
<b>ResponseTime</b>	タイマー	イベント完了までの経過時間。

以下の統計は、ENTRY、EXIT、および FAILURE の性質を備えたエレメントを持つサービス・コンポーネント・イベントだけに限られます。各統計は、アプリケーションにおいて、所定のサーバー・イベント・タイプの 1 イベントごとに作成されます。すべてのパフォーマンス測定は、カウンター (指定されたイベント・ポイントの実行累積数) かタイマー (2 つのイベント・ポイント実行間隔の期間 (単位: ミリ秒)) のどちらかです。モニターできる各イベント種類 (および関連するエレメント) を以下にリストします。

表 2. イベント・パフォーマンス統計を生成できるイベント・タイプおよびエレメント

イベント・タイプ	エレメント
ビジネス・プロセス	Process Invoke Staff Receive Wait Compensate Pick Scope
ヒューマン・タスク	Task
ビジネス・ルール	Operation
ビジネス状態マシン	Transition Guard Action EntryAction ExitAction
セレクター	Operation
マップ	Map Transformation
メディエーション	OperationBinding ParameterMediation
リソース・アダプター	InboundEventRetrieval InboundEventDelivery Outbound

## モニターするパフォーマンス統計の指定

管理コンソールを使用することで、パフォーマンス・モニター・インフラストラクチャーによるモニター用に 1 つの統計、複数の統計、または関連する統計のグループを指定することができます。

### 始める前に

このタスクを実行する前に、パフォーマンス・モニターを使用可能にしていること、およびモニター対象のイベントを少なくとも 1 度は必ず呼び出していることを確認します。

### プロシージャ

1. 管理コンソールを開きます。
2. 「モニターおよびチューニング」 > 「パフォーマンス・モニター・インフラストラクチャー (PMI)」を選択します。
3. モニターするイベント・ポイントを含んでいるサーバーまたはノード・エージェントを選択します。

**注:** クラスター上で統計のモニターを選択することはできません。それは特定のサーバーまたはノード上でのみ行うことができます。

4. **WBISStats.RootGroup** または **エンタープライズ Bean** など、いくつかのグループを展開します。モニター可能なすべての統計は、リストされたグループ内にあります。プロセス・サーバーの最後の始動以降呼び出されていないためにリストされない統計もあります。
5. パネルの左側にあるツリーからモニターする統計を選択し、右側で収集する統計を選択して、「使用可能にする」をクリックします。モニターするすべての統計に対してこの手順を繰り返します。
6. サーバー名のリンクをクリックして、「サーバー PMI 構成 (server PMI configuration)」ページに戻ります。
7. 「適用」または「OK」をクリックします。
8. 「保管」をクリックします。

### 結果

これで、Tivoli Performance Viewer で、選択した統計のパフォーマンスのモニターを開始できます。

**注:** これらの統計を表示する場合、カウンター型統計と期間型統計を混合しないでください。カウンターは累積されるため、アプリケーションによってはグラフのスケールがすぐに大きくなる可能性があります。反対に、期間型統計はシステムが各イベントを処理するのにかかる時間の平均を示すため、一定の範囲に保たれます。その結果、統計とその相対スケールの間不均衡が生じ、ビューアー・グラフで片方のタイプの統計がスキュー表示されます。

## チュートリアル: サービス・コンポーネントのパフォーマンス・モニター

このチュートリアルでは、パフォーマンス・モニター・インフラストラクチャー (PMI) にパブリッシュされるサービス・コンポーネント・イベント・モニターのセ

ットアップ方法、および Tivoli Performance Viewer (TPV) でパフォーマンス統計の結果を表示する方法について例を使用して説明します。この演習では、サービス・コンポーネント・イベント・ポイントのパフォーマンス・モニターと、Common Event Infrastructure (CEI) サーバーおよびロガーを使用したモニターとの違いを示します。大きな違いは、パフォーマンス・モニターの場合、特定の性質を持つ個別のイベントを選択するのではなく、サービス・コンポーネント・エレメント全体を選択する点です。WebSphere Process Server がパフォーマンスをモニターできるのは ENTRY、EXIT、および FAILURE という性質のイベントを持つサービス・コンポーネント・エレメントのみであるため、モニター対象として選択できるのはそのようなサービス・コンポーネント・エレメントのみです。

ENTRY、EXIT、および FAILURE の各サービス・コンポーネント・イベント・ポイントがすべてのモニター・タイプで同じであるのに対し、プロセス・サーバーのパフォーマンス・モニター機能では、CEI イベントに圧縮されている一部の情報を含む「最小化された」イベントを送出します。これらのイベントは PMI に送信され、対応するイベント・セットから次のパフォーマンス統計が計算されます。

- 成功した呼び出し: 対応する ENTRY イベントに続く、性質タイプが EXIT のイベントの送付。
- 失敗した呼び出し: 対応する ENTRY イベントに続く、性質タイプが FAILURE のイベントの送付。
- 正常終了の時間: ENTRY イベントの送付から、対応する EXIT イベント・ポイントの送付までの経過時間。

PMI は TPV に統計をパブリッシュし、TPV では、成功した呼び出しと失敗した呼び出しの数を示す累積カウンターおよび完了応答時間の実行平均を表示します。

## このチュートリアルの目的

このチュートリアルを完了すれば、次の操作を実行できるようになります。

- モニターするサービス・コンポーネント・エレメントのパフォーマンス統計を選択する。
- パフォーマンス統計結果を表示および解釈する。

## このチュートリアルを完了するのに必要な時間

このチュートリアルを完了するには、およそ 15 分から 20 分かかります。

## 前提条件

このチュートリアルを実行するには、次の条件を満たす必要があります。

- プロセス・サーバーを構成および始動済みである。
- プロセス・サーバーで PMI を使用可能に設定する。
- プロセス・サーバーでサンプル・ギャラリー・アプリケーションがインストールおよび始動済みである。
- プロセス・サーバーでビジネス・ルール・サンプル・アプリケーションがインストールおよび始動済みである。「サンプル・ギャラリー」ページの指示に従って、ビジネス・ルール・サンプル・アプリケーションをセットアップし、実行します。

これらのすべての前提条件が満たされたら、チュートリアルに進む前に、少なくとも 1 度サンプル・ギャラリーからビジネス・ルール・サンプル・アプリケーションを実行してください。

#### 例: サービス・コンポーネントのパフォーマンスのモニター: このタスクについて

このシナリオでは、ビジネス・ルール・サンプル・アプリケーションを使用して、3 つのすべてのパフォーマンス統計 (成功、失敗、および応答時間) をモニターします。このアプリケーションが配置されている Web ページを開いておいてください。モニターの開始後にサンプルを数回実行するため、ページは開いたままにしておいてください。サンプルを少なくとも 1 度実行しておいてください。実行しておくと、サンプルがモニター対象として選択可能な機能のリストに表示されます。

#### プロシージャ

1. 管理コンソールを開きます。
2. クラスタをモニターするには、コンソールのナビゲーション・ツリーで、「サーバー」 > 「クラスタ」 > *cluster\_name* をクリックします。単一のプロセス・サーバーをモニターするには、コンソールのナビゲーション・ツリーで、「サーバー」 > 「アプリケーション・サーバー」 > *server\_name* をクリックします。
3. 「ランタイム (Runtime)」タブをクリックします。
4. 「パフォーマンス (Performance)」の下で、「パフォーマンス・モニター・インフラストラクチャー」をクリックします。
5. 「カスタム」を選択します。
6. 「WBISStats.RootGroup」 > 「BR」 > 「brsample\_module.DiscountRuleGroup」 > 「操作」を展開し、「\_calculateDiscount」を選択します。
7. 「BadRequests」、「GoodRequests」、および「ResponseTime」の横のチェック・ボックスを選択して、「使用可能にする」を押します。
8. 管理コンソールのナビゲーション・ツリーで、「モニターとチューニング (Monitoring and Tuning)」 > 「Performance Viewer」 > 「現在のアクティビティ (Current Activity)」をクリックします。
9. 「*server\_name*」の横のチェック・ボックスを選択して、「モニターの開始」を押します。
10. *server\_name* をクリックします。
11. 「WBISStats.RootGroup」 > 「BR」 > 「brsample\_module.DiscountRuleGroup」 > 「操作」を展開し、「\_calculateDiscount」の横のチェック・ボックスを選択します。

#### 結果

空白のグラフが表示され、グラフの下には 3 つの統計の名前と値が表示されます。統計名の横のチェック・ボックスが選択されていない場合、それを選択します。これで、PMI では選択されたイベントのパフォーマンス・データのパブリッシュが、TPV では結果の表示ができます。

ビジネス・ルール・サンプル・アプリケーションを数回実行し、パフォーマンス・ビューアーが定期的に最新表示されるのを確認します。グラフには、成功した要求

の累積数と各成功した要求の平均応答時間を表す線が表示されます。また、グラフの下には、各統計の名前の横に値が表示されます。成功数を表す線はサンプルをさらに追加呼び出すにつれて上昇するのに対し、応答時間の線は、最新表示が 2、3 回行われた後で水平になります。

このタスクを完了すれば、WebSphere Process Server がサービス・コンポーネントのパフォーマンス・モニターをインプリメントする方法を理解したことになります。モニターするサービス・コンポーネントを選択する方法や、パフォーマンス統計の計算方法についても理解できます。また、パフォーマンス・モニターを開始し、アプリケーション・パフォーマンスのリアルタイムの測定を表示することもできます。

パフォーマンス・モニターはシステム・リソースに負荷をかける可能性があります。そのため、このタスクが完了したらモニターを停止してください。モニターを停止するには、Tivoli Performance Viewer リンクをクリックし、ノードとサーバーの両方を選択して「**モニターの停止 (Stop Monitoring)**」を押します。

## Service Component Architecture のアプリケーション応答測定統計

Service Component Architecture レベルで、25 件のパフォーマンス統計をモニターできます。これらのアプリケーション応答測定統計 (これらはカウンターかタイマーです) を使用して、さまざまなパターンのサービスの呼び出しと応答を測定することができます。

以下の表に示すアプリケーション応答測定 (ARM) 統計は、Service Component Architecture (SCA) 層に対する呼び出し元による呼び出しおよびサービスから戻される結果の、時間およびカウントの測定 (非常に単純な方法での測定) です。事実、多数のサービス呼び出しパターンがあります。それは、遅延応答、結果取得、コールバック、および片方向呼び出しの同期インプリメンテーションと非同期インプリメンテーションとは異なります。しかし、これらはすべて、呼び出し元による呼び出しとサービス、サービスからの応答、または場合によってはデータ・ソースとの間で、SCA 層を介して行われます。

モニターする ARM 統計を指定するには、管理コンソールで「**モニターおよびチューニング**」>「**要求メトリック**」パネルを開きます。要求メトリック情報は、後で取り出して分析できるようログ・ファイルに保存されるか、ARM エージェントへ送信されるか、あるいはその両方で処理されます。WebSphere Process Server は ARM エージェントを含んではいませんが、ARM 4.0 に準拠したエージェントの使用をサポートしています。ユーザーは独自の ARM 実装プロバイダーを選択して、ARM 実装ライブラリーを入手できます。ARM プロバイダーの説明に従って、ARM プロバイダーの ARM API Javaアーカイブ (JAR) ファイルが必ずクラス・パス上に存在し、WebSphere Process Server が必要なクラスをロードできるようにしてください。次に、以下の項目を各サーバーのシステム・プロパティに追加する必要があります。そのためには、管理コンソールから「**アプリケーション・サーバー**」> *server\_name* >「**プロセス定義**」>「**Java 仮想マシン**」>「**カスタム・プロパティ**」を選択します。その後、サーバー (単数または複数) を再始動します。

- `Arm40.ArmMetricFactory` - ARM 実装プロバイダーのメトリック・ファクトリーの完全 Javaクラス名。

- Arm40.ArmTranReportFactory - ARM 実装プロバイダーのトランザクション・レポート・ファクトリーの完全 Java クラス名。
- Arm40.ArmTransactionFactory - ARM 実装プロバイダーのトランザクション・ファクトリーの完全 Java クラス名。

ARM 統計を収集するためのサーバーの詳しい構成方法については、WebSphere Application Server の資料を参照してください。

表 3. ARM 統計を生成できるイベント・タイプおよびエレメント

イベント・タイプ	エレメント
ビジネス・プロセス	Process
ヒューマン・タスク	Task
ビジネス・ルール	Operation
ビジネス状態マシン	Transition Guard Action EntryAction ExitAction
セレクター	Operation
マップ	Map Transformation
メディエーション	OperationBinding ParameterMediation
リソース・アダプター	InboundEventRetrieval InboundEventDelivery Outbound

表 4. 共通：以下の統計は、すべてのサービス呼び出しパターンに共通です。

統計名	タイプ	説明
<b>GoodRequests</b>	カウンター	例外が発生しなかったサービス呼び出しの数。
<b>BadRequests</b>	カウンター	例外が発生したサーバー呼び出しの数。
<b>ResponseTime</b>	タイマー	サーバー・サイドで測定される、要求の受信と結果の計算の間の時間。
<b>TotalResponseTime</b>	タイマー	呼び出し側で測定される、呼び出し側がサービスを要求してから呼び出し側に結果が戻るまでの時間。呼び出し側による結果の処理は含まれません。
<b>RequestDeliveryTime</b>	タイマー	呼び出し側で測定される、呼び出し側がサービスを要求してからサーバー・サイドの実装環境に要求が受け渡されるまでの時間。分散環境では、この測定の正確性はシステム・クロックの同期の正確性によって異なります。
<b>ResponseDeliveryTime</b>	タイマー	結果がクライアントに戻るまでに必要な時間。遅延応答の場合、この時間には結果取得時間は含まれません。分散環境では、この測定の正確性は、システム・クロックの同期の正確性によって異なります。



表 5. 参照：以下の統計は、サービスからの応答なしに、呼び出し側が SCA 層またはデータ・ソースを呼び出した場合に作成されます。

統計名	タイプ	説明
<b>GoodRefRequests</b>	カウンター	例外を発生させない SCA 層への呼び出し側による呼び出しの数。
<b>BadRefRequests</b>	カウンター	例外を発生させる SCA 層への呼び出し側による呼び出しの数。
<b>RefResponseTime</b>	タイマー	呼び出し側で測定される、呼び出し側が SCA 層に要求を出してから呼び出し側に呼び出しの結果が戻るまでの時間。
<b>BadRetrieveResult</b>	カウンター	例外を発生させるデータ・ソースへの呼び出し側による呼び出しの数。
<b>GoodRetrieveResult</b>	カウンター	例外を発生させないデータ・ソースへの呼び出し側による呼び出しの数。
<b>RetrieveResultResponseTime</b>	タイマー	呼び出し側で測定される、呼び出し側がデータ・ソースに要求を出してから呼び出し側にデータ・ソースの応答が戻るまでの時間。
<b>RetrieveResultWaitTime</b>	タイマー	タイムアウトが発生した場合に、呼び出し側で測定される時間。

表 6. ターゲット：以下の統計は、サービスと SCA またはデータ・ソースとの間に発生する要求がある場合に作成されます。

統計名	タイプ	説明
<b>GoodTargetSubmit</b>	カウンター	例外を発生させないサービスへの SCA 呼び出しの数。
<b>BadTargetSubmit</b>	カウンター	例外を発生させるサービスへの SCA 呼び出しの数。
<b>TargetSubmitTime</b>	タイマー	サーバー・サイドで測定される、SCA がサービスに要求を出してからその呼び出しの結果が SCA に戻るまでの時間。
<b>GoodResultSubmit</b>	カウンター	例外を発生させないデータ・ソースへのサービス呼び出しの数。
<b>BadResultSubmit</b>	カウンター	例外を発生させるデータ・ソースへのサービス呼び出しの数。
<b>ResultSubmitTime</b>	タイマー	サーバー・サイドで測定される、サービスがデータ・ソースに要求を出してからその要求の結果がサービスに戻るまでの時間。

表 7. コールバック：以下の統計は、呼び出し側にコールバック（元の呼び出しと同種の呼び出し）が存在する場合に作成されます。

統計名	タイプ	説明
<b>GoodCB</b>	カウンター	例外を発生させないコールバックへの SCA 呼び出しの数。

表7. コールバック (続き): 以下の統計は、呼び出し側にコールバック (元の呼び出しと同種の呼び出し) が存在する場合に作成されます。

統計名	タイプ	説明
BadCB	カウンター	例外を発生させるコールバックへの SCA 呼び出しの数。
CBTime	タイマー	SCA がコールバックに要求を出してからコールバックからの結果が SCA に戻るまでの時間。
GoodCBSubmit	カウンター	サービスからコールバックを処理する SCA への例外を発生させない呼び出しの数。
BadCBSubmit	カウンター	サービスからコールバックを処理する SCA への例外を発生させる呼び出しの数。
CBSubmitTime	タイマー	サービスがコールバックを処理する SCA に要求を出してから、SCA からの結果がサービスに戻るまでの時間。

以下のトピックでは、これらの統計をさまざまな実装で使用方法を示します。

## 同期呼び出し

ここでは、サービスへの単純な SCA 呼び出しとサービスからの応答から取得できる ARM パフォーマンス統計について説明します。

## パラメーター

SCA コンポーネントのイベント・モニターには、黒で示されるイベント・ポイント



が含まれます。青で示されるイベント・ポイント



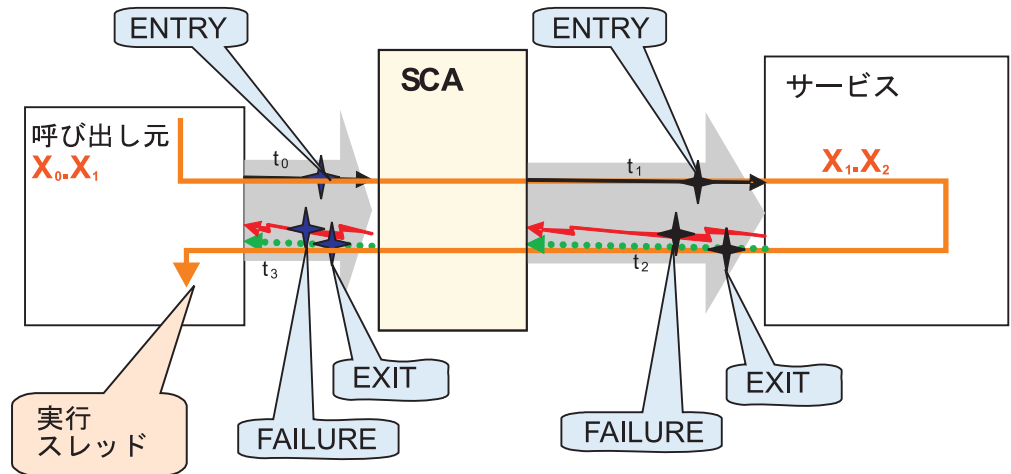
は、PMI/ARM 統計の計算と送付にのみ使用されます。

下記の表と図で、「現行」の ARM トランザクション ( $X_1$  として表記します) は、呼び出しサービス・コンポーネントが初めて呼び出されたときに作成されます。呼び出し元がサービス・コンポーネントではない場合、現行の ARM トランザクションが使用されるか、新規トランザクションが作成されます。開始トランザクションでないトランザクションには、親があります。これは、次の表と図に  $X_n \cdot X_{n+1}$  として表記されています。これらはトランザクションの血統を示すために使用されます。すべての SCA 呼び出しは新規トランザクションを開始し、これは、呼び出し元の現行トランザクションが親となります。ユーザーは新規トランザクションを作成でき、現行トランザクションにアクセスできますが、それによって SCA トランザクションの血統が変更されることはありません。

表8. SCA の同期呼び出しでの ARM 統計

統計	公式	ARM トランザクション
TotalResponseTime	$t_3 - t_0$	$X_0 \cdot X_1$
RequestDeliveryTime	$t_1 - t_0$	$X_1 \cdot X_2$
ResponseDeliveryTime	$t_3 - t_2$	
GoodRequests	Count <sub>EXIT</sub>	
BadRequests	Count <sub>FAILURE</sub>	
ProcessTime	$t_2 - t_1$	





### 同期実装での据え置き応答

ここでは、要求の同期呼び出しと、データ・ストアへの出力として送信された戻り結果によって取得できる ARM 統計を、同期実装について示します。

#### パラメーター

SCA コンポーネントのイベント・モニターには、黒で示されるイベント・ポイント



が含まれます。青で示されるイベント・ポイント



は、PMI/ARM 統計の計算と送出にのみ使用されます。

下記の表と図で、「現行」の ARM トランザクション ( $X_1$  として表記します) は、呼び出しサービス・コンポーネントが初めて呼び出されたときに作成されます。呼び出し元がサービス・コンポーネントではない場合、現行の ARM トランザクションが使用されるか、新規トランザクションが作成されます。開始トランザクションでないトランザクションには、親があります。これは、次の表と図に  $X_n, X_{n+1}$  として表記されています。これらはトランザクションの血統を示すために使用されます。すべての SCA 呼び出しは新規トランザクションを開始し、これは、呼び出し元の現行トランザクションが親となります。ユーザーは新規トランザクションを作成でき、現行トランザクションにアクセスできますが、それによって SCA トランザクションの血統が変更されることはありません。

表9. 要求の呼び出しと戻り結果

タイプ	統計	公式	ARM トランザクション
共通	TotalResponseTime	$t_3 - t_0$	$X_0, X_1$
	RequestDeliveryTime	$t'_0 - t_0$	$X_1, X_2$
	ResponseDeliveryTime	なし	なし
	GoodRequests	$\text{Count}_{\text{EXIT}}$	$X_1, X_2$
	BadRequests	$\text{Count}_{\text{FAILURE}}$	
	ResponseTime	$t'_1 - t'_0$	

表9. 要求の呼び出しと戻り結果 (続き)

タイプ	統計	公式	ARM トランザクション
参照 A	GoodRefRequest	$\text{Count}_{\text{EXIT}}$	$X_1 \cdot X_2$
	BadRefRequests	$\text{Count}_{\text{FAILURE}}$	
	RefResponseTime	$t_1 - t_0$	

