

WebSphere® IBM WebSphere Process Server for
Multiplatforms
バージョン 7.0.0

Common Event Infrastructure



**WebSphere® IBM WebSphere Process Server for
Multiplatforms**
バージョン 7.0.0

Common Event Infrastructure



本書は、WebSphere® Process Server for Multiplatforms バージョン 7、リリース 0、モディフィケーション 0 (製品番号 5724-L01)、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。

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® IBM WebSphere Process Server for Multiplatforms
Version 7.0.0
Common Event Infrastructure

発行： 日本アイ・ビー・エム株式会社

担当： トランスレーション・サービス・センター

第1刷 2010.4

© Copyright IBM Corporation 2007, 2010.

目次

第 1 章 Common Event Infrastructure 1

Common Base Event モデル 3

第 2 章 Common Event Infrastructure の構成 5

Common Event Infrastructure コンポーネント 5

管理コンソールを使用した Common Event Infrastructure の構成 7

Common Event Infrastructure アプリケーションのデプロイ 9

Common Event Infrastructure のクラスターへのデプロイ 10

イベント・メッセージングの構成 13

追加の JMS キューの構成 13

外部 JMS プロバイダーを使用したイベント・メッセージングの構成 14

JMS 認証別名の構成 15

イベント・データベースの構成 16

イベント・データベースの制限 16

Derby イベント・データベースの構成 18

DB2 イベント・データベースの構成

(Linux、UNIX、および Windows システム) 19

z/OS システムでの DB2 データベースの構成 21

iSeries システムでの DB2 データベースの構成 23

Informix イベント・データベースの構成 25

Oracle イベント・データベースの構成 27

SQL サーバー・イベント・データベースの構成 28

データベース構成スクリプトの手動実行 30

以前のバージョンからのイベント・データベースのアップグレード 38

WebSphere Business Monitor 用のクロスセル

Common Event Infrastructure 構成 43

Common Event Infrastructure 構成の除去 43

イベント・データベースの除去 44

Common Event Infrastructure アプリケーションの

除去 45

Common Event Infrastructure サーバーからのイベント・

メッセージングの除去 45

第 3 章 Common Event Infrastructure の管理 47

管理コンソールを使用したイベント・サービスの管理 47

管理コンソールによるイベント・サービスの使用可能と使用不可の切り替え 47

管理コンソールを使用したイベント・エミッター・ファクトリーの作成 48

管理コンソールを使用したイベント・グループの作成 49

管理コンソールを使用したイベント・フィルターの作成 50

スクリプトを使用したイベント・サービスの管理 50

スクリプトによるイベント・サービスの使用可能化 50

スクリプトによるイベント・サービスの使用不可への設定 51

スクリプトによるエミッター・ファクトリーの作成 52

スクリプトによるイベント・グループの作成 55

スクリプトによるイベント・フィルターの作成 57

Common Event Infrastructure コンポーネントのロギングとトレース 58

イベント・データベースの保守 59

DB2 イベント・データベース統計の更新 59

DB2 イベント・データベース表の再編成 60

イベント・データベースからのイベントのパージ 61

第 4 章 Common Event Infrastructure 機能へのアクセス保護 69

第 5 章 Common Event Infrastructure のトラブルシューティング 73

始動時の問題 73

イベント・サービスが始動しない (メッセージ CEIDS0058E) 73

イベント送信時の問題 73

イベントの送信時にエラーが発生する (メッセージ CEIDS0060E) 74

イベントの送信時にエラーが発生する (ServiceUnavailableException) 75

イベントの送信時にエラーが発生する (NameNotFoundException) 75

イベントの送信時にエラーが発生する (メッセージ CEIEM0025E) 76

イベントの送信時にエラーが発生する (メッセージ CEIEM0034E) 76

イベントが無効である (メッセージ CEIEM0027E) 同期モードがサポートされない (メッセージ CEIEM0015E) 78

トランザクション・モードがサポートされない (メッセージ CEIEM0016E) 79

イベントの受信または照会時の問題 80

イベントの照会時にエラーが発生する (メッセージ CEIDS0060E) 80

イベントが永続データ・ストアに保管されない 81

コンシューマーがイベントを受け取らない (エラー・メッセージなし) 82

コンシューマーがイベントを受け取らない (NameNotFoundException) 83

拡張データ・エレメントを持つイベント・グループにイベントが含まれない	84
イベント・グループの照会時にエラーが発生する (メッセージ CEIES0048E).	85
各種の問題	86
Windows システム上でイベント・カタログ・パターン照会が失敗する	86

第 6 章 Common Event Infrastructure コマンド 87

configEventServiceDB2DB コマンド	88
configEventServiceDB2iSeriesDB コマンド	90
configEventServiceDB2ZOSDB コマンド	93
configEventServiceDerbyDB コマンド	96
configEventServiceInformixDB コマンド	98
configEventServiceOracleDB コマンド	100
configEventServiceSQLServerDB コマンド	103

deployEventService コマンド	106
deployEventServiceMdb コマンド	108
setEventServiceJmsAuthAlias コマンド	109
enableEventService コマンド	111
disableEventService コマンド	112
showEventServiceStatus コマンド	113
removeEventService コマンド	115
removeEventServiceMdb コマンド	116
removeEventServiceDB2DB コマンド	117
removeEventServiceDB2iSeriesDB コマンド	119
removeEventServiceDB2ZOSDB コマンド	120
removeEventServiceDerbyDB コマンド	122
removeEventServiceInformixDB コマンド	124
removeEventServiceOracleDB コマンド	125
removeEventServiceSQLServerDB コマンド	127
eventbucket コマンド行ユーティリティ	128
eventpurge コマンド行ユーティリティ	129

第 1 章 Common Event Infrastructure

Common Event Infrastructure は、基本的なイベント管理サービスを、それらのサービスを必要とするアプリケーションに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、サブトピック『Common Base Event モデル』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が、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。

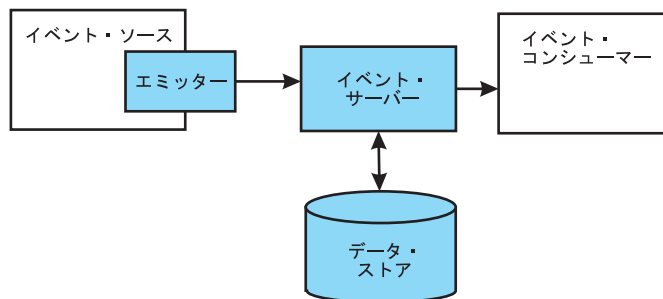
イベント・コンシューマー

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。

第 2 章 Common Event Infrastructure の構成

サーバー AdminTask オブジェクトを使用して、Common Event Infrastructure リソースを構成できるほか、既存のリソースを変更することもできます。

このタスクについて

Common Event Infrastructure (CEI) はデフォルト構成でインストールでき、デフォルト構成はスタンドアロン・サーバー構成で完全に機能します。この構成は、Profile Management Tool を使用してスタンドアロン・サーバー・プロファイルを作成する場合にのみ実行します。これ以外の場合 (Network Deployment 環境またはクラスターにインストールする場合など) は、管理コンソールを使用して CEI を構成し、システムに適した構成となるようにします。

また、wsadmin コマンドを使用して CEI を構成するか、既存の CEI 構成を変更するコマンドを使用することもできます。いずれの場合も、サーバー AdminTask オブジェクトを使用して管理コマンドを実行することにより、CEI の構成を変更します。

CEI 構成を変更した後、サーバーまたはクラスターを再始動する必要があります。

Common Event Infrastructure コンポーネント

Common Event Infrastructure コンポーネントは、一連のアプリケーション、サービス、およびリソースとしてサーバー上にインストールされます。

Common Event Infrastructure を構成すると、いくつかのコンポーネントが作成され、サーバー上にデプロイされます。

Common Event Infrastructure サービス

サーバーにインストールされるサービスで、アプリケーションおよびクライアントが Common Event Infrastructure を使用できるようにします。

Common Event Infrastructure サービスの構成は、次のようにして管理コンソールに表示できます。

- サーバーの場合は、「サーバー」>「アプリケーション・サーバー」>「*server_name*」>「ビジネス・インテグレーション」>「Common Event Infrastructure」>「Common Event Infrastructure サービス」を選択します。
- クラスターの場合は、「サーバー」>「クラスター」>「*cluster_name*」>「ビジネス・インテグレーション」>「Common Event Infrastructure」>「Common Event Infrastructure サービス」を選択します。

「イベント・インフラストラクチャー・サーバーを使用可能に設定」というチェック・ボックスが選択されている場合、サービスはインストール済みで稼働しているか、サーバーまたはクラスターを再始動すると始動します。こ

のチェック・ボックスがクリアされている場合、サービスはインストールされていないか、サーバーまたはクラスターを再始動するとアンインストールされます。

イベント・サービス設定

イベント・サービスが使用する一連のプロパティ。これらのプロパティにより、データ・ストアを使用して、イベントの配布および永続化が可能になります。通常、このリソースに構成は必要ありませんが、同じセルに複数のイベント・サービスをセットアップする場合は、追加のイベント・サービス設定を作成することが必要な場合があります。イベント・サービス設定を表示するには、「サービス統合」>「イベント・サービス」>「イベント・サービス設定」をクリックします。

イベント・メッセージング構成

Java Messaging Service (JMS) を使用して、イベントをイベント・サービスに非同期的に転送することが可能なリソース。デフォルトのメッセージング構成では、サーバー組み込みメッセージが使用されます。必要な場合は、イベント・メッセージング用に外部の JMS プロバイダーを構成することもできます。

イベント・データベース

イベント・データベースは、イベント・サービスが受け取ったイベントを永続的に保管するために使用されます。Derby データベースは、サーバーの一部として組み込まれていますが、実稼働環境での使用には推奨されません。その代わりとして、DB2[®]、Oracle、SQLServer、および Informix[®] 製品で外部イベント・データベースを構成できます。

イベント・フィルター・プラグイン

フィルター・プラグインは、XPath イベント・セレクターを使用して、ソースのイベントをフィルターに掛けるために使用されます。フィルター・プロパティを構成するには、「サービス統合」>「Common Event Infrastructure」>「イベント・エミッター・ファクトリー」>「イベント・フィルター設定」をクリックします。

エミッター・ファクトリー

エミッター・ファクトリーは、エミッターを作成するためにイベント・ソースで使用されるオブジェクトです。エミッターは、イベントをイベント・サービスに送信するために使用されます。エミッター・ファクトリーのプロパティは、そのエミッター・ファクトリーを使用して作成したすべてのエミッターの動作に影響を与えます。フィルター・プロパティを構成するには、「サービス統合」>「Common Event Infrastructure」>「イベント・エミッター・ファクトリー」をクリックします。

イベント・サービス伝送

イベント・サービス伝送は、エミッターが EJB 呼び出しを使用してイベント・サービスに同期的にアクセスする方法を決定するプロパティを定義するオブジェクトです。これらのプロパティは、エミッターを新規作成するときにエミッター・ファクトリーで使用されます。使用可能なイベント・サービス伝送は、エミッター・ファクトリー設定から表示または変更できます。

JMS 伝送

JMS 伝送は、エミッターが JMS キューを使用してイベント・サービスに非

同期にアクセスする方法を決定するプロパティを定義するオブジェクトです。これらのプロパティは、エミッターを新規作成するときにエミッター・ファクトリーで使用されます。使用可能な JMS 伝送は、エミッター・ファクトリー設定から表示または変更できます。

イベント・グループ

イベント・グループは、イベントの論理的な集合であり、イベントをその内容に応じて分類するために使用されます。イベント・サービスからイベントを照会する場合、またはイベント配布をサブスクライブする場合、イベント・コンシューマーは、イベント・グループを指定することにより、そのグループのイベントのみを取得できます。イベント・グループは、永続データ・ストアに保管するイベントを指定するために使用することもできます。使用可能なイベント・グループを管理コンソールで表示するには、「サービス統合」>「Common Event Infrastructure」>「イベント・サービス (Event service)」>「イベント・サービス (Event services)」>「event_service」>「イベント・グループ」をクリックします。

管理コンソールを使用した Common Event Infrastructure の構成

サーバー管理コンソールを使用して Common Event Infrastructure を構成します。

このタスクについて

管理コンソールの「Common Event Infrastructure サーバー」パネルを開きます。

サーバーを構成する場合は、「サーバー」>「サーバー・タイプ」>「WebSphere Application Server」> *server_name* >「ビジネス・インテグレーション」>「Common Event Infrastructure」>「Common Event Infrastructure サーバー」を選択します。

クラスターを構成する場合は、「サーバー」>「クラスター」>「WebSphere Application Server クラスター」> *cluster_name* >「ビジネス・インテグレーション」>「Common Event Infrastructure」>「Common Event Infrastructure サーバー」をクリックします。

手順

1. Common Event Infrastructure エンタープライズ・アプリケーションのデプロイメントを可能にするために、「イベント・インフラストラクチャー・サーバーを使用可能に設定」というチェック・ボックスを選択します。サーバーが既に構成済みである場合は、このチェック・ボックスを選択するかクリアすることによって、サーバーを使用可能または使用不可に設定できます。使用可能に設定するチェック・ボックスがクリアされている場合、Common Event Infrastructure はまだ構成されていないか、以前の構成が既に無効にされており、サーバーが再始動されていません。情報メッセージにより、このデプロイメント・ターゲットに Common Event Infrastructure が構成済みであるかどうかを示されます。サーバーが既に構成済みである場合は、イベント・データベースかメッセージ・ストア、またはその両方のデータ・ソース設定を変更できます。

注: サーバーがまだ構成されていない場合に、チェック・ボックスを選択して Common Event Infrastructure サーバーを使用可能にすると、表示されたパラメーターを変更しない限り、それらのパラメーターを使用してサーバーが構成されません。

- 初めて構成を行う場合は、共通データベースにイベント・データ・ソース・テーブルが作成されます。既に Common Event Infrastructure サーバー構成がある場合は、データベースを作成する必要があります。
- 共通データベースの下で、固有のスキーマの下にメッセージング・サービスが作成されます。

Common Event Infrastructure が構成されたサーバーまたはクラスターが再始動されると、新しい変更が有効になります。

2. イベント・データベースを構成 (または、既存構成の現行の設定を変更) するために、次のいずれかの方法で各フィールドに適切な設定値を入力します。
 - 「編集」をクリックして、このパネルよりも詳細なオプション・リストを持つデータベース構成パネルを表示します。
 - パネルの各フィールドを使用して、情報を入力します。
 - a. 「データベース名」 - イベントの保管に使用するデータベースの名前を入力します。
 - b. 「テーブルの作成」 - イベント・データベースにデータベース・テーブルを作成する場合は、このチェック・ボックスを選択します。

注: 別のサーバーにあるデータベースを使用するように Common Event Infrastructure を構成する場合は、このコントロールを使用してテーブルを作成することはできません。代わりに、この構成の残りの作業を完了した後に生成されるデータベース・スクリプトを使用する必要があります。その場合、「編集」をクリックしてデータ・ソース詳細パネルを表示すると、データベース作成スクリプトの場所が表示されます。

- c. 「ユーザー名」および「パスワード」 - イベント・データベースの認証用。
- d. 「サーバー」 - イベント・データベースが存在するサーバーの名前。
- e. 「プロバイダー」 - データベースのプロバイダーをメニューから選択します。

注: 「スキーマ」フィールドは、iSeries® または z/OS® プラットフォーム上で DB2 を使用してデータベースを作成する場合のみアクティブになります。それ以外の場合、「スキーマ」フィールドは使用できません。

重要: ターゲット・データベースにテーブルが存在する場合は、構成が失敗することがあります。

3. Common Event Infrastructure バスがサーバー上で「ローカル」として存在するか、あるいは「リモート」として他のサーバー上に存在するかを選択します。リモートを選択した場合は、メニューからリモート・ロケーションを選択するか、「新規作成 (New)」をクリックして新規リモート・バスを作成します。
4. メッセージング用の Common Event Infrastructure サポートを構成します。
 - 「編集」をクリックして、このパネルよりも詳細なオプション・リストを持つデータベース構成パネルを表示します。

- パネルの各フィールドを使用して、情報を入力します。
 - a. 「データベース名」 - メッセージの保管に使用するデータベースの名前を入力します。
 - b. 「スキーマ」 - スキーマの名前を入力するか、表示されたデフォルト名を受け入れます。
 - c. 「ユーザー名」および「パスワード」 - メッセージング・データベースの認証用。
 - d. 「サーバー」 - メッセージング・データベースが存在するサーバーの名前。
 - e. 「プロバイダー」 - データベースのプロバイダーをメニューから選択します。
- 5. Common Event Infrastructure バスのメッセージング認証別名を作成します。
 - a. 「追加プロパティ」 > 「JMS 認証別名」を選択します。
 - b. システム統合バスでのセキュア通信に使用するユーザー ID およびパスワードを入力します。 セキュリティーが使用不可の場合は、デフォルトの構成済み値である「CEI」をユーザー ID とパスワードの両方に受け入れることができます。セキュリティーが使用可能に設定されている場合は、バス認証に使用するユーザー ID とパスワードを入力します。実稼働環境では、独自のユーザー ID とパスワードを選択してシステムを保護します。
 - c. 「OK」をクリックします。
- 6. 「OK」または「適用」をクリックします。
- 7. サーバーまたはクラスターを再始動します。

タスクの結果

これで、Common Event Infrastructure の主要な部分がすべて構成されて、サーバーまたはクラスターで稼働するようになります。この構成にはイベント・データ・ストア、メッセージング・エンジン、およびイベント・アプリケーションが含まれます。この単一のパネルを、本来なら Common Event Infrastructure の構成に使用する多数のコマンドやステップの代わりに使用できます。

次のタスク

サーバーまたはクラスターを再始動した後、アプリケーションから出力されるサービス・コンポーネント・イベントを保管できるようになります。この時点で、「Common Event Infrastructure の宛先」パネルを選択することにより、Common Event Infrastructure サーバーのランタイム・プロパティを変更できます。また、始動時に Common Event Infrastructure サーバーを始動するかどうかを選択したり、イベントが送信されるエミッター・ファクトリー JNDI 名を指定することができます。

Common Event Infrastructure アプリケーションのデプロイ

Common Event Infrastructure を使用するには、まずイベント・サービスおよび関連するリソースをサーバー・ランタイム環境にデプロイしておく必要があります。

このタスクについて

Common Event Infrastructure エンタープライズ・アプリケーションには、イベント・サービスのランタイム・コンポーネントと、非同期イベント送信で使用されるデフォルトのメッセージング構成が含まれます。

イベント・サービスをデプロイするには、以下の手順を実行します。

手順

wsadmin ツールから **deployEventService** 管理コマンドをバッチ・モードまたは対話モードで実行します。 **deployEventService** 管理コマンドのパラメーターは、以下のとおりです。

nodeName

イベント・サービスをデプロイするノードの名前。このパラメーターは任意指定です。ノード名を指定しない場合、デフォルト値は現在のノードです。ノード名を指定する場合は、**serverName** パラメーターを使用してサーバー名も指定する必要があります。クラスターにイベント・サービスをデプロイする場合は、このパラメーターは無効です。

serverName

イベント・サービスをデプロイするサーバーの名前。このパラメーターは、ノードを指定する場合にのみ必要です。クラスターにイベント・サービスをデプロイする場合は、このパラメーターは無効です。

clusterName

イベント・サービスをデプロイするクラスターの名前。このパラメーターは任意指定です。ノードまたはサーバーの有効範囲にデプロイする場合は指定しないでください。

enable

サーバー始動時にイベント・サービスを自動的に開始するかどうかを示します。デフォルト値は **true** です。

タスクの結果

管理コマンドが完了すると、Common Event Infrastructure のイベント・サービスおよびデフォルトのメッセージング構成が、指定された有効範囲にデプロイされます。

次のタスク

WebSphere® セキュリティーが使用可能な場合は、**setEventServiceJmsAuthAlias** 管理コマンドを使用して JMS 認証別名とパスワードを構成する必要もあります。

クラスターにイベント・サービスをデプロイする場合は、イベント・データベースを手動で構成する必要もあります。

Common Event Infrastructure のクラスターへのデプロイ

Common Event Infrastructure リソースは、いくつかの方法でクラスター環境にデプロイできます。

Common Event Infrastructure の既存のクラスターへのデプロイ

既存のクラスターにイベント・サービス・アプリケーションをデプロイできます。

このタスクについて

イベント・サービス・アプリケーションをクラスターにデプロイすることは、アプリケーションをスタンドアロン・サーバーにデプロイすることと本質的には同じです。ただし、クラスター環境では、デフォルトのイベント・データベースは構成されません。

Common Event Infrastructure をクラスター環境にデプロイして構成するには、以下の手順を実行します。

手順

1. クラスターの名前を指定し、それ以外はスタンドアロン・サーバーの場合と同様に、**deployEventService** 管理コマンドを実行します。クラスターを指定するには、**clusterName** パラメーターを使用します。
2. デプロイメント・マネージャー・システムで、データベース構成管理コマンドを実行します。**clusterName** パラメーターを使用して、クラスター名を指定します。このコマンドにより、データベース構成スクリプトが生成されます。
3. 生成されたデータベース構成スクリプトをデータベース・システムにコピーします。
4. データベース・システムに対してデータベース構成スクリプトを実行して、イベント・データベースを作成します。
5. デプロイメント・マネージャー・システムで **enableEventService** コマンドを実行して、イベント・サービスを使用可能にします。**clusterName** パラメーターを使用して、クラスターの名前を指定します。

既存の Common Event Infrastructure サーバーの変換によるクラスターの作成

Common Event Infrastructure と共に構成されている既存のスタンドアロン・サーバーを変換して、クラスターを作成することができます。

始める前に

既存のサーバーを変換する前に、そのサーバーが Common Event Infrastructure 用に完全に構成されていることを確認してください。この構成には、イベント・サービス・アプリケーションのデプロイとイベント・データベースの構成も含まれます。

このタスクについて

クラスターを作成するには、以下の手順を実行します。

手順

1. 通常の WebSphere プロセスに従って、スタンドアロン・サーバーを新しいクラスターの最初のメンバーに変換します。サーバーを変換するとき、以下のステップが実行されます。
 - サーバーの有効範囲にある使用可能な Common Event Infrastructure リソースが、新しいクラスターの有効範囲に移動されます。

デフォルト・データベース: 既存のサーバーがデフォルトの Derby データベースと共に構成されている場合、データベース・リソースは、クラスターの有効範囲に移動されずに除去されます。デフォルトのデータベース構成は、クラスターでサポートされません。この場合、デフォルトでは、クラスター内のイベント・サービスは使用不可になります。

- デプロイするイベント・サービス・アプリケーションのターゲット・リストが変更され、変換されるサーバーが除去されて新しいクラスターが追加されず。
2. オプション: 変換されるサーバーがデフォルトの Derby データベースと共に構成されていた場合は、クラスターの新しいイベント・データベースを構成してから、イベント・サービスを使用可能にする必要があります。
 - a. デプロイメント・マネージャー・システムで、データベース構成管理コマンドを実行します。 `clusterName` パラメーターを使用して、クラスター名を指定します。このコマンドにより、データベース構成スクリプトが生成されます。
 - b. 生成されたデータベース構成スクリプトをデータベース・システムにコピーします。
 - c. データベース・システムに対してデータベース構成スクリプトを実行して、イベント・データベースを作成します。
 - d. デプロイメント・マネージャー・システムで `enableEventService` コマンドを実行して、イベント・サービスを使用可能にします。 `clusterName` パラメーターを使用して、クラスターの名前を指定します。

既存の Common Event Infrastructure サーバーをテンプレートとして使用してクラスターを作成する

既存の Common Event Infrastructure サーバーをテンプレートとして指定することにより、クラスターを作成できます。

始める前に

この方法を使用してクラスターを作成する前に、Common Event Infrastructure 用に完全に構成済みである既存のサーバーを用意する必要があります。この構成には、イベント・サービス・アプリケーションのデプロイとイベント・データベースの構成も含まれます。

このタスクについて

クラスターを作成するには、以下の手順を実行します。

手順

1. 通常の WebSphere プロセスに従い、既存の Common Event Infrastructure サーバーを最初のクラスター・メンバーのテンプレートとして使用して、新しいクラスターを作成します。最初のメンバーが作成されるときに、以下のステップが実行されます。
 - 既存のサーバーの有効範囲にある使用可能な Common Event Infrastructure リソースが、新しいクラスターの有効範囲にコピーされます。

デフォルト・データベース: 既存のサーバーがデフォルトの Derby データベースと共に構成されている場合、データベース・リソースは、クラスターの有効範囲にコピーされません。デフォルトのデータベース構成は、クラスターでサポートされません。この場合、デフォルトでは、クラスター内のイベント・サービスは使用不可になります。

- デプロイするイベント・サービス・アプリケーションのターゲット・リストが変更され、新しいクラスターが組み込まれます。
2. オプション: 既存のサーバーがデフォルトの Derby データベースと共に構成されていた場合は、クラスターの新しいイベント・データベースを構成してから、イベント・サービスを使用可能にする必要があります。
 - a. デプロイメント・マネージャー・システムで、データベース構成管理コマンドを実行します。 `clusterName` パラメーターを使用して、クラスター名を指定します。このコマンドにより、データベース構成スクリプトが生成されます。
 - b. 生成されたデータベース構成スクリプトをデータベース・システムにコピーします。
 - c. データベース・システムに対してデータベース構成スクリプトを実行して、イベント・データベースを作成します。
 - d. デプロイメント・マネージャー・システムで `enableEventService` コマンドを実行して、イベント・サービスを使用可能にします。 `clusterName` パラメーターを使用して、クラスターの名前を指定します。