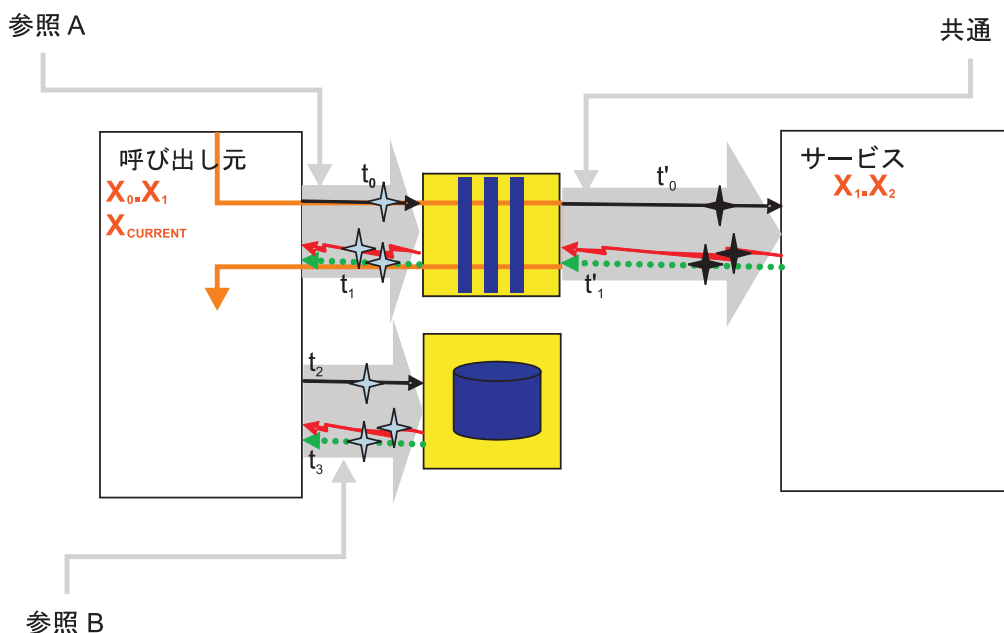


表10. データ・ソースへの出力の呼び出し

タイプ	統計	公式	ARM トランザクション
参照 B	GoodRetrieveResult	$\text{Count}_{\text{EXIT}}$	$X_1 \cdot X_2$
	BadRetrieveResult	$\text{Count}_{\text{FAILURE}}$	
	ResultRetrieveResponseTime	$\sum t_3 - t_2$	
	ResultRetrieveWaitTime	$\sum \text{timeout}$	

### 非同期実装での据え置き応答

ここでは、サービスの呼び出しと戻り結果が呼び出され、結果出力がサービスのターゲットからデータ・ストアへ送信される非同期実装からの ARM 統計を示します。

### パラメーター

SCA コンポーネントのイベント・モニターには、黒で示されるイベント・ポイント



が含まれます。青で示されるイベント・ポイント



は、PMI/ARM 統計の計算と送出しにのみ使用されます。

下記の表と図で、「現行」の ARM トランザクション ( $X_1$  として表記します) は、呼び出しサービス・コンポーネントが初めて呼び出されたときに作成されます。呼び出し元がサービス・コンポーネントではない場合、現行の ARM トランザクションが使用されるか、新規トランザクションが作成されます。開始トランザクションでないトランザクションには、親があります。これは、次の表と図に  $X_n, X_{n+1}$  として表記されています。これらはトランザクションの血統を示すために使用されます。すべての SCA 呼び出しは新規トランザクションを開始し、これは、呼び出し元の現行トランザクションが親となります。ユーザーは新規トランザクションを作成でき、現行トランザクションにアクセスできますが、それによって SCA トランザクションの血統が変更されることはありません。

表 11. 要求の呼び出しと戻り結果

タイプ	統計	公式	ARM トランザクション
共通	TotalResponseTime	$t_3 - t_0$	$X_0, X_1$
	RequestDeliveryTime	$t'_0 - t_0$	$X_1, X_2$
	ResponseDeliveryTime	$t'_{03} - t'_2$	
	GoodRequests	Count <sub>EXIT</sub>	
	BadRequests	Count <sub>FAILURE</sub>	
	ResponseTime	$t'_3 - t'_0$	
参照 A	GoodRefRequest	Count <sub>EXIT</sub>	$X_0, X_1$
	BadRefRequests	Count <sub>FAILURE</sub>	
	RefResponseTime	$t_1 - t_0$	
ターゲット A	GoodTargetSubmit	Count <sub>EXIT</sub>	$X_1, X_2$
	BadTargetSubmit	Count <sub>FAILURE</sub>	
	TargetSubmitTime	$t'_1 - t'_0$	

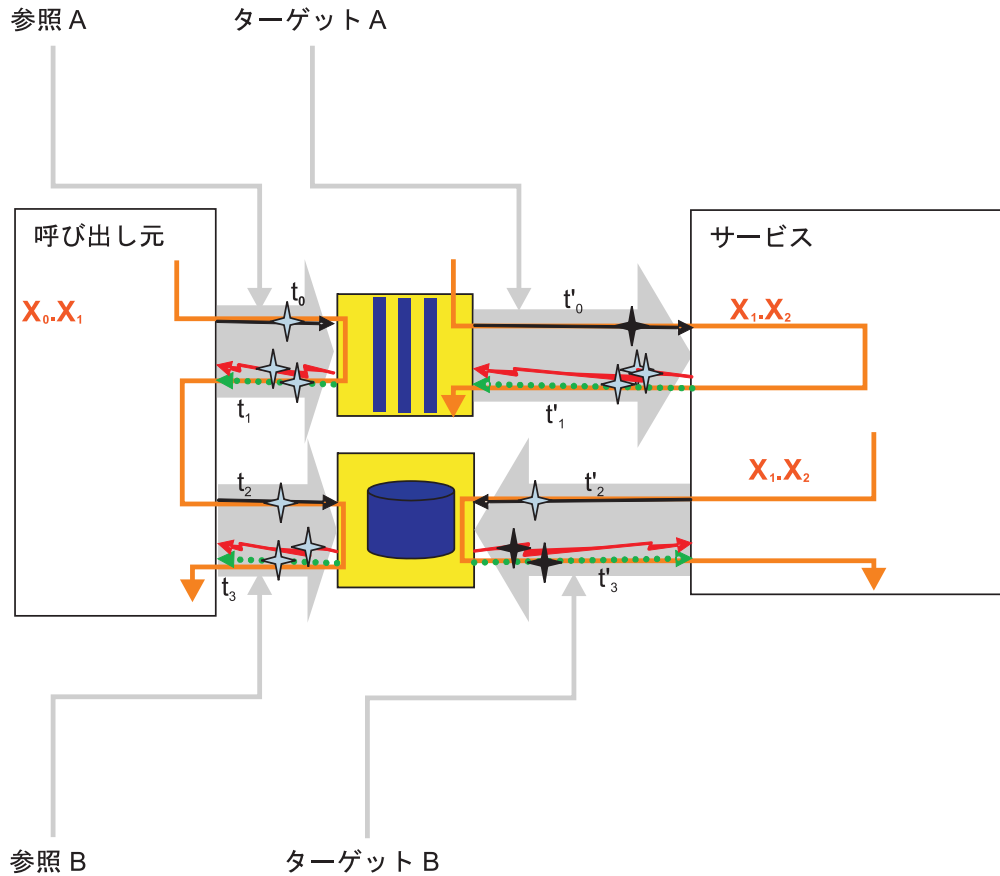


表 12. データ・ストアへの戻り結果の呼び出し

タイプ	統計	公式	ARM トランザクション
参照 B	GoodResultSubmit	$\text{Count}_{\text{EXIT}}$	$X_0.X_1$
	BadResultSubmit	$\text{Count}_{\text{FAILURE}}$	
	ResultResponseTime	$t'_3 - t'_2$	
ターゲット B	GoodResultRetrieve	$\text{Count}_{\text{EXIT}}$	$X_1.X_2$
	BadResultRetrieve	$\text{Count}_{\text{FAILURE}}$	
	ResultRetrieveResponseTime	$\sum t_3 - t_2$	
	ResultRetrieveWaitTime	$\sum \text{timeout}$	

### 非同期の結果取得での据え置き応答

ARM トランザクションを使用して、ResultRetrieve ARM 統計を何らかのオリジナル要求に相関させることができるのは、 $X_{\text{PARENT-1}}$  と  $X_{\text{PARENT-2}}$  が共通の上位トランザクションを持っている場合だけです。要求の呼び出しと結果の取得は、異なるスレッド上で行われます。

## パラメーター

SCA コンポーネントのイベント・モニターには、黒で示されるイベント・ポイント



が含まれます。青で示されるイベント・ポイント



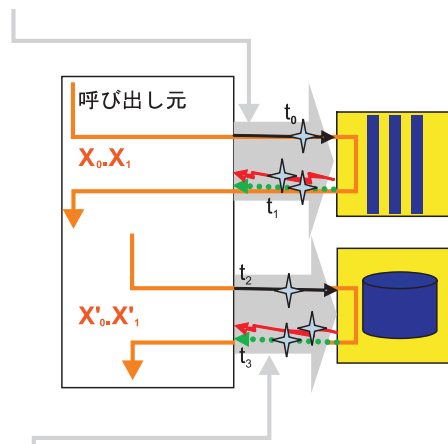
は、PMI/ARM 統計の計算と送出しにのみ使用されます。

下記の表と図で、「現行」の ARM トランザクション ( $X_1$  として表記します) は、呼び出しサービス・コンポーネントが初めて呼び出されたときに作成されます。呼び出し元がサービス・コンポーネントではない場合、現行の ARM トランザクションが使用されるか、新規トランザクションが作成されます。開始トランザクションでないトランザクションには、親があります。これは、次の表と図に  $X_n \cdot X_{n+1}$  として表記されています。これらはトランザクションの血統を示すために使用されます。すべての SCA 呼び出しは新規トランザクションを開始し、これは、呼び出し元の現行トランザクションが親となります。ユーザーは新規トランザクションを作成でき、現行トランザクションにアクセスできますが、それによって SCA トランザクションの血統が変更されることはありません。

表 13. 要求の呼び出しと戻り結果

タイプ	統計	公式	ARM トランザクション
共通	TotalResponseTime	$t_3 - t_0$	$X_0 \cdot X_1$
	RequestDeliveryTime	$t'_0 - t_0$	$X_1 \cdot X_2$
	ResponseDeliveryTime	なし	なし
	GoodRequests	Count <sub>EXIT</sub>	$X_1 \cdot X_2$
	BadRequests	Count <sub>FAILURE</sub>	
	ResponseTime	具体的な図を参照	
参照 A	GoodReferenceRequest	Count <sub>EXIT</sub>	$X_1 \cdot X_2$
	BadReferenceRequests	Count <sub>FAILURE</sub>	
	ReferenceResponseTime	$t_1 - t_0$	

参照 A



図の右側については、『同期/非同期実装での据え置き応答』の各トピック内の図を参照してください。

要求の受け取り時間  $t_0$  は、どちらの場合も同じです。

参照 B

表 14. 要求の呼び出しと戻り結果

タイプ	統計	公式	ARM トランザクション
参照 B	GoodRetrieveResult	Count <sub>EXIT</sub>	X' <sub>0</sub> .X' <sub>1</sub>
	BadRetrieveResult	Count <sub>FAILURE</sub>	
	RetrieveResultResponseTime	$\sum t_3 - t_2$	
	RetrieveResultWaitTime	$\sum \text{timeout}$	

## 同期実装での非同期コールバック

同期実装で、コールバック要求とコールバック実行で異なるスレッドを使用する場合に使用可能な ARM 統計。

### パラメーター

SCA コンポーネントのイベント・モニターには、黒で示されるイベント・ポイント



が含まれます。青で示されるイベント・ポイント



は、PMI/ARM 統計の計算と送出しにのみ使用されます。

下記の表と図で、「現行」の ARM トランザクション (X<sub>1</sub> として表記します) は、呼び出しサービス・コンポーネントが初めて呼び出されたときに作成されます。呼び出し元がサービス・コンポーネントではない場合、現行の ARM トランザクションが使用されるか、新規トランザクションが作成されます。開始トランザクションでないトランザクションには、親があります。これは、次の表と図に X<sub>n</sub>.X<sub>n+1</sub> として表記されています。これらはトランザクションの血統を示すために使用されます。すべての SCA 呼び出しは新規トランザクションを開始し、これは、呼び出し元の現行トランザクションが親となります。ユーザーは新規トランザクションを作成でき、現行トランザクションにアクセスできますが、それによって SCA トランザクションの血統が変更されることはありません。

表 15. 要求の呼び出しと戻り結果

タイプ	統計	公式	ARM トランザクション
共通	TotalResponseTime	t <sub>2</sub> - t <sub>0</sub>	X <sub>0</sub> .X <sub>1</sub>
	RequestDeliveryTime	t' <sub>0</sub> - t <sub>0</sub>	X <sub>1</sub> .X <sub>2</sub>
	ResponseDeliveryTime	t <sub>2</sub> - t' <sub>1</sub>	
	GoodRequests	Count <sub>EXIT</sub>	
	BadRequests	Count <sub>FAILURE</sub>	
	ResponseTime	t <sub>3</sub> - t <sub>2</sub>	
参照	GoodRefRequest	Count <sub>EXIT</sub>	
	BadRefRequests	Count <sub>FAILURE</sub>	
	RefResponseTime	t' <sub>1</sub> - t' <sub>0</sub>	



表 16. コールバックの呼び出し

タイプ	統計	公式	ARM トランザクション
コールバック	GoodCB	$\text{Count}_{\text{EXIT}}$	$X_1 \cdot X_3$
	BadCB	$\text{Count}_{\text{FAILURE}}$	
	CBTime	$t_3 - t_2$	

### 非同期実装での非同期コールバック

非同期実装で、異なるスレッドを使用したコールバック要求とコールバック実行について使用可能な ARM 統計

#### パラメーター

SCA コンポーネントのイベント・モニターには、黒で示されるイベント・ポイント



が含まれます。青で示されるイベント・ポイント



は、PMI/ARM 統計の計算と送出しにのみ使用されます。

下記の表と図で、「現行」の ARM トランザクション ( $X_1$  として表記します) は、呼び出しサービス・コンポーネントが初めて呼び出されたときに作成されます。呼び出し元がサービス・コンポーネントではない場合、現行の ARM トランザクションが使用されるか、新規トランザクションが作成されます。開始トランザクションでないトランザクションには、親があります。これは、次の表と図に  $X_n \cdot X_{n+1}$  として表記されています。これらはトランザクションの血統を示すために使用されます。すべての SCA 呼び出しは新規トランザクションを開始し、これは、呼び出し元の

現行トランザクションが親となります。ユーザーは新規トランザクションを作成でき、現行トランザクションにアクセスできますが、それによって SCA トランザクションの血統が変更されることはありません。

表 17. 要求の呼び出しと戻り結果

タイプ	統計	公式	ARM トランザクション
共通	TotalResponseTime	$t_2 - t_0$	$X_0 \cdot X_1$
	RequestDeliveryTime	$t'_0 - t_0$	$X_1 \cdot X_2$
	ResponseDeliveryTime	$t_2 - t'_2$	
	GoodRequests	$\text{Count}_{\text{EXIT}}$	
	BadRequests	$\text{Count}_{\text{FAILURE}}$	
	ResponseTime	$t'_3 - t'_0$	
参照 A	GoodRefRequest	$\text{Count}_{\text{EXIT}}$	
	BadRefRequests	$\text{Count}_{\text{FAILURE}}$	
	RefResponseTime	$t_1 - t_0$	
ターゲット A	GoodTargetSubmit	$\text{Count}_{\text{EXIT}}$	$X_1 \cdot X_2$
	BadTargetSubmit	$\text{Count}_{\text{FAILURE}}$	
	TargetSubmitTime	$t'_1 - t'_0$	

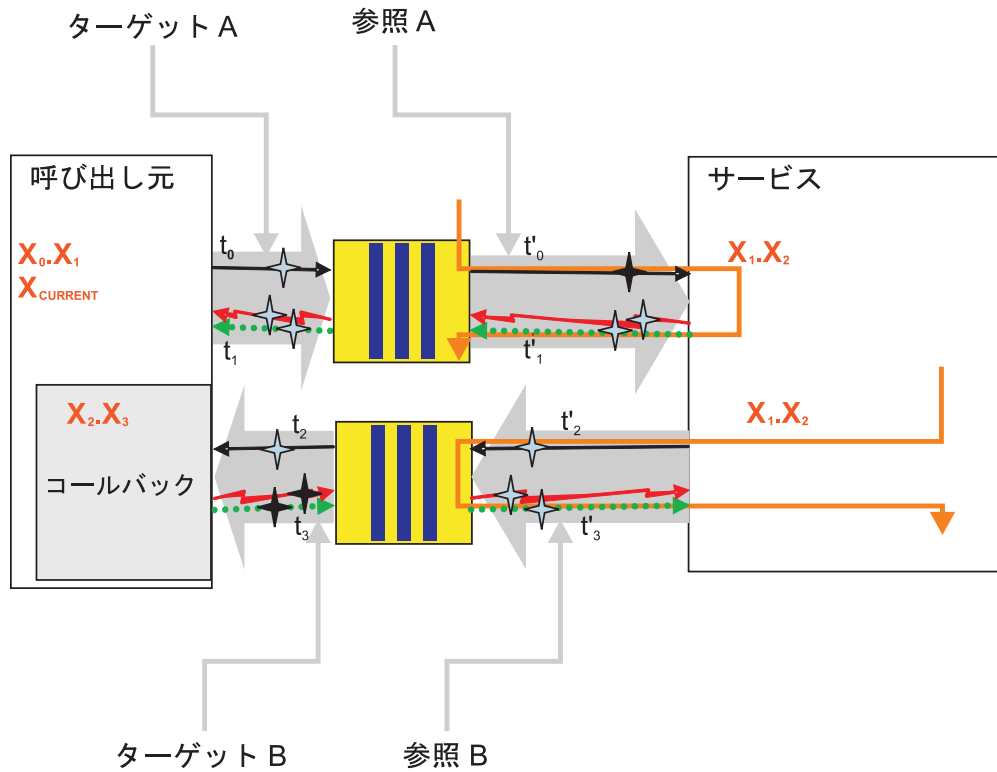




表 18. コールバックの呼び出し

タイプ	統計	公式	ARM トランザクション
参照 B	GoodCBSSubmit	Count <sub>EXIT</sub>	X <sub>1</sub> .X <sub>2</sub>
	BadCBSSubmit	Count <sub>FAILURE</sub>	
	CBSSubmitTime	t' <sub>3</sub> - t' <sub>2</sub>	
ターゲット B	GoodCB	Count <sub>EXIT</sub>	X <sub>0</sub> .X <sub>1</sub>
	BadCB	Count <sub>FAILURE</sub>	
	CBTime	t <sub>3</sub> - t <sub>2</sub>	

### 同期実装での非同期片方向

同期実装で、呼び出しが実行依頼 (応答不要送信) された場合の ARM 統計。

#### パラメーター

SCA コンポーネントのイベント・モニターには、黒で示されるイベント・ポイント



が含まれます。青で示されるイベント・ポイント

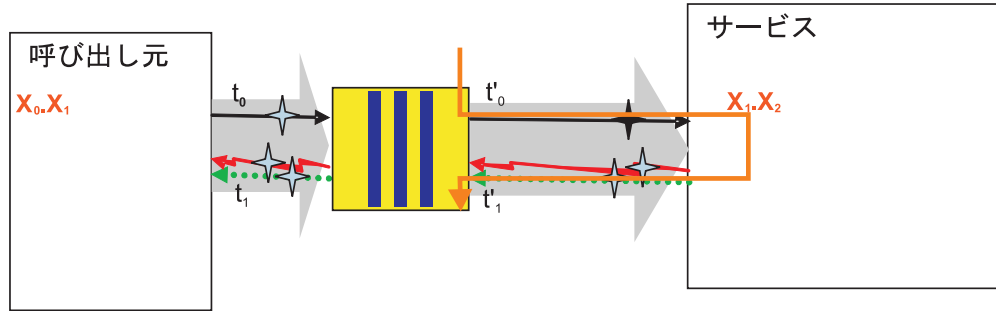


は、PMI/ARM 統計の計算と送出にのみ使用されます。

下記の表と図で、「現行」の ARM トランザクション (X<sub>1</sub> として表記します) は、呼び出しサービス・コンポーネントが初めて呼び出されたときに作成されます。呼び出し元がサービス・コンポーネントではない場合、現行の ARM トランザクションが使用されるか、新規トランザクションが作成されます。開始トランザクションでないトランザクションには、親があります。これは、次の表と図に X<sub>n</sub>.X<sub>n+1</sub> として表記されています。これらはトランザクションの血統を示すために使用されます。すべての SCA 呼び出しは新規トランザクションを開始し、これは、呼び出し元の現行トランザクションが親となります。ユーザーは新規トランザクションを作成でき、現行トランザクションにアクセスできますが、それによって SCA トランザクションの血統が変更されることはありません。

表 19. 要求の呼び出しと戻り結果

タイプ	統計	公式	ARM トランザクション
共通	TotalResponseTime	t <sub>1</sub> - t <sub>0</sub>	X <sub>0</sub> .X <sub>1</sub>
	RequestDeliveryTime	t' <sub>0</sub> - t <sub>0</sub>	X <sub>1</sub> .X <sub>2</sub>
	ResponseDeliveryTime	なし	なし
	GoodRequests	Count <sub>EXIT</sub>	X <sub>1</sub> .X <sub>2</sub>
	BadRequests	Count <sub>FAILURE</sub>	
	ResponseTime	t' <sub>1</sub> - t' <sub>0</sub>	



## 非同期実装での非同期片方向

非同期実装で、呼び出しが実行依頼 (応答不要送信) された場合の ARM 統計。

### パラメーター

SCA コンポーネントのイベント・モニターには、黒で示されるイベント・ポイント



が含まれます。青で示されるイベント・ポイント

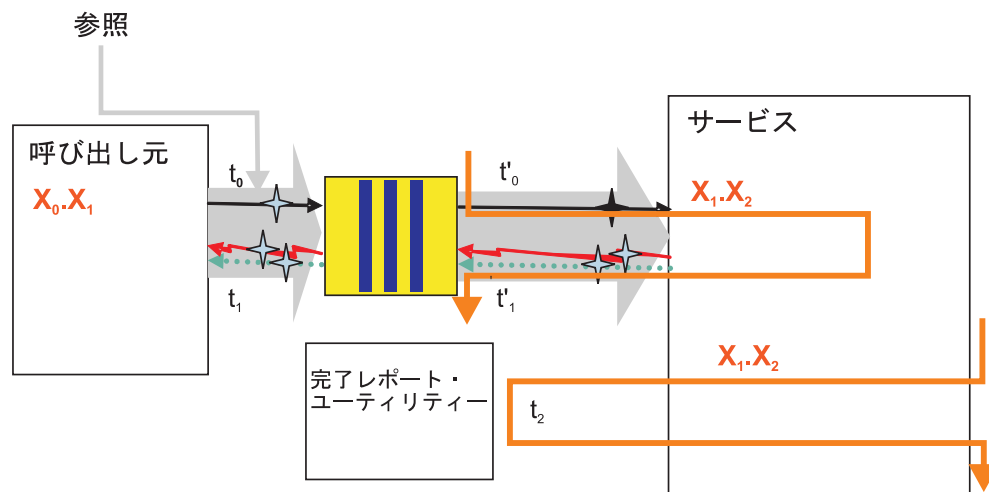


は、PMI/ARM 統計の計算と送出にのみ使用されます。

下記の表と図で、「現行」の ARM トランザクション ( $X_1$  として表記します) は、呼び出しサービス・コンポーネントが初めて呼び出されたときに作成されます。呼び出し元がサービス・コンポーネントではない場合、現行の ARM トランザクションが使用されるか、新規トランザクションが作成されます。開始トランザクションでないトランザクションには、親があります。これは、次の表と図に  $X_n, X_{n+1}$  として表記されています。これらはトランザクションの血統を示すために使用されます。すべての SCA 呼び出しは新規トランザクションを開始し、これは、呼び出し元の現行トランザクションが親となります。ユーザーは新規トランザクションを作成でき、現行トランザクションにアクセスできますが、それによって SCA トランザクションの血統が変更されることはありません。