イベント・メッセージングの構成

JMS でイベントをイベント・サービスに転送する場合に使用するメッセージングの構成を変更できます。

このタスクについて

管理コンソール・パネルを使用して、サーバー上に `Common Event Infrastructure` を構成するときは、`Common Event Infrastructure` のメッセージング・インフラストラクチャーを作成します。一般に、メッセージング構成は、イベント・サービスへのイベントの非同期伝送にデフォルトのメッセージング・プロバイダーを使用し、単一の JMS キューを作成します。必要であれば、このメッセージング構成を変更できます。

追加の JMS キューの構成

デフォルトのイベント・メッセージング構成を使用している場合は、イベントをイベント・サービスに転送するために JMS キューを追加できます。

このタスクについて

デフォルトのメッセージング構成を使用して追加の JMS キューを構成するには、サービス統合バスのキュー宛先に転送される複数の JMS キューをセットアップします。`Common Event Infrastructure` サービス統合バス のキュー宛先は、イベント・サービスがデプロイされる有効範囲によって決まります。

有効範囲	サービス統合バスのキュー宛先
サーバー	<code>node.server.CommonEventInfrastructureQueueDestination</code>
クラスター	<code>cluster.CommonEventInfrastructureQueueDestination</code>

サービス統合バスの構成の詳細については、資料を参照してください。

外部 JMS プロバイダーを使用したイベント・メッセージングの構成

デフォルトの組み込みメッセージング構成をイベント伝送で使用しない場合は、非同期メッセージ・トランスポートを構成することにより、外部の Java Messaging Service (JMS) プロバイダーを使用できます。

始める前に

外部 JMS プロバイダーを使用してイベント・メッセージングを構成する前に、まず使用している JMS プロバイダーに適したインターフェースを使用して JMS キューおよび接続ファクトリーを作成する必要があります。リスナー・ポートまたはアクティベーション・スペックを作成する必要もあります。

このタスクについて

外部 JMS プロバイダーを使用してイベント・メッセージングを構成するには、以下の手順を実行します。

手順

wsadmin ツールから **deployEventServiceMdb** 管理コマンドをバッチ・モードまたは対話モードで実行します。 **deployEventServiceMdb** コマンドのパラメーターは、以下のとおりです。

applicationName

デプロイするイベント・サービスのメッセージ駆動型 Bean のアプリケーション名。このパラメーターは必須です。

nodeName

イベント・サービスのメッセージ駆動型 Bean をデプロイするノードの名前を指定します。ノード名を指定する場合は、サーバー名も指定する必要があります。ノード名はオプション・パラメーターで、デフォルト値は現在のノードです。クラスターにアプリケーションをデプロイする場合は、このパラメーターを指定しないでください。

serverName

イベント・サービスのメッセージ駆動型 Bean をデプロイするサーバーの名前を指定します。このパラメーターは、サーバーの有効範囲にアプリケーションをデプロイする場合、必須です。それ以外の場合は、任意指定です。クラスターにアプリケーションをデプロイする場合は、サーバー名を指定しないでください。

clusterName

イベント・サービスのメッセージ駆動型 Bean をデプロイするクラスターの名前を指定します。このパラメーターは、アプリケーションをクラスターにデプロイする場合にのみ指定します。

listenerPort

イベントの公開時にイベント・サービスのメッセージ駆動型 Bean が使用するリスナー・ポートの名前を指定します。指定するリスナー・ポートは存在している必要があります。リスナー・ポートとアクティベーション・スペックのいずれかを指定する必要がありますが、両方を指定しないでください。

activationSpec

イベントの公開時にイベント・サービスのメッセージ駆動型 Bean が使用するアクティベーション・スペックの JNDI 名を指定します。指定するアクティベーション・スペックは存在している必要があります。リスナー・ポートとアクティベーション・スペックのいずれかを指定する必要がありますが、両方を指定しないでください。

qcfJndiName

イベント・サービスのメッセージ駆動型 Bean が使用する JMS キュー接続ファクトリーの JNDI 名。このパラメーターは、アクティベーション・スペックを指定する場合、必須です。それ以外の場合は、任意指定です。キュー接続ファクトリーとリスナー・ポートを指定する場合、キュー接続ファクトリーは、そのリスナー・ポート用に構成されたものと一致する必要があります。

タスクの結果

deployEventServiceMdb 管理コマンドは、指定したリスナー・ポートまたはアクティベーション・スペック用に構成された、イベント・サービスのメッセージ駆動型 Bean をデプロイします。外部の JMS 構成を使用して、エミッター・ファクトリーと JMS 伝送の作成も行います。アプリケーションは、(デフォルトのメッセージング構成を使用するように構成された) デフォルトのエミッター・ファクトリー、または (外部の JMS プロバイダーを使用する) 新しいエミッター・ファクトリーを使用できます。

次のタスク

イベント・サービスに対して複数の JMS キューをセットアップする場合は、異なるエンタープライズ・アプリケーション名と JMS キューを指定して、このコマンドを複数回実行します。スクリプトを実行するたびに、追加のメッセージ駆動型 Bean がデプロイされ、新規リソースで指定された JMS キューを使用するように構成されます。

JMS 認証別名の構成

WebSphere セキュリティーが使用可能で、非同期 JMS メッセージングを使用してイベントをイベント・サービスに送信する場合は、JMS 認証別名を構成する必要があります。

このタスクについて

JMS 認証別名を構成するには、以下の手順を実行します。

手順

wsadmin ツールから **setEventServiceJmsAuthAlias** 管理コマンドをバッチ・モードまたは対話モードで実行します。 **setEventServiceJmsAuthAlias** コマンドのパラメーターは、以下のとおりです。

userName

JMS 認証別名で使用するユーザーの名前。このパラメーターは必須です。

password

JMS 認証別名で使用するユーザーのパスワード。このパラメーターは必須です。

nodeName

JMS 認証別名を更新または作成するノードの名前。ノード名を指定する場合は、サーバー名も指定する必要があります。クラスターの認証別名を構成する場合は、ノード名を指定しないでください。

serverName

JMS 認証別名を更新または作成するサーバーの名前。このパラメーターは、ノードを指定する場合にのみ必要です。クラスターで認証別名を構成する場合は、このパラメーターは無効です。

clusterName

JMS 認証別名を更新または作成するクラスターの名前。このパラメーターは、クラスターで認証別名を構成する場合にのみ指定します。クラスター名を指定する場合は、ノード名またはサーバー名を指定しないでください。

タスクの結果

イベント・サービス・オブジェクトが使用する JMS 認証別名は、指定された有効範囲で更新されます。認証別名が存在しない場合は、指定された値を使用して認証別名が作成されます。

イベント・データベースの構成

サポートされている各データベース製品に固有のコマンドを使用して、イベント・データ・ソースを構成できます。

このタスクについて

イベント・データベースでは、イベントの永続性をサポートする必要があります。管理コンソールの **Common Event Infrastructure** 構成パネルを使用しなかった場合でも、ここで説明するコマンドを使用すればイベント・データベースを作成できます。

イベント・データベースの制限

特定のデータベース・ソフトウェアを使用してイベント・データベースを構成する場合は、いくつかの制限が適用されます。

ご使用の環境に適用される制限については、以下の表を参照してください。

表 1. イベント・データベースの制限

データベース・タイプ	制限
Oracle	<ul style="list-style-type: none"> • Oracle 11 JDBC Thin ドライバーでは、Unicode 文字セットを使用する場合にストリング値のサイズ制限があります。大きい値 (長メッセージ属性など) を含むイベントがイベント・データベースに保管されると、Oracle ORA-01461 エラーが発生する可能性があります。この制限についての詳細は、Oracle 11 の資料を参照してください。 <p>この問題を避けるには、Oracle 11 OCI ドライバーまたは Oracle 11 Thin ドライバーを使用してください。</p> • Oracle データベース・ソフトウェアは、ブランクのストリングを NULL 値として扱います。ブランクのストリングをイベント属性値として指定すると、そのストリングは、Oracle イベント・データベースに保管されるときに NULL に変換されます。
Informix	<ul style="list-style-type: none"> • JDBC 3.0 ドライバー (またはそれ以降) が必要です。以前のバージョンの JDBC ドライバーでは、必要な XA トランザクションが完全にはサポートされません。 • configEventServiceInformixDB 管理コマンドにより生成されるデータベース構成スクリプトおよびデータベース除去スクリプトでは、SQL スクリプトを実行するために dbaccess コマンドが必要です。このコマンドは、Informix サーバー上でしか使用できない可能性があります。したがって、Informix サーバーが WebSphere サーバーとは別のシステム上にある場合は、データベース構成スクリプトを Informix サーバーにコピーして、ローカルで実行する必要があります。

表 1. イベント・データベースの制限 (続き)

データベース・タイプ	制限
SQL サーバー	<ul style="list-style-type: none"> • SQL サーバー・データベースは、混在認証モードを使用するように構成する必要があります。トラステッド接続はサポートされません。 • XA ストアド・プロシージャをインストールする必要があります。このストアド・プロシージャは、Microsoft[®] Corporation が提供する JDBC ドライバーに同梱されています。 • PATH ステートメントで指定したディレクトリに sqljdbc.dll ファイルが含まれ、使用できる必要があります。このファイルは、Microsoft Corporation が提供する JDBC ドライバーに同梱されています。 • Distributed Transaction Coordinator (DTC) サービスを開始する必要があります。

Derby イベント・データベースの構成

Linux[®]、UNIX[®]、または Windows[®] システム上のサーバーまたはクラスターの有効範囲にある Derby イベント・データベースを構成できます。

このタスクについて

イベント・データベースに使用できる Derby データベースには、Derby Embedded と Derby Network の 2 つのタイプがあります。どちらのタイプも WebSphere Application Server に添付されていますが、機能が制限されており、稼働環境には適していません。したがって、Derby は開発やテストなどの目的でのみ、イベント・データベースとして使用してください。Derby データベースの詳細については、WebSphere Application Server の (このページの下部にリンクがある) 資料を参照してください。

Derby Embedded は、スタンドアロン・サーバーでのみ使用できます。したがって、スタンドアロン・サーバーをクラスター環境または ND 環境に統合する場合は、別のデータベース製品を使用してイベント・データ・ソースを完全に再構成する必要があります。この処理は、サーバーを始動すると自動的に開始されます。

Derby Network はクラスター環境または ND 環境で使用できますが、Derby Embedded と同じように、実際の稼働システムでの使用には適していません。これをサーバーで使用するには、データベースを手動で開始する必要があります。

Derby イベント・データベースを構成するには、以下の手順を実行します。

手順

1. wsadmin ツールを開始します。

- AdminTask オブジェクトを使用して、**configEventServiceDerbyDB** 管理コマンドをバッチ・モードまたは対話モードで実行します。**configEventServerDerbyDB** コマンドに最低限指定する必要のあるパラメーターを以下に示します。

createDB

データベース構成スクリプトを管理コマンドで作成して実行するかどうかを指定します。**true** または **false** を指定します。このパラメーターを **false** に設定すると、スクリプトは作成されますが、実行されません。この場合、データベース構成を完了するにはデータベース構成スクリプトを実行する必要があります。

nodeName

イベント・サービス・データ・ソースを作成するサーバーを含むノードの名前。ノード名を指定する場合は、サーバー名も指定する必要があります。以下のいずれかを指定してください。

- ・ ノード名およびサーバー名
- ・ クラスタ名

serverName

イベント・サービス・データ・ソースを作成するサーバーの名前。

clusterName

イベント・サービス・データ・ソースを作成するクラスタの名前。クラスタ名を指定する場合は、ノード名とサーバー名を指定しないでください。

環境によっては、その他のパラメーターが必要になる場合もあります。詳細なパラメーターのリストとその使用方法については、**configEventServiceDerbyDB** 管理コマンドのヘルプを参照してください。

タスクの結果

管理コマンドは、必要なデータ・ソースを指定された有効範囲に作成します。**createDB** パラメーターに **true** を指定した場合、コマンドは、生成されたデータベース構成スクリプトも実行し、データベースを作成します。

デフォルトでは、生成されたデータベース構成スクリプトは、*profile_root/databases/event/node/server/dbscripts/derby* ディレクトリーに保管されます。(Network Deployment 環境では、これらのスクリプトは、デプロイメント・マネージャーのプロファイル・ディレクトリーの下に保管されます。) 任意指定の **outputScriptDir** パラメーターに値を指定した場合は、その場所にスクリプトが保管されます。これらのスクリプトを使用すると、イベント・データベースをいつでも手動で構成できます。

DB2 イベント・データベースの構成 (Linux、UNIX、および Windows システム)

Linux、UNIX、または Windows システムで、DB2 Universal Database™ を使用して、外部イベント・データベースを構成できます。

このタスクについて

Linux、UNIX、または Windows システムで DB2 イベント・データベースを構成するには、以下の手順を実行します。

手順

1. wsadmin ツールを開始します。
2. AdminTask オブジェクトを使用して、**configEventServiceDB2DB** 管理コマンドをバッチ・モードまたは対話モードで実行します。**configEventServiceDB2DB** コマンドに最低限指定する必要があるパラメーターを以下に示します。

createDB

データベース構成スクリプトを管理コマンドで作成して実行するかどうかを指定します。 **true** または **false** を指定します。このパラメーターを **false** に設定すると、スクリプトは作成されますが、実行されません。この場合、データベース構成を完了するにはデータベース構成スクリプトを実行する必要があります。

nodeName

イベント・サービス・データ・ソースを作成するサーバーを含むノードの名前。ノード名を指定する場合は、サーバー名も指定する必要があります。以下のいずれかを指定してください。

- ノード名およびサーバー名
- クラスター名

serverName

イベント・サービス・データ・ソースを作成するサーバーの名前。

clusterName

イベント・サービス・データ・ソースを作成するクラスターの名前。クラスター名を指定する場合は、ノード名とサーバー名を指定しないでください。

jdbcClassPath

JDBC ドライバーへのパス。ドライバー・ファイルのパスのみを指定します。ファイル名は指定しないでください。

dbHostName

データベースがインストールされているサーバーのホスト名。

dbUser

イベント・データベースの作成時に使用する DB2 ユーザー ID。指定するユーザー ID には、データベースを作成および除去するための十分な特権が必要です。

dbPassword

使用する DB2 パスワード。

環境によっては、その他のパラメーターが必要になる場合もあります。詳細なパラメーターのリストとその用法については、**configEventServiceDB2DB** 管理コマンドのヘルプを参照してください。

タスクの結果

管理コマンドは、必要なデータ・ソースを指定された有効範囲に作成します。
createDB パラメーターに true を指定した場合、コマンドは、生成されたデータベース構成スクリプトも実行し、データベースを作成します。

デフォルトでは、生成されたデータベース構成スクリプトは、
profile_root/databases/event/node/server/dbscripts/db2 ディレクトリーに保管されます。
(Network Deployment 環境では、これらのスクリプトは、デプロイメント・マネージャーのプロファイル・ディレクトリーの下に保管されます。) 任意指定の
outputScriptDir パラメーターに値を指定した場合は、その場所にスクリプトが保管されます。これらのスクリプトを使用すると、イベント・データベースをいつでも手動で構成できます。

z/OS システムでの DB2 データベースの構成

DB2 データベース・ソフトウェアを使用して、z/OS システムでイベント・データベースを構成することができます。

始める前に

リモート・クライアントから DB2 データベースを構成するには、最新のフィックスパックを含む DB2 Connect 製品がインストールされている必要があります。

このタスクについて

イベント・データベースを構成するには、以下の手順を実行します。

手順

1. Linux UNIX Windows z/OS イベント・データベースを Linux、UNIX、または Windows クライアント・システムから構成する場合は、以下のステップを実行して、データベースを作成およびカタログします。
 - a. z/OS システム上で、DB2 管理メニューを使用してサブシステムを作成します。
 - b. オプション: イベント・データベースに使用するストレージ・グループを作成します。既存のストレージ・グループ (sysdeflt など) を使用することもできます。
 - c. イベント・データベースで使用する 4 K、8 K、および 16 K のバッファークラスタ・プールを使用可能にします。
 - d. データ・ソースが使用するユーザー ID に対して必要な権限を付与します。このユーザー ID には、作成したデータベースおよびストレージ・グループへのアクセス権限がなければなりません。また、データベース用新規テーブル、テーブル・スペース、および索引を作成する権限も必要です。
 - e. リモート・データベースをカタログします。スクリプトまたは DB2 コマンド行ウィンドウで、以下のコマンドを実行します。

```
catalog tcpip node zosnode remote hostname server IP_port
system db_subsystem
catalog database db_name as db_name at node zosnode authentication DCS
```

ノードとそのデータベースをカタログする方法については、DB2 Connect の資料を参照してください。

- f. リモート・サブシステムへの接続を確立できることを確認してください。この確認を行うには、以下のコマンドを実行します。

```
db2 connect to subsystem user userid using password
```

- g. ホスト・データベースにバインドします。以下のコマンドを実行します。

```
db2 connect to db_name user userid using password
db2 bind db2_root/bnd/@ddcsmvs.lst blocking all sqlerror continue message
      mvs.msg grant public
db2 connect reset
```

クライアントをホスト・データベースにバインドする方法について詳しくは、DB2 Connect の資料を参照してください。

2. WebSphere システムで、wsadmin ツールを開始します。
3. AdminTask オブジェクトを使用して、**configEventServiceDB2ZOSDB** 管理コマンドをバッチ・モードまたは対話モードで実行します。

configEventServiceDB2ZOSDB コマンドに最低限指定する必要があるパラメーターを以下に示します。

createDB

Linux **UNIX** **Windows** データベース構成スクリプトを管理コマンドで作成して実行するかどうかを指定します。このパラメーターは、管理コマンドを **Linux**、**UNIX**、または **Windows** クライアント・システムから実行する場合にのみ適用されます。 **true** または **false** を指定します。

このパラメーターを **false** に設定するか、z/OS システム上でコマンドを実行すると、スクリプトは作成されますが、実行されません。この場合、データベース構成を完了するにはデータベース構成スクリプトを実行する必要があります。

nodeName

イベント・サービス・データ・ソースを作成するサーバーを含むノードの名前。ノード名を指定する場合は、サーバー名も指定する必要があります。以下のいずれかを指定してください。

- ノード名およびサーバー名
- クラスタ名

serverName

イベント・サービス・データ・ソースを作成するサーバーの名前。

clusterName

イベント・サービス・データ・ソースを作成するクラスタの名前。クラスタ名を指定する場合は、ノード名とサーバー名を指定しないでください。

jdbcClassPath

JDBC ドライバーへのパス。ドライバー・ファイルのパスのみを指定します。ファイル名は指定しないでください。

dbHostName

データベースがインストールされているサーバーのホスト名。

dbUser

イベント・データベースの作成時に使用する DB2 ユーザー ID。指定するユーザー ID には、データベースを作成および除去するための十分な特権が必要です。

dbPassword

使用する DB2 パスワード。

環境によっては、その他のパラメーターが必要になる場合もあります。詳細なパラメーターのリストとその使用方法については、**configEventServiceDB2ZOSDB** 管理コマンドのヘルプを参照してください。

タスクの結果

管理コマンドは、必要なデータ・ソースを指定された有効範囲に作成します。また、コマンドを Linux、UNIX、または Windows DB2 クライアント上で実行している場合、createDB パラメーターに true を指定すると、コマンドは、生成されたデータベース構成スクリプトも実行し、データベースを作成します。z/OS システムの場合は、SQL Processor Using File Input (SPUFI) 機能を使用して、生成された DDL ファイルを実行する必要があります。DDL ファイルは、*profile_root/databases/event/node/server/db2zos/ddl* ディレクトリーに保管されます。

デフォルトでは、生成されたデータベース構成スクリプトは、*profile_root/databases/event/node/server/dbscripts/db2zos* ディレクトリーに保管されます。(Network Deployment 環境では、これらのスクリプトは、デプロイメント・マネージャーのプロファイル・ディレクトリーの下に保管されます。) 任意指定の outputScriptDir パラメーターに値を指定した場合は、その場所にスクリプトが保管されます。これらのスクリプトを使用すると、イベント・データベースをいつでも手動で構成できます。

次のタスク

データベースの構成が終了した後、サーバー管理コンソールを使用してデータベース構成をテストできます。このタスクを実行する場合、該当する JDBC データ・ソースにナビゲートして「テスト接続」オプションを選択します。

iSeries システムでの DB2 データベースの構成

DB2 データベース・ソフトウェアを使用して、iSeries システムでイベント・データベースを構成することができます。

このタスクについて

ローカル iSeries サーバーを使用してリモート iSeries サーバーを構成する場合は、ローカル・サーバー上のリモート・データベース項目をターゲット・データベースに対して別名として指定する必要があります。イベント・データベースを構成するには、以下の手順を実行します。

手順

1. wsadmin ツールを開始します。

- AdminTask オブジェクトを使用して、**configEventServiceDB2iSeriesDB** 管理コマンドをバッチ・モードまたは対話モードで実行します。

configEventServiceDB2iSeriesDB コマンドに対して最低限指定する必要のあるパラメーターを以下に示します。

createDB

データベース構成スクリプトを管理コマンドで作成して実行するかどうかを指定します。true または false を指定します。このパラメーターを false に設定すると、スクリプトは作成されますが、実行されません。この場合、データベース構成を完了するにはデータベース構成スクリプトを実行する必要があります。

制限: 管理コマンドでデータベース構成スクリプトを自動的に実行できるのは、iSeries システム上に限られます。クライアント・システム上でコマンドを実行した場合は、エラーが返されます。

nodeName

イベント・サービス・データ・ソースを作成するサーバーを含むノードの名前。ノード名を指定する場合は、サーバー名も指定する必要があります。以下のいずれかを指定してください。

- ノード名およびサーバー名
- クラスター名

serverName

イベント・サービス・データ・ソースを作成するサーバーの名前。

clusterName

イベント・サービス・データ・ソースを作成するクラスターの名前。クラスター名を指定する場合は、ノード名とサーバー名を指定しないでください。

toolboxJdbcClassPath

IBM Toolbox for Java の DB2 JDBC ドライバーのパス。このパラメーターは、ネイティブ JDBC ドライバーではなく、Toolbox for Java のドライバーを使用する場合にのみ指定します。ドライバー・ファイルのパスのみを指定します。ファイル名は含めないでください。

nativeJdbcClassPath

DB2 for iSeries のネイティブ JDBC ドライバーのパス。このパラメーターは、Toolbox for Java のドライバーではなく、ネイティブ JDBC ドライバーを使用する場合にのみ指定します。ドライバー・ファイルのパスのみを指定します。ファイル名は含めないでください。

dbHostName

データベースがインストールされているサーバーのホスト名。このパラメーターは、Toolbox for Java の JDBC ドライバーを使用する場合、必須です。

dbUser

イベント・データベースの作成時に使用する DB2 ユーザー ID。指定するユーザー ID には、データベースを作成および除去するための十分な特権が必要です。

dbPassword

使用する DB2 パスワード。

環境によっては、その他のパラメーターが必要になる場合もあります。詳細なパラメーターのリストとその使用法については、 **configEventServiceDB2iSeriesDB** 管理コマンドのヘルプを参照してください。

タスクの結果

管理コマンドは、必要なデータベースとデータ・ソースを指定された有効範囲に作成するスクリプトを生成します。デフォルトでは、これらのスクリプトは、*profile_root/databases/event/node/server/dbscripts/db2iseries* ディレクトリーに保管されます。任意指定の *outputScriptDir* パラメーターに値を指定した場合は、その場所にスクリプトが保管されます。これらのスクリプトを使用すると、イベント・データベースをいつでも手動で構成できます。

次のタスク

データベース構成管理コマンドをクライアント・システム上で実行した場合は、生成されたスクリプトを iSeries システムに転送してから実行して、必要なリソースを作成する必要があります。

データベースの構成が終了した後、サーバー管理コンソールを使用してデータベース構成をテストできます。構成をテストする場合、該当する JDBC データ・ソースにナビゲートして「テスト接続」オプションを選択します。

Informix イベント・データベースの構成

Linux、UNIX、または Windows システムで、IBM Informix Dynamic Server を使用して、外部イベント・データベースを構成できます。

このタスクについて

Informix イベント・データベースを構成するには、以下の手順を実行します。

手順

1. *wsadmin* ツールを開始します。
2. *AdminTask* オブジェクトを使用して、**configEventServiceInformixDB** 管理コマンドをバッチ・モードまたは対話モードで実行します。

configEventServiceInformixDB コマンドに最低限指定する必要のあるパラメーターを以下に示します。

createDB

データベース構成スクリプトを管理コマンドで作成して実行する必要があるかどうかを示します。 *true* または *false* を指定します。このパラメーターを *false* に設定すると、スクリプトは作成されますが、実行されません。この場合、データベース構成を完了するにはデータベース構成スクリプトを実行する必要があります。

特権: このパラメーターに *true* を指定する場合は、Informix データベース、DB スペース、テーブル、ビュー、索引、およびストアド・プロシージャを作成するための十分な特権がご自分のユーザー ID にあることを確認してください。

nodeName

イベント・サービス・データ・ソースを作成する必要があるサーバーを含むノードの名前。ノード名を指定する場合は、サーバー名も指定する必要があります。以下のいずれかを指定してください。

- ノード名およびサーバー名
- クラスター名

serverName

イベント・サービス・データ・ソースを作成する必要があるサーバーの名前。

clusterName

イベント・サービス・データ・ソースを作成する必要があるクラスターの名前。クラスター名を指定する場合は、ノード名とサーバー名を指定しないでください。

jdbcClassPath

JDBC ドライバーへのパス。ドライバー・ファイルのパスのみを指定します。ファイル名は指定しないでください。

dbInformixDir

Informix データベース・ソフトウェアがインストールされているディレクトリー。このパラメーターは、`createDB` パラメーターに `true` を指定した場合にのみ必要です。

dbHostName

データベース・サーバーがインストールされているシステムのホスト名。

dbServerName

Informix サーバー名 (`o1_servername` など)。

dbUser

イベント・データベース表を所有する Informix データベース・スキーマ・ユーザー ID。これは、データベースと DB スペースを作成するための十分な特権を持つユーザー ID である必要があります。WebSphere データ・ソースは、このユーザー ID を使用して Informix データベース接続を認証します。

dbPassword

指定したスキーマ・ユーザー ID のパスワード。

環境によっては、その他のパラメーターが必要になる場合もあります。詳細なパラメーターのリストとその使用方法については、`configEventServiceInformixDB` 管理コマンドのヘルプを参照してください。

タスクの結果

管理コマンドは、必要なデータ・ソースを指定された有効範囲に作成します。`createDB` パラメーターに `true` を指定した場合、コマンドは、生成されたデータベース構成スクリプトも実行し、データベースを作成します。

デフォルトでは、生成されたデータベース構成スクリプトは、`profile_root/databases/event/node/server/dbscripts/informix` ディレクトリーに保管されます。(Network Deployment 環境では、これらのスクリプトは、デプロイメント・マ

ネージャーのプロファイル・ディレクトリーの下に保管されます。) 任意指定の `outputScriptDir` パラメーターに値を指定した場合は、その場所にスクリプトが保管されます。これらのスクリプトを使用すると、イベント・データベースをいつでも手動で構成できます。

スクリプトの実行: `configEventServiceInformixDB` 管理コマンドにより生成されるデータベース構成スクリプトおよびデータベース除去スクリプトでは、SQL スクリプトを実行するために `dbaccess` コマンドが必要です。このコマンドは、Informix サーバー上でしか使用できない可能性があります。したがって、Informix サーバーがサーバーとは別のシステム上にある場合は、データベース構成スクリプトを Informix サーバーにコピーして、ローカルで実行する必要がある可能性があります。

Oracle イベント・データベースの構成

Linux、UNIX、または Windows システム上で、Oracle データベースを使用し、外部のイベント・データベースを構成できます。

始める前に

Oracle イベント・データベースを構成する前に、まずデータベースを作成しておく必要があります。イベント・データベースの構成コマンドを実行する前に、Oracle SID が既に存在している必要があります。イベント・データベースのデフォルトの SID は、`event` です。

このタスクについて

Oracle イベント・データベースを構成するには、以下の手順を実行します。

手順

1. `wsadmin` ツールを開始します。
2. `AdminTask` オブジェクトを使用して、`configEventServiceOracleDB` 管理コマンドをバッチ・モードまたは対話モードで実行します。 `configEventServiceOracleDB` コマンドに最低限指定する必要があるパラメーターを以下に示します。

createDB

データベース構成スクリプトを管理コマンドで作成して実行する必要があるかどうかを示します。 `true` または `false` を指定します。このパラメーターを `false` に設定すると、スクリプトは作成されますが、実行されません。この場合、データベース構成を完了するにはデータベース構成スクリプトを実行する必要があります。

nodeName

イベント・サービス・データ・ソースを作成する必要があるサーバーを含むノードの名前。ノード名を指定する場合は、サーバー名も指定する必要があります。以下のいずれかを指定してください。

- ノード名およびサーバー名
- クラスタ名

serverName

イベント・サービス・データ・ソースを作成する必要があるサーバーの名前。

clusterName

イベント・サービス・データ・ソースを作成する必要があるクラスターの名前。クラスター名を指定する場合は、ノード名とサーバー名を指定しないでください。

jdbcClassPath

JDBC ドライバーへのパス。ドライバー・ファイルのパスのみを指定します。ファイル名は指定しないでください。

oracleHome

ORACLE_HOME ディレクトリー。このパラメーターは、createDB パラメーターに true を指定した場合にのみ必要です。

dbPassword

データベース構成時に作成されるスキーマ・ユーザー ID に対して使用するパスワード (デフォルトのユーザー ID は ceiuser です)。このパスワードは、Oracle データベース接続を認証するために使用されます。

sysUser

Oracle SYSUSER ユーザー ID。このユーザー ID には、SYSDBA 特権が必要です。

sysPassword

指定した SYSUSER ユーザー ID のパスワード。

環境によっては、その他のパラメーターが必要になる場合もあります。詳細なパラメーターのリストとその用法については、**configEventServiceOracleDB** 管理コマンドのヘルプを参照してください。

タスクの結果

管理コマンドは、必要なデータ・ソースを指定された有効範囲に作成します。createDB パラメーターに true を指定した場合、コマンドは、生成されたデータベース構成スクリプトも実行し、データベースを作成します。

デフォルトでは、生成されたデータベース構成スクリプトは、*profile_root/databases/event/node/server/dbscripts/oracle* ディレクトリーに保管されます。(Network Deployment 環境では、これらのスクリプトは、デプロイメント・マネージャーのプロファイル・ディレクトリーの下に保管されます。) 任意指定の outputScriptDir パラメーターに値を指定した場合は、その場所にスクリプトが保管されます。これらのスクリプトを使用すると、イベント・データベースをいつでも手動で構成できます。

SQL サーバー・イベント・データベースの構成

Windows システムで、Microsoft SQL Server Enterprise を使用して、外部イベント・データベースを構成できます。

このタスクについて

SQL サーバー・イベント・データベースを構成するには、以下の手順を実行します。

手順

1. SQL サーバー・データベース・サーバー・システムで、データベース・ファイルを格納するために使用するディレクトリーを作成します。 デフォルトでは、ファイルは `c:\program files\ibm\event\ceiinst1\sqlserver_data` ディレクトリーに書き込まれます。別の場所を指定する必要がある場合は、生成されたデータベース構成スクリプトを編集して `ceiInstancePrefix` パラメーターの値を変更し、次にスクリプトを手動で実行する必要があります。
2. サーバー・システムで、`wsadmin` ツールを開始します。
3. `AdminTask` オブジェクトを使用して、`configEventServiceSQLServerDB` 管理コマンドをバッチ・モードまたは対話モードで実行します。

configEventServiceSQLServerDB コマンドに最低限指定する必要があるパラメーターを以下に示します。

createDB

データベース構成スクリプトを管理コマンドで作成して実行する必要があるかどうかを示します。 `true` または `false` を指定します。このパラメーターを `false` に設定すると、スクリプトは作成されますが、実行されません。この場合、データベース構成を完了するにはデータベース構成スクリプトを実行する必要があります。

nodeName

イベント・サービス・データ・ソースを作成する必要があるサーバーを含むノードの名前。ノード名を指定する場合は、サーバー名も指定する必要があります。以下のいずれかを指定してください。

- ノード名およびサーバー名
- クラスター名

serverName

イベント・サービス・データ・ソースを作成する必要があるサーバーの名前。サーバー名を指定する場合は、ノード名も指定する必要があります。

clusterName

イベント・サービス・データ・ソースを作成する必要があるクラスターの名前。クラスター名を指定する場合は、ノード名とサーバー名を指定しないでください。

dbServerName

SQL サーバー・データベースのサーバー名。このパラメーターは、`createDB` パラメーターに `true` を指定した場合にのみ必要です。

dbHostName

SQL サーバー・データベースが稼働しているサーバーのホスト名。

dbPassword

イベント・データベース表を所有するために作成されたユーザー ID に対して使用するパスワード (デフォルトのユーザー ID は `ceiuser` です)。
WebSphere データ・ソースは、このパスワードを使用して SQL サーバー・データベース接続を認証します。

saUser

データベースとユーザーを作成および除去する特権を持つユーザー ID。このパラメーターは、createDB パラメーターに true を指定した場合にのみ必要です。

saPassword

指定した SA ユーザーのパスワード。

環境によっては、その他のパラメーターが必要になる場合もあります。詳細なパラメーターのリストとその使用方法については、**configEventServiceSQLServerDB** 管理コマンドのヘルプを参照してください。

タスクの結果

管理コマンドは、必要なデータ・ソースを指定された有効範囲に作成します。createDB パラメーターに true を指定した場合、コマンドは、生成されたデータベース構成スクリプトも実行し、データベースを作成します。

デフォルトでは、生成されたデータベース構成スクリプトは、*profile_root/databases/event/nodeserver/dbscripts/dbscripts/sqlserver* ディレクトリーに保管されます。(Network Deployment 環境では、これらのスクリプトは、デプロイメント・マネージャーのプロファイル・ディレクトリーの下に保管されます。) 任意指定の outputScriptDir パラメーターに値を指定した場合は、その場所にスクリプトが保管されます。これらのスクリプトを使用すると、イベント・データベースをいつでも手動で構成できます。

データベース構成スクリプトの手動実行

データベース構成管理コマンドで生成されたスクリプトは、いつでも手動で実行できます。

このタスクについて

データベース構成には 2 つのステップがあります。まず、データベース構成管理コマンドが、環境に対応したデータベース固有のスクリプトを生成します。次に、この生成されたスクリプトが、イベント・データベースとデータ・ソースを構成します。管理コマンドの実行時に createDB パラメーターに true を指定した場合は、両方のステップが自動的に実行されます。

ただし、createDB パラメーターに false を指定した場合は、生成されたスクリプトをターゲット・システム上で手動で実行して、データベース構成を完了する必要があります。以下の状況の場合は、スクリプトを手動で実行する必要がある可能性があります。

- 管理コマンドを実行したシステムとは異なるシステムでイベント・データベースを構成する必要がある。
- イベント・データベースを後で再作成する必要がある。
- 生成されたスクリプトを実行する前に、そのスクリプトが使用するデフォルト・オプションを変更する必要がある。

Derby イベント・データベースの手動による作成

Derby イベント・データベース用のデータベース構成スクリプトを手動で生成するには、`cr_event_derby` コマンドを使用します。

このタスクについて

Derby イベント・データベース用に生成されたデータベース構成スクリプトを手動で実行するには、以下の手順を実行します。

手順

1. サーバー・システムで、生成されたスクリプトを含むディレクトリーに移動します。デフォルトの場所は、`install_root/profiles/profile_name/dbscripts/CEI_ceiDbName` ディレクトリーです。データベース構成管理コマンドの `outputScriptDir` パラメーターに値を指定した場合は、その場所にスクリプトが保管されています。
2. 必要に応じ、ASCII テキスト・エディターを使用して構成スクリプトを変更します。スクリプトの名前は、ご使用のオペレーティング・システムに応じて異なります。

- **Windows** `cr_event_derby.bat`

- **Linux** **UNIX** `cr_event_derby.sh`

- **i5/OS プラットフォームの場合:** `cr_event_derby`

3. オプション: iSeries システム上でデータベースを構成する場合は、Qshell インタープリターを始動します。
4. 以下の構文を使用して、データベース作成スクリプトを実行します (該当する場合は、ファイル拡張子を指定することを忘れないでください)。

```
cr_event_derby -p profile_path [-s server_name|-c cluster_name]
```

パラメーターは、以下のとおりです。

-p profile_path

WebSphere プロファイル・ディレクトリーへのパス。このパラメーターは必須です。

-s server_name

サーバーの名前。このパラメーターは、サーバーの有効範囲でデータベースを構成する場合、必須です。

-c cluster_name

クラスターの名前。このパラメーターは、クラスターの有効範囲でデータベースを構成する場合、必須です。

例えば、以下のコマンドは、`profile1` プロファイルを使用して、`server1` サーバーの有効範囲に Derby データベースを作成します。

```
cr_event_derby -p c:%WebSphere%appserver%profiles%myprofile -s server1
```

5. サーバーを再始動します。統合ノードの場合は、**stopNode** および **startNode** コマンドを使用して、ノード・エージェントを停止してから再始動する必要があります。

次のタスク

データベースの構成が完了したら、管理コンソールを使用してデータベース構成をテストできます。これを行うには、該当する JDBC データ・ソースにナビゲートし、「テスト接続」オプションを選択します。

Linux、UNIX、または Windows システム上での手動による DB2 イベント・データベースの作成

Linux、UNIX、または Windows サーバー上で DB2 イベント・データベース用のデータベース構成スクリプトを手動で生成するには、`cr_event_db2` を使用します。

このタスクについて

Linux、UNIX、または Windows システムの DB2 イベント・データベース用に生成されたデータベース構成スクリプトを手動で実行するには、以下の手順を実行します。

手順

1. サーバー・システムで、生成されたスクリプトを含むディレクトリに移動します。デフォルトの場所は、`install_root/profiles/profile_name/dbscripts/CEI_ceiDbName` ディレクトリです。データベース構成管理コマンドの `outputScriptDir` パラメーターに値を指定した場合は、その場所にスクリプトが保管されています。
2. 必要に応じ、ASCII テキスト・エディターを使用して構成スクリプトを変更します。スクリプトの名前は、ご使用のオペレーティング・システムに応じて異なります。
 - **Windows** Windows システムの場合: `cr_event_db2.bat`
 - **Linux** **UNIX** Linux および UNIX システムの場合: `cr_event_db2.sh`
3. 以下の構文を使用して、データベース作成スクリプトを実行します (該当する場合は、ファイル拡張子を指定することを忘れないでください)。

```
cr_event_db2 [client|server] db_user [db_password]
```

パラメーターは、以下のとおりです。

client|server

データベースがクライアントとサーバーのいずれであるかを示します。

client または **server** を指定する必要があります。

db_user

データベース・ユーザー ID。このパラメーターは必須です。

db_password

データベース・ユーザーのパスワード。クライアント・データベースのパスワードを指定しない場合は、パスワードの入力プロンプトが表示されます。

例えば、以下のコマンドは、ユーザー ID `db2admin` とパスワード `mypassword` を使用して、クライアント・データベース用の DB2 イベント・データベースを作成します。

```
cr_event_db2 client db2admin mypassword
```

4. サーバーを再始動します。統合ノードの場合は、**stopNode** および **startNode** コマンドを使用して、ノード・エージェントを停止してから再始動する必要があります。

次のタスク

データベースの構成が完了したら、管理コンソールを使用してデータベース構成をテストできます。これを行うには、該当する JDBC データ・ソースにナビゲートし、「テスト接続」オプションを選択します。

z/OS システム上での手動による DB2 イベント・データベースの作成

Linux、UNIX、または Windows クライアント・システムを使用して、z/OS システム上で DB2 イベント・データベース用のデータベース構成スクリプトを手動で生成するには、`cr_event_db2zos` を使用します。

このタスクについて

Linux、UNIX、または Windows クライアント・システムを使用して、z/OS システムの DB2 イベント・データベース用に生成されたデータベース構成スクリプトを手動で実行するには、以下の手順を実行します。

手順

1. サーバー・システムで、生成されたスクリプトを含むディレクトリーに移動します。デフォルトの場所は `install_root/profiles/profile_name/dbscripts/CEI_ceiDbName` です。データベース構成管理コマンドの `outputScriptDir` パラメーターに値を指定した場合は、その場所にスクリプトが保管されています。
2. 必要に応じ、ASCII テキスト・エディターを使用して構成スクリプトを変更します。スクリプトの名前は、ご使用のオペレーティング・システムに応じて異なります。
 - Windows システムの場合: `cr_event_db2zos.bat`
 - Linux および UNIX システムの場合: `cr_event_db2zos.sh`
3. 以下の構文を使用して、データベース作成スクリプトを実行します (該当する場合は、ファイル拡張子を指定することを忘れないでください)。

```
cr_event_db2zos [dbName=db_name] db_user [db_password]
```

パラメーターは、以下のとおりです。

db_name

使用するデータベース名。このパラメーターは任意指定です。データベース名を指定しない場合は、名前が生成されます。

db_user

使用するためのデータベース・ユーザー ID。このパラメーターは必須です。

db_password

データベース・ユーザーのパスワード。パスワードを指定しない場合、DB2 データベースは、パスワードの入力プロンプトを表示します。

例えば、以下のコマンドは、ユーザー ID db2admin とパスワード mypassword を使用して、event という名前の DB2 イベント・データベースを作成します。

```
cr_event_db2zos dbName=client db2admin mypassword
```

4. サーバーを再始動します。統合ノードの場合は、**stopNode** および **startNode** コマンドを使用して、ノード・エージェントを停止してから再始動する必要があります。

次のタスク

データベースの構成が完了したら、管理コンソールを使用してデータベース構成をテストできます。これを行うには、該当する JDBC データ・ソースにナビゲートし、「テスト接続」オプションを選択します。

iSeries システム上での手動による DB2 イベント・データベースの作成

iSeries システム上で DB2 イベント・データベース用のデータベース構成スクリプトを手動で生成するには、cr_event_db2iseries コマンドを使用します。

このタスクについて

iSeries システム上の DB2 イベント・データベース用に生成されたデータベース構成スクリプトを手動で実行するには、以下の手順を実行します。

手順

1. サーバー・システムで、生成されたスクリプトを含むディレクトリーに移動します。デフォルトの場所は `install_root/profiles/profile_name/dbscripts/CEI_ceiDbName` です。データベース構成管理コマンドの `outputScriptDir` パラメーターに値を指定した場合は、その場所にスクリプトが保管されています。
2. 必要に応じ、ASCII テキスト・エディターを使用して `cr_event_db2iseries` スクリプトを変更します。
3. Qshell インタープリターを開始します。
4. 以下の構文を使用して、データベース作成スクリプトを実行します。

```
cr_event_db2iseries db_user db_password
```

パラメーターは、以下のとおりです。

db_user

データベース・ユーザー ID。このパラメーターは必須です。

db_password

データベース・ユーザーのパスワード。このパラメーターは必須です。

例えば、以下のコマンドは、ユーザー ID db2admin とパスワード mypassword を使用して、DB2 イベント・データベースを作成します。

```
cr_event_db2iseries db2admin mypassword
```

5. サーバーを再始動します。統合ノードの場合は、**stopNode** および **startNode** コマンドを使用して、ノード・エージェントを停止してから再始動する必要があります。

次のタスク

データベースの構成が完了したら、管理コンソールを使用してデータベース構成をテストできます。これを行うには、該当する JDBC データ・ソースにナビゲートし、「テスト接続」オプションを選択します。

Informix イベント・データベースの手動による作成

データベース構成管理コマンドで生成されたスクリプトは、いつでも手動で実行できます。

このタスクについて

Informix イベント・データベース用に生成されたデータベース構成スクリプトを手動で実行するには、以下の手順を実行します。

手順

1. サーバー・システムで、生成されたスクリプトを含むディレクトリーに移動します。デフォルトの場所は `install_root/profiles/profile_name/dbscripts/CEI_ceiDbName` です。データベース構成管理コマンドの `outputScriptDir` パラメーターに値を指定した場合は、その場所にスクリプトが保管されています。
2. 必要に応じ、ASCII テキスト・エディターを使用して構成スクリプトを変更します。スクリプトの名前は、ご使用のオペレーティング・システムに応じて異なります。
 - **Windows** Windows システムの場合: `cr_event_informix.bat`
 - **Linux** **UNIX** Linux および UNIX システムの場合:
`cr_event_informix.sh`
3. パラメーターを指定せずに、データベース作成スクリプトを実行します。
4. サーバーを再始動します。統合ノードの場合は、`stopNode` および `startNode` コマンドを使用して、ノード・エージェントを停止してから再始動する必要があります。

次のタスク

データベースの構成が完了したら、管理コンソールを使用してデータベース構成をテストできます。これを行うには、該当する JDBC データ・ソースにナビゲートし、「テスト接続」オプションを選択します。

Oracle イベント・データベースの手動による作成

Oracle イベント・データベース用のデータベース構成スクリプトを手動で生成するには、`cr_event_oracle` コマンドを使用します。

このタスクについて

Oracle イベント・データベース用に生成されたデータベース構成スクリプトを手動で実行するには、以下の手順を実行します。

手順

1. サーバー・システムで、生成されたスクリプトを含むディレクトリーに移動します。デフォルトの場所は `install_root/profiles/profile_name/dbscripts/CEI_ceiDbName` です。データベース構成管理コマンドの `outputScriptDir` パラメーターに値を指定した場合は、その場所にスクリプトが保管されています。
2. 必要に応じ、ASCII テキスト・エディターを使用して構成スクリプトを変更します。スクリプトの名前は、ご使用のオペレーティング・システムに応じて異なります。

- **Windows** Windows システムの場合: `cr_event_oracle.bat`

- **Linux** **UNIX** Linux および UNIX システムの場合: `cr_event_oracle.sh`

3. 以下の構文を使用して、データベース作成スクリプトを実行します (該当する場合は、ファイル拡張子を指定することを忘れないでください)。

```
cr_event_oracle password sys_user
                sys_password [sid=sid]
                [oracleHome=oracle_home]
```

パラメーターは、以下のとおりです。

password

スキーマ・ユーザー ID のパスワード。このパラメーターは必須です。

sys_user

Oracle データベースの SYSDBA 特権を持つユーザー ID (通常は、システム・ユーザー)。このパラメーターは必須です。

sys_password

指定したシステム・ユーザー ID のパスワード。このユーザー ID でパスワードを使用しない場合は、`none` を入力します。

sid=sid

Oracle システム ID (SID)。このパラメーターはオプションです。

oracleHome=oracle_home

Oracle ホーム・ディレクトリー。このパラメーターは任意指定です。値を指定しない場合は、生成されたパスが使用されます。

例えば、以下のコマンドは、スキーマ・ユーザー ID `auser` とシステム・ユーザー ID `sys` を使用して、Oracle イベント・データベースを作成します。

```
cr_event_oracle auser sys syspassword sid=event oracleHome=c:\oracle
```

4. サーバーを再始動します。統合ノードの場合は、`stopNode` および `startNode` コマンドを使用して、ノード・エージェントを停止してから再始動する必要があります。

次のタスク

データベースの構成が完了したら、管理コンソールを使用してデータベース構成をテストできます。これを行うには、該当する JDBC データ・ソースにナビゲートし、「**テスト接続**」オプションを選択します。

SQL サーバー・イベント・データベースの手動による作成

SQL サーバー・イベント・データベース用のデータベース構成スクリプトを手動で生成するには、`cr_event_mssql` コマンドを使用します。

このタスクについて

SQL サーバー・イベント・データベース用に生成されたデータベース構成スクリプトを手動で実行するには、以下の手順を実行します。

手順

1. サーバー・システムで、生成されたスクリプトを含むディレクトリに移動します。デフォルトの場所は `install_root/profiles/profile_name/dbscripts/CEI_ceiDbName` です。データベース構成管理コマンドの `outputScriptDir` パラメーターに値を指定した場合は、その場所にスクリプトが保管されています。
2. 必要に応じ、ASCII テキスト・エディターを使用して `cr_event_mssql.bat` スクリプトを変更します。
3. 以下の構文を使用して、データベース作成スクリプトを実行します。

```
cr_event_mssql user_id password [server=server] sauser=sa_user  
sapassword=sapassword
```

パラメーターは、以下のとおりです。

user_id

作成されたテーブルを所有する SQL サーバー・ログイン・ユーザー ID。このユーザー ID は、データベースへの JDBC 接続ができるようにするために SQL サーバー内で作成する必要があります。(JDBC ドライバーは、トラステッド接続をサポートしません。)

password

作成する新しいログイン・ユーザー ID のパスワード。

server=server

SQL サーバー・データベースを含むサーバーの名前。このパラメーターは任意指定です。デフォルト値は、ローカル・ホストです。

sauser=sa_user

sa ユーザー ID。このユーザー ID には、データベースおよびユーザー・ログインを作成するための十分な特権が必要です。

sapassword=sapassword

sa パスワード (混在認証モードを使用する場合)。sa ユーザー ID にパスワードが設定されていない場合は、`sapassword=` に値を指定しないでください。トラステッド接続を使用する場合は、このパラメーターを省略します。

例えば、以下のコマンドは、ログイン・ユーザー ID `userid` を使用して、SQL サーバー・イベント・データベースを作成します。

```
cr_event_mssql userid apassword server=myserver sauser=sa sapassword=sapassword
```

4. サーバーを再始動します。統合ノードの場合は、`stopNode` および `startNode` コマンドを使用して、ノード・エージェントを停止してから再始動する必要があります。

次のタスク

データベースの構成が完了したら、管理コンソールを使用してデータベース構成をテストできます。これを行うには、該当する JDBC データ・ソースにナビゲートし、「テスト接続」オプションを選択します。