表 20. 要求の呼び出しと戻り結果

タイプ	統計	公式	ARM トランザクション
共通	TotalResponseTime	$t_1 - t_0$	$X_{\text{PARENT}} \cdot X_{\text{REQUEST}}$
	RequestDeliveryTime	$t'_0 - t_0$	$X_{\text{REQUEST}} \cdot X_{\text{PROCESS}}$
	ResponseDeliveryTime	なし	なし
	GoodRequests	$\text{Count}_{\text{EXIT}}$	$X_{\text{REQUEST}} \cdot X_{\text{PROCESS}}$
	BadRequests	$\text{Count}_{\text{FAILURE}}$	
	ResponseTime	$t'_2 - t_0$	
参照	GoodRefRequest	$\text{Count}_{\text{EXIT}}$	$X_{\text{PARENT}} \cdot X_{\text{REQUEST}}$
	BadRefRequest	$\text{Count}_{\text{FAILURE}}$	
	RefResponseDuration	$t_1 - t_0$	



## サービス・コンポーネント・イベントのモニター

WebSphere Process Server モニターでは、特定のイベント・ポイントでサービス・コンポーネントのデータを取り込むことができます。ログ・ファイル内の個々のイベントを表示するか、もっと多用途の Common Event Infrastructure サーバーのモニター機能を使用することができます。

プロセス・サーバーにデプロイされているアプリケーションには、アプリケーションが実行されている限りモニターされるサービス・コンポーネント・イベントの仕様が含まれています。WebSphere Integration Developer を使用してアプリケーションを開発した場合、サービス・コンポーネント・イベントを継続的にモニターするように指定できます。この仕様はアプリケーションの一部として組み込まれており、アプリケーションのデプロイ時にプロセス・サーバーによって読み取られる .mon 拡張子を持つファイルの形式で提供されます。アプリケーションを開始したら、.mon ファイルに指定されているサービス・コンポーネントのモニターをオフにすることはできません。WebSphere Process Server の資料では、このタイプの継続モニターについては扱っていません。この件についての詳細は、WebSphere Integration Developer の資料を参照してください。

WebSphere Process Server を使用して、アプリケーションの .mon ファイルに指定されていないサービス・コンポーネント・イベントをモニターすることができます。プロセス・サーバーでイベント・モニターの出力をログ・ファイルや Common Event Infrastructure サーバー・データベースに送信するように構成することができます。モニター対象イベントは、Common Base Event 標準を使用してフォーマット設定されますが、各イベントで保持する情報の量を規制することができます。WebSphere Process Server のモニター機能を使用して、問題の診断、アプリケーションのプロセス・フローの分析、またはアプリケーションの使用法の監査を行うことができます。

## ビジネス・プロセス・イベントとヒューマン・タスク・イベントのモニターの使用可能化

ビジネス・プロセスおよびヒューマン・タスクのサービス・コンポーネントのモニターをサポートするには、それらのサービス・コンポーネント種類のモニターを実際に開始する前に、WebSphere Process Server を構成する必要があります。

### 始める前に

ビジネス・プロセス・コンテナとヒューマン・タスク・コンテナがプロセス・サーバー上に事前に作成されている必要があります。

### このタスクについて

以下のタスクを実行して、WebSphere Process Server での Common Event Infrastructure モニターのサポートを使用可能に設定します。

### プロシージャ

1. 管理コンソールを開きます。
2. Business Process Choreographer が単一サーバー上に構成されている場合は、以下のステップを実行して、サーバーがビジネス・プロセス・イベントを生成できるようにします。
  - a. 左側のフレームで、「サーバー」を展開し、「アプリケーション・サーバー」 > *server\_name* をクリックします。
  - b. 「コンテナ設定」 > 「Business Process Choreographer Container の設定」 > 「Business Process Choreographer Container」を選択します。
  - c. 「状態監視者」をクリックします。
  - d. 「Business Flow Manager」と「Human Task Manager」の両方について、「監査ロギング」と「Common Event Infrastructure のロギング」のボックスにチェック・マークが付いていることを確認します。これらのチェック・ボックスが選択されていない場合は、これらを選択し、サーバーを再始動する必要があります。
3. Business Process Choreographer がクラスター上に構成されている場合は、以下のステップを実行して、クラスターがビジネス・プロセス・イベントを生成できるようにします。
  - a. 「クラスター」 > *cluster\_name* を選択します。
  - b. 「Business Process Choreographer Container の設定」 > 「Business Process Choreographer Container」を選択します。
  - c. 「状態監視者」をクリックします。
  - d. 「Business Flow Manager」と「Human Task Manager」の両方について、「監査ロギング」と「Common Event Infrastructure のロギング」のボックスにチェック・マークが付いていることを確認します。これらのチェック・ボックスが選択されていない場合は、これらを選択し、クラスターを再始動する必要があります。

### 次のタスク

これらのボックスのいずれかを選択する必要がある場合は、サーバーまたはクラスターを再始動して、変更を有効にする必要があります。

## サービス・コンポーネント・イベントのロギングの構成

プロセス・サーバー・モニターによって送出されるサービス・コンポーネント・イベントを収集するには、WebSphere Application Server のロギング機能を使用する方法を選択できます。アプリケーションの処理に関する問題を診断する場合は、ロガーを使用してイベント内のデータを表示します。

WebSphere Process Server では、基盤となる WebSphere Application Server の拡張ロギング機能を使用して、サービス・コンポーネント・イベント・ポイントでプロセス・サーバー・モニターによって送出されるイベントを収集できます。管理コンソールを使用して、モニターする特定のサービス・コンポーネント・イベント・ポイント、結果のサービス・コンポーネント・イベントに含まれる有効搭載量の詳細、および結果をパブリッシュする方法 (特定形式のファイルにパブリッシュしたり、コンソールに直接パブリッシュしたりするなど) を指定できます。モニター・ログには Common Base Event 形式でエンコードされたイベントが記録されており、イベント・エレメントに含まれている情報を使用して、サービス・コンポーネントの処理に関する問題をトレースすることができます。

WebSphere Application Server ロギング機能とトレース機能については、WebSphere Application Server の資料で詳細に文書化されており、製品全体でのロギングおよびトレースの使用法についても詳細に説明されています。このセクションでは、WebSphere Process Server に固有のサービス・コンポーネントに関連するロギングについて、補足情報のみを提供します。製品全体の他のコンポーネントでロギングおよびトレースを使用する場合については、WebSphere Application Server の資料を参照してください。

### 診断トレース・サービスの使用可能化

このタスクを使用して、診断トレース・サービスを使用可能に設定します。このサービスは、サービス・コンポーネント・イベントに含む詳細情報の量を管理できるロギング・サービスです。

#### 始める前に

Common Event Infrastructure (CEI) ロギングと監査ロギングを許可するには、ビジネス・プロセス・コンテナとヒューマン・タスク・コンテナを構成しておく必要があります。

#### このタスクについて

診断トレース・サービスは、サービス・コンポーネント・イベントのエレメントに含まれる詳細情報を取り込むために必要な詳細レベルを指定可能な唯一のロガー・タイプです。イベントをログに記録するには、プロセス・サーバーを始動する前に、診断トレース・サービスを使用可能にする必要があります。また、管理コンソールを使用して、CEI サーバーを使用してモニターするサービス・コンポーネント・イベント・ポイントを選択する場合も、このサービスを使用可能にする必要があります。

#### プロシージャ

1. ナビゲーション・ペインで、「サーバー」>「アプリケーション・サーバー」をクリックします。
2. 作業で使用するサーバーの名前をクリックします。
3. 「トラブルシューティング」の下で、「診断トレース・サービス (Diagnostic Trace service)」をクリックします。
4. 「構成」タブの「ログの使用可能化 (Enable log)」を選択します。
5. 「適用」をクリックし、次に「保管」をクリックします。
6. 「OK」をクリックします。

### 次のタスク

プロセス・サーバーがすでに始動している場合は、変更を有効にするためにプロセス・サーバーを再始動する必要があります。

### 管理コンソールを使用したロギング・プロパティの構成

このタスクを使用して、モニター機能がサービス・コンポーネント・イベントをロガー・ファイルにパブリッシュするように指定できます。

#### このタスクについて

WebSphere Process Server アプリケーションがモニター対象イベントをログに記録できるようにするには、モニターするサービス・コンポーネント・イベント・ポイント、イベントごとに必要な詳細レベル、およびイベントのログへのパブリッシュで使用する出力形式を指定する必要があります。管理コンソールを使用して、次の操作を実行できます。

- 特定のイベント・ログを使用可能または使用不可にする。
- ログの詳細レベルを指定する。
- ログ・ファイルの保管場所、保持するログ・ファイルの数、およびログ出力の形式を指定する。

ログ構成は、静的または動的に変更することができます。静的構成変更は、アプリケーション・サーバーの始動または再始動時にアプリケーションに反映されます。動的構成変更つまりランタイムの構成変更は、即座に適用されます。

ログの作成時、構成データに基づいてログのレベル値が設定されます。特定のログ名に対応する構成データがない場合、ログ・レベルの値はそのログの親から取得されます。親ログの構成データが存在しない場合は、さらにその親のログがチェックされ、ヌル以外のレベル値が検出されるまでツリーをさかのぼって同じ操作が実行されます。ログのレベルを変更すると、その変更はログの子に伝搬され、必要に応じてさらにその子に伝搬されます。

#### プロシージャ

1. ロギングを使用可能にし、ログの出力プロパティを設定します。
2. ナビゲーション・ペインで、「サーバー」>「アプリケーション・サーバー」をクリックします。
3. 作業で使用するサーバーの名前をクリックします。



4. 「トラブルシューティング」の下で、「**ロギングおよびトレース (Logging and tracing)**」をクリックします。
5. 「**ログ詳細レベルの変更 (Change Log Detail levels)**」をクリックします。
6. コンポーネント、パッケージ、およびグループのリストに、稼働中のサーバーに現在登録されているすべてのコンポーネントが表示されます。少なくとも 1 度呼び出されたプロセス・サーバー・イベントのみがこのリストに表示されます。ログ可能なイベント・ポイントを持つすべてのプロセス・サーバー・コンポーネントは、**WBILocationMonitor.LOG**、という名前前で始まるいずれかのコンポーネントの下にリストされます。
  - 構成を静的に変更するイベントを選択するには、「**構成**」タブをクリックします。
  - 構成を動的に変更するイベントを選択するには、「**ランタイム (Runtime)**」タブをクリックします。
7. ログに記録するイベントまたはイベント・グループを選択します。
8. イベントまたはイベント・グループごとにロギング・レベルを設定します。

注: CEI イベントのロギングでは、FINE、FINER、および FINEST レベルのみが有効です。

9. 「**適用**」をクリックします。
10. 「**OK**」をクリックします。
11. 静的な構成変更を有効にするには、プロセス・サーバーを停止して再始動します。

## 結果

デフォルトでは、ロガーは `trace.log` というファイルに出力をパブリッシュします。このファイルは、`install_root/profiles/profile_name/logs/server_name` フォルダにあります。

## チュートリアル: サービス・コンポーネント・イベントのロギング

このチュートリアルでは、ロガーにパブリッシュされるサービス・コンポーネント・イベント・モニターのセットアップ方法、およびログ・ファイルに記録されたイベントの表示方法について例を使用して説明します。この例のシナリオにより、プロセス・サーバーにすでにデプロイされ稼働しているアプリケーションで、モニターするサービス・コンポーネント・イベント・ポイントを選択する方法を示します。ここでは、アプリケーションの処理がそれらのイベント・ポイントのいずれかに到達した場合常にモニター機能がイベントを送出する仕組みについて知ることができます。送られる各イベントは標準化された Common Base Event 形式を取り、XML スtringとしてログ・ファイルに直接パブリッシュされます。

## このチュートリアルの目的

このチュートリアルを完了すれば、次の操作を実行できるようになります。

- モニターするサービス・コンポーネント・イベント・ポイントおよびプロセス・サーバーのロガーにパブリッシュされる出力を選択する。
- ログ・ファイルに保管されているイベントを表示する。

## このチュートリアルを完了するのに必要な時間

このチュートリアルを完了するには、およそ 15 分から 20 分かかります。

### 前提条件

このチュートリアルを実行するには、次の条件を満たす必要があります。

- プロセス・サーバーを構成および始動済みである。
- Common Event Infrastructure を構成済みである。
- プロセス・サーバーの診断トレース・サービスが使用可能に設定されている。
- プロセス・サーバーでサンプル・ギャラリー・アプリケーションがインストールおよび始動済みである。
- プロセス・サーバーでビジネス・ルール・サンプル・アプリケーションがインストールおよび始動済みである。「サンプル・ギャラリー」ページの指示に従って、ビジネス・ルール・サンプル・アプリケーションをセットアップし、実行します。

これらのすべての前提条件が満たされたら、チュートリアルに進む前に、少なくとも 1 度サンプル・ギャラリーからビジネス・ルール・サンプル・アプリケーションを実行してください。

### 例: ロガーによるイベントのモニター: このタスクについて

このシナリオではビジネス・ルール・サンプル・アプリケーションを使用するため、このアプリケーションが配置されている Web ページをあらかじめ開いておいてください。この Web ページは開いたままにしておいてください。モニター・パラメーターを指定した後、このサンプルを実行します。サンプルがモニター対象として選択可能な機能リストに表示されるように、少なくとも 1 度実行されていることを確認してください。

### プロシージャ

1. 管理コンソールを開きます。
2. ナビゲーション・ペインで、「サーバー」>「アプリケーション・サーバー」をクリックします。
3. `server_name` をクリックします。
4. 「トラブルシューティング」の下で、「ロギングおよびトレース (Logging and tracing)」をクリックします。
5. 「ログ詳細レベルの変更 (Change Log Detail levels)」をクリックします。
6. 「ランタイム (Runtime)」タブを選択します。
7. **WBILocationMonitor.LOG.BR** のツリーを展開します。  
**WBILocationMonitor.LOG.BR.brsample.\*** エレメントの下に 7 つのイベント・タイプが表示されます。最初のイベントは **WBILocationMonitor.LOG.BR.brsample\_module.DiscountRuleGroup** と呼ばれ、次の性質を持つ **Operation\_calculateDiscount** という名前の 1 つの関数が含まれています。
  - ENTRY
  - EXIT



- FAILURE
  - SelectionKeyExtracted
  - TargetFound
8. 各イベントをクリックして 「**finest**」 を選択します。
  9. 「**OK**」 をクリックします。
  10. 「ビジネス・ルール・サンプル・アプリケーション (business rules sample application)」 ページに切り替え、アプリケーションを 1 度実行します。
  11. テキスト・エディターを使用して、システムの `profile_root/logs/server_name` フォルダーに格納されている `trace.log` ファイルを開きます。

## 結果

サンプル・アプリケーションを実行したときにモニターによって送出されたビジネス・ルール・イベントを含むログ内の行が表示されます。これを見て気付く大きな点は、出力が Common Base Event 標準に準拠した長い未解析 XML ストリングで構成されていることです。ENTRY イベントと EXIT イベントを調べてください。ビジネス・オブジェクトが 16 進形式でエンコードされていることがわかります (詳細レベルとして「**finest**」を選択したためにビジネス・オブジェクトが含まれています)。この出力を、Common Event Infrastructure サーバーにパブリッシュされたイベントと比較してください。このサーバーは、XML を読み取り可能な表に解析し、ビジネス・オブジェクト・データを読み取り可能な形式にデコードします。この演習内の前のステップに戻り、詳細レベルを「**finest**」から「**fine**」または「**finer**」に変更して、イベント間の違いを比較することもできます。

この演習を完了すれば、ロガーを使用してモニターするサービス・コンポーネント・イベント・ポイントを選択する方法を理解したことになります。このタイプのモニターで送出されるイベントは標準形式で、結果は未加工の XML 形式のストリングとしてログ・ファイルに直接パブリッシュされることがわかりました。パブリッシュされたイベントの表示は、テキスト・エディターでログ・ファイルを開き、個別のイベントの内容を復号するという単純な作業で実行できます。

## 次のタスク

ビジネス・ルール・サンプル・アプリケーションのモニターを終了する場合は、ここで概要を示したステップを逆戻りして、サンプル・イベントの詳細レベルを「**info**」 にリセットしてください。

## ビジネス・ルールおよびセクターの監査ロギング

WebSphere Process Server で、ビジネス・ルールやセクターへの変更が自動的にログに記録されるようにセットアップすることができます。

サーバーで自動的にビジネス・ルールやセクターへの変更を検出したり、変更の詳細を説明するログ・ファイルにエントリーを作成するように構成することができます。標準 JVM SystemOut.log ファイルと指定したカスタム監査ログ・ファイルのどちらにログ・エントリーを書き込むかを選択できます。変更方法によって、各ビジネス・ルールやセクターを変更したプロセス・サーバーでは以下の内容がログに記録されます。

- 変更を行ったユーザーの名前

- 変更要求が出された場所
- 旧ビジネス・ルール・オブジェクトまたはセレクター・オブジェクト
- 旧オブジェクトから置き換えられた新規ビジネス・ルールまたはセレクター

ビジネス・ルール・オブジェクトおよびセレクター・オブジェクトは、置き換えられたビジネス・ルールまたはセレクターの場合でも、置き換えた新規バージョンの場合でも、完全なビジネス・ルール・セット、デシジョン・テーブル、ビジネス・ルール・グループ、またはセレクターです。ログを調べて (監査出力は Common Event Infrastructure データベースに出力できません)、旧/新ビジネス・ルールまたはセレクターを比較して変更箇所を判別してください。以下のシナリオで、ロギングが発生する状況 (ロギングが構成されている場合) と、ログ・エントリーの内容について説明します。

シナリオ	結果	ログ・エントリーの内容
ビジネス・ルール・マネージャーを使用したビジネス・ルールのパブリッシュ	要求	ユーザー ID、サーバー名 (該当する場合、セルとノードを含む)、旧ビジネス・ルール・ルールセット、新規ルールセット。
	失敗	ユーザー ID、サーバー名 (該当する場合、セルとノードを含む)、旧ビジネス・ルール・ルールセット、新規ルールセット。
リポジトリ・データベースの更新とコミット (ビジネス・ルール・マネージャーを使用したパブリッシュから)	成功	ユーザー ID、旧ルールセット、新規ルールセット。
	失敗	ユーザー ID、新規ルールセット。
セレクター・グループまたはビジネス・ルール・グループのエクスポート	要求	ユーザー ID、セレクター、またはビジネス・ルール・グループ名。
	成功	ユーザー ID、サーバー名 (該当する場合、セルとノードを含む)、エクスポートされたセレクター・グループまたはビジネス・ルール・グループのコピー。
	失敗	ユーザー ID、サーバー名 (該当する場合、セルとノードを含む)、セレクター・グループまたはビジネス・ルール・グループ名。

シナリオ	結果	ログ・エントリーの内容
セレクター・グループまたはビジネス・ルール・グループのインポート	要求	ユーザー ID、新規セレクター・グループまたはビジネス・ルール・グループのコピー。
	成功	ユーザー ID、サーバー名 (該当する場合、セルとノードを含む)、インポートされたセレクター・グループまたはビジネス・ルール・グループのコピー、インポートされたバージョンに置き換えられたセレクター・グループまたはビジネス・ルール・グループのコピー。
	失敗	ユーザー ID、サーバー名 (該当する場合、セルとノードを含む)、インポートされる予定だったセレクター・グループまたはビジネス・ルール・グループのコピー。
アプリケーションのインストール	成功	ユーザー ID、サーバー名 (該当する場合、セルとノードを含む)、セレクター・グループまたはビジネス・ルール・グループ名。
	失敗	ユーザー ID、サーバー名 (該当する場合、セルとノードを含む)、セレクター・グループまたはビジネス・ルール・グループ名。
アプリケーションの更新 (管理コンソールまたは wsadmin コマンドから)	成功	ユーザー ID、サーバー名 (該当する場合、セルとノードを含む)、新規セレクター・グループまたはビジネス・ルール・グループのコピー、旧セレクター・グループまたはビジネス・ルール・グループのコピー。
	失敗	ユーザー ID、サーバー名 (該当する場合、セルとノードを含む)、新規セレクター・グループまたはビジネス・ルール・グループのコピー。

シナリオ	結果	ログ・エントリーの内容
既存のビジネス・ルールかセレクター、またはその両方が開始されている以前にアプリケーションをデプロイ	成功	サーバー名 (該当する場合、セルとノードを含む)、セレクター・グループまたはビジネス・ルール・グループのコピー。
	失敗	サーバー名 (該当する場合、セルとノードを含む)、セレクター・グループまたはビジネス・ルール・グループのコピー。

## Common Event Infrastructure サーバーでのサービス・コンポーネントのモニター

サービス・コンポーネントのモニター結果を Common Event Infrastructure サーバーにパブリッシュするように選択することができます。サービス・コンポーネント・イベント・ポイントを Common Event Infrastructure サーバーによるモニター対象として指定できます。モニターは、アプリケーション・フローの表示および管理を行う永続ベースか、問題のトラブルシューティングを行う随時ベースかどちらかです。

WebSphere Process Server モニターを使用して、Common Event Infrastructure (CEI) バスを通して送られるサービス・コンポーネント・イベント内のサービス・コンポーネント・イベント・ポイントのデータをパブリッシュすることができます。このモニター方法によって、システムでのサービス・コンポーネントのアクティビティの分析をより柔軟に行うことができます。また、Common Base Event ブラウザーなどの CEI イベント用に最適化されたブラウザーを使用することもできます。Common Base Event ブラウザーはプロセス・サーバーに組み込まれています。