以前のバージョンからのイベント・データベースのアップグレード

以前のバージョンの Common Event Infrastructure からマイグレーションした場合、イベント永続性を使用するには、既存のイベント・データベースのアップグレードが必要になる場合があります。

このタスクについて

イベント・データベースのアップグレードが必要になるのは、Common Event Infrastructure バージョン 5.1 以前からマイグレーションする場合です。

データベースのアップグレード・プロセスでは、現在のイベント・データを保持しながら、既存のイベント・データベースのスキーマとメタデータが現行バージョンにアップグレードされます。

データベースのアップグレード・スクリプトでは、既存のイベント・データベースのスキーマとメタデータが現行バージョンにアップグレードされます。

サポートされないバージョン: Common Event Infrastructure 6.0 でサポートされなくなったバージョンのデータベース・ソフトウェアをイベント・データベースが使用している場合は、まず、そのデータベース・ソフトウェア用の適切な手順を使用して、データベースをサポートされるバージョンへとマイグレーションする必要があります。次に、イベント・データベースのアップグレード・プロセスに従って、データベースをアップグレードします。

以前のバージョンからの DB2 イベント・データベースのアップグレード

Linux、UNIX、または Windows システムに Common Event Infrastructure バージョン 5.1 の既存の DB2 イベント・データベースがある場合は、それを現行バージョンにアップグレードする必要があります。

このタスクについて

Linux または UNIX システムで DB2 イベント・データベースをアップグレードするには、以下の手順を実行します。

手順

1. 既存のイベント・データベースのバックアップ・コピーを作成します。
2. `profile_root/bin` ディレクトリに移動します。
3. 次のように、ご使用のオペレーティング・システム用の DB2 アップグレード・スクリプトを実行します。

- **Windows** Windows システムの場合:

```
eventUpgradeDB2 runUpgrade=[true|false] dbUser=user
[dbName=name] [dbPassword=pw]
[dbNode=node] [scriptDir=dir]
```

- **Linux** **UNIX** Linux および UNIX システムの場合:

```
eventUpgradeDB2.sh runUpgrade=[true|false] dbUser=user
[dbName=name] [dbPassword=pw]
[dbNode=node] [scriptDir=dir]
```

通常、必要なパラメーターは以下のとおりです。

runUpgrade

生成された DDL スクリプトをアップグレード・スクリプトで自動的に実行してデータベース・アップグレードを完了するかどうかを指定します。このパラメーターは必須です。データベース・アップグレードを後で、または別のシステム上で手動で実行する場合は、false を指定します。

dbUser

使用する DB2 ユーザー ID を指定します。このパラメーターは必須です。

dbName

DB2 データベース名を指定します。デフォルトのイベント・データベース名は event です。このパラメーターは、runUpgrade=true を指定した場合、必須です。

dbPassword

指定した DB2 ユーザー ID のパスワードを指定します。このパラメーターは任意指定です。パスワードを指定しない場合、DB2 は、パスワードの入力プロンプトを表示します。

dbNode

データベース・ノード名を指定します。このパラメーターは、DB2 クライアント・システムからアップグレード・スクリプトを実行する場合、必須です。

scriptDir

生成された DDL スクリプトを含めるディレクトリーを指定します。このパラメーターは任意指定です。ディレクトリーを指定しない場合は、`.$eventDBUpgrade¥db2` ディレクトリーにスクリプトが保管されます。

詳細なパラメーターのリストとその使用方法を確認するには、パラメーターを指定せずに **eventUpgradeDB2** スクリプトを実行します。

タスクの結果

アップグレード・スクリプトは、イベント・データベースをアップグレードするために必要な DDL スクリプトを生成します。runUpgrade=true を指定した場合は、DDL スクリプトが自動的に実行され、アップグレードが完了します。

例

Windows システムの既存の DB2 データベースをアップグレードする例を以下に示します。

```
eventUpgradeDB2 runUpgrade=true dbUser=db2inst1 dbName=event
```

次のタスク

`runUpgrade=false` を指定した場合は、データベース・システム上で DDL スクリプトを手動で実行して、データベース・アップグレードを完了する必要があります。

以前のバージョンからの DB2 for z/OS イベント・データベースのアップグレード

z/OS システムに Common Event Infrastructure バージョン 5.1 の既存の DB2 イベント・データベースがある場合は、それを現行バージョンにアップグレードする必要があります。

このタスクについて

z/OS システムの DB2 イベント・データベースをアップグレードするには、以下の手順を実行します。

手順

1. 既存のイベント・データベースのバックアップ・コピーを作成します。
2. `profile_root/bin` ディレクトリに移動します。
3. ご使用のクライアント・オペレーティング・システム用の DB2 for z/OS アップグレード・スクリプトを以下のように実行します。

- **Windows** Windows システムの場合:

```
eventUpgradeDB2ZOS runUpgrade=[true|false] dbUser=user  
[dbName=name] [dbPassword=pw]  
[scriptDir=dir] storageGroup=group  
bufferPool14K=4kbufpool bufferPool18k=8kbufpool  
bufferPool16K=16kbufpool
```

- **Linux** **UNIX** Linux および UNIX システムの場合:

```
eventUpgradeDB2ZOS.sh runUpgrade=[true|false] dbUser=user  
[dbName=name] [dbPassword=pw]  
[scriptDir=dir] storageGroup=group  
bufferPool14K=4kbufpool bufferPool18k=8kbufpool  
bufferPool16K=16kbufpool
```

通常、必要なパラメーターは以下のとおりです。

runUpgrade

生成された DDL スクリプトをアップグレード・スクリプトで自動的に実行してデータベース・アップグレードを完了するかどうかを指定します。このパラメーターは必須です。データベースを後で、または別のシステム上で手動でアップグレードする場合は、`false` を指定します。

z/OS システム: このパラメーターは、ネイティブ z/OS システムでは無視されます。生成された DDL スクリプトの自動実行は、クライアント・システムでのみサポートされます。

dbUser

使用する DB2 ユーザー ID を指定します。このパラメーターは必須です。

dbName

DB2 データベース名を指定します。デフォルトのイベント・データベース名は `event` です。このパラメーターは、`runUpgrade=true` を指定した場合、必須です。

dbPassword

指定した DB2 ユーザー ID のパスワードを指定します。このパラメーターは任意指定です。パスワードを指定しない場合、DB2 は、パスワードの入力プロンプトを表示します。

scriptDir

生成された DDL スクリプトを含めるディレクトリーを指定します。このパラメーターは任意指定です。ディレクトリーを指定しない場合は、`¥eventDBUpgrade¥db2zos` ディレクトリーにスクリプトが保管されます。

storageGroup

ストレージ・グループの名前を指定します。このパラメーターは必須です。

bufferPool4K

4K バッファ・プールの名前を指定します。このパラメーターは必須です。

bufferPool8K

8K バッファ・プールの名前を指定します。このパラメーターは必須です。

bufferPool16K

16K バッファ・プールの名前を指定します。このパラメーターは必須です。

詳細なパラメーターのリストとその使用方法を確認するには、パラメーターを指定せずに `eventUpgradeDB2ZOS` スクリプトを実行します。

タスクの結果

アップグレード・スクリプトは、イベント・データベースをアップグレードするために必要な DDL スクリプトを生成します。クライアント・システムで `runUpgrade=true` を指定した場合は、DDL スクリプトが自動的に実行され、アップグレードが完了します。

例

Windows クライアント・システムから DB2 for z/OS イベント・データベースをアップグレードする例を以下に示します。

```
eventUpgradeDB2ZOS runUpgrade=true dbUser=db2inst1 dbName=event
storageGroup=sysdef1t bufferPool4K=BP9 bufferPool8K=BP8K9 bufferPool16K=BP16K9
```

次のタスク

`runUpgrade=false` を指定した場合、または z/OS システム上でアップグレード・スクリプトを実行した場合は、生成された DDL スクリプトを、SQL Processor Using File Input (SPUFI) 機能を使用して z/OS システムで手動で実行する必要があります。この手順によって、データベース・アップグレードが完了します。

Oracle イベント・データベースのバージョン 5 からのアップグレード

Common Event Infrastructure バージョン 5.1 の既存の Oracle イベント・データベースを使用している場合は、それを現行バージョンにアップグレードする必要があります。

このタスクについて

Oracle イベント・データベースをアップグレードするには、以下の手順を実行します。

手順

1. 既存のイベント・データベースのバックアップ・コピーを作成します。
2. `profile_root/bin` ディレクトリに移動します。
3. 次のように、ご使用のオペレーティング・システム用の Oracle アップグレード・スクリプトを実行します。

- Windows システムの場合:

```
eventUpgradeOracle runUpgrade=[true|false] schemaUser=schemauser  
[oracleHome=dir] [dbName=name]  
[dbUser=sysuser] [dbPassword=pw]  
[scriptDir=dir]
```

- Linux および UNIX システムの場合:

```
eventUpgradeOracle.sh runUpgrade=[true|false] schemaUser=schemauser  
[oracleHome=dir] [dbName=name]  
[dbUser=sysuser] [dbPassword=pw]  
[scriptDir=dir]
```

通常、必要なパラメーターは以下のとおりです。

runUpgrade

生成された DDL スクリプトをアップグレード・スクリプトで自動的に実行してデータベース・アップグレードを完了するかどうかを指定します。このパラメーターは必須です。データベースを後で、または別のシステム上で手動でアップグレードする場合は、`false` を指定します。

schemaUser

データベース表を所有する Oracle ユーザー ID を指定します。このパラメーターは必須です。

oracleHome

Oracle ホーム・ディレクトリを指定します。このパラメーターは、`runUpgrade=true` を指定した場合、必須です。

dbName

Oracle データベース名を指定します。デフォルトのイベント・データベース名は `event` です。このパラメーターは、`runUpgrade=true` を指定した場合、必須です。

dbUser

Oracle システム・ユーザー ID を指定します。このパラメーターは、`runUpgrade=true` を指定した場合、必須です。

dbPassword

システム・ユーザー ID のパスワードを指定します。システム・ユーザー ID にパスワードがない場合は、このパラメーターを指定しないでください。

scriptDir

生成された DDL スクリプトを含めるディレクトリーを指定します。このパラメーターは任意指定です。ディレクトリーを指定しない場合は、`¥eventDBUpgrade¥oracle` ディレクトリーにスクリプトが保管されます。

詳細なパラメーターのリストとその使用方法を確認するには、パラメーターを指定せずに `eventUpgradeOracle` スクリプトを実行します。

タスクの結果

アップグレード・スクリプトは、イベント・データベースをアップグレードするために必要な DDL スクリプトを生成します。 `runUpgrade=true` を指定した場合は、DDL スクリプトが自動的に実行され、アップグレードが完了します。

例

Windows システムの既存の Oracle データベースをアップグレードする例を以下に示します。

```
eventUpgradeOracle runUpgrade=true schemaUser=cei
dbName=event dbUser=sys
```

次のタスク

`runUpgrade=false` を指定した場合は、データベース・システム上で DDL スクリプトを手動で実行して、データベース・アップグレードを完了する必要があります。

WebSphere Business Monitor 用のクロスセル Common Event Infrastructure 構成

Common Event Infrastructure (CEI) イベントを生成するリモート・サーバーと、WebSphere Business Monitor サーバーの間に、接続を構成する必要があります。

このタスクについて

マルチサーバー環境で複数のセルを横断した CEI の構成方法について詳しくは、IBM WebSphere Business Monitor インフォメーション・センターのトピック『WebSphere Business Monitor を使用するようにリモート CEI サーバーを構成』を参照してください。

Common Event Infrastructure 構成の除去

サーバーをアンインストールする準備として Common Event Infrastructure 構成の除去が必要な場合は、まず最初に、デプロイされたエンタープライズ・アプリケーションおよびデータベース構成を除去しなければなりません。

このタスクについて

Common Event Infrastructure はサーバーの基本インストールにインストールされますが、まだ構成されていない場合はアクティブにはなりません。このトピックでは、以前に構成された Common Event Infrastructure のインスタンスを除去する方法のみについて説明します。Common Event Infrastructure の構成を除去するには、以下のステップを実行します。

イベント・データベースの除去

イベント・データベースを除去するには、データベースのタイプに合った適切な管理コマンドを使用します。

このタスクについて

イベント・データベースを除去するには、以下の手順を実行します。

手順

1. wsadmin ツールを開始します。
2. AdminTask オブジェクトを使用して、イベント・データベースに対して適切な管理コマンドを次のように実行します。

データベース・タイプ	コマンド
Derby	122 ページの『removeEventServiceDerbyDB コマンド』
DB2 (Linux、UNIX、および Windows システムの場合)	117 ページの『removeEventServiceDB2DB コマンド』
DB2 (z/OS システムの場合)	120 ページの『removeEventServiceDB2ZOSDB コマンド』
DB2 (iSeries システムの場合)	119 ページの『removeEventServiceDB2iSeriesDB コマンド』
Informix	124 ページの『removeEventServiceInformixDB コマンド』
Oracle	125 ページの『removeEventServiceOracleDB コマンド』
SQL サーバー	127 ページの『removeEventServiceSQLServerDB コマンド』

データベース固有の注意点:

- z/OS システムの場合、管理コマンドでは、JDBC データ・ソースのみが除去されます。データベースを除去するには、SPUFI を使用して、データベース作成時に生成されたデータベース除去スクリプトを実行する必要があります。デフォルトでは、このスクリプトは、*profile_root/databases/event/node/server/dbscripts/db2zos* ディレクトリーに置かれています。

- iSeries システムの場合、管理コマンドでは、JDBC データ・ソースのみが除去されます。iSeries システム上のデータベースを除去するには、そのデータベース用に作成されたコレクションを除去します。

必須パラメーター (ユーザー ID、パスワードなど) は、データベース・タイプごとに異なります。詳細なパラメーターのリストとその使用法については、管理コマンドのヘルプを参照してください。

Common Event Infrastructure アプリケーションの除去

イベント・サービス・エンタープライズ・アプリケーションおよびリソースをサーバーから手動で除去するには、**removeEventService** 管理コマンドを使用します。

このタスクについて

イベント・サービス・エンタープライズ・アプリケーションを除去するには、以下の手順を実行します。

手順

1. wsadmin ツールを開始します。
2. AdminTask オブジェクトを使用して、**removeEventService** 管理コマンドをバッチ・モードまたは対話モードで実行します。**removeEventService** 管理コマンドのパラメーターを以下に示します。

nodeName

イベント・サービスがデプロイされているノードの名前。このパラメーターは任意指定です。ノード名を指定しない場合、デフォルト値は現在のノードです。ノード名を指定する場合は、**serverName** パラメーターを使用してサーバー名も指定する必要があります。このパラメーターは、イベント・サービスをクラスターから除去する場合は無効です。

serverName

イベント・サービスがデプロイされているサーバーの名前。このパラメーターは、ノードを指定する場合にのみ必要です。イベント・サービスをクラスターから除去する場合は、このパラメーターは無効です。

clusterName

イベント・サービスがデプロイされているクラスターの名前。このパラメーターは任意指定です。イベント・サービスをサーバーから除去する場合は指定しないでください。

Common Event Infrastructure サーバーからのイベント・メッセージングの除去

外部 JMS プロバイダーのイベント・サービスのメッセージング構成を除去するには、**removeEventServiceMdb** 管理コマンドを使用します。

このタスクについて

このコマンドは、JMS 構成用にデプロイされたメッセージ駆動型 Bean を除去します。イベント・サービスのメッセージ駆動型 Bean を除去するには、以下の手順を実行します。

手順

1. wsadmin ツールを開始します。
2. AdminTask オブジェクトを使用して、**removeEventServiceMdb** 管理コマンドをバッチ・モードまたは対話モードで実行します。**removeEventServiceMdb** 管理コマンドのパラメーターは、以下のとおりです。

applicationName

デPLOYされたイベント・サービスのメッセージ駆動型 Bean のアプリケーション名。

nodeName

イベント・サービスのメッセージ駆動型 Bean がデPLOYされているノードの名前。ノード名を指定する場合は、サーバー名も指定する必要があります。ノード名はオプション・パラメーターで、デフォルト値は現在のノードです。クラスターからアプリケーションを除去する場合は、このパラメーターを指定しないでください。

serverName

イベント・サービスのメッセージ駆動型 Bean がデPLOYされているサーバーの名前。このパラメーターは、サーバーからアプリケーションを除去する場合に必要です。それ以外の場合は任意指定です。クラスターからアプリケーションを除去する場合は、サーバー名を指定しないでください。

clusterName

イベント・サービスのメッセージ駆動型 Bean がデPLOYされているクラスターの名前。このパラメーターは、クラスターからアプリケーションを除去する場合にのみ指定します。

第 3 章 Common Event Infrastructure の管理

以下のトピックでは、Common Event Infrastructure コンポーネントの実行時の操作を制御するために使用できる管理タスクをいくつか説明します。

管理コンソールを使用したイベント・サービスの管理

Web ベースの管理コンソールを使用して、イベント・サービスを管理できます。

管理コンソールによるイベント・サービスの使用可能と使用不可の切り替え

イベント・サービスを使用可能および使用不可にするには、サーバー管理コンソールでイベント・サービスのプロパティを変更します。

このタスクについて

イベント・サービスが使用可能な場合は、サーバーが始動したときに、イベント・サービスが自動的に始動します。

管理コンソールからイベント・サービスを使用可能または使用不可にするには、以下の手順を実行します。

手順

1. 次のいずれかの方法を選択して、該当する管理コンソール・パネルへ進みます。
 - 管理コンソールの「Common Event Infrastructure サーバー」パネルを開きます。
 - サーバーの場合は、「サーバー」 → 「サーバー・タイプ」 → 「WebSphere Application Server」 → *server_name* → 「ビジネス・インテグレーション」 → 「Common Event Infrastructure」 → 「Common Event Infrastructure サーバー」を選択します。
 - クラスターの場合は、「サーバー」 → 「クラスター」 → 「WebSphere Application Server クラスター」 → *cluster_name* → 「ビジネス・インテグレーション」 → 「Common Event Infrastructure」 → 「Common Event Infrastructure サーバー」を選択します。
 - 別の方法として、「コンテナ・サービス」を開いてこのタスクを行うこともできます。
 - サーバーの場合は、「サーバー」 → 「サーバー・タイプ」 → 「WebSphere Application Server」 → *server_name* → 「コンテナ・サービス」 → 「Common Event Infrastructure サービス」をクリックします。
 - クラスターの場合は、「サーバー」 → 「クラスター」 → 「WebSphere Application Server クラスター」 → *cluster_name* → 「クラスター・メンバー」 → *server* → 「コンテナ・サービス」 → 「Common Event Infrastructure サービス」をクリックします。

2. 「サーバー起動時にサービスを使用可能にする」プロパティを選択または選択解除します。このチェック・ボックスを選択すると、サーバーが始動したときに Common Event Infrastructure サービスが始動します。
3. 構成変更を保管します。
4. Network Deployment 環境の場合のみ、ノードを同期させます。
5. サーバーまたはクラスターを再始動します。

管理コンソールを使用したイベント・エミッター・ファクトリーの作成

エミッター・ファクトリーは、エミッターを作成するためにイベント・ソースで使用されます。

このタスクについて

エミッター・ファクトリーのプロパティは、そのエミッター・ファクトリーを使用して作成したすべてのエミッターの動作に影響を与えます。デフォルトのエミッター・ファクトリーを使用することも、ユーザーのイベント・ソースで使用するエミッター・ファクトリーを別途作成することもできます。エミッター・ファクトリーを追加で作成すれば、別のトランザクション・モードまたはイベント伝送を指定することもできます。「Common Event Infrastructure の宛先」パネルで Common Event Infrastructure (CEI) を構成した後に作成されたイベント・エミッターを表示するには、以下の手順に従います。

1.
 - 単一サーバーの場合は、「サーバー」>「サーバー・タイプ」>「WebSphere Application Server」> *server_name* を選択します。
 - クラスターの場合は、「サーバー」>「クラスター」>「WebSphere Application Server クラスター」 *cluster_name* を選択します。
2. 「構成」タブで、「ビジネス・インテグレーション」>「Common Event Infrastructure」>「Common Event Infrastructure の宛先」を選択します。
3. 既存のイベント・エミッター・ファクトリーを JNDI 名のメニューから選択するか、テキスト・フィールドで指定できます。

CEI サーバーがローカルでない場合は、JNDI 名をリモート・サーバーへ解決する必要があります。ND 環境内でのネーミングの詳細については、WebSphere Application Server の資料を参照してください。エミッター・ファクトリーを作成するには、以下の手順を実行します。

手順

1. サーバー管理コンソールで、「サービス統合」>「Common Event Infrastructure」>「イベント・エミッター・ファクトリー」>「新規」をクリックします。
2. 新しいエミッター・ファクトリーのプロパティを指定します。これらのプロパティについて詳しくは、エミッター・ファクトリーの設定ページについてのオンライン・ヘルプを参照してください。
3. 構成変更を保管します。
4. Network Deployment 環境の場合のみ、ノードを同期させます。

5. サーバーを再始動します。

タスクの結果

これでイベント・ソースが、構成されたエミッター・ファクトリーを使用してエミッターを作成できるようになりました。

管理コンソールを使用したイベント・グループの作成

イベント・グループは、イベントのプロパティ・データの内容に基づいて、イベントの論理的な集合を定義します。イベント・グループは、イベント・サービスからイベントを照会するときに使用できます。イベント・グループを必要に応じて JMS 宛先に関連付けることにより、イベントを非同期的に配布することもできます。

このタスクについて

イベント・グループを作成するには、以下の手順を実行します。

手順

1. オプション: イベント・グループの JMS 宛先を 1 つ以上セットアップします。イベント・グループは、1 つの JMS トピックと、1 つ以上の JMS キューに関連付けることができます。JMS 宛先と接続ファクトリーを作成する方法、およびそれらを JNDI 名前空間にバインドする方法については、JMS プロバイダーの資料を参照してください。

セキュリティ: WebSphere セキュリティーが使用可能な場合は、JMS 宛先の構成で認証別名を指定する必要があります。

2. 新規イベント・グループを作成します。サーバー管理コンソールで、「サービス統合」>「Common Event Infrastructure」>「イベント・サービス (Event Service)」>「イベント・サービス (Event Services)」>「event_service」>「イベント・グループ」>「新規」をクリックします。
3. イベント・セレクターおよび任意指定の JMS 宛先をはじめとする、新しいイベント・グループのプロパティを指定します。
4. 構成変更を保管します。
5. Network Deployment 環境の場合のみ、ノードを同期させます。
6. サーバーを再始動します。

タスクの結果

これでイベント・コンシューマーは、イベント照会時にイベント・グループを指定できるようになりました。イベント・サービス設定でイベント配布が使用可能になっている場合、イベント・グループに属するイベントは、そのイベント・グループで指定されたすべての JMS 宛先にも公開されます。これにより、イベント・コンシューマーは、適切な宛先をサブスクライブすることにより、イベントを非同期的に受け取ることができます。

管理コンソールを使用したイベント・フィルターの作成

イベント・フィルターは、デフォルトのフィルター・プラグインで使用されるプロパティを定義します。フィルター・プラグインは、ソースのイベントをフィルタリングするためにエミッターで使用されます。

このタスクについて

イベント・フィルターは、エミッター・ファクトリーの構成の一部として指定できます。このエミッター・ファクトリーを使用して作成されたすべてのエミッターは、指定されたフィルターを使用して、イベント・サービスに送信するイベントを決定します。

イベント・フィルターを作成するには、以下の手順を実行します。

手順

1. サーバー管理コンソールで、「サービス統合」>「**Common Event Infrastructure**」>「イベント・エミッター・ファクトリー」>「*emitter_factory*」>「イベント・フィルター」>「新規」をクリックします。
2. 新しいイベント・フィルターのプロパティを指定します。これらのプロパティについて詳しくは、イベント・フィルターの設定ページについてのオンライン・ヘルプを参照してください。
3. 構成変更を保管します。
4. Network Deployment 環境の場合のみ、ノードを同期させます。
5. サーバーを再始動します。

タスクの結果

これでイベント・エミッターが、構成されたフィルターを使用して、イベント・サービスに送信するイベントを決定できるようになりました。

スクリプトを使用したイベント・サービスの管理

スクリプト・インターフェースを使用して、イベント・サービスを管理できます。

スクリプトによるイベント・サービスの使用可能化

イベント・サービスを使用可能にするには、wsadmin ツールを使用して AdminTask 管理コマンドを実行します。

このタスクについて

イベント・サービスが使用可能な場合は、サーバーが始動したときに、イベント・サービスが自動的に始動します。

wsadmin ツールを使用してイベント・サービスを使用可能にするには、以下の手順を実行します。

手順

1. wsadmin ツールを開始します。

- AdminTask オブジェクトを使用して、**enableEventService** 管理コマンドを実行します。

- Jacl を使用する場合:

```
# サーバーの有効範囲のイベント・サービスを使用可能にします
$AdminTask enableEventService { -nodeName node1 -serverName server1 }

# クラスターの有効範囲のイベント・サービスを使用可能にします
$AdminTask enableEventService { -clusterName cluster1 }
```

- Jython を使用する場合:

```
# サーバーの有効範囲のイベント・サービスを使用可能にします
AdminTask.enableEventService([' -nodeName node1 -serverName server1 '])

# クラスターの有効範囲のイベント・サービスを使用可能にします
AdminTask.enableEventService([' -clusterName cluster1 '])
```

enableEventService コマンドのパラメーターは、以下のとおりです。

nodeName

イベント・サービスを使用可能にする必要のあるノードの名前。これはオプション・パラメーターで、デフォルト値は現在のノードです。ノード名を指定する場合は、サーバー名も指定する必要があります。クラスターのイベント・サービスを使用可能にする場合は、ノードを指定しないでください。

serverName

イベント・サービスを使用可能にする必要のあるサーバーの名前。このパラメーターは、ノード名を指定する場合、必須です。クラスターのイベント・サービスを使用可能にする場合は、サーバー名を指定しないでください。

clusterName

イベント・サービスを使用可能にする必要のあるクラスターの名前。このパラメーターは、クラスターのイベント・サービスを使用可能にする場合、必須です。

- サーバーを再始動します。

スクリプトによるイベント・サービスの使用不可への設定

イベント・サービスを使用不可にするには、wsadmin ツールを使用して AdminTask 管理コマンドを実行します。

このタスクについて

イベント・サービスが使用不可の場合は、サーバーが始動しても、そのイベント・サービスが自動的に始動しません。

wsadmin ツールを使用してイベント・サービスを使用不可にするには、以下の手順を実行します。

手順

- wsadmin ツールを開始します。
- AdminTask オブジェクトを使用して、**disableEventService** 管理コマンドを実行します。
 - Jacl を使用する場合:

```
# サーバーの有効範囲のイベント・サービスを使用不可にします
$AdminTask disableEventService { -nodeName node1 -serverName server1 }
```

```
# クラスターの有効範囲のイベント・サービスを使用不可にします
$AdminTask disableEventService { -clusterName cluster1 }
```

- Jython を使用する場合:

```
# サーバーの有効範囲のイベント・サービスを使用不可にします
AdminTask.disableEventService([' -nodeName node1 -serverName server1 '])
```

```
# クラスターの有効範囲のイベント・サービスを使用不可にします
AdminTask.disableEventService([' -clusterName cluster1 '])
```

disableEventService コマンドのパラメーターは、以下のとおりです。

nodeName

イベント・サービスを使用不可にする必要のあるノードの名前。これはオプション・パラメーターで、デフォルト値は現在のノードです。ノード名を指定する場合は、サーバー名も指定する必要があります。クラスターのイベント・サービスを使用不可にする場合は、ノードを指定しないでください。

serverName

イベント・サービスを使用不可にする必要のあるサーバーの名前。このパラメーターは、ノード名を指定する場合、必須です。クラスターのイベント・サービスを使用不可にする場合は、サーバー名を指定しないでください。

clusterName

イベント・サービスを使用不可にする必要のあるクラスターの名前。このパラメーターは、クラスターのイベント・サービスを使用不可にする場合、必須です。

3. サーバーを再始動します。

スクリプトによるエミッター・ファクトリーの作成

Jacl または Jython スクリプトを使用してエミッター・ファクトリーを作成できます。

始める前に

このタスクを開始するには、wsadmin ツールが実行中である必要があります。詳しくは、WebSphere Application Server の資料を参照してください。

このタスクについて

新しいエミッター・ファクトリーを構成するには、以下の手順を実行します。

手順

1. Common Event Infrastructure のプロバイダー ID を指定します。

- Jacl を使用する場合:

```
set providerid [$AdminConfig getid %
/Cell:mycell/Node:mynode/Server:myserver/ %
EventInfrastructureProvider:/]
```

- Jython を使用する場合:

```
providerid =
AdminConfig.getid('/Cell:mycell/Node:mynode/Server:myserver/EventInfrastructureProvider:/')
print providerid
```

出力例:

```
EventInfrastructureProvider(cells/mycell/nodes/mynode/servers/myserver|resources-cei.xml#
EventInfrastructureProvider_1)
```

2. 必要な属性を設定します。

- Jacl を使用する場合:

```
set Name [list name "EmitterName"]
set JndiName [list jndiName "Put JNDI name for new emitter factory here"]
set Description [list description "Put description here"]
set Category [list category "Put category here"]
```

```
# 新しいトランザクション内で各イベントを送信するには、
# TransactionMode に true を設定します
set TransactionMode [list preferredTransactionMode "false"]
```

```
# 同期イベント伝送を優先同期モードとして使用するには、
# SynchronizationMode を true に設定します
set SynchronizationMode [list preferredSynchronizationMode "true"]
```

```
# 同期伝送がサポートされない場合は、ブランクのままにします
set SyncJNDIName [list synchronousTransmissionProfileJNDIName ¥
"Put JNDI name of synchronous transmission profile here "]
```

```
# 非同期伝送がサポートされない場合は、ブランクのままにします
set AsyncJNDIName [list asynchronousTransmissionProfileJNDIName ¥
"Put JNDI name of asynchronous transmission profile here "]
```

```
set FilteringEnabled [list filteringEnabled "false"]
```

```
# フィルタリングが使用できない場合は、ブランクのままにします
set FilterJNDIName [list filterFactoryJNDIName ¥
"Put JNDI name of event filter here"]
```

```
# カスタム・プロパティには、互換モードが含まれます
set CompatibilityMode [list [list name compatibilityMode] ¥
[list description ""] ¥
[list required false] ¥
[list type java.lang.Boolean] ¥
[list value "false"] ] ¥
set resProp [list [list resourceProperties [list ¥
$CompatibilityMode ]]]
```

- Jython を使用する場合:

```
Name = ['name', 'EmitterName']
JndiName = ['jndiName', 'Put JNDI name for new emitter factory here']
Description = ['description', 'Put description here']
Category = ['category', 'Put category here']
```

```
# 新しいトランザクション内で各イベントを送信するには、
# TransactionMode に true を設定します
TransactionMode = ['preferredTransactionMode', 'false']
```

```
# 同期イベント伝送を優先同期モードとして使用するには、
# SynchronizationMode を true に設定します
SynchronizationMode = ['preferredSynchronizationMode', 'true']
```

```
# 同期伝送がサポートされない場合は、ブランクのままにします
SyncJNDIName = ['synchronousTransmissionProfileJNDIName', ¥
'Put JNDI name of synchronous transmission profile here']
```

```
# 非同期伝送がサポートされない場合は、ブランクのままにします
```

```

AsyncJNDIName = ['asynchronousTransmissionProfileJNDIName', ¥
                'Put JNDI name of asynchronous transmission profile here']

FilteringEnabled = ['filteringEnabled', 'false']

# フィルタリングが使用できない場合は、ブランクのままにします
FilterJNDIName = ['filterFactoryJNDIName', 'Put JNDI name of event filter here']

# カスタム・プロパティには、互換モードが含まれます
compatibilityName = ['name','compatibilityMode']
compatibilityDescription = ['description','']
compatibilityRequired = ['required', 'false']
compatibilityType = ['type', 'java.lang.Boolean']
compatibilityValue = ['value', 'false']
CompatibilityMode = [compatibilityName, compatibilityDescription, compatibilityRequired, ¥
                    compatibilityType, compatibilityValue]
customProperties = ['propertySet', [['resourceProperties', [CompatibilityMode]]]]

```

3. 新しいエミッター・ファクトリーのプロパティを設定します。

- Jacl を使用する場合:

```

set properties "[list $Name $JndiName $Description $Category ¥
                $TransactionMode $SynchronizationMode $AsyncJNDIName ¥
                $SyncJNDIName $FilteringEnabled $FilterJNDIName ¥
                [list propertySet $resProp]]"

```

- Jython を使用する場合:

```

properties = [Name,JndiName,Description,Category,TransactionMode,
SynchronizationMode,AsyncJNDIName,SyncJNDIName,FilteringEnabled,
FilterJNDIName, customProperties]
print properties

```

出力例:

```

[['name', 'EmitterName'], ['jndiName', 'Put JNDI name for new
emitter factory here'], ['description', 'Put description here'],
['category', 'Put category here'], ['preferredTransactionMode',
'false'], ['preferredSynchronizationMode', 'true'],
['asynchronousTransmissionProfileJNDIName', 'Put JNDI name of
asynchronous transmission profile here '],
['synchronousTransmissionProfileJNDIName', 'Put JNDI name of
synchronous transmission profile here '], ['filteringEnabled',
'false'], ['filterFactoryJNDIName', 'Put JNDI name of event
filter here'], ['propertySet', [['resourceProperties', [['name',
'compatibilityMode'], ['description', ''], ['required', 'false'],
['type', 'java.lang.Boolean'], ['value', 'false']]]]]]]

```

4. エミッター・ファクトリーを作成します。

- Jacl を使用する場合:

```

set emitterProf [$AdminConfig create EmitterFactoryProfile ¥
$providerid $properties]

```

- Jython を使用する場合:

```

print AdminConfig.create('EmitterFactoryProfile', providerid, properties)

```

出力例:

```

EmitterName(cells/mycell/nodes/mynode/servers/myserver|resources-cei.xml#EmitterFactoryProfile_1)

```

5. 構成変更を保管します。

6. Network Deployment 環境の場合のみ、ノードを同期させます。

7. サーバーを再始動します。

スクリプトによるイベント・グループの作成

Jacl スクリプトまたは Jython スクリプトを使用してイベント・グループを作成できます。

始める前に

このタスクを開始するには、wsadmin ツールが実行中である必要があります。詳しくは、WebSphere Application Server の資料を参照してください。

このタスクについて

新しいイベント・グループを構成するには、以下の手順を実行します。

手順

1. Common Event Infrastructure のプロバイダー ID を指定します。

- Jacl を使用する場合:

```
set providerid [AdminConfig getid ¥  
/Cell:mycell/Node:mynode/Server:myserver/ ¥  
EventInfrastructureProvider:/]
```

- Jython を使用する場合:

```
providerid =  
AdminConfig.getid  
( '/Cell:mycell/Node:mynode/Server:myserver/EventInfrastructureProvider:/' )  
print providerid
```

出力例:

```
EventInfrastructureProvider(cells/mycell/nodes/mynode/servers/myserver|  
resources-cei.xml#EventInfrastructureProvider_1)
```

2. イベント・グループ・リストを取得します。

- Jacl を使用する場合:

```
set eventGroupProfileId [lindex [AdminConfig list  
EventGroupProfileList $providerid] 0]
```

- Jython を使用する場合:

```
eventGroupProfileId = AdminConfig.list('EventGroupProfileList',providerid)
```

3. 必要な属性を設定します。

- Jacl を使用する場合:

```
set name [ list eventGroupName "EventGroupName" ]  
  
# イベント・セレクター・ストリングの '[' 文字は、すべてエスケープします。  
set selectorString [ list eventSelectorString "Set event selector here" ]  
  
# JMS を使用してイベントを公開しない場合は、ブランクのままにします  
set JNDIName [ list topicJNDIName "Set topic JNDI name here" ]  
  
# JMS を使用してイベントを公開しない場合は、ブランクのままにします  
set connectionFactoryJNDIName [ list topicConnectionFactoryJNDIName ¥  
"Set topic connection factory JNDI name here" ]  
  
set persistEventsFlag [ list persistEvents "true" ]  
  
# カスタム・プロパティには、互換モードが含まれます  
set CompatibilityMode [list [list name compatibilityMode] ¥  
[list description "" ] ¥  
[list required false] ¥
```

```

        [list type java.lang.Boolean] ¥
        [list value "false" ] ¥
        set resProp [list [list resourceProperties [list ¥
        $CompatibilityMode ]]]

```

- Jython を使用する場合:

```

Name = ['eventGroupName', 'EventGroupName' ]
SelectorString = ['eventSelectorString', 'Set event selector here']

```

```

# JMS を使用してイベントを公開しない場合は、空白のままにします
JNDIName = [ 'topicJNDIName', 'Set topic JNDI name here']

```

```

# JMS を使用してイベントを公開しない場合は、空白のままにします
ConnectionFactoryJNDIName = ['topicConnectionFactoryJNDIName', ¥
    'Set topic connection factory JNDI name here']

```

```

PersistEventsFlag = ['persistEvents', 'true']

```

```

# カスタム・プロパティには、互換モードが含まれます

```

```

compatibilityName = ['name','compatibilityMode']
compatibilityDescription = ['description','']
compatibilityRequired = ['required', 'false']
compatibilityType = ['type', 'java.lang.Boolean']
compatibilityValue = ['value', 'false']
CompatibilityMode = [compatibilityName,
    compatibilityDescription,
    compatibilityRequired, ¥
    compatibilityType,
    compatibilityValue]
customProperties = ['propertySet', [[ 'resourceProperties',
    [CompatibilityMode]]]]

```

4. 新しいイベント・グループのプロパティを設定します。

- Jacl を使用する場合:

```

set properties [ list $name $selectorString $JNDIName ¥
$connectionFactoryJNDIName $persistEventsFlag [list propertySet ¥
$resProp]]

```

- Jython を使用する場合:

```

properties = [Name,SelectorString,JNDIName,ConnectionFactoryJNDIName, ¥
PersistEventsFlag,customProperties]

```

出力例:

```

[['eventGroupName', 'EventGroupName'], ['eventSelectorString',
'Set event selector here'], ['topicJNDIName', 'Set topic JNDI
name here'], ['topicConnectionFactoryJNDIName', 'Set topic
connection factory JNDI name here'], ['persistEvents','true'],
['propertySet', [[ 'resourceProperties', [[['name',
'compatibilityMode'], ['description', ''], ['required', 'false'],
['type', 'java.lang.Boolean'], ['value', 'false']]]]]]]]

```

5. イベント・グループを作成します。

- Jacl を使用する場合:

```

set result [ $AdminConfig create EventGroupProfile
$eventGroupProfileId $properties ]

```

- Jython を使用する場合:

```

print AdminConfig.create('EventGroupProfile',
eventGroupProfileId, properties)

```

出力例:

```

(cells/mycell/nodes/mynode/servers/myserver|resources-cei.xml#EventGroupProfile_1)

```

6. 構成変更を保管します。
7. Network Deployment 環境の場合のみ、ノードを同期させます。
8. サーバーを再始動します。

スクリプトによるイベント・フィルターの作成

Jacl または Jython スクリプトを使用してイベント・フィルターを作成できます。

始める前に

このタスクを開始するには、wsadmin ツールが実行中である必要があります。詳しくは、WebSphere Application Server の資料を参照してください。

このタスクについて

新しいイベント・フィルターを構成するには、以下の手順を実行します。

手順

1. Common Event Infrastructure のプロバイダー ID を指定します。

- Jacl を使用する場合:

```
set providerid [$AdminConfig getid ¥  
/Cell:mycell/Node:mynode/Server:myserver/  
EventInfrastructureProvider:/]
```

- Jython を使用する場合:

```
providerid =  
AdminConfig.getid('/Cell:mycell/Node:mynode/Server:myserver/EventInfrastructureProvider:/')  
print providerid
```

出力例:

```
EventInfrastructureProvider(cells/mycell/nodes/mynode/servers/myserver|resources-cei.xml#  
EventInfrastructureProvider_1)
```

2. 必要な属性を設定します。

- Jacl を使用する場合:

```
set Name [list name "EventFilterName"]  
set JndiName [list jndiName "Put JNDI name for new event filter here"]  
set Description [list description "Set description of event filter here"]  
set Category [list category "Set category for event filter here"]
```

構成ストリングの '[' 文字は、すべてエスケープします

(例: "CommonBaseEvent¥[@severity=50¥]"

```
set filterConfigurationString [list filterConfigurationString ¥  
"Set filter configuration string here"]
```

- Jython を使用する場合:

```
Name = ['name', 'EventFilterName']  
JndiName = ['jndiName', 'Put JNDI name for new event filter here']  
Description = ['description', 'Set description of event filter here']  
Category = ['category', 'Set category for event filter here']  
FilterConfigurationString = ['filterConfigurationString', ¥  
"Set filter configuration string here"]
```

3. 新規イベント・フィルターのプロパティを設定します。

- Jacl を使用する場合:

```
set properties [list $name $jndiName $description $category ¥  
$filterConfigurationString]
```

- Jython を使用する場合:

```
properties = [Name,JndiName,Description,Category,FilterConfigurationString]
print properties
```

出力例:

```
[['name', 'EventFilterName'], ['jndiName', 'Put JNDI name for new
event filter here'], ['description', 'Set description of event
filter here'], ['category', 'Set category for event filter
here'], ['filterConfigurationString', 'Set filter configuration
string here']]
```

4. イベント・フィルターを作成します。

- Jacl を使用する場合:

```
set filterProf [$AdminConfig create FilterFactoryProfile ¥
$providerid $properties]
```

- Jython を使用する場合:

```
print AdminConfig.create('FilterFactoryProfile', providerid, properties)
```

出力例:

```
EventFilterName(cells/mycell/nodes/mynode/servers/myserver|resources-cei.xml#FilterFactoryProfile_1)
```

5. 構成変更を保管します。
6. Network Deployment 環境の場合のみ、ノードを同期させます。
7. サーバーを再始動します。

Common Event Infrastructure コンポーネントのロギングとトレース

ロギングとトレースを使用可能して、Common Event Infrastructure を使用するアプリケーションの問題をデバッグできます。

このタスクについて

Common Event Infrastructure コンポーネントは、Common Event Infrastructure のサーバーおよびクライアント環境で使用可能な JSR47 Java ロギング・フレームワークを使用します。ロギング・フレームワークの使用方法については、WebSphere Application Server Network Deployment バージョン 6.1 のトラブルシューティングの資料を参照してください。

以下の表は、Common Event Infrastructure コンポーネントが使用するロガー名を示します。

表2. ロガー名

コンポーネント	ロガー名
ルート・ロガー名	com.ibm.events
イベント・カタログ	com.ibm.events.catalog
イベント・サービス・サブコンポーネント	com.ibm.events.access com.ibm.events.bus com.ibm.events.distribution com.ibm.events.server
デフォルトのデータ・ストア・プラグイン	com.ibm.events.datastore
イベント・エミッター	com.ibm.events.emitter

表 2. ロガー名 (続き)

コンポーネント	ロガー名
通知ヘルパー	com.ibm.events.notification
構成	com.ibm.events.configuration com.ibm.events.admintask
データベース構成	com.ibm.events.install.db
マイグレーション	com.ibm.events.migration
各種のユーティリティー	com.ibm.events.util

イベント・データベースの保守

外部のイベント・データベースを使用する場合は、提供されるスクリプトを定期的
に実行して、データベースを保守する必要があります。

DB2 イベント・データベース統計の更新

Linux、UNIX、または Windows で DB2 データベースを使用可能にして、照会を最
適化し、フリー・スペースを見つけるには、**runstats** スクリプトを実行してデー
タベース統計を更新します。

このタスクについて

DB2 データベース統計は、定期的に更新することを推奨します。特に、以下の状況
では、定期的な更新が重要です。

- イベント・サービスのイベント削除インターフェース、またはデフォルトのデー
タ・ストア・プラグインの高速イベント・ページ・ユーティリティーを使用し
て、データベースからイベントが削除された場合
- 多数のイベントがデータベースに挿入された場合
- テーブルが、**reorg** スクリプトを使用して再編成された場合
- 索引がテーブルに追加されたか、テーブルから除去された場合

runstats スクリプトは、*profile_root/event/node_name/server_name* または
cluster_name/dbscripts/db2 ディレクトリーにあります。

手順

データベース統計を更新するには、以下のいずれかのコマンドを実行します。

- **Windows** Windows システムの場合:
`runstats.bat db_user [password=db_password]`
- **Linux** **UNIX** Linux および UNIX システムの場合:
`runstats.sh db_user [password=db_password]`

パラメーターは、以下のとおりです。

db_user

使用するためのデータベース・ユーザー ID。このパラメーターは必須で
す。

db_password

データベース・パスワード。このパラメーターは任意指定です。コマンド行でパスワードを指定しないと、DB2 データベースは、パスワードの入力プロンプトを表示します。

例

例えば、以下のコマンドを実行すると、データベース・ユーザー ID `dbadmin` とパスワード `mypassword` を使用して、Windows システムの DB2 データベース統計が更新されます。

```
runstats.bat dbadmin mypassword
```

DB2 イベント・データベース表の再編成

Linux、UNIX、または Windows 上の DB2 イベント・データベースからイベントをページまたは削除した後に、**reorg** スクリプトを使用してデータベース表を再編成できます。

このタスクについて

reorg スクリプトは、`profile_root/event/node_name/server_name` または `cluster_name/dbscripts/db2` ディレクトリーにあります。

手順

イベント・データベース表を再編成するには、以下のいずれかのコマンドを実行します。

- **Windows** Windows システムの場合:

```
reorg.bat db_alias db_user [password=db_password]
```
- **Linux** **UNIX** Linux および UNIX システムの場合:

```
reorg.sh db_alias db_user [password=db_password]
```

パラメーターは、以下のとおりです。

db_alias

データベース別名。DB2 クライアントではイベント・データベースがカタログされている必要があります。DB2 サーバーでスクリプトを実行する場合は、既にデータベースはカタログされています。

db_user

使用するためのデータベース・ユーザー ID。このパラメーターは必須です。

db_password

データベース・パスワード。このパラメーターは任意指定です。コマンド行でパスワードを指定しないと、DB2 データベースは、パスワードの入力プロンプトを表示します。

例

例えば、以下のコマンドは、データベース別名 `eventdb`、ユーザー ID `dbadmin`、およびパスワード `mypassword` を使用して、Windows システム上のイベント・データベース表を再編成します。

```
reorg.bat eventdb dbadmin mypassword
```

次のタスク

`reorg` スクリプトを実行したら、`runstats` スクリプトを使用して、データベース統計を更新する必要があります。詳しくは、59 ページの『DB2 イベント・データベース統計の更新』を参照してください。

イベント・データベースからのイベントのパージ

提供されているスクリプトを実行することにより、イベント・データベースから多数のイベントをパージできます。

このタスクについて

デフォルトのデータ・ストア・プラグインが提供する一連のユーティリティを定期的に実行すると、多数の古いイベントをイベント・データベースからパージできます。これらのユーティリティは、特定の基準に一致したイベントを削除する `eventpurge` イベント・サービス・コマンドとは異なります。

Derby データベース: Derby イベント・データベースでは、データベース・パージ・ユーティリティはサポートされません。

このデータベース・パージ機能では、バケットという概念が使用されます。バケットはテーブルのセットで、イベント・データベースのイベントの保管に使用されます。デフォルトのデータ・ストア・プラグインは 2 つのバケットを使用します。

- アクティブ・バケットは、最新のイベントを収容するバケットです。新しいイベントはアクティブ・バケットに格納されます。このアクティブ・バケットは、データベース・パージ・ユーティリティを使用してパージすることはできません。
- 非アクティブ・バケットは、古いイベントを収容します。非アクティブ・バケットに格納されるイベントは、照会、削除、または変更できますが、通常、新規のイベントは非アクティブ・バケットには格納されません。非アクティブ・バケットは、データベース・パージ・ユーティリティでパージできます。

各イベントは 1 つのバケットにのみ格納されます。イベント・コンシューマーから見れば、アクティブ・バケットと非アクティブ・バケットの違いはありません。コンシューマーは、イベントが保管されているバケットを知らなくても、特定のイベントを照会、変更、または削除できます。この方法の利点は、データベース固有のインターフェースを使用して、アクティブ・バケットに影響を与えることなく、非アクティブ・バケットをパージできる点です。通常のイベント・トラフィックは、パージ操作の実行中も継続して実行できます。

非アクティブ・バケットのパージ後、バケットをスワップして、アクティブ・バケットを非アクティブ・バケットに、非アクティブ・バケットをアクティブ・バケットにすることができます。バケットのスワップができるのは、非アクティブ・バケットが空の場合に限ります。

新規イベントはアクティブ・バケットにのみ保管されますが、ある状況下では、バケット交換の直後にイベントが非アクティブ・バケットに保管される場合があります。データ・ストア・プラグインは、定期的にチェックを実行し、どのバケットが現在アクティブとマークされているか判別します。ただし、次のチェックが実行されるまで、一部のイベントは非アクティブ・バケット内に継続して格納されます。また、バッチの一部として送信されるイベントは、すべて同じバケット内に格納されます。これは、バッチの処理中にバケットが非アクティブになった場合も同様です。

この高速パージ機能を使用する場合、バケットをスワップする頻度、または非アクティブ・バケットをパージする頻度を決定するのはユーザーの責任になります。イベント・トラフィック、ストレージ・スペース、アーカイブ要件、およびその他の考慮事項に基づき頻度を決定してください。

イベント・データベースのアクティブ・バケット状況の表示または変更

アクティブ・バケット状況では、現在どのバケットが使用可能か、および現在どのバケットが使用不可かが示されます。

手順

アクティブ・バケット状況を表示または変更するには、`eventbucket` コマンドを使用します。

```
eventbucket [-status] [-change]
```

このコマンドには、以下のオプションがあります。

-status

このオプションを使用すると、現在のバケット構成についての情報を表示できます。例えば、アクティブ・バケットの設定、およびバケットのチェック間隔 (データ・ストア・プラグインが、どのバケットがアクティブか判定する頻度) を表示できます。