イベントは、ロガーに送信されるイベントと一致した内容で構造化されていますが、サービス・コンポーネント・イベントの分析のために特別に設計されたビューアーがアクセスできるように、データベースに保管されます。アプリケーションの作成時にサービス・コンポーネント・イベント・ポイントをアプリケーション内に指定して、アプリケーションがデプロイされ、サーバー上で稼働するようになった後にモニターが常時継続的に行われるようにします。これは「静的」モニターと呼ばれる方法です。静的モニターは、システム上のコンポーネント処理の適切なフローで特に重要なサービス・コンポーネント・イベント・ポイントで実行します。この情報を使用して、システムで稼働するサービス・コンポーネント・プロセスのすべてのアクション、およびそれらの相互作用を容易に監視できます。また、これらのプロセスの通常フローからの逸脱を素早く検出することもできます。この場合は、サービス・コンポーネントが正常に作動していない可能性があります。

サービス・コンポーネントの静的モニターを構成するには、WebSphere Integration Developer を使用して、プロセス・サーバーにデプロイするアプリケーションのサービス・コンポーネント・イベント・ポイントを選択します。選択されたサービス・コンポーネント・イベント・ポイントは、アプリケーションと共にデプロイされる、.mon という拡張子を持つ XML ファイルの形式で指定されます。稼働中のサーバーにデプロイした後は、アプリケーションの .mon ファイルに指定されているイ

メント・モニターの詳細レベルをオフにしたり、レベルを下げたりすることはできません。このようなモニターを停止するには、サーバーを停止してアプリケーションをアンデプロイする必要があります。 .mon ファイルを使用するアプリケーションの作成およびデプロイについて詳しくは、WebSphere Integration Developer インフォメーション・センターを参照してください。

「動的」モニター用のサービス・コンポーネント・イベント・ポイントも選択できます。この機能は、稼働中のサーバーにデプロイ済みのアプリケーション上で使用可能または使用不可能にすることができます。CEI サーバーを使用して動的モニターを実行する仕組みは、ロギングによってシステムの問題を診断およびトラブルシューティングする場合と基本的に同じです。出力はロガーにパブリッシュされるものと基本的に同じです。CEI バスを通して送られるイベントごとに構造を構成する Common Base Event エlementを使用します。また、ロギング・データと同様、詳細レベルの違いはイベント内でエンコードされる有効搭載量にのみ影響します。

## 管理コンソールを使用したサービス・コンポーネント・イベント・モニターの構成

管理コンソールを使用して、モニター機能がサービス・コンポーネント・イベントを Common Event Infrastructure サーバーにパブリッシュするように動的に指定することができます。

### 始める前に

ロガーの場合と同様に、診断トレース・サービスを使用可能にする必要があります。サーバーの再始動後、モニターするイベントを 1 回呼び出し、モニター可能なイベントのリストに表示されるようにします。

### このタスクについて

モニター対象のイベントを選択する方法は、プロセス・サーバーにデプロイ済みのアプリケーションに対して使用します。アプリケーションと共にプロセス・サーバーにデプロイされた .mon ファイルに指定されているイベントは、ここでの変更に関係なく、Common Event Infrastructure (CEI) データベースによってモニターされます。このようなイベントの場合は、CEI データベースに収集およびパブリッシュするための詳細レベルの値を大きくするだけで済みます。CEI データベースにパブリッシュされる出力は、ロガーによってパブリッシュされる出力に非常によく似ています。

### プロシージャ

1. 管理コンソールから、「トラブルシューティング」>「ロギングおよびトレース」をクリックします。
2. 「ログ詳細レベルの変更 (Change Log Detail levels)」をクリックします。
3. コンポーネント、パッケージ、およびグループのリストに、稼働中のサーバーに現在登録されているすべてのコンポーネントが表示されます。少なくとも 1 回呼び出されたプロセス・サーバー・イベントのみがこのリストに表示されます。ログ可能なすべてのプロセス・サーバー・イベントは、**WBILocationMonitor.CEI** という名前が始まるいずれかのコンポーネントの下にリストされます。
  - 構成を静的に変更するには、「構成」タブをクリックします。

- ・ 構成を動的に変更するには、「ランタイム (Runtime)」タブをクリックします。
4. モニターするイベントまたはイベント・グループを選択します。
  5. イベントごとに収集する情報の詳細レベルをクリックします。

注: CEI イベントの場合、FINE、FINER、および FINEST レベルのみが有効です。

6. 「適用」をクリックし、次に「保管」をクリックします。
7. 「OK」をクリックします。
8. 構成を静的に変更した場合は、変更を有効にするためにプロセス・サーバーを再始動する必要があります。

## 結果

モニター対象イベントの結果は、Common Base Event Browser で表示できます。

## チュートリアル: イベント・モニターでの Common Event Infrastructure サーバーの使用

このチュートリアルでは、Common Event Infrastructure (CEI) サーバーにパブリッシュされるサービス・コンポーネント・イベント・モニターのセットアップ方法、および Common Base Event ブラウザーで保管されたイベントを表示する方法について例を使用して説明します。このシナリオで使用する例では静的モニターは使用しないため、.mon ファイルを使用してデプロイされたアプリケーションは継続して特定のサービス・コンポーネント・イベント・ポイントモニターします。静的モニターの実行方法について詳しくは、IBM WebSphere Integration Developer インフォメーション・センターを参照してください。

その代わりに、この例で使用するシナリオでは、プロセス・サーバーにすでにデプロイされて実行されているアプリケーションのサービス・コンポーネントで、モニター・イベント・ポイントを選択する方法を示します。ここでは、アプリケーションの処理がそれらのイベント・ポイントのいずれかに到達した場合常にモニター機能がイベントを送出する仕組みについて知ることができます。送られる各イベントは CEI サーバーにパブリッシュされます。CEI サーバーにはデータベースに関するイベント情報が保管されます。イベントを表示するには、Common Base Event ブラウザーを使用します。

### このチュートリアルの目的

このチュートリアルを完了すれば、次の操作を実行できるようになります。

- ・ モニターするサービス・コンポーネント・イベント・ポイントを、CEI サーバーにパブリッシュされるイベントと共に選択する。
- ・ 保管されているイベントを Common Base Event ブラウザーによって表示する。

### このチュートリアルを完了するのに必要な時間

このチュートリアルを完了するには、およそ 15 分から 20 分かかります。



## 前提条件

このチュートリアルを実行するには、次の条件を満たす必要があります。

- プロセス・サーバーを構成および始動済みである。
- CEI およびそのデータベースを構成済みである。
- プロセス・サーバーの診断トレース・サービスが使用可能に設定されている。
- プロセス・サーバーでサンプル・ギャラリー・アプリケーションがインストールおよび始動済みである。
- プロセス・サーバーでビジネス・ルール・サンプル・アプリケーションがインストールおよび始動済みである。「サンプル・ギャラリー」ページの指示に従って、ビジネス・ルール・サンプル・アプリケーションをセットアップし、実行します。

これらのすべての前提条件が満たされたら、チュートリアルに進む前に、少なくとも 1 度サンプル・ギャラリーからビジネス・ルール・サンプル・アプリケーションを実行してください。

### 例: Common Event Infrastructure サーバーによるモニター: このタスクについて

このシナリオではビジネス・ルール・サンプル・アプリケーションを使用するため、このアプリケーションが配置されている Web ページをあらかじめ開いてください。この Web ページは開いたままにしておいてください。モニター・パラメーターを指定した後、このサンプルを実行します。サンプルを少なくとも 1 度実行しておいてください。実行しておくこと、サンプルがモニター対象として選択可能な機能のリストに表示されます。

### プロシージャ

1. 管理コンソールを開きます。
2. ナビゲーション・ペインで、「サーバー」>「アプリケーション・サーバー」をクリックします。
3. `server_name` をクリックします。
4. 「トラブルシューティング」の下で、「ロギングおよびトレース (Logging and tracing)」をクリックします。
5. 「ログ詳細レベルの変更 (Change Log Detail levels)」をクリックします。
6. 「ランタイム (Runtime)」タブを選択します。
7. **WBILocationMonitor.CEI.BR** のツリーを展開します。  
**WBILocationMonitor.CEI.BR.brssample.\*** エレメントの下に以下の 5 つのイベント・タイプが表示されます。各イベント・タイプには、**Operation.\_calculateDiscount** 関数によって付加された名前 **WBILocationMonitor.CEI.BR.brssample\_module.DiscountRuleGroup** と、次の性質が含まれています。
  - ENTRY
  - EXIT
  - FAILURE
  - SelectionKeyExtracted
  - TargetFound



8. 各イベントをクリックして「**finest**」を選択します。
9. 「**OK**」をクリックします。
10. 「ビジネス・ルール・サンプル・アプリケーション (business rules sample application)」ページに切り替え、アプリケーションを 1 度実行します。
11. 管理コンソールに戻り、ナビゲーション・ペインから、「統合アプリケーション」>「**Common Base Event ブラウザー**」を選択します。
12. Network Deployment 環境内のノードでサーバーを実行している場合は、「イベント・データ・ストア」フィールドにサーバーとノードの名前が含まれるように変更する必要がある場合もあります。cell/nodes/node\_name/servers/server\_name/ejb/com/ibm/events/access/EventAccess というストリングを入力します。
13. 「イベントの取得」を押します。

## 結果

Common Base Event ブラウザーの上側のペインに、サンプル・アプリケーションを実行したときに CEI サーバーにパブリッシュされた 4 つのビジネス・ルール・イベントのリストが表示されます。イベントの 1 つを選択してください。下側のペインにイベントの内容が表示されます。このイベントとロガーにパブリッシュされたイベントを比較してください。CEI サーバーにパブリッシュされた元の XML ストリングはブラウザーによって解析済みで、ENTRY イベントと EXIT イベントのビジネス・オブジェクト・コードは、元の 16 進形式から読み取り可能な XML に変換されています。この演習内の前のステップに戻り、詳細レベルを「**finest**」から「**fine**」または「**finer**」に変更して、イベント間の違いを比較することもできます。

この演習を完了すれば、CEI サーバーを使用してモニターするサービス・コンポーネント・イベント・ポイントを選択する方法を理解したことになります。このタイプのモニターで送出されるイベントは標準形式で、結果はデータベースにパブリッシュされることがわかりました。また、Common Base Event ブラウザーを使用してデータベースからイベントを取得し、個別のイベント情報を解析済み表形式でブラウザー上に表示することもできるようになります。

## 次のタスク

ビジネス・ルール・サンプル・アプリケーションのモニターを終了する場合は、ここで概要を示したステップを逆戻りして、サンプル・イベントの詳細レベルを「**info**」にリセットしてください。

## セッション・モニター

Common Base Event ブラウザーを使用して、Common Event Infrastructure データベースで同じセッション ID 属性を持つすべてのイベントを検索することによって、同じセッションの一部である複数のイベントをモニターすることができます。

WebSphere Process Server には、単一セッションの一部であるすべてのサービス・コンポーネント・イベントを識別できる拡張機能があります。Common Base Event の標準エレメントには、contextDataElement エレメントの下の WBISESSIONID という名前の属性があります。各セッションの固有 ID はこの属性に保管されているので、そのセッションの一部であったすべてのサービス・コンポーネント・イベントを識別できます。Common Base Event ブラウザーの「**SessionID**」フィールドを使用

して、Common Event Infrastructure (CEI) データベースに保管されていて指定したセッション ID と一致するイベントを検索できます。この機能を使用すると、すべてのサービス・コンポーネント・イベントのプロセス・フローおよびプロセス・コンテンツを簡単に確認することができます。この情報を使用して、アプリケーションの効率性を評価したり、特定の環境の下でのみ発生する問題を診断したりできます。

**Common Base Event** ブラウザーを使用して、イベントおよび関連するコンテンツに関する戻されたリストを表示できます。「すべてのイベント」ビューをクリックすると、「失敗」と「ビジネス・プロセス」という 2 つの列が表示されます。特定のイベントの「失敗」列にリンクがある場合、そのリンクをクリックして失敗したイベントの詳細を表示できます。同様に、特定のイベントに関連付けられた「ビジネス・プロセス」にリンクがある場合、そのリンクをクリックして **Business Process Explorer** を開き、ビジネス・プロセス・イベントまたはヒューマン・タスク・イベントの詳細を表示できます。



---

## 第 3 章 モニター対象イベントの表示

モニター対象イベントのパブリッシュ結果を表示する方法は多数あり、使用するモニターのタイプによって異なります。このセクションでは、Common Event Infrastructure データベースに保管されているパフォーマンス・データ、イベント・ログ、およびサービス・コンポーネント・イベントを表示するのに使用できる方法について説明します。

---

### Tivoli Performance Viewer によるパフォーマンス・メトリックの表示

このトピックでは、Tivoli Performance Viewer を使用してパフォーマンス・モニターを開始および停止する方法、パフォーマンス・モニター・インフラストラクチャー・データをグラフ形式または表形式でリアルタイムに表示する方法、およびオプションで、後で同じビューアーで再確認できるファイルにデータを記録する方法について説明します。

#### 始める前に

ここでは、ノード上に 1 つ以上のサーバーが作成されて稼働しており、パフォーマンス・モニター・インフラストラクチャー (PMI) が使用可能で、モニターするサービス・コンポーネント・イベント・ポイントが、ビューアー内から選択できるように少なくとも 1 回呼び出されていることが前提となっています。

#### このタスクについて

Tivoli Performance Viewer (TPV) は、プロセス・サーバーのパフォーマンスの全局面について、さまざまな詳細情報を表示することができる強力なアプリケーションです。『Tivoli Performance Viewer (TPV) を使用したパフォーマンスのモニター (Monitoring performance with Tivoli Performance Viewer)』セクションで、このツールの目的別使用方法について詳しく説明しています。このプログラムを使用する場合の詳細な説明については、この部分を参照してください。このセクションでは、WebSphere Process Server の固有イベントのパフォーマンス・データの表示についてのみ説明します。

Performance Viewer により、管理者およびプログラマーは、WebSphere Process Server の現在の正常性をモニターできます。データの収集と表示はプロセス・サーバーで行われるため、パフォーマンスに影響があります。パフォーマンスへの影響を最小限に抑えるには、モニターするアクティビティーが行われるサーバーのみをモニターしてください。

**注:** これらの統計を表示する場合、カウンター型統計と期間型統計を混合しないでください。カウンターは累積されるため、アプリケーションによってはグラフのスケールがすぐに大きくなる可能性があります。反対に、期間型統計はシステムが各イベントを処理するのにかかる時間の平均を示すため、一定の範囲に保たれます。その結果、統計とその相対スケールの間不均衡が生じ、ビューアー・グラフで片方のタイプの統計がスキュー表示されます。

- 現在のパフォーマンス・アクティビティーの表示

1. 管理コンソールのナビゲーション・ツリーで、「**モニターとチューニング (Monitoring and Tuning)**」>「**Performance Viewer**」>「**現在のアクティビティ (Current Activity)**」をクリックします。
2. 「**サーバー**」を選択し、モニターするアクティビティがあるサーバーの名前をクリックします。別の方法として、モニターするアクティビティがあるサーバーの**チェック・ボックス**を選択し、「**モニターの開始 (Start Monitoring)**」をクリックすることもできます。複数のサーバーのモニターを同時に開始するには、複数のサーバーを選択し、「**モニターの開始 (Start Monitoring)**」をクリックします。
3. 「**パフォーマンス・モジュール (Performance Modules)**」を選択します。
4. 表示する各パフォーマンス・モジュール名の横にある**チェック・ボックス**を選択します。パフォーマンス統計を生成し、少なくとも 1 回呼び出されたすべてのプロセス・サーバー固有のイベントが、**WBISStats.RootGroup** 階層の下にリストされます。ツリーを展開する場合はノードの横の **+** をクリックし、縮小させる場合はノードの横の **-** をクリックします。
5. 「**モジュールの表示 (View Modules)**」をクリックします。ページの右側に、要求されたデータを示すグラフまたは表が表示されます。デフォルトではグラフが表示されます。

各モジュールには、複数のカウンターが関連付けられています。カウンターは、データのグラフまたは表の下にある表に表示されます。選択されたカウンターは、グラフまたは表内に表示されます。カウンターの横にある**チェック・ボックス**を選択またはクリアすることによって、グラフや表にカウンターを追加またはグラフや表からカウンターを削除できます。デフォルトでは、モジュールごとに最初の 3 つのカウンターが表示されます。

カウンターは 20 個まで選択可能で、「**現在のアクティビティ (Current Activity)**」モードの TPV に表示できます。

6. オプション: グラフまたは表からモジュールを削除するには、モジュールの横の**チェック・ボックス**をクリアし、再度「**モジュールの表示 (View Modules)**」をクリックします。
  7. オプション: データを表形式で表示するには、カウンターを選択する表で「**表の表示 (View Table)**」をクリックします。切り替えてグラフに戻すには、「**グラフの表示 (View Graph)**」をクリックします。
  8. オプション: グラフの凡例を表示するには、「**凡例の表示 (Show Legend)**」をクリックします。凡例を非表示にするには、「**凡例の非表示 (Hide Legend)**」をクリックします。
  9. イベントのパフォーマンスのモニターが終了したら、「**Tivoli Performance Viewer**」をクリックし、モニターしていたサーバーを選択して、「**モニターの停止 (Stop Monitoring)**」をクリックします。
- **パフォーマンス統計のログ記録**

サーバー上でモニターがアクティブな間は、現在使用可能なすべての PMI カウンターのデータをログに記録し、結果を TPV ログ・ファイルに記録することができます。毎回 20 個までのカウンターを組み合わせ、特定期間の TPV ログ・ファイルを複数回表示することができます。特定期間内のサーバーでの異なるパフォーマンス測定の間を柔軟に監視できます。

1. サマリー・レポートまたはパフォーマンス・モジュールを表示する場合は、「**ロギングの開始 (Start Logging)**」をクリックします。
2. 終了したら、「**ロギングの停止 (Stop Logging)**」をクリックします。 デフォルトでは、ログ・ファイルは、サーバーが稼働しているノードの `profile_root/logs/tpv` ディレクトリーに保管されます。TPV は保管スペースへの書き込みが終了すると、ログ・ファイルを自動的に圧縮します。この時点で、各圧縮ファイルに含まれるログ・ファイルが 1 つのみで、その名前は圧縮ファイルと同じになっている必要があります。
3. ログを表示するには、管理コンソールのナビゲーション・ツリーで、「**モニターとチューニング (Monitoring and Tuning)**」>「**Performance Viewer**」>「**ログの表示 (View Logs)**」をクリックします。

---

## サービス・コンポーネント・イベント・ログ・ファイルの表示と解釈

このトピックでは、サービス・コンポーネント・モニターによって生成されたログ・ファイルにある情報を解釈する方法について説明します。ログ・ファイルは、管理コンソールのログ・ビューアーで表示したり、任意の個別のテキスト・ファイル・エディターで表示したりできます。

サービス・コンポーネント・モニターによってロガーに送出されるイベントは、Common Base Event 形式でエンコードされます。ログ・ファイルへのパブリッシュ時に、イベントは XML タグ付け形式の単一の長いテキスト行として組み込まれます。これには、以下のような複数のロガー固有のフィールドも含まれます。ログに記録されたイベントの Common Base Event コーディングを復号する方法について詳しくは、この資料のイベント・カタログのセクションを参照してください。このセクションを参照することにより、ログ・ファイルの各項目に含まれる他のフィールドや、ロガーの構成時に選択したログ・ファイル用の形式がどのように構造化されるかについて理解できます。

### 基本形式フィールドと拡張形式フィールド

プロセス・サーバーでは、ファイルまたはメモリー内の循環バッファーにロギング出力を送信できます。トレース出力をメモリー内の循環バッファーに送信する場合は、表示できるようにするためにまずトレース出力をファイルにダンプする必要があります。出力はプレーン・テキストとして生成されます。形式は基本、拡張、またはログ・アナライザーのうちからユーザーが指定した形式になります。出力の基本形式と拡張形式は、メッセージ・ログで使用可能な基本形式と拡張形式に似ています。基本形式と拡張形式で使用するフィールドおよびフォーマット手法の多くは同一のものです。これらの形式で使用可能なフィールドは次の通りです。

#### TimeStamp

タイム・スタンプは、フォーマット設定するプロセスのロケールを使用してフォーマット設定されます。タイム・スタンプには完全修飾日付 (YYMMDD)、ミリ秒単位までの 24 時間表示、および時間帯が含まれます。

#### ThreadId

トレース・イベントを発行したスレッドのハッシュ・コードから生成される 8 文字の 16 進値。

#### ThreadName

メッセージまたはトレース・イベントを発行した Java スレッドの名前。

**ShortName**

トレース・イベントを発行したロギング・コンポーネントの省略名。一般には、WebSphere Process Server の内部コンポーネントのクラス名ですが、ユーザー・アプリケーションのその他の ID である場合もあります。

**LongName**

トレース・イベントを発行したロギング・コンポーネントの絶対パス名。一般には、WebSphere Process Server の内部コンポーネントの完全修飾クラス名ですが、ユーザー・アプリケーションのその他の ID である場合もあります。

**EventType**

トレース・イベントのタイプを示す 1 文字フィールド。トレース・タイプは小文字です。次の値があります。

- 1 FINE または EVENT タイプのトレース・エントリ。
- 2 FINER タイプのトレース・エントリ。
- 3 FINEST、DEBUG、または DUMP タイプのトレース・エントリ。
- Z トレース・タイプが認識されなかったことを示すプレースホルダー。

**ClassName**

メッセージまたはトレース・イベントを発行したクラス。

**MethodName**

メッセージまたはトレース・イベントを発行したメソッド。

**Organization**

メッセージまたはトレース・イベントを発行したアプリケーションを所有する組織。

**Product**

メッセージまたはトレース・イベントを発行した製品。

**Component**

メッセージまたはトレース・イベントを発行した製品内のコンポーネント。

**基本形式**

基本形式で表示されるトレース・イベントでは、次の形式が使用されます。

```
<timestamp><threadId><shortName><eventType>[className] [methodName] <textmessage>
    [parameter 1]
    [parameter 2]
```

**拡張形式**

拡張形式で表示されるトレース・イベントでは、次の形式が使用されます。

```
<timestamp><threadId><eventType><UOW><source=longName>[className] [methodName]
<Organization><Product><Component> [thread=threadName]
<textMessage>[parameter 1=parameterValue] [parameter 2=parameterValue]
```

**ログ・アナライザー形式**

ログ・アナライザー形式を指定することにより、WebSphere Application Server に組み込まれたアプリケーションである Log Analyzer ツールを使用して、トレース出力を開くことができます。この形式では、Log Analyzer のマージ機能を使用できるため、2 つの異なるサーバー・プロセスのトレースを関連させる場合に便利です。



---

## Common Base Event ブラウザーでのイベントの表示

イベントを選択、ソート、および表示するには、Common Base Event ブラウザーを使用します。

### 始める前に

このタスクでは、管理コンソールにログインしていることを前提としています。

### このタスクについて

イベント・ブラウザーは、イベント・アクセス・インターフェースを使用して、イベント・データを照会します。照会の結果はブラウザーに表示されます。

### プロシージャ

1. まずイベント・ブラウザーを開きます。管理コンソールのナビゲーション・ペインで、「統合アプリケーション」を選択してから「**Common Base Event ブラウザー**」をクリックします。
2. 表示するイベントを指定します。
3. 戻されたイベントのビューを選択します。
4. どのブラウザー・パネルでも、検索基準またはソート基準の選択を完了してから、ブラウザー・パネルの下部にある「**イベントの取得**」ボタンをクリックして、イベントを表示します。

## 表示するイベントの指定

イベント・データベース内のイベントを照会する検索条件を指定する場合の Common Base Event ブラウザーの使用方法

### 始める前に

このタスクでは、イベント・ブラウザーがすでに開かれており、「イベントの取得」パネルが表示されていることが前提となります。

「イベント・データ・ストア・プロパティ」フィールドの入力は必須です。「イベント・フィルター・プロパティ」フィールドはオプションで、時刻、日付、サーバー名、サブコンポーネント名、およびイベント重大度の各パラメーターを基にしてイベント検索を絞り込むことができます。

### プロシージャ

1. 必須: 検索するイベント・データ・ストアを指定します。

このフィールドは Java Naming and Directory Interface (JNDI) 名です。つまり、管理コンソールで構成可能な Enterprise JavaBeans (EJB) 参照です。サーバーのデフォルトは `java:comp/env/eventsaccess` ですが、Network Deployment 環境で作業する場合は別の名前の指定が必要になることがあります (JNDI のネーミングについての詳細は、WebSphere Application Server Network Deployment バージョン 6.1 の資料を参照してください)。

2. 必須: 検索するイベント・グループを指定します。

これは、イベントの取得元となるイベント・グループです。デフォルト・グループは、All events です。

3. 必須: 取得するイベント数を指定します。

検索するイベントの最大数は 500 です。

4. オプション: レポートの作成日 (カレンダー期間) を指定します。

開始日と終了日を入力します。

5. オプション: レポートの作成時間 (時間枠) を指定します。

開始時刻と終了時刻を入力します。

6. オプション: サーバー名を指定します。

7. オプション: 該当する場合は、サブコンポーネント名を指定します。

8. オプション: イベントの優先順位を指定します。取得するイベント優先順位の範囲は、0 (優先順位が最も低い) から 100 (優先順位が最も高い) です。

9. オプション: イベントの重大度を指定します。

取得するイベント重大度の範囲は、0 (重大度が最も低い) から 70 (重大度が最も高い) です。

10. 「イベントの取得」をクリックします。

## 結果

検索条件に一致する Common Base Event の数が表示されます。照会した結果が表示されない場合は、「トラブルシューティングとサポート」PDF ファイルの『Common Base Event ブラウザーのトラブルシューティング』のトピックを参照してください。

戻されたイベントを表示するには、ナビゲーション・バーからビューを選択します。「すべてのイベント」、「BPEL プロセス・イベント」、「ユーザー・データ・イベント」、または「サーバー・イベント」の中からクリックできます。イベント・データを表示する場合は、「イベントの取得」をクリックすることによっていつでも検索条件を変更することができます。

## 次のタスク

イベントが戻されたら、それら进行处理してさまざまなレベルのイベント詳細を取得できます。

## イベント・ブラウザーから戻されるイベントの処理

照会により戻されるイベントを表示するには、イベント・ブラウザーを使用します。

### 始める前に

このタスクでは、実行依頼した照会から戻されるデータ (トピック『表示するイベントの指定』で説明) を扱います。

### このタスクについて

照会により、基準を満たすすべてのイベントが戻されます。

### プロシージャ

1. ナビゲーション・バーからビューをクリックしてください。

ナビゲーション・バーには、戻された照会の次に示すビューが表示されます。

#### すべてのイベント

戻されたすべてのイベントです。

#### BPEL プロセス・イベント

特定のプロセス・インスタンスの Business Process Choreographer イベント。

#### ユーザー・データ・イベント

拡張名 ECS:UserDataEvent を持つイベントです。このイベント・タイプは、ECSEmitter クラスの addUserDataEvent メソッドによって作成されます。

#### サーバー・イベント

特定サーバーのイベント。

2. 以下のいずれかのアクションを実行します。
  - ステップ 1 で「**BPEL プロセス・イベント**」をクリックした場合、プロセス・テンプレートをクリックしてから、プロセス・インスタンスをクリックする必要があります。
  - ステップ 1 で「**サーバー・イベント**」 をクリックした場合、サーバーをクリックする必要があります。
3. イベントをクリックして、ブラウザ・ウィンドウの下部にあるペインにイベント・データを表示します。

## Common Base Event ブラウザーのトラブルシューティング

Common Base Event ブラウザーにアクセスできなくなる主な条件は 4 つあります。

### 条件

#### 「サーバーが見つからない」

サーバーが使用不可です。イベント・ブラウザの URI を起動しようとするすると、「サーバーが見つかりません」というブラウザ・ページが戻され、サーバーが使用不可であることが示されます。この場合、サーバー管理者に連絡して、問題の原因を判別する必要があります。

#### 「ファイルが見つからない」

サーバーは使用可能だが、イベント・ブラウザ・アプリケーションがサーバーにインストールされていないか、開始していない可能性があります。イベント・ブラウザの URI を起動しようとするすると、「ファイルが見つかりません」というブラウザ・ページが戻され、サーバーは使用可能だが、そのサーバーでその URI は使用不可であることが示されます。この場合、サーバー管理者に連絡して、問題の原因を判別する必要があります。

#### ログオン・パネルが表示される

サーバーとイベント・ブラウザは使用可能だが、ユーザーがイベント・ブ

ブラウザにアクセスするための適切なロールにマップされていません。ログイン・パネルに入力するようプロンプトが出されます。ユーザー ID とパスワードを入力してログインしようとする、失敗します。この場合、サーバー管理者に連絡して、イベント・ブラウザを起動するための適切な許可を得る必要があります。

#### 「イベントの取得」パネルにエラー・メッセージが表示される

サーバーとイベント・ブラウザは使用可能であり、ユーザーはアクセスするための適切な権限を持っているが、Common Event Infrastructure サーバーが使用不可です。「イベントの取得」ボタンをクリックすると、イベント・ブラウザの「イベントの取得」パネルにエラー・メッセージが表示されます。エラー情報はメッセージ・ログに記録されます。

## 第 4 章 イベント・カタログ

イベント・カタログには、サービス・コンポーネント・タイプごとにモニター可能なすべてのイベントの仕様、および各イベントによって作成される、関連付けられた Common Base Event 拡張データ・エレメントが収容されています。このセクションで提供されている情報を参照資料として使用してください。それにより、個別のイベントの構造が理解できます。この知識によって、各イベントに含まれている情報を復号し、各イベントで生成される比較的大容量のデータから必要な情報を素早く識別することができます。このセクションには、Common Base Event の構造および標準エレメント、Business Process Choreographer サービス・コンポーネントおよび WebSphere Process Server 固有サービス・コンポーネントのイベント・リスト、および各イベント・タイプに固有の Common Base Event への拡張機能などの情報が含まれています。また、サービス・コンポーネントで処理されるビジネス・オブジェクトがサービス・コンポーネント・イベントに取り込まれる方法についても説明します。

指定されたタイプのイベントは、Common Event Infrastructure (CEI) バスを通して CEI サーバーまたはロガーに送出される場合、Common Base Event の形式を取ります。これは、基本的には、イベント・カタログの仕様に従って作成されたイベント・エレメントのカプセル化形式の XML です。Common Base Event には、標準エレメント、プロセス・サーバー・コンポーネント識別エレメント、イベント関連領域 ID、および各イベント・タイプに固有の追加エレメントが含まれています。これらのすべてのエレメントは、イベントがサービス・コンポーネント・モニターによって送出されたときにはいつでも、CEI サーバーまたはロガーに渡されます。ただし例外として、イベントに有効搭載量内のビジネス・オブジェクト・コードが組み込まれている場合は、イベントに組み込むビジネス・オブジェクト・データの量を指定します。

### Common Base Event の標準エレメント

ここでは、サービス・コンポーネント・モニターから送出されるすべてのイベントに含まれる Common Base Event のエレメントをリストします。

属性	説明
version	1.0.1 に設定します。
creationTime	イベントが作成された時刻 (UTC 形式)。
globalInstanceId	Common Base Event インスタンスの ID。この ID は自動的に生成されます。
localInstanceId	この ID は自動的に生成されます (空白の場合もあり)。
severity	ビジネス・プロセスまたはヒューマン・タスクにイベントが及ぼす影響。この属性は 10 (情報) に設定されます。それ以外の場合、これは使用されません。
priority	未使用。
reporterComponentId	未使用。
locationType	Hostname に設定します。

属性	説明
location	実行中のサーバーのホスト名に設定します。
application	未使用。
executionEnvironment	オペレーティング・システムを示すストリング。
component	プロセス・サーバーのバージョン。ビジネス・プロセスとヒューマン・タスクの場合、WPS# と、その後に現行プラットフォームの ID と基盤となるソフトウェア・スタックのバージョン ID を設定します。
componentType	Apache QName 形式を基にしたコンポーネント QName。  ビジネス・プロセスの場合、次のように設定します。  www.ibm.com/namespaces/autonomic/Workflow_Engine  ヒューマン・タスクの場合、次のように設定します。  www.ibm.com/xmlns/prod/websphere/scdl/human-task
subComponent	監視可能なエレメント名。  ビジネス・プロセスの場合、BFM に設定します。 ヒューマン・タスクの場合、HTM に設定します。
componentIdType	ProductName に設定します。
instanceId	サーバーの ID。この ID の形式は、 <i>cell_name/node_name/server_name</i> です。区切り文字はオペレーティング・システムによって異なります。
processId	オペレーティング・システムのプロセス ID。
threadId	Java 仮想マシン (JVM) のスレッド ID。
Situation Type	イベントが報告される原因となったシチュエーションのタイプ。プロセス・サーバー固有コンポーネントの場合は、ReportSituation に設定します。
Situation Category	イベントが報告される原因となったシチュエーション・タイプのカテゴリ。プロセス・サーバー固有コンポーネントの場合は、STATUS に設定します。
Situation Reasoning Scope	報告されたシチュエーションの影響の有効範囲。プロセス・サーバー固有コンポーネントの場合は、EXTERNAL に設定します。
ECSCurrentID	現在のイベント相関範囲 ID の値。
ECSParentID	親イベント相関範囲 ID の値。
WBISessionID	現在のセッション ID の値。
extensionName	イベント名に設定します。

## イベント内のビジネス・オブジェクト

ビジネス・オブジェクト・データは、バージョン 6.1 以降、イベントの中で XML 形式で搬送されます。Common Base Event 形式は xs:any スキーマを含んでおり、これは、ビジネス・オブジェクト・ペイロードを XML エレメント内にカプセル化します。

サービス・コンポーネント・イベントに収集するビジネス・オブジェクトの詳細レベルを指定します。この詳細レベルは、イベントに渡されるビジネス・オブジェクト・コードの量にのみ影響します。その他のすべての Common Base Event エlement (標準とイベント固有の両方) は、イベントにパブリッシュされます。サービス・コンポーネント・イベントに適用可能な詳細レベルの名前は、WebSphere Integration Developer を使用して静的モニターを作成したか、または管理コンソールで動的モニターを作成したかに応じて異なりますが、次の表に示すように対応しています。

管理コンソールの詳細レベル	Common Base Event/WebSphere Integration Developer の詳細レベル	パブリッシュされる有効搭載量情報
FINE	EMPTY	なし。
FINER	DIGEST	有効搭載量の説明のみです。
FINEST	FULL	有効搭載量のすべてです。

詳細レベルは、イベント・インスタンス・データに含まれている PayloadType Element によって指定されます。実際のビジネス・オブジェクト・データは、モニターが FULL/FINEST の詳細を記録する設定になっていれば、イベントのみに組み込まれます。ビジネス・オブジェクト・データ自体は、xsd:any スキーマの下で Common Base Event に組み込まれています。実際には、プロセス・サーバーのビジネス・オブジェクト・ペイロードは wbi:event という名前のルート・Element によって表示されます。イベント出力をロガーにパブリッシュすると、ログ・ファイルの参照時に出力が表示されます。イベントが CEI サーバーに対してパブリッシュされている場合は、Common Base Event ブラウザーを使用してイベントを表示できます。その場合、wbi:event リンクをクリックすると、ビジネス・オブジェクト・データが表示されます。

## Business Process Choreographer イベント

WebSphere Process Server には、ビジネス・プロセスとヒューマン・タスクで使用する Business Process Choreographer サービス・コンポーネントが組み込まれています。このセクションでは、これらのコンポーネントでモニター可能なイベント・ポイントについて説明します。

### ビジネス・プロセス・イベントのモニター

ビジネス・プロセスのために発行されるイベントは、状態非依存データとビジネス・プロセス・イベントに固有のデータで構成されています。ここでは、ビジネス・プロセス・イベントに固有の属性と Element について説明します。

ビジネス・プロセス・イベントには、次に示すイベント内容のカテゴリがあります。

#### ビジネス・プロセス固有のイベント・データ

ビジネス・プロセスでは、イベントはプロセス、アクティビティ、スコープ、リンク、および変数に関連付けられます。これらの各イベント・タイプのオブジェクト固有の内容について説明します。



Business Process Choreographer バージョン 6.1 では、2 つのイベント形式が使用されることがあります。

### WebSphere Business Monitor 6.0.2 形式

WebSphere Business Monitor 6.0.2 形式のイベントは、WebSphere Integration Developer 6.0.2 でモデリングされているプロセスがある場合、または WebSphere Integration Developer 6.1 で WebSphere Business Monitor 6.0.2 形式モードが使用可能になっている場合に発生します。特に明記されていない限り、これらのイベントのオブジェクト固有の内容は、タイプがストリングの *extendedDataElement* XML エレメントとして書き込まれます。

### WebSphere Business Monitor 6.1 形式

WebSphere Business Monitor 6.1 形式のイベントは、WebSphere Integration Developer 6.1 でモデリングされたプロセスがあり、WebSphere Business Monitor 6.1 形式モードが使用可能になっている場合に発生します。これらのイベントのオブジェクト固有の内容は、Common Base Event の *eventPointData* フォルダの *xs:any* スロットに XML エレメントとして書き込まれます。また、有効搭載量のメッセージは *applicationData* セクションに書き込まれます。XML の構造は、XML スキーマ定義 (XSD) ファイル *BFMEvents.xsd* で定義されています。このファイルは、*install\_root\ProcessChoreographer\client* ディレクトリーにあります。

## プロセス

プロセス・インスタンスのイベントには、以下のようなオブジェクト固有のイベント内容が含まれています。

属性	説明
<i>processTemplateName</i>	インスタンスが派生したプロセス・テンプレートの名前
<i>processTemplateValidFrom</i>	テンプレートの有効日
<i>processTemplateId</i>	プロセス・テンプレートの ID
<i>processInstanceDescription</i>	オプション: プロセス・インスタンスの説明
<i>processInstanceExecutionState</i>	プロセスの状態を表すストリング値。フォーマットは次のとおりです。 <i>state number-state description</i> 。この属性は、以下の値のうちのいずれかをとることができます。  1 - STATE_READY 2 - STATE_RUNNING 3 - STATE_FINISHED 4 - STATE_COMPENSATING 5 - STATE_FAILED 6 - STATE_TERMINATED 7 - STATE_COMPENSATED 8 - STATE_TERMINATING 9 - STATE_FAILING 10 - STATE_INDOUBT 11 - STATE_SUSPENDED 12 - STATE_COMPENSATION_FAILED
<i>principal</i>	現行の実行ステップの実行対象プリンシパル。これは通常、プロセスのスターターです。

属性	説明
PayloadType	有効搭載量タイプ。ストリングの値は、none、digest、または full のいずれかです。この値は、WebSphere Integration Developer の設定によって決まります。使用可能な明示的なモニター定義がない場合、例えば「 <b>デフォルトのイベントを有効にする (Enable default events)</b> 」が選択されている場合は、デフォルトの有効搭載量タイプは full になります。

## アクティビティおよびスコープ

アクティビティおよびスコープには、以下のようなオブジェクト固有のイベント内容が含まれています。

属性	説明
processTemplateName	インスタンスが派生したプロセス・テンプレートの名前。
processTemplateValidFrom	テンプレートの有効日。
activityTemplateName	オプション: インスタンスが派生したアクティビティ・テンプレートの名前。
activityInstanceDescription	オプション: アクティビティ・インスタンスの説明。
activityKind	<p>アクティビティの種類を示すストリング値。この値のフォーマットは、次のとおりです。 <i>kind number-kind description</i>。この属性は、以下の値のうちのいずれかをとることができます。</p> <ul style="list-style-type: none"> <li>3 - KIND_EMPTY</li> <li>21 - KIND_INVOKE</li> <li>23 - KIND_RECEIVE</li> <li>24 - KIND_REPLY</li> <li>25 - KIND_THROW</li> <li>26 - KIND_TERMINATE</li> <li>27 - KIND_WAIT</li> <li>29 - KIND_COMPENSATE</li> <li>30 - KIND_SEQUENCE</li> <li>32 - KIND_SWITCH</li> <li>34 - KIND_WHILE</li> <li>36 - KIND_PICK</li> <li>38 - KIND_FLOW</li> <li>40 - KIND_SCOPE</li> <li>42 - KIND_SCRIPT</li> <li>43 - KIND_STAFF</li> <li>44 - KIND_ASSIGN</li> <li>45 - KIND_CUSTOM</li> <li>46 - KIND_RETHROW</li> <li>47 - KIND_FOR_EACH_SERIAL</li> <li>48 - KIND_FOR_EACH_PARALLEL</li> <li>1000 - SQLSnippet</li> <li>1001 - RetrieveSet</li> <li>1002 - InvokeInformationService</li> <li>1003 - AtomicSQLSnippetSequence</li> </ul>

属性	説明
state	<p>アクティビティの状態を表すstring値。フォーマットは次のとおりです。<i>state number-state description</i>。アクティビティの状態コードは、プロセスで使用する状態コードとは異なります。この属性は、以下の値のうちのいずれかをとることができます。</p> <p>1 - STATE_INACTIVE  2 - STATE_READY  3 - STATE_RUNNING  4 - STATE_SKIPPED  5 - STATE_FINISHED  6 - STATE_FAILED  7 - STATE_TERMINATED  8 - STATE_CLAIMED  9 - STATE_TERMINATING  10 - STATE_FAILING  11 - STATE_WAITING  12 - STATE_EXPIRED  13 - STATE_STOPPED</p>
bpellId	アクティビティの wpc:id 属性を表すstring値。
PayloadType	<p>有効搭載量タイプ。stringの値は、none、digest、または full のいずれかです。この値は、WebSphere Integration Developer の設定によって決まります。使用可能な明示的なモニター定義がない場合、例えば「<b>デフォルトのイベントを有効にする (Enable default events)</b>」が選択されている場合は、デフォルトの有効搭載量タイプは full になります。</p>

## リンク

リンクには、以下のようなオブジェクト固有のイベント内容が含まれています。

属性	説明
processTemplateName	インスタンスが派生したプロセス・テンプレートの名前
processTemplateValidFrom	テンプレートの有効日
flowBpellId	リンクを含むフロー・アクティビティの wpc:id 属性を表すstring値
elementName	評価されたリンクの名前
description	リンクの説明。プロセス・モデルに指定した場合のみ、この属性が含まれます。
PayloadType	<p>有効搭載量タイプ。stringの値は、none、digest、または full のいずれかです。この値は、WebSphere Integration Developer の設定によって決まります。使用可能な明示的なモニター定義がない場合、例えば「<b>デフォルトのイベントを有効にする (Enable default events)</b>」が選択されている場合は、デフォルトの有効搭載量タイプは full になります。</p>

## 変数

変数には、以下のようなオブジェクト固有のイベント内容が含まれています。

属性	説明
processTemplateName	インスタンスが派生したプロセス・テンプレートの名前。
processTemplateValidFrom	テンプレートの有効日。
variableName	変更された変数の名前。
variableData	<p>WBI Monitor の互換性のあるイベントが要求されると発行されます。変数の内容の XML 表現。データ・オブジェクトの各プロパティが、ネスト化拡張データ・エレメントの形式でレポートされます。エレメント・タイプは、「ブール」または「ストリング」タイプで、適切な値を持ちます。変数 <i>variableName</i> が初期化されていない場合、<i>variableData</i> エレメントも存在しません。</p> <p>この属性は、WebSphere Business Monitor 6.0.2 形式のイベントでのみ使用されます。WebSphere Business Monitor 6.1 形式のイベントの場合、変数の内容が <i>applicationData</i> セクションに書き込まれます。このセクションには、名前が変数名に設定されている 1 つのコンテンツ・エレメントが含まれています。</p>
variableData_BO	<p>WBI Monitor の互換性のあるイベント以外が要求されると発行されます。このエレメントのタイプは「noValue」で、値には変数の内容の XML 表現が含まれます。データ・オブジェクトの各プロパティが、ネスト化拡張データ・エレメントの形式でレポートされます。変数 <i>variableName</i> が初期化されていない場合、<i>VariableData_BO</i> エレメントも存在しません。</p> <p>この属性は、WebSphere Business Monitor 6.0.2 形式のイベントでのみ使用されます。WebSphere Business Monitor 6.1 形式のイベントの場合、変数の内容が <i>applicationData</i> セクションに書き込まれます。このセクションには、名前が変数名に設定されている 1 つのコンテンツ・エレメントが含まれています。</p>
bpelId	アクティビティの wpc:id 属性を表すストリング値。
PayloadType	有効搭載量タイプ。ストリングの値は、none、digest、または full のいずれかです。この値は、WebSphere Integration Developer の設定によって決まります。使用可能な明示的なモニター定義がない場合、例えば「 <b>デフォルトのイベントを有効にする (Enable default events)</b> 」が選択されている場合は、デフォルトの有効搭載量タイプは full になります。