-change

このオプションを使用すると、アクティブ・バケットと非アクティブ・バケットをスワップできます。非アクティブ・バケットは空でないと、このオプションは使用できません。

イベント・データベースの非アクティブ・バケットのパージ

非アクティブ・バケットのパージ方法は、データベース・ソフトウェアにより異なります。

このタスクについて

Derby データベース: 高速パージ・ユーティリティーは、Derby イベント・データベースに対してサポートされていません。

DB2 イベント・データベースの非アクティブ・バケットのパージ (Linux、UNIX、または Windows システム):

Linux、UNIX、および Windows システムでは、DB2 データベース用のデータベース・パージ・ユーティリティーは、シェル・スクリプトまたはバッチ・ファイルとして実装されています。

手順

非アクティブ・バケットをパージするには、以下のコマンドのいずれかを実行します。

- 非アクティブ・バケットをパージするには、以下のコマンドを実行します。

```
fastpurge dbalias dbuser [password=dbpassword] [copydir=copydir]
```

このコマンドのパラメーターは、以下のとおりです。

dbalias

データベース別名。DB2 クライアントではイベント・データベースがカタログされている必要があります。DB2 サーバーでスクリプトを実行する場合は、既にデータベースはカタログされています。

dbuser

イベント・データベースへの接続時に使用するデータベース・ユーザー ID。

dbpassword

指定したユーザー ID に使用するパスワード。このパラメーターは任意指定です。パスワードを指定しない場合、DB2 データベースは、パスワードの入力プロンプトを表示します。

copydir

ロード・ユーティリティーが生成するファイルに使用するディレクトリーへのパス。このパラメーターは、イベント・データベースの順方向リカバリーを使用可能にしているときのみ必要です (LOGRETAIN または USEREXIT データベース構成設定を使用可能にしている場合)。デフォルトでは、イベント・データベースは順方向リカバリーを使用しません。

DB2 イベント・データベースの非アクティブ・バケットのパージ:

DB2 イベント・データベース用のデータベース・パージ・ユーティリティーは、DB2 ロード・ユーティリティーを使用して実装されています。

このタスクについて

非アクティブ・バケットをパージする方法

手順

1. `eventbucket` コマンドを実行して非アクティブ・バケット (バケット 0 またはバケット 1) を識別します。
2. 適切なユーティリティー制御ファイルをアップロードします。これらのファイルは、データベース構成時に生成され、`/event/dbscripts/db2zos` ディレクトリーに格納されます。以下ファイルのいずれかをアップロードします。
 - バケット 0 が非アクティブな場合、`fastpurge00.ctl` ファイルをアップロードします。

- バケット 1 が非アクティブな場合、fastpurge01.ctl ファイルをアップロードします。

アップロード時のフォーマット: 制御ファイルは、論理レコード長 80 の固定レコード・フォーマットを使用してアップロードする必要があります。

3. ISPF DB2I 基本オプション・メニューに移動し、「ユーティリティー」オプションを選択します。
4. 以下の情報を指定します。

フィールド	値
機能	EDITJCL
ユーティリティー	LOAD
ステートメント・データ・セット	アップロードした制御ファイルを収容するデータ・セットの名前
LISTDEF	NO
テンプレート	NO

5. Enter を押して、次のパネルに移動します。
6. recdsn 入力フィールドで、アップロードした制御ファイルを格納しているデータ・セットの名前を入力します。
7. Enter を押します。非アクティブ・バケットをパージする JCL スクリプトが生成されます。
8. Enter を押して、出力メッセージをクリアします。
9. 必要に応じて生成された JCL スクリプトを編集します。
10. JCL スクリプトを実行依頼します。

タスクの結果

生成される JCL スクリプトは、再利用できます。両方のバケット (0 および 1) をパージするためのスクリプトの作成後は、この手順全体を繰り返す必要はありません。

DB2 イベント・データベース (iSeries システム) の非アクティブ・バケットのパージ:

iSeries システムの場合、DB2 イベント・データベースのデータベース・パージ・ユーティリティーは、ストアード・プロシージャーとして実装されます。

このタスクについて

非アクティブ・バケットをパージする方法

手順

1. 5250 端末をエミュレートする端末アプリケーション (IBM パーソナル・コミュニケーションズなど) を使用して、iSeries システムに接続します。
2. 高速パージ・ストアード・プロシージャーを実行するための十分な特権を持つユーザー ID を使用してログインします。イベント・データベースを作成するために使用したユーザー ID を推奨します。

3. `strsql` コマンドを実行します。
4. 以下の SQL ステートメントを入力します。

```
call collection/fast_purge()
```

`collection` は、イベント・データベースを含むコレクションの名前です。デフォルトのコレクションは `event` です。

```
call event/fast_purge()
```

Oracle イベント・データベースの非アクティブ・バケットのパージ:

Oracle イベント・データベース用のデータベース・パージ・ユーティリティは、ストアード・プロシージャとして実装されています。

始める前に

Oracle データベースのデータベース・パージ・ユーティリティを使用する前に、SQL*Plus を Oracle クライアントにインストールし、このクライアントを Oracle データベースと通信するように構成する必要があります (`tnsnames.ora` ファイルを適切に構成する必要があります)。詳しくは Oracle のドキュメンテーションを参照してください。

手順

非アクティブ・バケットをパージするには、SQL*Plus を使用してストアード・プロシージャを実行します。

```
sqlplus connect_string @fastpurge.sql
```

`fastpurge.sql` ファイルは、データベース構成時に生成されるその他のスクリプトと同じ場所に保管されます (デフォルトの場所は、`profile_root/databases/event/node_name/ server_name` または `cluster_name/dbscripts/oracle` です)。

`connect_string` パラメーターは、Oracle 接続文字列です。イベント・データベースの作成に使用したデータベース・ユーザー ID と同じユーザー ID を使用します。

Informix イベント・データベースの非アクティブ・バケットのパージ:

Informix イベント・データベース用のデータベース・パージ・ユーティリティは、ストアード・プロシージャとして実装されています。

始める前に

Informix データベースのユーティリティを使用するには、Informix 環境変数が適切に設定された環境で、Informix `dbaccess` コマンドを実行する必要があります。詳しくは、Informix の資料を参照してください。

手順

1. 非アクティブ・バケットをパージするには、次のように `dbaccess` を使用してストアード・プロシージャを実行します。

```
dbaccess - fastpurge.sql
```

fastpurge.sql ファイルは、データベース構成時に生成されるその他のスクリプトと同じ場所に保管されます (デフォルトの場所は、*profile_root/databases/event/nodelserver/dbscripts/informix* です)。

2. 次のように eventpurgepool スクリプトを実行して、接続プールをパージします。

- **Windows** Windows システムの場合: eventpurgepool.bat
- **Linux** **UNIX** Linux または UNIX システムの場合:
eventpurgepool.sh

SQL サーバー・イベント・データベースの非アクティブ・バケットのパージ:

SQL サーバー・イベント・データベースのデータベース・パージ・ユーティリティは、ストアード・プロシージャとして実装されています。

始める前に

SQL サーバー・データベースのユーティリティを使用するには、osql ユーティリティを SQL サーバー・クライアントにインストールしておく必要があります。詳しくは、SQL サーバーの資料を参照してください。

注: isql ユーティリティを使用してデータベース・パージ・ストアード・プロシージャを実行しないでください。

手順

非アクティブ・バケットをパージするには、次のように osql ユーティリティを使用してストアード・プロシージャを実行します。

```
osql -Sserver_name -Udbuser -Pdbpassword -deventdb -Q"fast_purge"
```

fast_purge ファイルは、データベース構成時に生成されるその他のスクリプトと同じ場所に保管されます (デフォルトの場所は、*profile_root/databases/event/nodelserver/dbscripts/sqlserver* です)。

このコマンドのパラメーターは、以下のとおりです。

server_name

データベース・サーバー名。

dbuser

イベント・データベースへの接続時に使用するデータベース・ユーザー ID。

dbpassword

指定したユーザー ID に使用するパスワード。このパラメーターは任意指定です。パスワードを指定しない場合、osql ユーティリティは、パスワードの入力プロンプトを表示します。

eventdb

イベント・データベースの名前 (通常は event)。

イベント・データベースのバケット確認間隔の変更

バケット確認間隔は、データ・ストア・プラグインが、どのバケットがアクティブであるのかを確認する頻度を示します。この値は、データ・ストア設定でカスタム・プロパティとして指定されます。

このタスクについて

デフォルトのバケット確認間隔は 5 分 (300 秒) です。これより短い間隔を設定すると、スワッピング後にイベントが非アクティブ・バケットに格納される可能性は低下しますが、パフォーマンスも低下します。

バケット確認間隔の変更方法。

手順

1. 管理コンソールで「サービス統合」>「**Common Event Infrastructure**」>「イベント・サービス (Event service)」>「イベント・サービス (Event services)」>「*event_service*」>「イベント・データ・ストア」をクリックします。
2. `BucketCheckInterval` プロパティの値を変更してバケット確認間隔 (秒) を指定します。

タスクの結果

この変更は、次回サーバーを始動したときに有効になります。

第 4 章 Common Event Infrastructure 機能へのアクセス保護

WebSphere メソッド・レベルの宣言セキュリティーを使用して、Common Event Infrastructure 機能へのアクセスを保護することができます。

Common Event Infrastructure には 6 つのセキュリティー・ロールが定義されており、各ロールには関係する機能グループが関連付けられています。それらのセキュリティー・ロールにより、プログラミング・インターフェースとコマンド両方へのアクセスが制御されます。

次の表に、セキュリティー・ロールと、それぞれのロールに関連付けられているユーザーのタイプを示します。

表 3. セキュリティー・ロールとユーザーのタイプ

セキュリティー・ロール	ユーザーのタイプ
eventCreator	<p>同期 EJB 呼び出しを使用してエミッターにイベントを送信する必要があるイベント・ソース。このロールのユーザーは、次のインターフェースにアクセスできます。</p> <ul style="list-style-type: none">• Emitter.sendEvent()• Emitter.sendEvents()• eventemit コマンド <p>eventCreator ロールでは、同期 EJB 呼び出しを使用してイベントを送信するようにエミッターが構成されている場合にのみ、イベント送信へのアクセスを制限します。エミッターがイベント送信の際に非同期 JMS メッセージングを使用する場合は、JMS セキュリティーを使用して、イベントの送信で使用される宛先へのアクセスを制限する必要があります。</p>
eventUpdater	<p>イベント・データベースに保管されているイベントを更新する必要があるイベント・コンシューマー。このロールのユーザーは、次のインターフェースにアクセスできます。</p> <ul style="list-style-type: none">• EventAccess.updateEvents()• EventAccess.eventExists()• EventAccess.queryEventByGlobalInstanceId()• EventAccess.queryEventsByAssociation()• EventAccess.queryEventsByEventGroup()• eventquery コマンド

表 3. セキュリティー・ロールとユーザーのタイプ (続き)

セキュリティ・ロール	ユーザーのタイプ
eventConsumer	<p>イベント・データベースに保管されているイベントを照会する必要があるイベント・コンシューマー。このロールのユーザーは、次のインターフェースにアクセスできます。</p> <ul style="list-style-type: none"> • EventAccess.eventExists() • EventAccess.queryEventByGlobalInstanceId() • EventAccess.queryEventsByAssociation() • EventAccess.queryEventsByEventGroup() • eventquery コマンド
eventAdministrator	<p>イベント・データベースに保管されているイベントを照会、更新、および削除する必要があるイベント・コンシューマー。このロールのユーザーは、次のインターフェースにアクセスできます。</p> <ul style="list-style-type: none"> • EventAccess.purgeEvents() • EventAccess.eventExists() • EventAccess.queryEventByGlobalInstanceId() • EventAccess.queryEventsByAssociation() • EventAccess.queryEventsByEventGroup() • EventAccess.updateEvents() • Emitter.sendEvent() • Emitter.sendEvents() • eventquery コマンド • eventpurge コマンド • eventemit コマンド • eventbucket コマンド
catalogReader	<p>イベント・カタログからイベント定義を取得する必要があるイベント・カタログ・アプリケーション。このロールのユーザーは、次のインターフェースにアクセスできます。</p> <ul style="list-style-type: none"> • EventCatalog.getAncestors() • EventCatalog.getChildren() • EventCatalog.getDescendants() • EventCatalog.getEventDefinition() • EventCatalog.getEventDefinitions() • EventCatalog.getEventExtensionNamesForSourceCategory() • EventCatalog.getEventExtensionToSourceCategoryBindings() • EventCatalog.getParent() • EventCatalog.getRoot() • EventCatalog.getSourceCategoriesForEventExtension() • eventcatalog コマンド (-listdefinitions オプション) • eventcatalog コマンド (-listcategories オプション) • eventcatalog コマンド (-exportdefinitions オプション)

表 3. セキュリティー・ロールとユーザーのタイプ (続き)

セキュリティ・ロール	ユーザーのタイプ
catalogAdministrator	イベント・カタログのイベント定義を作成、更新、削除、または取得する必要のあるイベント・カタログ・アプリケーション。このロールのユーザーは、EventCatalog インターフェースのすべてのメソッド、および eventcatalog コマンドのすべての関数にアクセスできます。イベント・カタログに変更が生じるとイベントが生成されるので、このロールではイベント送信インターフェースにもアクセスできます。

イベント・サービスのメッセージ駆動型 Bean は、サーバーのユーザー ID を使用して実行されます。非同期 JMS 送信を使用してイベントをイベント・サービスに送信し、メソッド・ベースのセキュリティを使用可能にしている場合は、このユーザー ID を eventCreator ロールにマップする必要があります。

セキュリティ: イベント・ソースが Java セキュリティーを有効にして稼働している場合に、ユーザー独自のグローバル固有 ID (GUID) を生成するには、適切な処理を行うようにポリシー・ファイルを変更する必要があります。以下の項目を追加してください。

```
permission java.io.FilePermission "${java.io.tmpdir}${/}guid.lock",
    "read, write, delete";
permission java.net.SocketPermission "*", "resolve";
```

第 5 章 Common Event Infrastructure のトラブルシューティング

以下のトピックでは、問題の発生時に実行していたタスクまたはアクティビティーに基づいて、イベント・サービスのトラブルシューティング情報について説明します。

始動時の問題

Common Event Infrastructure サーバーの始動に関する問題のトラブルシューティングを行います。

イベント・サービスが始動しない (メッセージ CEIDS0058E)

イベント・サービスが始動せず、メッセージ *CEIDS0058E* が *WebSphere* ログ・ファイルに出力されます。

原因

イベント・サービスは、ユーザー名で修飾された SQL ステートメントを使用します。このエラーは、イベント・データベースに接続するためにイベント・サービスが使用したユーザー名が、データベースを作成するために使用したユーザー ID と同じでないことを示しています。

対応策

イベント・データベースに接続するために使用するユーザー ID は、イベント・データベースを作成するために使用したユーザー ID と同じである必要があります。この問題を訂正するには、以下の手順を実行します。

1.
 - 単一サーバーの場合は、「サーバー」>「アプリケーション・サーバー」>「*server_name*」を選択します。
 - クラスターの場合は、「サーバー」>「クラスター」>「*cluster_name*」を選択します。
2. 「構成」タブから、「ビジネス・インテグレーション」>「Common Event Infrastructure」>「Common Event Infrastructure サーバー」を選択します。
3. 指定したユーザー ID とパスワードを、データベースを作成するために使用したものと一致するように変更します。
4. 構成変更を保管します。
5. サーバーを再始動します。

イベント送信時の問題

Common Event Infrastructure サーバーでのイベント送信に関する問題のトラブルシューティングを行います。

イベントの送信時にエラーが発生する (メッセージ CEIDS0060E)

イベントを送信しようとする、イベント・ソースがエラーを検出し、メッセージ CEIDS0060E が WebSphere ログ・ファイルに示されます。

原因

イベント・サービスは、イベント・データベースに保管されたメタデータを使用して、Common Base Event のエレメントと属性をデータベースのテーブルと列にマップします。この情報は、イベント・サービスの始動後にアプリケーションがそれを最初に使用しようとするときに、データベースから読み取られます。

メタデータ・テーブルにデータが設定されるのは、イベント・データベースの作成時です。このエラーは、実行時に必要なメタデータがテーブルに含まれない場合に発生します。

対応策

この問題を訂正するには、必要なメタデータを再作成する必要があります。イベント・データベースを作成するときに、データベース構成管理コマンドは、後でメタデータにデータを再設定するために使用できるデータベース・スクリプトも生成します。このスクリプトの名前は、次のようにデータベース・タイプに応じて異なります。

データベース・タイプ	スクリプト名
Derby	ins_metadata.derby
DB2	ins_metadata.db2
Informix	ins_metadata.sql
Oracle	ins_metadata.ora
SQL サーバー	ins_metadata.mssql
DB2 UDB for iSeries	ins_metadata.db2

デフォルトでは、このスクリプトは、*profile_root/dbscripts/CEI_database_namenode* ディレクトリーに作成されます。このスクリプトは、いつでも実行できます。

メタデータを再作成するには、以下の適切な SQL プロセッサを使用してスクリプトを実行します。

- Windows Linux UNIX DB2: db2
- Windows Linux UNIX Oracle: SQL*Plus
- Windows Linux UNIX Informix: dbaccess
- Windows SQL サーバー: osql

メタデータにデータを再設定したら、サーバーを再始動します。

イベントの送信時にエラーが発生する (ServiceUnavailableException)

イベントをイベント・サーバーに送信しようとする、イベント・ソース・アプリケーションがエラーを検出します。ログ・ファイルでは、`ServiceUnavailableException` とメッセージ「プロバイダー URL を指定して最初のコンテキストを取得しようとしている間に、通信障害が発生しました。(A *communication failure occurred while attempting to obtain an initial context with the provider URL.*)」が示されています。

原因

この問題は、イベント・ソース・アプリケーションがイベント・サーバーに接続できないことを示します。これは、以下の条件によって生じます。

- イベント・サーバーが稼働していません。
- イベント・ソース・アプリケーションが、正しい JNDI プロバイダー URL を使用するよう構成されていません。

対応策

この問題を訂正するには、以下の手順を実行します。

1. イベント・サーバーの状況を確認するには、`profile_root/bin` ディレクトリーに移動して、以下の `serverStatus` コマンドを実行します。

```
serverStatus servername
```
2. イベント・サーバーが稼働していない場合は、次のように `startServer` コマンドを使用してサーバーを始動します。

```
startServer servername
```
3. イベント・サーバーに接続できないアプリケーションを含むサーバーのホスト名とリモート・メソッド呼び出し (RMI) ポートを調べます。イベント・ソース・アプリケーション用に構成された JNDI URL に同じ値が指定されていることを確認します。CEI サーバーが別のサーバー上にある場合、JNDI は、そのリモート・デプロイメント・ターゲットを使用して解決される必要があります。

イベントの送信時にエラーが発生する (NameNotFoundException)

イベントをイベント・サービスに送信しようとする、イベント・ソース・アプリケーションがエラーを検出します。ログ・ファイルでは、`NameNotFoundException` と「名前 `events/configuration/emitter/Default` の最初のコンポーネントが見つかりません。(First component in name `events/configuration/emitter/Default` not found.)」というようなメッセージが示されています。

原因

この問題は、イベント・サービスが使用できないことを示しており、以下のいずれかの条件が原因となっている可能性があります。

- イベント・サービスがデプロイされていません。
- イベント・サービスが使用不可になっています。

対応策

イベント・サービスをデプロイするには、以下の手順を実行します。

1. wsadmin ツールを開始します。
2. AdminTask オブジェクトを使用して、deployEventService 管理コマンドを実行します。
3. サーバーを再始動します。

wsadmin ツールを使用してイベント・サービスを使用可能にするには、以下の手順を実行します。

1. wsadmin ツールを開始します。
2. AdminTask オブジェクトを使用して、enableEventService 管理コマンドを実行します。
3. サーバーを再始動します。

管理コンソールを使用してイベント・サービスを使用可能にするには、以下の手順を実行します。

1. 「アプリケーション」 → 「アプリケーション・タイプ」 → 「WebSphere エンタープライズ・アプリケーション」 → 「server」 → 「コンテナ・サービス」 → 「Common Event Infrastructure サービス」をクリックします。
2. 「サーバー始動時にサービスを使用可能にする」プロパティを選択します。
3. 「OK」をクリックして変更を保存します。
4. サーバーを再始動します。

イベントの送信時にエラーが発生する (メッセージ CEIEM0025E)

イベントをイベント・サーバーに送信しようとする、イベント・ソース・アプリケーションがエラーを検出します。ログ・ファイルでは、*DuplicateGlobalInstanceIdException* が示されています。

原因

この問題は、エミッターがイベントを送信したが、同じグローバル・インスタンス ID を持つ別のイベントが既に存在しているために、イベント・サービスがそのイベントを拒否したことを示しています。各イベントは、*globalInstanceId* プロパティで指定された固有のグローバル・インスタンス ID を持つ必要があります。

対応策

この問題を訂正するには、以下のいずれかの手順を実行します。

- イベント・ソース・アプリケーションがイベントごとに固有のグローバル・インスタンス ID を生成することを確認します。
- 送信するイベントの *globalInstanceId* プロパティを空のままにします。これにより、エミッターがイベントごとに固有の ID を自動的に生成します。

イベントの送信時にエラーが発生する (メッセージ CEIEM0034E)

イベントをイベント・サービスに送信しようとする、イベント・ソースがエラーを検出します。ログ・ファイルでは、*EmitterException* とメッセージ「エミッター・

プロファイルに定義された JNDI 名が JNDI でバインドされていないために、JMS キューを JNDI で検索できませんでした。(The JNDI lookup of a JMS queue failed because the JNDI name defined in the emitter profile is not bound in the JNDI.) が示されています。

原因

この問題は、エミッターが使用する JMS 伝送構成で、JMS 構成に定義されていない JMS リソースが 1 つ以上指定されていることを示します。

対応策

この問題を訂正するには、以下の手順を実行します。

1. 管理コンソールで「サービス統合」>「Common Event Infrastructure」>「イベント・エミッター・ファクトリー」>「emitter_factory」>「JMS 伝送設定」をクリックします。イベント・ソース・アプリケーションが使用するエミッター・ファクトリーの JMS 伝送が表示されていることを確認します。
2. 「キュー JNDI 名」プロパティと「キュー接続ファクトリー JNDI 名」プロパティに指定された値を確認します。指定した JNDI 名が JNDI 名前空間に存在していて、それが有効な JMS オブジェクトであることを確認します。必要な場合は、これらのプロパティを変更するか、必要な JMS リソースを作成します。

イベントが無効である (メッセージ CEIEM0027E)

イベント・ソースがイベントの送信を試みても、エミッターがイベントをイベント・サービスに送信しません。ログ・ファイルには、メッセージ CEIEM0027E (「Common Base Event が無効であるため、エミッターがイベントをイベント・サーバーに送信できませんでした (The emitter did not send the event to the event server because the Common Base Event is not valid)」) が出力されます。

原因

このメッセージは、Common Base Event 仕様に準拠しないデータが、1 つ以上のイベント・プロパティに含まれていることを示します。イベント・データが無効になる状態は多数あります。以下に例を示します。

- グローバル・インスタンス ID の長さは、32 文字以上、64 文字以下である必要があります。
- 重大度は、0 から 70 の範囲内である必要があります。

対応策

この問題を訂正するには、以下の手順を実行します。

1. ログ・ファイルで例外メッセージの詳細を調べて、無効なイベント・プロパティを突き止めます。例えば、以下のメッセージは、グローバル・インスタンス ID (ABC) の長さが無効であることを示します。

```
例外: org.eclipse.hyades.logging.events.cbe.ValidationException
(Exception: org.eclipse.hyades.logging.events.cbe.ValidationException)
: IWAT0206E 指定した Common Base Event プロパティの ID の長さが、
32 文字から 64 文字の有効な範囲内に収まっていません。
(The length of the identifier in the specified Common Base Event
```

```
property is outside the valid range of 32 to 64 characters.)
プロパティ: CommonBaseEvent.globalInstanceId
(Property: CommonBaseEvent.globalInstanceId)
値: ABC
(Value: ABC)
```

2. Common Base Event 仕様に準拠するように、ソースのイベントの内容を訂正します。
3. イベントを再送信します。

同期モードがサポートされない (メッセージ CEIEM0015E)

イベント・ソースがイベントの送信を試みても、エミッターがイベントをイベント・サービスに送信せず、ログ・ファイルに、メッセージ *CEIEM0015E* (「指定された同期モードは、エミッターでサポートされません (*The emitter does not support the specified synchronization mode*)」) を出力します。

原因

この問題は、イベント送信時にイベント・ソースが渡すパラメーターで、エミッターでサポートされない同期モードが指定されていることを示します。これは、以下のいずれかの条件によって生じます。

- イベント・ソースで指定されている同期モードが無効です。これは、`IllegalArgumentException` とメッセージ「同期モード *mode* は無効です。 (*Synchronization mode mode is not valid.*)」によって示されます。
- イベント・ソースで指定されている同期モードが、エミッターでサポートされるように構成されていません。これは、`SynchronizationModeNotSupportedException` とメッセージ「指定された同期モード *mode* は、エミッターでサポートされません。 (*The emitter does not support the specified synchronization mode: mode.*)」によって示されます。

対応策

イベント・ソースで指定されている同期モードが無効であること (`IllegalArgumentException`) が例外メッセージで示された場合は、イベント送信を試みているメソッド呼び出しを調べます。メソッド・パラメーターに以下の有効な同期モードのいずれかが指定されていることを確認します。

- `SynchronizationMode.ASYNCHRONOUS`
- `SynchronizationMode.SYNCHRONOUS`
- `SynchronizationMode.DEFAULT`

これらの定数は、`com.ibm.events.emitter.SynchronizationMode` インターフェースで定義されています。

指定された同期モードがエミッターでサポートされないこと (`SynchronizationModeNotSupportedException`) が例外メッセージで示された場合は、以下の手順でエミッター・ファクトリー構成を確認します。

1. 管理コンソールで「サービス統合」 > 「Common Event Infrastructure」 > 「イベント・エミッター・ファクトリー」 > 「*emitter_factory*」をクリックします。イベント・ソース・アプリケーションが使用するエミッター・ファクトリーが表示されていることを確認します。

2. エミッター・ファクトリーの設定を調べて、サポートされる同期モードを確認します。

- 「イベント・サービス伝送をサポート (Support Event Service transmission)」プロパティーが選択されている場合は、同期モードがサポートされます。
- 「JMS 伝送をサポート (Support JMS transmission)」プロパティーが選択されている場合は、非同期モードがサポートされます。

トランザクション・モードの照会: イベント・ソースは、`isSynchronizationModeSupported()` メソッドを使用することにより、特定のエミッターでサポートされるトランザクション・モードをプログラマチックに照会できます。詳しくは、Javadoc API の資料を参照してください。

3. 使用しようとする同期モードがエミッターでサポートされない場合は、エミッター・ファクトリー構成を変更するか、サポートされる同期モードを使用するようにイベント・ソースを変更します。

トランザクション・モードがサポートされない (メッセージ CEIEM0016E)

イベント・ソースがイベントの送信を試みても、エミッターがイベントをイベント・サービスに送信せず、メッセージ *CEIEM0016E* (「指定されたトランザクション・モードは、エミッターでサポートされません (*The emitter does not support the specified transaction mode*)」) をログ・ファイルに出力します。

原因

この問題は、イベント送信時にイベント・ソースが渡すパラメーターで、エミッターでサポートされないトランザクション・モードが指定されていることを示します。これは、以下のいずれかの条件によって生じます。

- イベント・ソースで指定されているトランザクション・モードが無効です。
- イベント・ソースで指定されている同期モードが、そのエミッター環境でサポートされません。トランザクションは、Java EE コンテナ内でのみサポートされます。

対応策

この問題を訂正するには、イベントの送信を試みるメソッド呼び出しを調べて、メソッド・パラメーターに正しいトランザクション・モードが指定されていることを確認します。

- エミッターが Java EE コンテナで動作している場合は、メソッド・パラメーターに以下の有効なトランザクション・モードのいずれかが指定されていることを確認します。
 - `TransactionMode.NEW`
 - `TransactionMode.SAME`
 - `TransactionMode.DEFAULT`

これらの定数は、`com.ibm.events.emitter.TransactionMode` インターフェイスで定義されています。

- エミッターが Java EE コンテナで動作していない場合は、メソッド・パラメーターに `TransactionMode.DEFAULT` が指定されていることを確認します。

イベントの受信または照会時の問題

Common Event Infrastructure サーバーでのイベントの受信および照会に関する問題のトラブルシューティングを行います。

イベントの照会時にエラーが発生する (メッセージ CEIDS0060E)

イベント・サービスからイベントを照会しようとする、イベント・コンシューマーがエラーを検出して、*WebSphere* ログ・ファイルにメッセージ *CEIDS0060E* が示されます。

原因

イベント・サービスは、イベント・データベースに保管されたメタデータを使用して、*Common Base Event* のエレメントと属性をデータベースのテーブルと列にマップします。この情報は、イベント・サービスの始動後にアプリケーションがそれを最初に使用しようとするときに、データベースから読み取られます。

メタデータ・テーブルにデータが設定されるのは、イベント・データベースの作成時です。このエラーは、実行時に必要なメタデータがテーブルに含まれない場合に発生します。

対応策

この問題を訂正するには、必要なメタデータを再作成する必要があります。イベント・データベースを作成するときに、データベース構成管理コマンドは、後でメタデータにデータを再設定するために使用できるデータベース・スクリプトも生成します。このスクリプトの名前は、次のようにデータベース・タイプに応じて異なります。

データベース・タイプ	スクリプト名
Derby	ins_metadata.derby
DB2	ins_metadata.db2
Informix	ins_metadata.sql
Oracle	ins_metadata.ora
SQL サーバー	ins_metadata.mssql
DB2 UDB for iSeries	ins_metadata.db2

デフォルトでは、このスクリプトは、*profile_root/dbscripts/CEI_database_namenode* ディレクトリーに作成されます。このスクリプトは、いつでも実行できます。

メタデータを再作成するには、以下の適切な SQL プロセッサを使用してスクリプトを実行します。

- Windows Linux UNIX DB2: db2
- Windows Linux UNIX Oracle: SQL*Plus

- **Windows** **Linux** **UNIX** Informix: dbaccess
- **Windows** SQL サーバー: osql

メタデータにデータを再設定したら、サーバーを再始動します。

イベントが永続データ・ストアに保管されない

イベント・ソース・アプリケーションが正常にイベントをエミッターに送信しますが、イベント・ソースがイベントを照会すると、そのイベントが永続データ・ストアに入っていません。

原因

この問題は、エミッターがイベントをイベント・サービスに送信していないか、イベント・サービスがイベントを永続データ・ストアに保管していないことを示します。これは、以下の条件のいずれかによって生じます。

- 永続データ・ストアがイベント・サービスに対して使用可能になっていません。
- イベントが、イベントを永続化するように構成されたイベント・グループに属していません。
- イベントがエミッターによってフィルター操作で除去されます。

対応策

永続データ・ストアがイベント・サービスに対して使用可能になっていることを確認するには、以下の手順を実行します。

1. 管理コンソールで「サービス統合」>「Common Event Infrastructure」>「イベント・サービス (Event service)」>「イベント・サービス (Event services)」>「event_service」をクリックします。
2. 「イベント・データ・ストアを使用可能にする」チェック・ボックスが選択されていることを確認します。
3. 「OK」をクリックして変更を保存します。

イベントを永続化するようにイベント・グループが構成されていることを確認するには、以下の手順を実行します。

1. 管理コンソールで「サービス統合」>「Common Event Infrastructure」>「イベント・サービス (Event service)」>「イベント・サービス (Event services)」>「event_service」>「イベント・グループ」> event_group をクリックします。
2. 「イベントをイベント・データ・ストアに永続化する (Persist events to event data store)」チェック・ボックスが選択されていることを確認します。
3. 「OK」をクリックして変更を保存します。

複数のイベント・グループ: イベントが複数のイベント・グループに属している場合があります。該当するイベント・グループが永続化を行うように構成されていて、データ・ストアが使用可能な場合は、イベントはデータ・ストアに保管されません。

フィルター設定を確認するには、以下の手順を実行します。

1. 管理コンソールで「サービス統合」>「Common Event Infrastructure」>「イベント・エミッター・ファクトリー」>「emitter_factory」>「イベント・フィルター」をクリックします。(イベント・ソース・アプリケーションが使用しているエミッター・ファクトリーの設定が表示されていることを確認してください。)
2. コンシューマーに送信を試みているイベントが、フィルター構成ストリングによって除外されるかどうかを確認します。除外される設定の場合は、フィルター構成ストリングを変更するか、イベント・データを変更して、イベントがフィルター操作で除去されないようにします。
3. 「OK」をクリックして変更を保存します。

コンシューマーがイベントを受け取らない (エラー・メッセージなし)

イベント・ソース・アプリケーションが正常にイベントをエミッターに送信しますが、コンシューマーが JMS インターフェースを使用してイベントを受け取りません。

原因

この問題は、以下のいずれかの条件によって生じます。

- イベント配布がイベント・サービスに対して使用可能になっていません。
- イベントがエミッターによってフィルター操作で除去されます。
- イベントが通知ヘルパーによってフィルター操作で除去されます。
- イベント・コンシューマーが正しいイベント・グループを指定していません。
- JMS 接続が開始していません。

対応策

この問題の対応策は、元になる原因に応じて異なります。

- イベント配布がイベント・サービスに対して使用可能になっていることを確認するには、以下の手順を実行します。
 1. 管理コンソールで「サービス統合」>「Common Event Infrastructure」>「イベント・サービス (Event service)」>「イベント・サービス (Event services)」>「event_service」をクリックします。
 2. 「イベント配布を使用可能にする」プロパティが選択されていない場合は、このチェック・ボックスを選択します。
 3. 「OK」をクリックして変更を保存します。
- エミッターのイベント・フィルター設定を確認するには、以下の手順を実行します。
 1. 管理コンソールで「サービス統合」>「Common Event Infrastructure」>「イベント・エミッター・ファクトリー」>「emitter_factory」>「イベント・フィルター」をクリックします。(イベント・ソース・アプリケーションが使用しているエミッター・ファクトリーの設定が表示されていることを確認してください。)
 2. コンシューマーに送信を試みているイベントが、フィルター構成ストリングによって除外されるかどうかを確認します。除外される設定の場合は、フィルタ

一構成ストリングを変更するか、イベント・データを変更して、イベントがフィルター操作で除去されないようにします。

3. 「OK」をクリックして変更を保存します。
- 通知ヘルパーのイベント・フィルター設定を確認するには、以下の手順を実行します。
 1. `NotificationHelper.setEventSelector` メソッドを使用してイベント・コンシューマー・アプリケーションを調べ、通知ヘルパー用にイベント・セレクターが指定されているかどうかを確認します。
 2. イベント・セレクターが指定されている場合は、受け取りを試みるイベントが除外されていないことを確認します。(イベント・セレクターがヌルの場合は、すべてのイベントが渡されます。)
 - イベント・コンシューマーが指定したイベント・グループを確認するには、以下の手順を実行します。
 1. 管理コンソールで「サービス統合」>「Common Event Infrastructure」>「イベント・サービス (Event service)」>「イベント・サービス (Event services)」>「event_service」>「イベント・グループ」をクリックします。イベント・サービス用に定義されたすべてのイベント・グループのリストが表に示されます。
 2. イベント・コンシューマーがサブスクライブするイベント・グループを選択します。
 3. 「イベント・セレクター・ストリング」プロパティを探します。
 4. 指定したイベント・セレクターが、受け取りを試みるイベントの内容と一致することを確認します。一致しない場合は、必要に応じて以下のいずれかの変更を行います。
 - イベントがイベント・グループに含まれるようにイベント・セレクターを変更します。
 - イベントがイベント・グループに一致するようにイベント・データを変更します。
 - イベントを含む別のイベント・グループをサブスクライブするようにイベント・コンシューマーを変更します。
 - JMS 接続を開始するには、以下の手順を実行します。

イベント・コンシューマーで、イベントの受け取りを試みる前に `QueueConnection.start()` メソッドまたは `TopicConnection.start()` メソッドを使用します。

コンシューマーがイベントを受け取らない (`NameNotFoundException`)

イベント・ソース・アプリケーションが正常にイベントをエミッターに送信しますが、JMS インターフェースを使用してイベントがコンシューマーに公開されません。ログ・ファイルでは、`NameNotFoundException` が示されています。

原因

この問題は、存在しない 1 つ以上の JMS リソースがイベント・グループ構成で指定されていることを示します。

対応策

この問題を訂正するには、以下の手順を実行します。

1. 管理コンソールで「サービス統合」>「Common Event Infrastructure」>「イベント・サービス (Event service)」>「イベント・サービス (Event services)」>「event_service」>「イベント・グループ」> event_group をクリックします。

複数のイベント・グループ: イベントが複数のイベント・グループに属している場合があります。

2. 「トピック JNDI 名」プロパティと「トピック接続ファクトリー JNDI 名」プロパティの値を確認します。指定した JMS リソースが存在することを確認します。必要な場合は、JMS プロバイダーの構成インターフェースを使用して、必要なリソースを作成します。

拡張データ・エレメントを持つイベント・グループにイベントが含まれない

拡張データ・エレメント述部を指定するイベント・グループを定義しましたが、このイベント・グループを照会しても、予期するイベントが返されません。

原因

イベント・データが有効な XML であっても、Common Base Event 仕様に準拠していない可能性があります。この場合は、エラー・メッセージが表示されずに、予期しない結果が発生することがあります。

以下の内容のイベントがあるとします。

```
<?xml version="1.0" encoding="ASCII"?>
<!-- XPath 式 CommonBaseEvent[@globalInstanceId] に一致するイベント -->
<CommonBaseEvent
  xmlns:xsi="http://www.w3.org/TR/xmlschema-1/"
  xmlns="http://www.ibm.com/AC/commonbaseevent1_0_1"
  version="1.0.1"
  creationTime="2005-10-17T12:00:01Z"
  severity="10"
  priority="60"
>
  <situation categoryName="RequestSituation">
    <situationType xsi:type="RequestSituation"
      reasoningScope="INTERNAL"
      successDisposition="Succeeded"
      situationQualifier="TEST"
    />
  </situation>
  <sourceComponentId
    component="component"
    subComponent="subcomponent"
    componentIdType="componentIdType"
    location="localhost"
    locationType="Hostname"
    componentType="sourceComponentType"
  />
</CommonBaseEvent>
```

```
<extendedDataElement name="color" type="string">
  <values>red</values>
</extendedDataElement>
</CommonBaseEvent>
```

このイベントには、1 つの子エレメントを持つ単一の拡張データ・エレメントが含まれます。

次に、以下の XPath イベント・セレクター・ストリングで構成されたイベント・グループ定義があるとします。

```
CommonBaseEvent[extendedDataElements[@name='color' and @type='string'
and @values='red']]
```

イベントの XML 定義にミススペルが含まれているために、このイベント・セレクターはイベントに一致しません。イベント・データで、`extendedDataElements` エレメントが `extendedDataElement` とミススペルされています。これは整形形式の XML であるため、エラーが発生せずに any エレメントとして扱われます。このエレメントは検索できません。

対応策

送信されるイベントの XML データが Common Base Event 仕様に準拠していることを確認してください。

イベント・グループの照会時にエラーが発生する (メッセージ CEIES0048E)

イベント・グループからイベントの照会を試みると、イベント・コンシューマー・アプリケーションがエラーを検出します。ログ・ファイルでは、`EventGroupNotDefinedException` とメッセージ `CEIES0048E` (「イベント・サーバー・インスタンスが使用しているイベント・グループ・リストにイベント・グループが定義されていません。(The event group is not defined in the event group list that the event server instance is using.)」) が示されています。

原因

この問題は、イベント・コンシューマー・アプリケーションが `EventAccess Bean` を使用して照会を実行したが、コンシューマーが指定したイベント・グループ名に対応する既存のイベント・グループが存在しなかったことを示します。

対応策

この問題を訂正するには、以下の手順を実行します。

1. 管理コンソールで「サービス統合」>「Common Event Infrastructure」>「イベント・サービス (Event service)」>「イベント・サービス (Event services)」>「event_service」>「イベント・グループ」をクリックします。イベント・サービス用に定義されたすべてのイベント・グループのリストが表に示されます。
2. イベント・ソースが、定義済みのイベント・グループ名を照会メソッド呼び出しのパラメーターに指定していることを確認します。

各種の問題

Common Event Infrastructure サーバーに関する各種の問題のトラブルシューティングを行います。

Windows システム上でイベント・カタログ・パターン照会が失敗する

Windows システムで `eventcatalog` コマンドを使用して、イベント定義のパターン照会を実行しようとしています。例えば、`eventcatalog -listdefinitions -name EVENT% -pattern` と入力します。しかし、予期する結果が得られません。

原因

パーセント文字 (%) は、Windows コマンド行インターフェースの予約文字であるため、`eventcatalog` コマンドに適切に渡されません。

対応策

Windows システムでは、パーセント文字をエスケープするために、次のように %% と入力する必要があります。

```
eventcatalog -listdefinitions -name EVENT%% -pattern
```

第 6 章 Common Event Infrastructure コマンド

Common Event Infrastructure でイベント・データベースとサービスの作成および除去に使用するコマンド

特定のデータベース上にイベント・データ・ソースを作成するためのコマンド:

- DB2 マルチプラットフォーム — 88 ページの『configEventServiceDB2DB コマンド』
- DB2 iSeries — 90 ページの『configEventServiceDB2iSeriesDB コマンド』
- DB2 z/OS — 93 ページの『configEventServiceDB2ZOSDB コマンド』
- Derby — 96 ページの『configEventServiceDerbyDB コマンド』
- Informix — 98 ページの『configEventServiceInformixDB コマンド』
- Oracle — 100 ページの『configEventServiceOracleDB コマンド』
- SQL Server — 103 ページの『configEventServiceSQLServerDB コマンド』

サーバーがイベント・サービスの処理に使用する汎用 Common Event Infrastructure コマンド:

- イベント・サービス・アプリケーションのデプロイ — 106 ページの『deployEventService コマンド』
- イベント・サービス・アプリケーションのメッセージ駆動型 Bean のデプロイ — 108 ページの『deployEventServiceMdb コマンド』
- メッセージングを行うために Common Event Infrastructure が使用する認証別名の作成 — 109 ページの『setEventServiceJmsAuthAlias コマンド』
- サーバーの再始動時のイベント・サービスの使用可能化 — 111 ページの『enableEventService コマンド』
- サーバーの再始動時のイベント・サービスの使用不可化 — 112 ページの『disableEventService コマンド』
- イベント・サービス状況のリターン — 113 ページの『showEventServiceStatus コマンド』
- イベント・サービス・アプリケーションの除去 — 115 ページの『removeEventService コマンド』
- イベント・サービス・アプリケーションのメッセージ駆動型 Bean の除去 — 116 ページの『removeEventServiceMdb コマンド』

特定のデータベース上にあるイベント・データ・ソースを除去するためのコマンド:

- DB2 マルチプラットフォーム — 117 ページの『removeEventServiceDB2DB コマンド』
- DB2 iSeries — 119 ページの『removeEventServiceDB2iSeriesDB コマンド』
- DB2 z/OS — 120 ページの『removeEventServiceDB2ZOSDB コマンド』
- Derby — 122 ページの『removeEventServiceDerbyDB コマンド』
- Informix — 124 ページの『removeEventServiceInformixDB コマンド』

- Oracle — 125 ページの『removeEventServiceOracleDB コマンド』
- SQL Server — 127 ページの『removeEventServiceSQLServerDB コマンド』

configEventServiceDB2DB コマンド

configEventServiceDB2DB コマンドを使用して、DB2 データベースを使用する Common Event Infrastructure を構成します。

目的

configEventServiceDB2DB コマンドは、AdminTask オブジェクトに使用可能な Common Event Infrastructure 管理コマンドです。このコマンドは、サーバーまたはクラスター上に DB2 用のイベント・サービス・データベースとデータ・ソースを作成するために使用します。AdminTask オブジェクトの詳細については、WebSphere Application Server Network Deployment バージョン 6.1 の資料を参照してください。

パラメーター

- createDB

このパラメーターを `true` に設定すると、コマンドは DDL データベース・スクリプトとデータベースを作成します。このパラメーターを `false` に設定すると、コマンドは DDL データベース・スクリプトのみを作成します。データベースを作成するには、現在のサーバーが、データベース・コマンドを実行するように既に構成済みである必要があります。指定しない場合、デフォルト値は `false` です。

- overrideDataSource

データベース・スクリプトの任意指定の出力ディレクトリー。このパラメーターを指定すると、コマンドは、指定されたディレクトリーにイベント・サービス・データベース・スクリプトを生成します。指定されたディレクトリーに絶対パスが含まれない場合、コマンドは、`profile_root/bin` 内に、指定されたディレクトリーを作成します。このパラメーターを指定しない場合、データベース・スクリプトのデフォルトの出力ディレクトリーは、`profile_root/databases/event/node/server/dbscripts/dbtype` です。

- nodeName

イベント・サービス・データ・ソースを作成する必要があるサーバーを含むノードの名前。このパラメーターを指定する場合は、`serverName` パラメーターも設定する必要があります。`clusterName` パラメーターを指定する場合は、このパラメーターを指定しないでください。

- serverName

イベント・サービス・データ・ソースを作成する必要があるサーバーの名前。`nodeName` パラメーターを指定せずにこのパラメーターを指定した場合、コマンドは、現在の WebSphere プロファイルのノード名を使用します。`clusterName` パラメーターを指定する場合は、このパラメーターを指定しないでください。

- clusterName

イベント・サービス・データ・ソースを作成する必要があるクラスターの名前。このパラメーターを指定する場合は、`serverName` パラメーターと `nodeName`

パラメーターを設定しないでください。 **serverName** パラメーターと **nodeName** パラメーターを指定する場合は、このパラメーターを指定しないでください。

- **jdbcClassPath**

JDBC ドライバーへのパス。ドライバー・ファイルのパスのみを指定します。パスにファイル名を含めないでください。このパラメーターは必須です。

- **dbName**

DB2 ノード名 (8 文字以下で指定する必要があります)。このノードは、既にカタログ済みで、DB2 サーバーと通信するように構成済みである必要があります。このパラメーターを設定する必要があるのは、現在のサーバーが DB2 クライアントとして構成されていて、パラメーター **createDB** が **true** に設定されている場合です。

- **dbHostName**

データベース・サーバーがインストールされているサーバーのホスト名。このパラメーターは必須です。

- **dbPort**

DB2 インスタンス・ポート。指定しない場合、デフォルト値は 50 000 です。

- **dbName**

作成するデータベースの名前。指定しない場合、デフォルト値は **event** です。

- **dbUser**

データベースを作成および除去する特権を持つ DB2 ユーザー ID。指定しない場合、デフォルト値は **db2inst1** です。

- **dbPassword**

DB2 パスワード。このパラメーターは必須です。

- **outputScriptDir**

データベース・スクリプトの任意指定の出力ディレクトリー。このパラメーターを指定すると、コマンドは、指定されたディレクトリーにイベント・サービス・データベース・スクリプトを生成します。指定されたディレクトリーに絶対パスが含まれない場合、コマンドは、*profile_root/bin* 内に、指定されたディレクトリーを作成します。このパラメーターを指定しない場合、データベース・スクリプトのデフォルトの出力ディレクトリーは、*profile_root/databases/event/node/server/dbscripts/db2* です。

サンプル

バッチ・モードの使用例:

- Jacl を使用する場合:

```
$AdminTask configEventServiceDB2DB {-createDB true
  -overrideDataSource true
  -nodeName nodename
  -serverName servername
  -jdbcClassPath c:%sqllib%java
  -dbUser db2inst1
  -dbPassword dbpassword
  -dbHostName host_name
  -dbPort 50000 }
```

- Jython スtringを使用する場合:

```
AdminTask.configEventServiceDB2DB(['-createDB true
-overrideDataSource true
-nodeName nodename
-serverName servername
-jdbcClassPath c:%sqllib%java
-dbUser db2inst1
-dbPassword dbpassword
-dbHostName host_name
-dbPort 50000 ]
```

- Jython リストを使用する場合:

```
AdminTask.configEventServiceDB2DB(['-createDB', 'true',
'-overrideDataSource', 'true',
'-nodeName', 'nodename',
'-serverName', 'servername',
'-jdbcClassPath', 'c:%sqllib%java',
'-dbUser', 'db2inst1',
'-dbPassword', 'dbpassword ',
'-dbHostName', 'host_name',
'-dbPort', '50000 '])
```

対話モードの使用例:

- Jacl を使用する場合:

```
$AdminTask configEventServiceDB2DB -interactive
```

- Jython スtringを使用する場合:

注: **Vista** **Windows 7** この製品は、Microsoft Windows 2003、Windows Vista および Windows 7 オペレーティング・システムをサポートしない Jython バージョンを使用します。

```
AdminTask.configEventServiceDB2DB(['-interactive'])
```

- Jython リストを使用する場合:

```
AdminTask.configEventServiceDB2DB(['-interactive'])
```

configEventServiceDB2iSeriesDB コマンド

configEventServiceDB2iSeriesDB コマンドを使用して、DB2 for iSeries データベースを使用する Common Event Infrastructure を構成します。

目的

configEventServiceDB2iSeriesDB コマンドは、AdminTask オブジェクトに使用可能な Common Event Infrastructure 管理コマンドです。これを使用して、リモート・データベース・サーバー上で使用する DDL データベース・スクリプトを生成し、ローカル・サーバー上の DB2 iSeries 用のイベント・サービス・データベースを作成し、サーバーまたはクラスター上にデータ・ソースを作成します。AdminTask オブジェクトの詳細については、WebSphere Application Server Network Deployment バージョン 6.1 の資料を参照してください。

注: **Vista** **Windows 7** この製品は、Microsoft Windows 2003、Windows Vista および Windows 7 オペレーティング・システムをサポートしない Jython バージョンを使用します。

パラメーター

- createDB

このパラメーターを `true` に設定すると、コマンドは DDL データベース・スクリプトとデータベースを作成します。このパラメーターを `false` に設定すると、コマンドは DDL データベース・スクリプトのみを作成します。データベースを作成するには、現在のサーバーが、データベース・コマンドを実行するように既に構成済みである必要があります。指定しない場合、デフォルト値は `false` です。