## ビジネス・プロセス・イベントの拡張子名

拡張子名は、イベントの有効搭載量を示します。ビジネス・プロセス・イベントのすべての拡張子名と、対応する有効搭載量のリストがここに表示されます。

拡張子名には、Common Base Event の *extensionName* 属性の値として使用されるストリング値が含まれています。これは、イベントに関する追加データを提供する

XML エLEMENTの名前でもあります。イベント・ELEMENTの名前は大文字 (例: BPC.BFM.BASE) で、XML ELEMENTの名前は大/小文字混合 (例: *BPCEventCode*) です。指示されている場合以外は、すべてのデータ・ELEMENTのタイプはストリングです。

ビジネス・プロセス・イベントに使用できる拡張子名を以下に示します。

- 『BPC.BFM.BASE』
- 61 ページの 『BPC.BFM.PROCESS.BASE』
- 61 ページの 『BPC.BFM.PROCESS.STATUS』
- 62 ページの 『BPC.BFM.PROCESS.START』
- 62 ページの 『BPC.BFM.PROCESS.FAILURE』
- 62 ページの 『BPC.BFM.PROCESS.CORREL』
- 62 ページの 『BPC.BFM.PROCESS.WISTATUS』
- 63 ページの 『BPC.BFM.PROCESS.WITRANSFER』
- 63 ページの 『BPC.BFM.PROCESS.ESCALATED』
- 63 ページの 『BPC.BFM.PROCESS.EVENT』
- 64 ページの 『BPC.BFM.PROCESS.PARTNER』
- 64 ページの 『BPC.BFM.ACTIVITY.BASE』
- 66 ページの 『BPC.BFM.ACTIVITY.STATUS』
- 66 ページの 『BPC.BFM.ACTIVITY.FAILURE』
- 66 ページの 『BPC.BFM.ACTIVITY.MESSAGE』
- 67 ページの 『BPC.BFM.ACTIVITY.CLAIM』
- 67 ページの 『BPC.BFM.ACTIVITY.WISTATUS』
- 67 ページの 『BPC.BFM.ACTIVITY.WITRANSFER』
- 68 ページの 『BPC.BFM.ACTIVITY.FOREACH』
- 68 ページの 『BPC.BFM.ACTIVITY.ESCALATED』
- 68 ページの 『BPC.BFM.ACTIVITY.EVENT』
- 68 ページの 『BPC.BFM.LINK.STATUS』
- 69 ページの 『BPC.BFM.VARIABLE.STATUS』

## BPC.BFM.BASE

BPC.BFM.BASE は、WBIMonitoringEvent の XML ELEMENTを継承します。

表 21. BPC.BFM.BASE の XML ELEMENT

XML ELEMENT	説明
<i>BPCEventCode</i>	イベントの特性を示す Business Process Choreographer イベント・コード。
<i>processTemplateName</i>	プロセス・テンプレートの名前。この名前は、表示名とは異なることがあります。
<i>processTemplateValidFrom</i>	プロセス・テンプレートの有効開始日属性。

表 21. BPC.BFM.BASE の XML エlement (続き)

XML エlement	説明
<i>eventLocalCounter</i>	ローカル・カウンターは、同一トランザクション内で発生する 2 つのイベントの順序を検出するときに使用されます。Microflow インスタンスでは、このカウンターによりすべての発行済みイベントの順序が再構成されます。長期実行プロセスの場合、ローカル・カウンターは現行ナビゲーション・トランザクションにおける順序を示します。

## BPC.BFM.PROCESS.BASE

BPC.BFM.PROCESS.STATUS は、60 ページの『BPC.BFM.BASE』の XML エlement を継承します。

表 22. BPC.BFM.PROCESS.BASE の XML エlement

XML エlement	説明
<i>processInstanceExecutionState</i>	<p>プロセスの現在の実行状態。形式は次のとおりです。&lt;state code&gt;-&lt;state name&gt; この属性は、以下の値のうちのいずれかをとることができます。</p> <ul style="list-style-type: none"> <li>1 - STATE_READY</li> <li>2 - STATE_RUNNING</li> <li>3 - STATE_FINISHED</li> <li>4 - STATE_COMPENSATING</li> <li>5 - STATE_FAILED</li> <li>6 - STATE_TERMINATED</li> <li>7 - STATE_COMPENSATED</li> <li>8 - STATE_TERMINATING</li> <li>9 - STATE_FAILING</li> <li>10 - STATE_INDOUBT</li> <li>11 - STATE_SUSPENDED</li> <li>12 - STATE_COMPENSATION_FAILED</li> </ul>

## BPC.BFM.PROCESS.STATUS

BPC.BFM.PROCESS.STATUS は、『BPC.BFM.PROCESS.BASE』の XML エlement を継承します。

表 23. BPC.BFM.PROCESS.STATUS の XML エlement

XML エlement	説明
<i>processTemplateId</i>	プロセス・テンプレートの ID。
<i>processInstanceDescription</i>	プロセス・インスタンスの説明。
<i>principal</i>	このイベントに関連したユーザーの名前。

## BPC.BFM.PROCESS.START

BPC.BFM.PROCESS.START は、61 ページの『BPC.BFM.PROCESS.STATUS』の XML エlement を継承します。

表 24. BPC.BFM.PROCESS.START の XML Element

XML Element	説明
<i>username</i>	BPC.BFM.PROCESS.START の場合、これは、プロセスの開始または再開を要求したユーザーの名前です。

## BPC.BFM.PROCESS.FAILURE

BPC.BFM.PROCESS.FAILURE は、61 ページの『BPC.BFM.PROCESS.STATUS』の XML Element を継承します。

表 25. BPC.BFM.PROCESS.FAILURE の XML Element

XML Element	説明
<i>processFailedException</i>	プロセスの失敗に先立つ例外メッセージ。

## BPC.BFM.PROCESS.CORREL

BPC.BFM.PROCESS.CORREL は、61 ページの『BPC.BFM.PROCESS.STATUS』の XML Element を継承します。

表 26. BPC.BFM.PROCESS.CORREL の XML Element

XML Element	説明
<i>correlationSet</i>	相関セット・インスタンス。フォーマットは以下のとおりです。 <pre>&lt;?xml version="1.0"?&gt; &lt;correlationSet name="correlation set name"&gt;   &lt;property name="property name" value="property value"/&gt;* &lt;/correlationSet&gt;</pre>

## BPC.BFM.PROCESS.WISTATUS

BPC.BFM.PROCESS.WISTATUS は、61 ページの『BPC.BFM.PROCESS.STATUS』の XML Element を継承します。

表 27. BPC.BFM.PROCESS.WISTATUS の XML Element

XML Element	説明
<i>username</i>	BPC.BFM.PROCESS.WISTATUS の場合、これは、作業項目が作成または削除されたユーザーのリストです。



## BPC.BFM.PROCESS.WITRANSFER

BPC.BFM.PROCESS.WITRANSFER は、61 ページの『BPC.BFM.PROCESS.STATUS』の XML エlement を継承します。

表 28. BPC.BFM.PROCESS.WITRANSFER の XML Element

XML Element	説明
<i>current</i>	作業項目の現在の所有者のユーザー名。これは、他のユーザーに転送された作業項目があるユーザーです。
<i>target</i>	作業項目の新規所有者のユーザー名。

## BPC.BFM.PROCESS.ESCALATED

BPC.BFM.PROCESS.ESCALATED は、61 ページの『BPC.BFM.PROCESS.STATUS』の XML Element を継承します。

表 29. BPC.BFM.PROCESS.ESCALATED の XML Element

XML Element	説明
<i>escalationName</i>	エスカレーションの名前。
<i>operation</i>	BPC.BFM.PROCESS.ESCALATED の場合、インライン呼び出しタスクがエスカレートされるイベント・ハンドラーに関連付けられている操作です。
<i>portTypeName</i>	インライン呼び出しタスクがエスカレートされるイベント・ハンドラーに関連付けられている操作のポート・タイプ名です。
<i>portTypeNamespace</i>	インライン呼び出しタスクがエスカレートされるイベント・ハンドラーに関連付けられている操作のポート・タイプ・ネーム・スペースです。

## BPC.BFM.PROCESS.EVENT

BPC.BFM.PROCESS.EVENT は、61 ページの『BPC.BFM.PROCESS.STATUS』の XML Element を継承します。

表 30. BPC.BFM.PROCESS.EVENT の XML エレメント

XML エレメント	説明
<i>message</i> または <i>message_BO-</i>	<p>ストリングまたはビジネス・オブジェクト (BO) 表現としてのサービスの入力メッセージまたは出力メッセージ。形式は、WebSphere Integration Developer の「イベント・モニター (Event Monitor)」タブで「互換性のあるイベントのモニター (Monitor Compatible Events)」オプションが選択されたかどうかによって異なります。</p> <p>この属性は、WebSphere Business Monitor 6.0.2 形式のイベントでのみ使用されます。WebSphere Business Monitor 6.1 形式のイベントの場合、メッセージの内容が <i>applicationData</i> セクションに書き込まれます。このセクションには、名前がメッセージ名に設定されている 1 つのコンテンツ・エレメントが含まれています。</p>
<i>operation</i>	受信したイベントでの操作名。
<i>portTypeName</i>	イベント・ハンドラーに関連付けられている操作のポート・タイプ名です。
<i>portTypeNamespace</i>	イベント・ハンドラーに関連付けられている操作のポート・タイプ・ネーム・スペースです。

## BPC.BFM.PROCESS.PARTNER

BPC.BFM.PROCESS.PARTNER は、61 ページの『BPC.BFM.PROCESS.STATUS』の XML エレメントを継承します。

表 31. BPC.BFM.PROCESS.PARTNER の XML エレメント

XML エレメント	説明
<i>partnerLinkName</i>	パートナー・リンクの名前。

## BPC.BFM.ACTIVITY.BASE

BPC.BFM.ACTIVITY.BASE は、60 ページの『BPC.BFM.BASE』の XML エレメントを継承します。

表 32. BPC.BFM.ACTIVITY.BASE の XML エレメント

XML エレメント	説明
<p><i>activityKind</i></p>	<p>アクティビティーの種類。例えば、sequence や invoke など。形式は、&lt;kind code&gt;-&lt;kind name&gt; です。この属性は、以下の値のうちのいずれかをとることができます。</p> <ul style="list-style-type: none"> <li>3 - KIND_EMPTY</li> <li>21 - KIND_INVOKE</li> <li>23 - KIND_RECEIVE</li> <li>24 - KIND_REPLY</li> <li>25 - KIND_THROW</li> <li>26 - KIND_TERMINATE</li> <li>27 - KIND_WAIT</li> <li>29 - KIND_COMPENSATE</li> <li>30 - KIND_SEQUENCE</li> <li>32 - KIND_SWITCH</li> <li>34 - KIND_WHILE</li> <li>36 - KIND_PICK</li> <li>38 - KIND_FLOW</li> <li>40 - KIND_SCOPE</li> <li>42 - KIND_SCRIPT</li> <li>43 - KIND_STAFF</li> <li>44 - KIND_ASSIGN</li> <li>45 - KIND_CUSTOM</li> <li>46 - KIND_RETHROW</li> <li>47 - KIND_FOR_EACH_SERIAL</li> <li>48 - KIND_FOR_EACH_PARALLEL</li> <li>1000 - SQLSnippet</li> <li>1001 - RetrieveSet</li> <li>1002 - InvokeInformationService</li> <li>1003 - AtomicSQLSnippetSequence</li> </ul>
<p><i>state</i></p>	<p>アクティビティー・インスタンスの現在の状態。形式は次のとおりです。&lt;state code&gt;-&lt;state name&gt;。この属性は、以下の値のうちのいずれかをとることができます。</p> <ul style="list-style-type: none"> <li>1 - STATE_INACTIVE</li> <li>2 - STATE_READY</li> <li>3 - STATE_RUNNING</li> <li>4 - STATE_SKIPPED</li> <li>5 - STATE_FINISHED</li> <li>6 - STATE_FAILED</li> <li>7 - STATE_TERMINATED</li> <li>8 - STATE_CLAIMED</li> <li>9 - STATE_TERMINATING</li> <li>10 - STATE_FAILING</li> <li>11 - STATE_WAITING</li> <li>12 - STATE_EXPIRED</li> <li>13 - STATE_STOPPED</li> </ul>

表 32. BPC.BFM.ACTIVITY.BASE の XML エlement (続き)

XML エlement	説明
<i>bpellId</i>	BPEL ファイルのアクティビティの <i>wpc:id</i> 属性。これは、プロセス・モデル内のアクティビティに固有です。

## BPC.BFM.ACTIVITY.STATUS

BPC.BFM.ACTIVITY.STATUS は、64 ページの『BPC.BFM.ACTIVITY.BASE』の XML エlementを継承します。

表 33. BPC.BFM.ACTIVITY.STATUS の XML エlement

XML エlement	説明
<i>activityTemplateName</i>	アクティビティ・テンプレートの名前。この名前は、表示名とは異なることがあります。
<i>activityTemplateId</i>	アクティビティ・テンプレートの内部 ID。
<i>activityInstanceDescription</i>	アクティビティ・インスタンスの説明。
<i>principal</i>	アクティビティを要求したユーザーの名前。

## BPC.BFM.ACTIVITY.FAILURE

BPC.BFM.ACTIVITY.FAILURE は、『BPC.BFM.ACTIVITY.STATUS』の XML エlementを継承します。

表 34. BPC.BFM.ACTIVITY.FAILURE の XML エlement

XML エlement	説明
<i>activityFailedException</i>	アクティビティが失敗する原因となった例外。

## BPC.BFM.ACTIVITY.MESSAGE

BPC.BFM.ACTIVITY.MESSAGE は、『BPC.BFM.ACTIVITY.STATUS』の XML エlementを継承します。

表 35. BPC.BFM.ACTIVITY.MESSAGE の XML エlement

XML エlement	説明
<i>message</i> または <i>message_BO</i>	<p>ストリングまたはビジネス・オブジェクト (BO) 表現としてのサービスの入力または出力メッセージ。形式は、WebSphere Integration Developer の「イベント・モニター (Event Monitor)」タブで「互換性のあるイベントのモニター (Monitor Compatible Events)」オプションが選択されたかどうかによって異なります。</p> <p>この属性は、WebSphere Business Monitor 6.0.2 形式のイベントでのみ使用されます。WebSphere Business Monitor 6.1 形式のイベントの場合、メッセージの内容が <i>applicationData</i> セクションに書き込まれます。このセクションには、名前がメッセージ名に設定されている 1 つのコンテンツ・Elementが含まれています。</p>

### BPC.BFM.ACTIVITY.CLAIM

BPC.BFM.ACTIVITY.CLAIM は、66 ページの『BPC.BFM.ACTIVITY.STATUS』の XML Elementを継承します。

表 36. BPC.BFM.ACTIVITY.CLAIM の XML Element

XML Element	説明
<i>username</i>	BPC.BFM.ACTIVITY.CLAIM の場合、タスクを要求したユーザー。

### BPC.BFM.ACTIVITY.WISTATUS

BPC.BFM.ACTIVITY.WISTATUS は、66 ページの『BPC.BFM.ACTIVITY.STATUS』の XML Elementを継承します。

表 37. BPC.BFM.ACTIVITY.WISTATUS の XML Element

XML Element	説明
<i>username</i>	BPC.BFM.ACTIVITY.WISTATUS の場合、作業項目に関連付けられているユーザーのリスト。

### BPC.BFM.ACTIVITY.WITRANSFER

BPC.BFM.ACTIVITY.WITRANSFER は、66 ページの『BPC.BFM.ACTIVITY.STATUS』の XML Elementを継承します。

表 38. BPC.BFM.ACTIVITY.WITRANSFER の XML エlement

XML エlement	説明
<i>current</i>	作業項目の現在の所有者のユーザー名。これは、他のユーザーに転送された作業項目があるユーザーです。
<i>target</i>	作業項目の新規所有者のユーザー名。

### BPC.BFM.ACTIVITY.FOREACH

BPC.BFM.ACTIVITY.FOREACH は、66 ページの『BPC.BFM.ACTIVITY.STATUS』の XML エlementを継承します。

表 39. BPC.BFM.ACTIVITY.FOREACH の XML エlement

XML エlement	説明
<i>parallelBranchesStarted</i>	開始したブランチの数。

### BPC.BFM.ACTIVITY.ESCALATED

BPC.BFM.ACTIVITY.ESCALATED は、66 ページの『BPC.BFM.ACTIVITY.STATUS』の XML エlementを継承します。

表 40. BPC.BFM.ACTIVITY.ESCALATED の XML エlement

XML エlement	説明
<i>escalationName</i>	エスカレーションの名前。
<i>operation</i>	BPC.BFM.ACTIVITY.ESCALATED の場合、インライン呼び出しタスクがエスカレートされるイベント・ハンドラーに関連付けられている操作です。

### BPC.BFM.ACTIVITY.EVENT

BPC.BFM.ACTIVITY.EVENT は、66 ページの『BPC.BFM.ACTIVITY.MESSAGE』の XML エlementを継承します。

表 41. BPC.BFM.ACTIVITY.EVENT の XML エlement

XML エlement	説明
<i>operation</i>	受信したイベントでの操作名。

### BPC.BFM.LINK.STATUS

BPC.BFM.LINK.STATUS は、60 ページの『BPC.BFM.BASE』の XML エlementを継承します。

表 42. BPC.BFM.LINK.STATUS の XML エlement

XML エlement	説明
<i>elementName</i>	リンクの名前。
<i>description</i>	リンクの説明。

表 42. BPC.BFM.LINK.STATUS の XML エlement (続き)

XML エlement	説明
<i>flowBpelId</i>	リンクを定義する flow アクティビティの ID。

## BPC.BFM.VARIABLE.STATUS

BPC.BFM.VARIABLE.STATUS は、60 ページの『BPC.BFM.BASE』の XML エlement を継承します。

表 43. BPC.BFM.VARIABLE.STATUS の XML エlement

XML エlement	説明
<i>variableName</i>	変数の名前。
<i>variableData</i> または <i>variableData_BO</i>	変数 <i>variableName</i> が初期化されていない場合、 <i>variableData</i> エlement も <i>VariableData_BO</i> エlement も存在しません。変数のデータは、ストリングまたはビジネス・オブジェクト (BO) のいずれかとして表現されます。形式は、WebSphere Integration Developer の「イベント・モニター (Event Monitor)」タブで「互換性のあるイベントのモニター (Monitor Compatible Events)」オプションが選択されたかどうかによって異なります。  この属性は、WebSphere Business Monitor 6.0.2 形式のイベントでのみ使用されます。WebSphere Business Monitor 6.1 形式のイベントの場合、変数の内容が <i>applicationData</i> セクションに書き込まれます。このセクションには、名前が変数名に設定されている 1 つのコンテンツ・Element が含まれています。
<i>bpelId</i>	変数の Business Process Choreographer ID。
<i>principal</i>	変数を更新したユーザーの名前。

## ビジネス・プロセス・イベント

WebSphere Integration Developer 内のビジネス・プロセス・Element のモニターが要求された場合、ビジネス・プロセス・イベントが送信されます。ここでは、ビジネス・プロセスによって発行可能なすべてのイベントのリストを示します。

プロセスまたはアクティビティの状態が変更された時点でイベントが発行されます。ビジネス・プロセスは、以下のイベント・タイプを発行することができます。

- 70 ページの『プロセス・イベント』
- 73 ページの『アクティビティ・イベント』
- 77 ページの『アクティビティ・スコープ・イベント』
- 78 ページの『リンク・イベント』



- 78 ページの『変数イベント』

## XML スキーマ定義 (XSD) ファイル

イベントの構造は、XML スキーマ定義 (XSD) ファイル BFMEvents.xsd に記述されています。このファイルは、`install_root¥ProcessChoreographer¥client` ディレクトリにあります。

## テーブル列のキー

以下のテーブルの列の内容は、次のとおりです。

**コード** イベントの番号。WebSphere Business Monitor 6.0.2 形式のイベントの場合、この値は *BPCEventCode* という名前の拡張データ・エレメントとして Common Base Event に書き込まれます。WebSphere Business Monitor 6.1 形式のイベントの場合、この値は Common Base Event の `xs:any` スロットに書き込まれます。

## 拡張子名

*extensionName* には、Common Base Event に含まれているイベント固有情報を定義する文字列値が含まれています。これは、イベントに関する追加データを提供する XML エレメントの名前でもあります。拡張子名についての詳細は、59 ページの『ビジネス・プロセス・イベントの拡張子名』を参照してください。

**状態** ビジネス・プロセス・イベントの状態名を指します。状態の詳細については、78 ページの『ビジネス・プロセス・イベントの状態』を参照してください。

## イベント性質

WebSphere Integration Developer に表示されるときの、EventNature パラメーター内のビジネス・プロセス・エレメントのイベント状態を指すポインター。

## プロセス・イベント

以下の表で、すべてのプロセス・イベントを説明します。

コード	拡張子名	状態	イベント性質	説明
21000	BPC.BFM.PROCESS.START	開始	ENTRY	プロセスが開始された
21001	BPC.BFM.PROCESS.STATUS	レポート	SUSPENDED	プロセスが中断された。プロセス・インスタンスを中断するには、Business Process Choreographer Explorer を使用します。

コード	拡張子名	状態	イベント性質	説明
21002	BPC.BFM.PROCESS.STATUS	レポート	RESUMED	プロセスが再開された。中断されたプロセスのみを再開できます。プロセス・インスタンスを再開するには、Business Process Choreographer Explorer を使用します。
21004	BPC.BFM.PROCESS.STATUS	停止	EXIT	プロセスが完了した
21005	BPC.BFM.PROCESS.STATUS	停止	TERMINATED	プロセスが強制終了した。プロセス・インスタンスを強制終了するには、Business Process Choreographer Explorer を使用します。
21019	BPC.BFM.PROCESS.START	レポート	RESTARTED	プロセスが再始動した
21020	BPC.BFM.PROCESS.STATUS	破棄	DELETED	プロセスが削除された
42001	BPC.BFM.PROCESS. FAILURE	失敗	FAILED	プロセスが失敗した
42003	BPC.BFM.PROCESS.STATUS	レポート	COMPENSATING	プロセスが補正中。プロセス・インスタンスを補正するには、Business Process Choreographer Explorer を使用します。
42004	BPC.BFM.PROCESS.STATUS	停止	COMPENSATED	プロセスが補正された
42009	BPC.BFM.PROCESS.STATUS	レポート	TERMINATING	プロセスが強制終了中
42010	BPC.BFM.PROCESS.STATUS	レポート	FAILING	プロセスが失敗する

コード	拡張子名	状態	イベント性質	説明
42027	BPC.BFM.PROCESS.CORREL	レポート	CORRELATION	相関セットが初期化された。プロセス・インスタンスの新規相関セットが初期化された時点で発行されます。これは、例えば開始相関セットがある受信アクティビティーがメッセージを受信した場合などに該当します。
42041	BPC.BFM.PROCESS.WISTATUS	レポート	WI_DELETED	プロセス作業項目が削除された
42042	BPC.BFM.PROCESS.WISTATUS	レポート	WI_CREATED	プロセス作業項目が作成された
42046	BPC.BFM.PROCESS.STATUS	失敗	COMPFAILED	プロセス補正が失敗した
42047	BPC.BFM.PROCESS.EVENT	レポート	EV_RECEIVED	プロセス・イベントを受信した。イベントを定義するにはプロセス・インターフェースを使用します。プロセスに関連付けられているイベント・ハンドラーがアクティブになると、このイベントが生成されます。
42049	BPC.BFM.PROCESS.ESCALATED	レポート	EV_ESCALATED	プロセス・イベントがエスカレートされた。このイベントは、インライン呼び出しタスクがエスカレートされると生成されます。このイベントはプロセス・レベルで定義されており、onEvent イベント・ハンドラーに関連付けられています。
42056	BPC.BFM.PROCESS.WITRANSFER	レポート	WI_TRANSFERRED	プロセス作業項目が転送された

コード	拡張子名	状態	イベント性質	説明
42058	BPC.BFM.PROCESS.PARTNER	レポート	PA_CHANGE	プロセス・パートナーが変更された。このイベントは、新規エンドポイント参照がパートナー・リンクに割り当てられた時点で生成されます。

プロセス・イベントの場合、以下のイベント関連範囲 ID の内容は次のとおりです。

- ECSCurrentID は、プロセス・インスタンスの ID です。
- ECSParentID は、現行プロセスのプロセス・インスタンス開始イベント前の ECSCurrentID の値です。

### アクティビティ・イベント

以下の表で、すべてのアクティビティ・イベントを説明します。

コード	拡張子名	状態	イベント性質	説明
21006	BPC.BFM.ACTIVITY.MESSAGE	開始	CREATED	アクティビティが実行可能である。このイベントは、human task アクティビティの開始時に生成されます。
21007	invoke アクティビティの場合: BPC.BFM.ACTIVITY.MESSAGE。その他すべてのアクティビティ・タイプの場合、 BPC.BFM.ACTIVITY.STATUS	開始	ENTRY	アクティビティが開始した。 invoke アクティビティでは、ビジネス・オブジェクト・ペイロードが使用可能です。
21011	invoke、human task、receive、および reply の各アクティビティの場合: BPC.BFM.ACTIVITY.MESSAGE。pick アクティビティの場合、 BPC.BFM.ACTIVITY.EVENT。その他すべてのアクティビティ・タイプの場合、 BPC.BFM.ACTIVITY.STATUS	停止	EXIT	アクティビティが完了した。 invoke、human task、receive、および reply アクティビティでは、ビジネス・オブジェクト・ペイロードが使用可能です。

コード	拡張子名	状態	イベント性質	説明
21021	BPC.BFM.ACTIVITY.STATUS	レポート	DEASSIGNED	要求がキャンセルされた。このイベントは、human task アクティビティに対する要求が取り消された時点で生成されます。
21022	BPC.BFM.ACTIVITY.CLAIM	レポート	ASSIGNED	アクティビティが要求された。このイベントは、human task アクティビティが要求された時点で生成されます。
21027	BPC.BFM.ACTIVITY.STATUS	停止	TERMINATED	アクティビティが強制終了した。長期実行アクティビティが、割り当てられている有効範囲またはプロセスでの障害処理の影響で強制終了することがあります。
21080	BPC.BFM.ACTIVITY.FAILURE	失敗	FAILED	アクティビティが失敗した
21081	BPC.BFM.ACTIVITY.STATUS	レポート	EXPIRED	アクティビティが期限切れになった。invoke アクティビティとインライン・ヒューマン・タスク・アクティビティに有効期限を定義できます。

コード	拡張子名	状態	イベント性質	説明
42005	BPC.BFM.ACTIVITY.STATUS	レポート	SKIPPED	アクティビティーがスキップされた。イベントは、結合動作定義されているアクティビティーにのみ適用できます。結合動作が false と評価された場合、アクティビティーはスキップされ、スキップされたイベントが発行されません。
42012	BPC.BFM.ACTIVITY.MESSAGE	レポート	OUTPUTSET	アクティビティー出力メッセージが設定された。ビジネス・オブジェクト・ペイロードが使用可能です。
42013	BPC.BFM.ACTIVITY.MESSAGE	レポート	FAULTSET	アクティビティー障害メッセージが設定された。ビジネス・オブジェクト・ペイロードが使用可能です。
42015	BPC.BFM.ACTIVITY.STATUS	停止	STOPPED	アクティビティーが停止した
42031	BPC.BFM.ACTIVITY.STATUS	レポート	FRETRIED	アクティビティーが強制的に再試行された。アクティビティーを強制的に再試行するには、Business Process Choreographer Explorer を使用します。
42032	BPC.BFM.ACTIVITY.STATUS	停止	FCOMPLETED	アクティビティーが強制終了された。アクティビティーを強制終了するには、Business Process Choreographer Explorer を使用します。

コード	拡張子名	状態	イベント性質	説明
42036	BPC.BFM.ACTIVITY.MESSAGE	レポート	EXIT	アクティビティーがメッセージを受信
42037	BPC.BFM.ACTIVITY.STATUS	レポート	CONDTRUE	ループ条件が true
42038	BPC.BFM.ACTIVITY.STATUS	レポート	CONDFALSE	ループ条件が false
42039	BPC.BFM.ACTIVITY. WISTATUS	レポート	WI_DELETED	作業項目が削除された。このイベントは、pick、インライン・ヒューマン・タスク、および受信イベントにのみ適用できます。
42040	BPC.BFM.ACTIVITY. WISTATUS	レポート	WI_CREATED	作業項目が作成された。このイベントは、pick、インライン・ヒューマン・タスク、および受信イベントにのみ適用できます。
42050	BPC.BFM.ACTIVITY.ESCALATED	レポート	ESCALATED	アクティビティーがエスカレートされた。このイベントは、pick、インライン・ヒューマン・タスク、および受信イベントにのみ適用できます。
42054	BPC.BFM.ACTIVITY. WISTATUS	レポート	WI_REFRESHED	アクティビティー作業項目が更新された。このイベントは、pick、インライン・ヒューマン・タスク、および受信イベントにのみ適用できます。
42055	BPC.BFM.ACTIVITY. WITRANSFER	レポート	WI_TRANSFERRED	作業項目が転送された。このイベントは、pick、インライン・ヒューマン・タスク、および受信イベントにのみ適用できます。



コード	拡張子名	状態	イベント性質	説明
42057	BPC.BFM.ACTIVITY.FOREACH	レポート	BRANCHES_STARTED	各アクティビティ・ブランチが開始された

アクティビティ・イベントの場合、以下のイベント関連範囲 ID の内容は次のとおりです。

- *ECSCurrentID* は、アクティビティの ID です。
- *ECSParentID* は、収容プロセスの ID です。

### アクティビティ・スコープ・イベント

以下の表で、すべてのアクティビティ・スコープ・イベントを説明します。

コード	拡張子名	状態	イベント性質	説明
42020	BPC.BFM.ACTIVITY.STATUS	開始	ENTRY	スコープが開始した
42021	BPC.BFM.ACTIVITY.STATUS	レポート	SKIPPED	スコープがスキップされた
42022	BPC.BFM.ACTIVITY.FAILURE	失敗	FAILED	スコープが失敗した
42023	BPC.BFM.ACTIVITY.STATUS	レポート	FAILING	スコープが強制終了中
42024	BPC.BFM.ACTIVITY.STATUS	停止	TERMINATED	スコープが強制終了した
42026	BPC.BFM.ACTIVITY.STATUS	停止	EXIT	スコープが完了した
42043	BPC.BFM.ACTIVITY.STATUS	レポート	COMPENSATING	スコープが補正中
42044	BPC.BFM.ACTIVITY.STATUS	停止	COMPENSATED	スコープが補正された
42045	BPC.BFM.ACTIVITY.STATUS	失敗	COMPFAILED	スコープ補正が失敗した
42048	BPC.BFM.ACTIVITY.EVENT	レポート	EV_RECEIVED	アクティビティ・イベントを受信
42051	BPC.BFM.ACTIVITY.ESCALATED	レポート	EV_ESCALATED	スコープ・イベントがエスカレートされた

アクティビティ・スコープ・イベントは、上記の BPC.BFM.ACTIVITY.STATUS で構文が説明されているアクティビティ・イベントの一種です。

アクティビティ有効範囲イベントの場合、以下のイベント関連範囲 ID の内容は次のとおりです。

- *ECSCurrentID* は、スコープの ID です。
- *ECSParentID* は、収容プロセスの ID です。

## リンク・イベント

以下の表で、すべてのリンク・イベントを説明します。

コード	拡張子名	状態	イベント性質	説明
21034	BPC.BFM.LINK.STATUS	レポート	CONDTRUE	リンクが true と評価された
42000	BPC.BFM.LINK.STATUS	レポート	CONDFALSE	リンクが false と評価された

リンク・イベントの場合、以下のイベント関連範囲 ID の内容は次のとおりです。

- ECSCurrentID は、リンクのソース・アクティビティの ID です。
- ECSParentID は、収容プロセスの ID です。

## 変数イベント

以下の表で、変数イベントを説明します。

コード	拡張子名	状態	イベント性質	説明
21090	BPC.BFM.VARIABLE.STATUS	レポート	CHANGED	変数が更新された。ビジネス・オブジェクト・ペイロードが使用可能です。

変数イベントの場合、以下のイベント関連範囲 ID の内容は次のとおりです。

- ECSCurrentID は、収容プロセスの ID です。
- ECSParentID は、現行プロセスのプロセス・インスタンス開始イベントの前の ECSCurrentID です。

## ビジネス・プロセス・イベントの状態

ビジネス・プロセス・イベントは、さまざまな状態で発行可能です。これらの状態のデータについては、状態エレメントで説明されています。

ビジネス・プロセス・イベントには、次の状態エレメントのいずれかが含まれます。

状態名	Common Base Event の内容	
開始	categoryName は StartSituation に設定されます。	
	situationType	
	Type	StartSituation
	reasoningScope	EXTERNAL
	successDisposition	SUCCESSFUL
	situationQualifier	START_COMPLETED

状態名	Common Base Event の内容	
停止	categoryName は StopSituation に設定されます。	
	situationType	
	Type	StopSituation
	reasoningScope	EXTERNAL
	successDisposition	SUCCESSFUL
	situationQualifier	STOP_COMPLETED
破棄	categoryName は DestroySituation に設定されます。	
	situationType	
	Type	DestroySituation
	reasoningScope	EXTERNAL
	successDisposition	SUCCESSFUL
失敗	categoryName は StopSituation に設定されます。	
	situationType	
	Type	StopSituation
	reasoningScope	EXTERNAL
	successDisposition	UNSUCCESSFUL
	situationQualifier	STOP_COMPLETED
レポート	categoryName は ReportSituation に設定されます。	
	situationType	
	Type	ReportSituation
	reasoningScope	EXTERNAL
	reportCategory	STATUS

## ヒューマン・タスク・イベントのモニター

ヒューマン・タスクのために発行されるイベントは、状態非依存データとヒューマン・タスク・イベントに固有のデータで構成されています。ここでは、ヒューマン・タスク・イベントに固有の属性とエレメントについて説明します。

ヒューマン・タスク・イベントには、次に示すイベント内容のカテゴリーがあります。

### ヒューマン・タスク固有のイベント・データ

タスクおよびエスカレーションのためにイベントが作成されます。これらの各イベント・タイプのオブジェクト固有の内容について説明します。

Business Process Choreographer バージョン 6.1 では、2 つのイベント形式が使用されることがあります。

#### WebSphere Business Monitor 6.0.2 形式

WebSphere Business Monitor 6.0.2 形式のイベントは、WebSphere Integration Developer 6.0.2 でモデリングされているタスクがある場合、または WebSphere Integration Developer 6.1 で WebSphere Business Monitor 6.0.2 形式モードが使用可能になっている場合に発生します。特に明記されていない

い限り、これらのイベントのオブジェクト固有の内容は、タイプがストリングの *extendedDataElement* XML エlementとして書き込まれます。

### WebSphere Business Monitor 6.1 形式

WebSphere Business Monitor 6.1 形式のイベントは、WebSphere Integration Developer 6.1 でモデリングされたタスクがあり、WebSphere Business Monitor 6.1 形式モードが使用可能になっている場合に発生します。これらのイベントのオブジェクト固有の内容は、Common Base Event の *eventPointData* フォルダの *xs:any* スロットに XML Elementとして書き込まれます。XML の構造は、XML スキーマ定義 (XSD) ファイル *HTMEvents.xsd* で定義されています。このファイルは、*install\_root\ProcessChoreographer\client* ディレクトリにあります。

### タスク

タスク・イベントには、以下のようなオブジェクト固有のイベント内容が含まれています。

属性	説明
<i>taskTemplateName</i>	インスタンスが派生したタスク・テンプレートの名前。
<i>taskTemplateValidFrom</i>	テンプレートの有効日。
<i>taskTemplateId</i>	インスタンスが派生したタスク・テンプレートの ID。
<i>taskInstanceDescription</i>	デフォルト・ロケールでのタスク・インスタンスの記述。
<i>PayloadType</i>	有効搭載量タイプ。ストリングの値は、 <i>none</i> 、 <i>digest</i> 、または <i>full</i> のいずれかです。この値は、WebSphere Integration Developer の設定によって決まります。使用可能な明示的なモニター定義がない場合、デフォルトの有効搭載量タイプは <i>full</i> になります。

### エスカレーション

エスカレーションには、以下のようなオブジェクト固有のイベント内容が含まれています。

属性	説明
<i>taskTemplateName</i>	インスタンスが派生したタスク・テンプレートの名前。
<i>taskTemplateValidFrom</i>	テンプレートの有効日。
<i>taskTemplateId</i>	インスタンスが派生したタスク・テンプレートの ID。
<i>escalationName</i>	エスカレーションの名前。
<i>escalationInstanceDescription</i>	オプション: エスカレーション・インスタンスの説明。
<i>PayloadType</i>	有効搭載量タイプ。ストリングの値は、 <i>none</i> 、 <i>digest</i> 、または <i>full</i> のいずれかです。この値は、WebSphere Integration Developer の設定によって決まります。使用可能な明示的なモニター定義がない場合、デフォルトの有効搭載量タイプは <i>full</i> になります。

## ヒューマン・タスク・イベントの拡張子名

拡張子名は、ヒューマン・タスク・イベントの有効搭載量を示します。ヒューマン・タスク・イベントのすべての拡張子名と、対応する有効搭載量のリストがここに表示されます。

拡張子名には、Common Base Event の *extensionName* 属性の値として使用されるストリング値が含まれています。これは、イベントに関する追加データを提供する XML エLEMENT の名前でもあります。イベント・ELEMENT の名前は大文字 (例えば、BPC.HTM.BASE) で、XML エLEMENT の名前は大/小文字混合 (例えば、HTMEventCode) です。指示されている場合以外は、すべてのデータ・ELEMENT のタイプはストリングです。

ヒューマン・タスク・イベントに使用できる拡張子名を以下に示します。

- 『BPC.HTM.BASE』
- 『BPC.HTM.TASK.BASE』
- 82 ページの 『BPC.HTM.TASK.STATUS』
- 82 ページの 『BPC.HTM.TASK.FOLLOW』
- 82 ページの 『BPC.HTM.TASK.MESSAGE』
- 83 ページの 『BPC.HTM.TASK.INTERACT』
- 83 ページの 『BPC.HTM.TASK.FAILURE』
- 83 ページの 『BPC.HTM.TASK.WISTATUS』
- 83 ページの 『BPC.HTM.TASK.WITRANSFER』
- 84 ページの 『BPC.HTM.ESCALATION.STATUS』
- 84 ページの 『BPC.HTM.ESCALATION.WISTATUS』
- 84 ページの 『BPC.HTM.ESCALATION.WITRANSFER』

### BPC.HTM.BASE

BPC.HTM.BASE は、WBIMonitoringEvent の XML エLEMENT を継承します。

表 44. BPC.HTM.BASE の XML エLEMENT

XML エLEMENT	説明
<i>HTMEventCode</i>	イベント・タイプの数を識別する Business Process Choreographer イベント・コード。以下の表に、可能なイベント・コードをリストします。

### BPC.HTM.TASK.BASE

BPC.HTM.TASK.BASE は、『BPC.HTM.BASE』の XML エLEMENT を継承します。

表 45. BPC.HTM.TASK.BASE の XML エLEMENT

XML エLEMENT	説明
<i>taskTemplateId</i>	テンプレートの ID。

表 45. BPC.HTM.TASK.BASE の XML エlement (続き)

XML エlement	説明
<i>taskTemplateName</i>	タスク・テンプレートの名前。この名前は、表示名とは異なることがあります。
<i>taskTemplateValidFrom</i>	タスク・テンプレートが有効になる日時。

## BPC.HTM.TASK.STATUS

BPC.HTM.TASK.STATUS は、81 ページの『BPC.HTM.TASK.BASE』の XML エlement を継承します。

表 46. BPC.HTM.TASK.STATUS の XML エlement

XML エlement	説明
<i>taskInstanceDescription</i>	タスクの説明。

## BPC.HTM.TASK.FOLLOW

BPC.HTM.TASK.FOLLOW は、81 ページの『BPC.HTM.TASK.BASE』の XML エlement を継承します。

表 47. BPC.HTM.TASK.FOLLOW の XML エlement

XML エlement	説明
<i>followTaskId</i>	後続タスクとして開始されたタスクの ID。

## BPC.HTM.TASK.MESSAGE

BPC.HTM.TASK.MESSAGE は、『BPC.HTM.TASK.STATUS』の XML エlement を継承します。

表 48. BPC.HTM.TASK.FOLLOW の XML エlement

XML エlement	説明
<i>message</i> または <i>message_BO</i>	<p>入力または出力メッセージを含むストリングまたはビジネス・オブジェクトの表現。形式は、WebSphere Integration Developer の「イベント・モニター (Event Monitor)」タブで「互換性のあるイベントのモニター (Monitor Compatible Events)」オプションが選択されたかどうかによって異なります。</p> <p>この属性は、WebSphere Business Monitor 6.0.2 形式のイベントでのみ使用されます。WebSphere Business Monitor 6.1 形式のイベントの場合、メッセージの内容が <i>applicationData</i> セクションに書き込まれます。このセクションには、名前がメッセージ名に設定されている 1 つのコンテンツ・Element が含まれています。</p>

## BPC.HTM.TASK.INTERACT

BPC.HTM.TASK.INTERACT は、82 ページの『BPC.HTM.TASK.STATUS』の XML エlementを継承します。

表 49. BPC.HTM.TASK.INTERACT の XML エlement

XML エlement	説明
<i>username</i>	BPC.HTM.TASK.INTERACT の場合、これは、タスクに関連付けられているユーザーの名前です。

## BPC.HTM.TASK.FAILURE

BPC.HTM.TASK.FAILURE は、82 ページの『BPC.HTM.TASK.STATUS』の XML エlementを継承します。

表 50. BPC.HTM.TASK.FAILURE の XML エlement

XML エlement	説明
<i>taskFailedException</i>	セミコロン (;) で区切られた <i>faultNameSpace</i> および <i>faultName</i> を含むストリング。

## BPC.HTM.TASK.WISTATUS

BPC.HTM.TASK.WISTATUS は、82 ページの『BPC.HTM.TASK.STATUS』の XML エlementを継承します。

表 51. BPC.HTM.TASK.WISTATUS の XML エlement

XML エlement	説明
<i>username</i>	BPC.BPC.TASK.WISTATUS の場合、これは、作業項目が作成または削除されたユーザーのリストです。

## BPC.HTM.TASK.WITRANSFER

BPC.HTM.TASK.WITRANSFER は、82 ページの『BPC.HTM.TASK.STATUS』の XML エlementを継承します。

表 52. BPC.HTM.TASK.WITRANSFER の XML エlement

XML エlement	説明
<i>current</i>	BPC.HTM.TASK.WITRANSFER の場合、これは、現行ユーザーの名前です。これは、他のユーザーに転送された作業項目があるユーザーです。
<i>target</i>	BPC.HTM.TASK.WITRANSFER の場合、これは、作業項目の受け取り側のユーザー名です。



## BPC.HTM.ESCALATION.STATUS

BPC.HTM.ESCALATION.STATUS は、81 ページの『BPC.HTM.TASK.BASE』の XML エlement を継承します。

表 53. BPC.HTM.ESCALATION.STATUS の XML Element

XML Element	説明
<i>escalationName</i>	エスカレーションの名前。
<i>escalationInstanceDescription</i>	エスカレーションの説明。

## BPC.HTM.ESCALATION.WISTATUS

BPC.HTM.ESCALATION.WISTATUS は、『BPC.HTM.ESCALATION.STATUS』の XML Element を継承します。

表 54. BPC.HTM.ESCALATION.WISTATUS の XML Element

XML Element	説明
<i>username</i>	BPC.HTM.ESCALATION.WISTATUS の場合、これは、作業項目がエスカレーションされたユーザーのリストです。

## BPC.HTM.ESCALATION.WITRANSFER

BPC.HTM.ESCALATION.WITRANSFER は、『BPC.HTM.ESCALATION.STATUS』の XML Element を継承します。

表 55. BPC.HTM.ESCALATION.WITRANSFER の XML Element

XML Element	説明
<i>current</i>	BPC.HTM.ESCALATION.WITRANSFER の場合、これは、現行ユーザーの名前です。これは、他のユーザーに転送された作業項目があるユーザーです。
<i>target</i>	BPC.HTM.ESCALATION.WITRANSFER の場合、これは、作業項目の受け取り側のユーザー名です。

## ヒューマン・タスク・イベント

WebSphere Integration Developer 内のタスクの Element についてモニターが要求された場合、ヒューマン・タスク・イベントが送信されます。ここでは、ヒューマン・タスクによって発行可能なすべてのイベントのリストを示します。

タスクの状態が変更された時点でイベントが発行されます。ヒューマン・タスクは、以下のイベント・タイプを発行することができます。

- 85 ページの『タスク・イベント』
- 86 ページの『エスカレーション・イベント』

注: タスク・モデルにおいてビジネス関連性フラグが true に設定されている場合、イベントは随時タスクの場合のみ発行されます。

インライン・タスクのイベントはアクティビティ・イベントとして発行されます。これらのイベントのリストについては、69 ページの『ビジネス・プロセス・イベント』を参照してください。

## XML スキーマ定義 (XSD) ファイル

イベントの構造は、XML スキーマ定義 (XSD) ファイル `HTMEvents.xsd` に記述されています。このファイルは、`install_root¥ProcessChoreographer¥client` ディレクトリにあります。

## テーブル列のキー

以下のテーブルの列の内容は、次のとおりです。

**コード** イベントの番号。WebSphere Business Monitor 6.0.2 形式のイベントの場合、この値は `HTMEventCode` という名前の拡張データ・エレメントとして Common Base Event に書き込まれます。WebSphere Business Monitor 6.1 形式のイベントの場合、この値は Common Base Event の `xs:any` スロットに書き込まれます。

### 拡張子名

Common Base Event の `extensionName` 属性の値として使用されるストリング値。

WebSphere Business Integration Modeler を使用して基本のタスク・モデルを作成する場合、有効搭載量にメッセージ・データを含むイベントの拡張名は、ハッシュ文字 (#) とそれに続く追加文字によって拡張できます。これらの追加文字は、各種メッセージ・オブジェクトを送達する Common Base Event を識別するのに使用します。メッセージ・データを出力するイベントには、データ・オブジェクトの内容をレポートするために、ネストされた追加の `extendedDataElements` も含まれます。詳しくは、WebSphere Business Integration Modeler の資料を参照してください。

**状態** ヒューマン・タスク・イベントの状態名を指します。状態の詳細については、87 ページの『ヒューマン・タスク・イベントの状態』を参照してください。

### イベント性質

WebSphere Integration Developer に表示されるとき、`EventNature` パラメーター内のビジネス・プロセス・エレメントのイベント状態を指すポインター。

## タスク・イベント

以下の表で、すべてのタスク・イベントを説明します。

コード	拡張子名	状態	イベント性質	説明
51001	BPC.HTM.TASK.INTERACT	レポート	CREATED	タスクが作成された
51002	BPC.HTM.TASK.STATUS	破棄	DELETED	タスクが削除された
51003	BPC.HTM.TASK.STATUS	開始	ENTRY	タスクが開始された
51004	BPC.HTM.TASK.STATUS	停止	EXIT	タスクが完了した

コード	拡張子名	状態	イベント性質	説明
51005	BPC.HTM.TASK.STATUS	レポート	DEASSIGNED	要求がキャンセルされた
51006	BPC.HTM.TASK. INTERACT	レポート	ASSIGNED	タスクが要求された
51007	BPC.HTM.TASK.STATUS	停止	TERMINATED	タスクが強制終了された
51008	BPC.HTM.TASK. FAILURE	失敗	FAILED	タスクが失敗した
51009	BPC.HTM.TASK.STATUS	レポート	EXPIRED	タスクの期限切れ
51010	BPC.HTM.TASK.STATUS	レポート	WAITFORSUBTASK	サブタスクを待機中
51011	BPC.HTM.TASK.STATUS	停止	SUBTASKCOMPLETED	サブタスクが完了した
51012	BPC.HTM.TASK.STATUS	レポート	RESTARTED	タスクが再始動した
51013	BPC.HTM.TASK.STATUS	レポート	SUSPENDED	タスクが一時停止した
51014	BPC.HTM.TASK.STATUS	レポート	RESUMED	タスクが再開した
51015	BPC.HTM.TASK. FOLLOW	レポート	COMPLETEDFOLLOW	タスクが完了し、追加タスクが開始した
51101	BPC.HTM.TASK.STATUS	レポート	UPDATED	タスク・プロパティが更新された
51103	BPC.HTM.TASK. MESSAGE	レポート	OUTPUTSET	出力メッセージが更新された。ビジネス・オブジェクト・ペイロードが使用可能です。
51104	BPC.HTM.TASK. MESSAGE	レポート	FAULTSET	障害メッセージが更新された。ビジネス・オブジェクト・ペイロードが使用可能です。
51201	BPC.HTM.TASK. WISTATUS	破棄	WI_DELETED	作業項目が削除された
51202	BPC.HTM.TASK. WISTATUS	レポート	WI_CREATED	作業項目が作成された
51204	BPC.HTM.TASK. WITRANSFER	レポート	WI_TRANSFERRED	作業項目が転送された
51205	BPC.HTM.TASK. WISTATUS	レポート	WI_REFRESHED	作業項目が更新された

タスク・イベントの場合、以下のイベント関連範囲の ID の内容は次のとおりです。

- ESCcurrentID は、タスク・インスタンスの ID です。
- ECSParentID は、タスク・インスタンス・イベントの前の ECSCurrentID です。

### エスカレーション・イベント

以下の表で、すべてのタスク・エスカレーション・イベントを説明します。

コード	拡張子名	状態	イベント性質	説明
53001	BPC.HTM.ESCALATION. STATUS	レポート	ENTRY	エスカレーションが起動された
53201	BPC.HTM.ESCALATION. WISTATUS	破棄	WI_DELETED	作業項目が削除された
53202	BPC.HTM.ESCALATION. WISTATUS	レポート	WI_CREATED	作業項目が作成された
53204	BPC.HTM.ESCALATION. WITRANSFER	レポート	WI_TRANS-FERRED	エスカレーションが転送された
53205	BPC.HTM.ESCALATION. WISTATUS	レポート	WI_REFRESH-ED	作業項目が更新された

タスク・イベントの場合、以下のイベント関連範囲の ID の内容は次のとおりです。

- ESCcurrentID は、エスカレーションの ID です。
- ECSParentID は、関連したタスク・インスタンスの ID です。

### ヒューマン・タスク・イベントの状態

ヒューマン・タスク・イベントは、さまざまな状態で発行可能です。これらの状態のデータについては、状態エレメントで説明されています。

ヒューマン・タスク・イベントには、次の状態エレメントのいずれかが含まれます。

状態名	Common Base Event の内容	
開始	categoryName は StartSituation に設定されます。	
	situationType	
	Type	StartSituation
	reasoningScope	EXTERNAL
	successDisposition	SUCCESSFUL
	situationQualifier	START_COMPLETED
停止	categoryName は StopSituation に設定されます。	
	situationType	
	Type	StopSituation
	reasoningScope	EXTERNAL
	successDisposition	SUCCESSFUL
	situationQualifier	STOP_COMPLETED
破棄	categoryName は DestroySituation に設定されます。	
	situationType	
	Type	DestroySituation
	reasoningScope	EXTERNAL
	successDisposition	SUCCESSFUL

状態名	Common Base Event の内容	
失敗	categoryName は StopSituation に設定されます。	
	situationType	
	Type	StopSituation
	reasoningScope	EXTERNAL
	successDisposition	UNSUCCESSFUL
	situationQualifier	STOP_COMPLETED
レポート	categoryName は ReportSituation に設定されます。	
	situationType	
	Type	ReportSituation
	reasoningScope	EXTERNAL
	reportCategory	STATUS

## プロセス・サーバーのイベント

WebSphere Process Server は独自のサービス・コンポーネントが特徴で、各サービス・コンポーネントには、モニター可能な独自のイベント・ポイント・セットがあります。サービス・コンポーネントにはエレメントが 1 つ以上あります。それらは各サービス・コンポーネントが処理する異なるステップの集合です。同様に、各エレメントには独自のイベント性質セットがあります。イベント性質は、サービス・コンポーネント・エレメントの処理時に到達するキーポイントです。すべてのサービス・コンポーネント、そのエレメントと関連するイベント性質、および各イベントに固有の拡張データ・エレメントがリストされています。

## リソース・アダプター・イベント

ここでは、モニター可能なリソース・アダプター・コンポーネント (ベース名 eis:WBI.JCAAdapter) のエレメントを、関連するイベント性質、イベント名、および各イベントに固有の拡張データ・エレメントとともにリストします。

イベント名	イベント性質	イベント内容	タイプ
<b>InboundEventRetrieval エレメント</b>			
eis:WBI.JCAAdapter. InboundEventRetrieval. ENTRY	ENTRY	pollQuantity	int
		status	int
		eventTypeFilters	string
eis:WBI.JCAAdapter. InboundEventRetrieval. EXIT	EXIT	なし	
eis:WBI.JCAAdapter. InboundEventRetrieval. FAILURE	FAILURE	FailureReason	例外
<b>InboundEventDelivery エレメント</b>			
eis:WBI.JCAAdapter. InboundEventDelivery.ENTRY	ENTRY	なし	
eis:WBI.JCAAdapter. InboundEventDelivery.EXIT	EXIT	なし	

イベント名	イベント性質	イベント内容	タイプ
eis:WBI.JCAAdapter. InboundEventDelivery.FAILURE	FAILURE	FailureReason	例外
<b>Outbound エlement</b>			
eis:WBI.JCAAdapter. Outbound.ENTRY	ENTRY	なし	
eis:WBI.JCAAdapter. Outbound.EXIT	EXIT	なし	
eis:WBI.JCAAdapter. Outbound.FAILURE	FAILURE	FailureReason	例外
<b>InboundCallbackAsyncDeliverEvent エlement</b>			
eis:WBI.JCAAdapter. InboundCallbackAsyncDeliverEvent. ENTRY	ENTRY	なし	
eis:WBI.JCAAdapter. InboundCallbackAsyncDeliverEvent. EXIT	EXIT	なし	
eis:WBI.JCAAdapter. InboundCallbackAsyncDeliverEvent. FAILURE	FAILURE	FailureReason	例外
<b>InboundCallbackSyncDeliverEvent エlement</b>			
eis:WBI.JCAAdapter. InboundCallbackSyncDeliverEvent. ENTRY	ENTRY	なし	
eis:WBI.JCAAdapter. InboundCallbackSyncDeliverEvent. EXIT	EXIT	なし	
eis:WBI.JCAAdapter. InboundCallbackSyncDeliverEvent. FAILURE	FAILURE	FailureReason	例外
<b>Polling エlement</b>			
eis:WBI.JCAAdapter. Polling.STARTED	STARTED	PollFrequency	int
		PollQuantity	int
eis:WBI.JCAAdapter. Polling.STOPPED	STOPPED	なし	
<b>Delivery エlement</b>			
eis:WBI.JCAAdapter. Delivery.EXIT	EXIT	なし	
eis:WBI.JCAAdapter. Delivery.FAILURE	FAILURE	EventID	string
		FailureReason	例外
<b>Retrieval エlement</b>			
eis:WBI.JCAAdapter. Retrieval.FAILURE	FAILURE	EventID	string
		FailureReason	例外
<b>Endpoint エlement</b>			
eis:WBI.JCAAdapter. Endpoint.FAILURE	FAILURE	FailureReason	例外
<b>Recovery エlement</b>			
eis:WBI.JCAAdapter. Recovery.EXIT	EXIT	なし	

イベント名	イベント性質	イベント内容	タイプ
eis:WBI.JCAAdapter. Recovery.FAILURE	FAILURE	FailureReason	例外
<b>EventFailure エlement</b>			
eis:WBI.JCAAdapter. EventFailure.FAILURE	FAILURE	FailureReason	例外
<b>Connection エlement</b>			
eis:WBI.JCAAdapter. Connection.FAILURE	FAILURE	FailureReason	例外

## ビジネス・ルール・イベント

ビジネス・ルール・コンポーネント (ベース名 br:WBI.BR) には、モニター可能な単一エレメントが含まれています。ここでは、このエレメントのすべてのイベント・タイプを、関連するイベント性質、イベント名、および各イベントに固有の拡張データ・エレメントと共にリストします。

イベント名	イベント性質	イベント内容	タイプ
br:WBI.BR.ENTRY	ENTRY	operationName	string
br:WBI.BR.EXIT	EXIT	operationName	string
br:WBI.BR.FAILURE	FAILURE	ErrorReport	例外
		operationName	string
WBI.BR. br:SelectionKeyExtracted	SelectionKeyExtracted	operationName	string
br:WBI.BR.TargetFound	TargetFound	operationName	string
		target	string

## ビジネス状態マシン・イベント

ここでは、モニター可能なビジネス状態マシン・コンポーネント (ベース名 bsm:WBI.BSM) のエレメントを、関連するイベント性質、イベント名、および各イベントに固有のすべての拡張データ・エレメントと共にリストします。

イベント名	イベント性質	イベント内容	タイプ
<b>StateMachineDefinition エlement</b>			
bsm:WBI.BSM. StateMachineDefinition. ALLOCATED	ALLOCATED	instanceID	string
bsm:WBI.BSM. StateMachineDefinition. RELEASED	RELEASED	instanceID	string
<b>Transition エlement</b>			
bsm:WBI.BSM.Transition.ENTRY	ENTRY	instanceID	string
		name	string
bsm:WBI.BSM.Transition.EXIT	EXIT	instanceID	string
		name	string



イベント名	イベント性質	イベント内容	タイプ
bsm:WBI.BSM.Transition.FAILURE	FAILURE	ErrorReport	例外
		instanceID	string
		name	string
<b>State エlement</b>			
bsm:WBI.BSM.State.ENTRY	ENTRY	instanceID	string
		name	string
bsm:WBI.BSM.State.EXIT	EXIT	instanceID	string
		name	string
bsm:WBI.BSM.State.FAILURE	FAILURE	ErrorReport	例外
		instanceID	string
		name	string
<b>Guard Element</b>			
bsm:WBI.BSM.Guard.ENTRY	ENTRY	instanceID	string
		name	string
bsm:WBI.BSM.Guard.EXIT	EXIT	instanceID	string
		name	string
		result	boolean
bsm:WBI.BSM.Guard.FAILURE	FAILURE	ErrorReport	例外
		instanceID	string
		name	string
<b>Action Element</b>			
bsm:WBI.BSM.Action.ENTRY	ENTRY	instanceID	string
		name	string
bsm:WBI.BSM.Action.EXIT	EXIT	instanceID	string
		name	string
bsm:WBI.BSM.Action.FAILURE	FAILURE	ErrorReport	例外
		instanceID	string
		name	string
<b>EntryAction Element</b>			
bsm:WBI.BSM.EntryAction. ENTRY	ENTRY	instanceID	string
		name	string
bsm:WBI.BSM.EntryAction. EXIT	EXIT	instanceID	string
		name	string
bsm:WBI.BSM.EntryAction. FAILURE	FAILURE	ErrorReport	例外
		instanceID	string
		name	string
<b>ExitAction Element</b>			
bsm:WBI.BSM.ExitAction.ENTRY	ENTRY	instanceID	string
		name	string
bsm:WBI.BSM.ExitAction.EXIT	EXIT	instanceID	string
		name	string

イベント名	イベント性質	イベント内容	タイプ
bsm:WBI.BSM.ExitAction. FAILURE	FAILURE	ErrorReport	例外
		instanceID	string
		name	string
<b>Timer エlement</b>			
bsm:WBI.BSM.Timer.START	START	instanceID	string
		name	string
		duration	string
bsm:WBI.BSM.Timer.STOPPED	STOPPED	instanceID	string
		name	string
		duration	string

## マップ・イベント

ここでは、モニター可能なマップ・コンポーネント (ベース名 map:WBI.MAP) のエレメントを、そのイベント性質、イベント名、および各イベントに固有のすべての拡張データ・エレメントと共にリストします。

表 56. 基本エレメント

イベント名	イベント性質	イベント内容	タイプ
map:WBI.MAP.ENTRY	ENTRY	なし	なし
map:WBI.MAP.EXIT	EXIT	なし	なし
map:WBI.MAP.FAILURE	FAILURE	FailureReason	例外
<b>Transformation エlement</b>			
map:WBI.MAP.Transformation. ENTRY	ENTRY	なし	なし
map:WBI.MAP.Transformation. EXIT	EXIT	なし	なし
map:WBI.MAP.Transformation. FAILURE	FAILURE	FailureReason	例外

## メディエーション・イベント

ここでは、モニター可能なメディエーション・コンポーネント (ベース名 ifm:WBI.MEDIATION) のエレメントを、関連するイベント性質、名前、および各イベントに固有のすべての拡張データ・エレメントと共にリストします。

イベント名	イベント性質	イベント内容	タイプ
<b>OperationBinding エlement</b>			
ifm:WBI.MEDIATION. OperationBinding.ENTRY	ENTRY	InteractionType	string
		TicketID	string
		Source	string
		Target	string

イベント名	イベント性質	イベント内容	タイプ
ifm:WBI.MEDIATION. OperationBinding.EXIT	EXIT	InteractionType	string
		TicketID	string
		Source	string
		Target	string
ifm:WBI.MEDIATION. OperationBinding.FAILURE	FAILURE	InteractionType	string
		TicketID	string
		Source	string
		Target	string
		ErrorReport	例外
<b>ParameterMediation エlement</b>			
ifm:WBI.MEDIATION. ParameterMediation. ENTRY	ENTRY	Type	string
		TransformName	string
WBI.MEDIATION. ParameterMediation. EXIT	EXIT	Type	string
		TransformName	string
ifm:WBI.MEDIATION. ParameterMediation. FAILURE	FAILURE	Type	string
		TransformName	string
		ErrorReport	例外

## リカバリー・イベント

リカバリー・コンポーネント (ベース名 `recovery:WBI.Recovery`) には、モニター可能な単一エレメントが含まれています。ここでは、このエレメントのすべてのイベント・タイプを、関連するイベント性質、イベント名、および各イベントに固有の拡張データ・エレメントと共にリストします。

イベント名	イベント性質	イベント内容	タイプ
recovery:WBI.Recovery. FAILURE	FAILURE	MsgId	string
		DestModuleName	string
		DestComponentName	string
		DestMethodName	string
		SourceModuleName	string
		SourceComponentName	string
		ResubmitDestination	string
		ExceptionDetails	string
		SessionId	string
		FailureTime	dateTime
		ExpirationTime	dateTime
		Status	int
		MessageBody	byteArray
		Deliverable	boolean

イベント名	イベント性質	イベント内容	タイプ
recovery:WBI.Recovery. DEADLOOP	DEADLOOP	DeadloopMsgId	string
		SIBusName	string
		QueueName	string
		Reason	string
recovery:WBI.Recovery. RESUBMIT	RESUBMIT	MsgId	string
		OriginalMesId	string
		ResubmitCount	int
		Description	string
recovery:WBI.Recovery. DELETE	DELETE	MsgId	string
		deleteTime	dateTime
		Description	string

## Service Component Architecture イベント

Service Component Architecture (SCA) には、`sca:WBI.SCA.MethodInvocation` というベース名を持つ単一エレメントが含まれています。ここでは、このエレメントのすべてのイベントと関連する性質を、各イベントに固有のすべての拡張データ・エレメントと共にリストします。

注: これらのイベントを、SCA 固有の ARM パフォーマンス統計と混同しないでください。

イベント名	イベント性質	イベント内容	タイプ
WBI.SCA. MethodInvocation. ENTRY	ENTRY	SOURCE COMPONENT	string
		SOURCE INTERFACE	string
		SOURCE METHOD	string
		SOURCE MODULE	string
		SOURCE REFERENCE	string
		TARGET COMPONENT	string
		TARGET INTERFACE	string
		TARGET METHOD	string
WBI.SCA. MethodInvocation. EXIT	EXIT	SOURCE COMPONENT	string
		SOURCE INTERFACE	string
		SOURCE METHOD	string
		SOURCE MODULE	string
		SOURCE REFERENCE	string
		TARGET COMPONENT	string
		TARGET INTERFACE	string
		TARGET METHOD	string

イベント名	イベント性質	イベント内容	タイプ
WBI.SCA. MethodInvocation. FAILURE	FAILURE	SOURCE COMPONENT	string
		SOURCE INTERFACE	string
		SOURCE METHOD	string
		SOURCE MODULE	string
		SOURCE REFERENCE	string
		TARGET COMPONENT	string
		TARGET INTERFACE	string
		TARGET METHOD	string
		TARGET MODULE	string
		例外	string

## セレクター・イベント

セレクター・コンポーネントには、モニター可能な単一エレメントが含まれています。ここでは、このエレメントのすべてのイベント・タイプを、関連するイベント性質、イベント名、および各イベントに固有の拡張データ・エレメントと共にリストします。すべてのセレクター・イベントのベース名は `sel:WBI.SEL` です。

イベント名	イベント性質	イベント内容	タイプ
<code>sel:WBI.SEL.ENTRY</code>	ENTRY	operationName	string
<code>sel:WBI.SEL.EXIT</code>	EXIT	operationName	string
<code>sel:WBI.SEL.FAILURE</code>	FAILURE	ErrorReport	例外
		operationName	string
<code>sel:WBI.SEL.SelectionKeyExtracted</code>	SelectionKeyExtracted	operationName	string
<code>sel:WBI.SEL.TargetFound</code>	TargetFound	operationName	string
		target	string



---

## 特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒106-8711

東京都港区六本木 3-2-12

日本アイ・ビー・エム株式会社

法務・知的財産

知的財産権ライセンス渉外

以下の保証は、国または地域の法律に沿わない場合は、適用されません。IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Corporation  
577 Airport Blvd., Suite 800  
Burlingame, CA 94010  
U.S.A.

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性があります。その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほめかしたり、保証することはできません。



それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生した創作物にも、次のように、著作権表示を入れていただく必要があります。(c) (お客様の会社名) (西暦年). このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。(C) Copyright IBM Corp. \_年を入れる\_. All rights reserved.

この情報をソフトコピーでご覧になっている場合は、写真やカラーの図表は表示されない場合があります。

## プログラミング・インターフェース情報

プログラミング・インターフェース情報は、プログラムを使用してアプリケーション・ソフトウェアを作成する際に役立ちます。

一般使用プログラミング・インターフェースにより、お客様はこのプログラム・ツール・サービスを含むアプリケーション・ソフトウェアを書くことができます。

ただし、この情報には、診断、修正、および調整情報が含まれている場合があります。診断、修正、調整情報は、お客様のアプリケーション・ソフトウェアのデバッグ支援のために提供されています。

**警告:** 診断、修正、調整情報は、変更される場合がありますので、プログラミング・インターフェースとしては使用しないでください。

## 商標

IBM、IBM logo、developerWorks、Tivoli、および WebSphere は、International Business Machines Corporation の米国およびその他の国における商標です。

Adobe は、Adobe Systems Incorporated の米国およびその他の国における登録商標または商標です。

Java およびすべての Java 関連の商標およびロゴは、Sun Microsystems, Inc. の米国およびその他の国における商標です。

他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。

この製品には、Eclipse Project (<http://www.eclipse.org>) により開発されたソフトウェアが含まれています。



IBM WebSphere Process Server for Multiplatforms バージョン 6.1.0







Printed in Japan