- overrideDataSource

このパラメーターを `true` に設定すると、コマンドは、指定された有効範囲にある既存のイベント・サービス・データ・ソースをすべて除去してから、新しいイベント・サービス・データ・ソースを作成します。このパラメーターを `false` に設定すると、指定された有効範囲に別のイベント・サービス・データ・ソースがある場合、その同じ有効範囲にイベント・サービス・データ・ソースを作成しません。指定しない場合、デフォルト値は `false` です。

- nodeName

イベント・サービス・データ・ソースを作成する必要があるサーバーを含むノードの名前。このパラメーターを指定する場合は、**serverName** パラメーターも設定する必要があります。**clusterName** パラメーターを指定する場合は、このパラメーターを指定しないでください。

- serverName

イベント・サービス・データ・ソースを作成する必要があるサーバーの名前。**nodeName** パラメーターを指定せずにこのパラメーターを指定した場合、コマンドは、現在の WebSphere プロファイルのノード名を使用します。**clusterName** パラメーターを指定する場合は、このパラメーターを指定しないでください。

- clusterName

イベント・サービス・データ・ソースを作成する必要があるクラスターの名前。このパラメーターを指定する場合は、**serverName** パラメーターと **nodeName** パラメーターを設定しないでください。**serverName** パラメーターと **nodeName** パラメーターを指定する場合は、このパラメーターを指定しないでください。

- toolboxJdbcClassPath

IBM Toolbox for Java の DB2 JDBC ドライバーのパス。ドライバー・ファイルのパスのみを指定します。ファイル名は含めないでください。このパラメーターまたは **jdbcClassPath** パラメーターのいずれか一方を指定する必要があります。

注: iSeriesサーバー上にデータベースを作成する場合は、**toolboxJdbcClassPath** パスを指定する必要があります。

- nativeJdbcClassPath

DB2 for iSeries のネイティブ JDBC ドライバーのパス。ドライバー・ファイルのパスのみを指定します。パスにファイル名を含めないでください。このパラメーターまたは **toolboxJdbcClassPath** パラメーターのいずれか一方を指定する必要があります。

注: iSeries サーバーを使用して iSeries 以外のサーバー上にデータベースを作成する場合は、**nativeJdbcClassPath** を指定する必要があります。DB2 Universal Database (マルチプラットフォーム版) または DB2 Universal Database for z/OS 上にデータベースを作成する場合は、DB2 Universal Driver を指定します。Informix、Oracle または Microsoft SQL Server 上にデータベースを作成する場合は、タイプ 4 のドライバーを指定します。

- **dbHostName**

DB2 for iSeries データベース・サーバーがインストールされているサーバーのホスト名。このパラメーターは、IBM Toolbox for Java の DB2 JDBC ドライバーを使用する場合には必須です。

- **dbName**

DB2 for iSeries のデータベース名。指定しない場合、デフォルト値は *LOCAL です。

- **collection**

DB2 for iSeries ライブラリーの SQL コレクション。コレクション名の長さは、最長 10 文字です。指定しない場合、デフォルト値は空ストリングです。

- **dbUser**

データベースを作成および除去する特権を持つ DB2 ユーザー ID。このパラメーターは必須です。

- **dbPassword**

データベース・ユーザー ID のパスワード。このパラメーターは必須です。

- **outputScriptDir**

データベース・スクリプトの任意指定の出力ディレクトリー。このパラメーターを指定すると、コマンドは、指定されたディレクトリーにイベント・サービス・データベース・スクリプトを生成します。指定されたディレクトリーに絶対パスが含まれない場合、コマンドは、*profile_root/bin* 内に、指定されたディレクトリーを作成します。このパラメーターを指定しない場合、データベース・スクリプトのデフォルトの出力ディレクトリーは、*profile_root/databases/event/node/server/dbscripts/db2iseries* です。

サンプル

バッチ・モードの使用例:

- Jacl を使用する場合:

```
$AdminTask configEventServiceDB2iSeriesDB {createDB true
  -overrideDataSource true
  -nodeName nodename
  -serverName servername
  -dbUser db2user
  -dbPassword dbpassword
  -nativeJdbcClassPath /myDB2ClassPath
  -collection event}
```

- Jython ストリングを使用する場合:

```
AdminTask.configEventServiceDB2iSeriesDB('[-createDB true
  -overrideDataSource true
  -nodeName nodename
  -serverName servername
  -nativeJdbcClassPath /myDB2ClassPath
  -collection event]')
```

- Jython リストを使用する場合:

```
AdminTask.configEventServiceDB2iSeriesDB(['-createDB', 'true',  
'-overrideDataSource', 'true',  
'-nodeName', 'nodename',  
'-serverName', 'servername',  
'-nativeJdbcClassPath', '/myDB2ClassPath',  
'-collection', 'event'])
```

対話モードの使用例:

- Jacl を使用する場合:

```
$AdminTask configEventServiceDB2iSeriesDB -interactive
```

- Jython スtringを使用する場合:

```
AdminTask.configEventServiceDB2iSeriesDB(['-interactive'])
```

- Jython リストを使用する場合:

```
AdminTask.configEventServiceDB2iSeriesDB(['-interactive'])
```

configEventServiceDB2ZOSDB コマンド

configEventServiceDB2ZOSDB コマンドを使用して、DB2 for z/OS データベースを使用する Common Event Infrastructure を構成します。

目的

configEventServiceDB2ZOSDB コマンドは、AdminTask オブジェクトに使用可能な Common Event Infrastructure 管理コマンドです。このコマンドは、サーバーまたはクラスター上に DB2 z/OS 用のイベント・サービス・データベースとデータ・ソースを作成するために使用します。AdminTask オブジェクトの詳細については、WebSphere Application Server Network Deployment バージョン 6.1 の資料を参照してください。

注: **Vista** **Windows 7** この製品は、Microsoft Windows 2003、Windows Vista および Windows 7 オペレーティング・システムをサポートしない Jython バージョンを使用します。

パラメーター

- createDB

このパラメーターを true に設定すると、コマンドは DDL データベース・スクリプトとデータベースを作成します。このパラメーターを false に設定すると、コマンドは DDL データベース・スクリプトのみを作成します。データベースを作成するには、現在のサーバーが、データベース・コマンドを実行するように既に構成済みである必要があります。指定しない場合、デフォルト値は false です。

- overrideDataSource

このパラメーターを true に設定すると、コマンドは、指定された有効範囲にある既存のイベント・サービス・データ・ソースをすべて除去してから、新しいイベント・サービス・データ・ソースを作成します。このパラメーターを false に設定すると、指定された有効範囲に別のイベント・サービス・データ・ソースがある場合、その同じ有効範囲にイベント・サービス・データ・ソースを作成しません。指定しない場合、デフォルト値は false です。

- **nodeName**
イベント・サービス・データ・ソースを作成する必要があるサーバーを含むノードの名前。このパラメーターを指定する場合は、**serverName** パラメーターも設定する必要があります。**clusterName** パラメーターを指定する場合は、このパラメーターを指定しないでください。
- **serverName**
イベント・サービス・データ・ソースを作成する必要があるサーバーの名前。**nodeName** パラメーターを指定せずにこのパラメーターを指定した場合、コマンドは、現在の WebSphere プロファイルのノード名を使用します。**clusterName** パラメーターを指定する場合は、このパラメーターを指定しないでください。
- **clusterName**
イベント・サービス・データ・ソースを作成する必要があるクラスターの名前。このパラメーターを指定する場合は、**serverName** パラメーターと **nodeName** パラメーターを設定しないでください。**serverName** パラメーターと **nodeName** パラメーターを指定する場合は、このパラメーターを指定しないでください。
- **jdbcClassPath**
JDBC ドライバーへのパス。ドライバー・ファイルのパスのみを指定します。パスにファイル名を含めないでください。このパラメーターは必須です。
- **dbHostName**
データベースがインストールされているサーバーのホスト名。このパラメーターは必須です。
- **dbPort**
DB2 for z/OS のインスタンス・ポート。指定しない場合、デフォルト値は 5027 です。
- **dbName**
DB2 データベース名。これは、DB2 クライアント上では、カタログされたデータベースの名前です。ネイティブ z/OS サーバー上では、データベース・サブシステムの名前です。指定しない場合、デフォルト値は event です。
- **dbDiskSizeInMB**
イベント・サービス・データベースのディスク・サイズ (MB) を指定します。この値は、10 MB 以上である必要があります。指定しない場合、デフォルト値は 100 MB です。
- **dbUser**
データベースを作成および除去する特権を持つ DB2 ユーザー ID。このパラメーターは必須です。
- **dbPassword**
データベース・ユーザー ID のパスワード。このパラメーターは必須です。
- **storageGroup**
イベント・データベースとイベント・カタログ・データベースのストレージ・グループ。ストレージ・グループは、既に作成済みで、アクティブになっている必要があります。

- **bufferPool4K**

4K バッファース・プールの名前。このバッファース・プールは、データベース DDL スクリプトを実行する前にアクティブにしておく必要があります。

- **bufferPool8K**

8K バッファース・プールの名前。このバッファース・プールは、データベース DDL スクリプトを実行する前にアクティブにしておく必要があります。

- **bufferPool16K**

16K バッファース・プールの名前。このバッファース・プールは、データベース DDL スクリプトを実行する前にアクティブにしておく必要があります。

- **outputScriptDir**

データベース・スクリプトの任意指定の出力ディレクトリー。このパラメーターを指定すると、コマンドは、指定されたディレクトリーにイベント・サービス・データベース・スクリプトを生成します。指定されたディレクトリーに絶対パスが含まれない場合、コマンドは、*profile_root/bin* 内に、指定されたディレクトリーを作成します。このパラメーターを指定しない場合、データベース・スクリプトのデフォルトの出力ディレクトリーは、*profile_root/databases/event/node/server/dbscripts/db2zos* です。

サンプル

バッチ・モードの使用例:

- Jacl を使用する場合:

```
$AdminTask configEventServiceDB2ZOSDB {-createDB true -overrideDataSource true
-nodeName nodename -serverName servename -jdbcClassPath c:%sqllib%java
-dbUser db2user -dbPassword dbpassword -dbHostName host_name -dbPort 5027
-storageGroup sysdeflt -bufferPool4K BP9 -bufferPool8K BP8K9
-bufferPool16K BP16K9}
```

- Jython スtringを使用する場合:

```
AdminTask.configEventServiceDB2ZOSDB(['-createDB true -overrideDataSource true
-nodeName nodename -serverName servename -jdbcClassPath c:%sqllib%java
-dbUser db2user -dbPassword dbpassword -dbHostName host_name -dbPort 5027
-storageGroup sysdeflt -bufferPool4K BP9 -bufferPool8K BP8K9
-bufferPool16K BP16K9'])
```

- Jython リストを使用する場合:

```
AdminTask.configEventServiceDB2ZOSDB(['-createDB', 'true', '-overrideDataSource',
'true', '-nodeName', 'nodename', '-serverName', 'servename', '-jdbcClassPath',
'c:%sqllib%java', '-dbUser', 'db2user', '-dbPassword', 'dbpassword',
'-dbHostName', 'host_name', '-dbPort', '5027', '-storageGroup', 'sysdeflt',
'-bufferPool4K', 'BP9', '-bufferPool8K', 'BP8K9', '-bufferPool16K', 'BP16K9'])
```

対話モードの使用例:

- Jacl を使用する場合:

```
$AdminTask configEventServiceDB2ZOSDB -interactive
```

- Jython スtringを使用する場合:

```
AdminTask.configEventServiceDB2ZOSDB(['-interactive'])
```

- Jython リストを使用する場合:

```
AdminTask.configEventServiceDB2ZOSDB ['-interactive'])
```

configEventServiceDerbyDB コマンド

configEventServiceDerbyDB コマンドを使用して、Derby データベースを使用する Common Event Infrastructure を構成します。

目的

configEventServiceDerbyDB コマンドは、AdminTask オブジェクトに使用可能な Common Event Infrastructure 管理コマンドです。このコマンドは、サーバーまたはクラスター上に Derby 用のイベント・サービス・データベースとデータ・ソースを作成するために使用します。AdminTask オブジェクトの詳細については、WebSphere Application Server Network Deployment バージョン 6.1 の資料を参照してください。

注: **Vista** **Windows 7** この製品は、Microsoft Windows 2003、Windows Vista および Windows 7 オペレーティング・システムをサポートしない Jython バージョンを使用します。

パラメーター

- createDB

このパラメーターを true に設定すると、コマンドは DDL データベース・スクリプトとデータベースを作成します。このパラメーターを false に設定すると、コマンドは DDL データベース・スクリプトのみを作成します。データベースを作成するには、現在のサーバーが、データベース・コマンドを実行するように既に構成済みである必要があります。指定しない場合、デフォルト値は false です。

- overrideDataSource

このパラメーターを true に設定すると、コマンドは、指定された有効範囲にある既存のイベント・サービス・データ・ソースをすべて除去してから、新しいイベント・サービス・データ・ソースを作成します。このパラメーターを false に設定すると、指定された有効範囲に別のイベント・サービス・データ・ソースがある場合、その同じ有効範囲にイベント・サービス・データ・ソースを作成しません。指定しない場合、デフォルト値は false です。

- nodeName

イベント・サービス・データ・ソースを作成する必要があるサーバーを含むノードの名前。このパラメーターを指定する場合は、**serverName** パラメーターも設定する必要があります。**clusterName** パラメーターを指定する場合は、このパラメーターを指定しないでください。

- serverName

イベント・サービス・データ・ソースを作成する必要があるサーバーの名前。**nodeName** パラメーターを指定せずにこのパラメーターを指定した場合、コマンドは、現在の WebSphere プロファイルのノード名を使用します。**clusterName** パラメーターを指定する場合は、このパラメーターを指定しないでください。

- clusterName

イベント・サービス・データ・ソースを作成する必要があるクラスターの名前。このパラメーターを指定する場合は、**serverName** パラメーターと **nodeName**

パラメーターを設定しないでください。 **serverName** パラメーターと **nodeName** パラメーターを指定する場合は、このパラメーターを指定しないでください。

- **dbHostName**

ネットワーク Derby データベースのホスト名。 Derby ネットワーク・データ・ソースを作成するには、このパラメーターと **dbPort** パラメーターを指定します。 Derby ローカル・データ・ソースを作成する場合は、このパラメーターと **dbPort** パラメーターを指定しないでください。

- **dbPort**

ネットワーク Derby データベースのポート番号。 Derby ネットワーク・データ・ソースを作成するには、このパラメーターと **dbHostName** パラメーターを指定します。 Derby ローカル・データ・ソースを作成する場合は、このパラメーターと **dbHostName** パラメーターを指定しないでください。

- **dbName**

作成するデータベースの名前。指定しない場合、デフォルト値は `event` です。

- **dbUser**

データ・ソースで使用される Derby データベース認証用のユーザー ID。 WebSphere ドメイン・セキュリティーが使用不可の場合、このパラメーターは任意指定です。このパラメーターを指定する場合は、**dbPassword** パラメーターも指定する必要があります。このパラメーターは、WebSphere ドメイン・セキュリティーが使用可能な場合は必須です。

- **dbPassword**

データ・ソースで使用される Derby データベース認証用のパスワード。 WebSphere ドメイン・セキュリティーが使用不可の場合、このパラメーターは任意指定です。このパラメーターを指定する場合は、**dbUser** パラメーターも指定する必要があります。このパラメーターは、WebSphere ドメイン・セキュリティーが使用可能な場合は必須です。

- **outputScriptDir**

データベース・スクリプトの任意指定の出力ディレクトリー。このパラメーターを指定すると、コマンドは、指定されたディレクトリーにイベント・サービス・データベース・スクリプトを生成します。指定されたディレクトリーに絶対パスが含まれない場合、コマンドは、*profile_root/bin* 内に、指定されたディレクトリーを作成します。このパラメーターを指定しない場合、データベース・スクリプトのデフォルトの出力ディレクトリーは、*profile_root/databases/event/node/server/dbscripts/derby* です。

サンプル

バッチ・モードの使用例:

- Jacl を使用する場合:

```
$AdminTask configEventServiceDerbyDB {-createDB true -overrideDataSource true
-nodeName nodename -serverName servername}
```

- Jython ストリングを使用する場合:

```
AdminTask.configEventServiceDerbyDB( '[-createDB true -overrideDataSource true
-nodeName nodename -serverName servername]')
```

- Jython リストを使用する場合:

```
AdminTask.configEventServiceDerbyDB(['-createDB', 'true', '-overrideDataSource',
'true', '-nodeName', 'nodename', '-serverName', 'servername'])
```

対話モードの使用例:

- Jacl を使用する場合:
`$AdminTask configEventServiceDerbyDB -interactive`
- Jython スtringを使用する場合:
`AdminTask.configEventServiceDerbyDB(['-interactive'])`
- Jython リストを使用する場合:
`AdminTask.configEventServiceDerbyDB(['-interactive'])`

configEventServiceInformixDB コマンド

configEventServiceInformixDB コマンドを使用して、Informix データベースを使用する Common Event Infrastructure を構成します。

目的

configEventServiceInformixDB コマンドは、AdminTask オブジェクトに使用可能な Common Event Infrastructure 管理コマンドです。このコマンドは、サーバーまたはクラスター上に Informix 用のイベント・サービス・データベースとデータ・ソースを作成するために使用します。AdminTask オブジェクトの詳細については、WebSphere Application Server Network Deployment バージョン 6.1 の資料を参照してください。

注: **Vista** **Windows 7** この製品は、Microsoft Windows 2003、Windows Vista および Windows 7 オペレーティング・システムをサポートしない Jython バージョンを使用します。

パラメーター

- createDB

このパラメーターを true に設定すると、コマンドは DDL データベース・スクリプトとデータベースを作成します。このパラメーターを false に設定すると、コマンドは DDL データベース・スクリプトのみを作成します。データベースを作成するには、現在のサーバーが、データベース・コマンドを実行するように既に構成済みである必要があります。指定しない場合、デフォルト値は false です。

- overrideDataSource

データベース・スクリプトの任意指定の出力ディレクトリー。このパラメーターを指定すると、コマンドは、指定されたディレクトリーにイベント・サービス・データベース・スクリプトを生成します。指定されたディレクトリーに絶対パスが含まれない場合、コマンドは、*profile_root/bin* 内に、指定されたディレクトリーを作成します。このパラメーターを指定しない場合、データベース・スクリプトのデフォルトの出力ディレクトリーは、*profile_root/databases/event/node/server/dbscripts/dbtype* です。

- nodeName

イベント・サービス・データ・ソースを作成する必要があるサーバーを含むノー

ドの名前。このパラメーターを指定する場合は、**serverName** パラメーターも設定する必要があります。**clusterName** パラメーターを指定する場合は、このパラメーターを指定しないでください。

- **serverName**

イベント・サービス・データ・ソースを作成する必要があるサーバーの名前。

nodeName パラメーターを指定せずにこのパラメーターを指定した場合、コマンドは、現在の WebSphere プロファイルのノード名を使用します。

clusterName パラメーターを指定する場合は、このパラメーターを指定しないでください。

- **clusterName**

イベント・サービス・データ・ソースを作成する必要があるクラスターの名前。

このパラメーターを指定する場合は、**serverName** パラメーターと **nodeName** パラメーターを設定しないでください。**serverName** パラメーターと

nodeName パラメーターを指定する場合は、このパラメーターを指定しないでください。

- **jdbcClassPath**

JDBC ドライバーへのパス。ドライバー・ファイルのパスのみを指定します。パスにファイル名を含めないでください。このパラメーターは必須です。

- **dbInformixDir**

Informix データベースをインストールするディレクトリー。このパラメーターは、**createDB** パラメーターを **true** に設定した場合、指定する必要があります。このパラメーターは必須です。

- **dbHostName**

データベースがインストールされているサーバーのホスト名。このパラメーターは必須です。

- **dbServerName**

Informix サーバー名 (ol_servername など)。このパラメーターは必須です。

- **dbPort**

Informix インスタンス・ポート。指定しない場合、デフォルト値は 1526 です。

- **dbName**

作成するデータベースの名前。指定しない場合、デフォルト値は **event** です。

- **dbUser**

イベント・サービス・データベース表を所有する Informix データベース・スキーマ・ユーザー ID。WebSphere データ・ソースは、このユーザー ID を使用して Informix データベース接続を認証します。このパラメーターは必須です。

- **dbPassword**

イベント・サービス Informix テーブルを所有するスキーマ・ユーザー ID のパスワード。WebSphere データ・ソースは、このパスワードを使用して Informix データベース接続を認証します。このパラメーターは必須です。

- **ceiInstancePrefix**

コマンドは、ディレクトリー内のデータベース・ファイルを固有の名前でグループ化するためにイベント・サービス・インスタンス名を使用します。指定しない場合、デフォルト値は **ceiinst1** です。

- outputScriptDir

データベース・スクリプトの任意指定の出力ディレクトリー。このパラメーターを指定すると、コマンドは、指定されたディレクトリーにイベント・サービス・データベース・スクリプトを生成します。指定されたディレクトリーに絶対パスが含まれない場合、コマンドは、*profile_root/bin* 内に、指定されたディレクトリーを作成します。このパラメーターを指定しない場合、データベース・スクリプトのデフォルトの出力ディレクトリーは、*profile_root/databases/event/node/server/dbscripts/informix* です。

サンプル

バッチ・モードの使用例:

- Jacl を使用する場合:

```
$AdminTask configEventServiceInformixDB {-createdB true
-overrideDataSource true -nodeName nodename -serverName servername
-jdbcClassPath "c:%program files%ibm%informix%jdbc%lib"
-dbInformixDir "c:%program files%ibm%informix" -dbUser informix
-dbPassword dbpassword -dbHostName host_name -dbPort 1526
-dbServerName ol_server }
```

- Jython スtringを使用する場合:

```
AdminTask.configEventServiceInformixDB(['-createdB true
-overrideDataSource true
-nodename nodename
-serverName servername
-jdbcClassPath "c:%program files%ibm%informix%jdbc%lib"
-dbInformixDir "c:%program files%ibm%informix"
-dbUser informix
-dbPassword dbpassword
-dbHostName host_name
-dbPort 1526
-dbServerName ol_server'])
```

- Jython リストを使用する場合:

```
AdminTask.configEventServiceInformixDB(['-createdB', 'true',
'-overrideDataSource', 'true', '-nodeName', 'nodename',
'-serverName', 'servername', '-jdbcClassPath',
'c:%program files%ibm%informix%jdbc%lib', '-dbInformixDir',
'c:%program files%ibm%informix', '-dbUser', 'informix ',
'-dbPassword', 'dbpassword', '-dbHostName', 'host_name',
'-dbPort', '1526', '-dbServerName', 'ol_server'])
```

対話モードの使用例:

- Jacl を使用する場合:

```
$AdminTask configEventServiceInformixDB -interactive
```

- Jython スtringを使用する場合:

```
AdminTask.configEventServiceInformixDB(['-interactive'])
```

- Jython リストを使用する場合:

```
AdminTask.configEventServiceInformixDB(['-interactive'])
```

configEventServiceOracleDB コマンド

configEventServiceOraclesDB コマンドを使用して、Oracle データベースを使用する Common Event Infrastructure を構成します。

目的

configEventServiceOracleDB コマンドは、AdminTask オブジェクトに使用可能な Common Event Infrastructure 管理コマンドです。このコマンドは、サーバーまたはクラスター上に Oracle 用のイベント・サービス・テーブルとデータ・ソースを作成するために使用します。このコマンドでは、データベースは作成されません。Oracle SID が既に存在している必要があります。AdminTask オブジェクトの詳細については、WebSphere Application Server Network Deployment バージョン 6.1 の資料を参照してください。

注: **Vista** **Windows 7** この製品は、Microsoft Windows 2003、Windows Vista および Windows 7 オペレーティング・システムをサポートしない Jython バージョンを使用します。

パラメーター

- createDB

このパラメーターを true に設定すると、コマンドは DDL データベース・スクリプトとデータベースを作成します。このパラメーターを false に設定すると、コマンドは DDL データベース・スクリプトのみを作成します。データベースを作成するには、現在のサーバーが、データベース・コマンドを実行するように既に構成済みである必要があります。指定しない場合、デフォルト値は false です。

- overrideDataSource

データベース・スクリプトの任意指定の出力ディレクトリー。このパラメーターを指定すると、コマンドは、指定されたディレクトリーにイベント・サービス・データベース・スクリプトを生成します。指定されたディレクトリーに絶対パスが含まれない場合、コマンドは、*profile_root/bin* 内に、指定されたディレクトリーを作成します。このパラメーターを指定しない場合、データベース・スクリプトのデフォルトの出力ディレクトリーは、*profile_root/databases/event/node/server/dbscripts/dbtype* です。

- nodeName

イベント・サービス・データ・ソースを作成する必要があるサーバーを含むノードの名前。このパラメーターを指定する場合は、**serverName** パラメーターも設定する必要があります。**clusterName** パラメーターを指定する場合は、このパラメーターを指定しないでください。

- serverName

イベント・サービス・データ・ソースを作成する必要があるサーバーの名前。**nodeName** パラメーターを指定せずにこのパラメーターを指定した場合、コマンドは、現在の WebSphere プロファイルのノード名を使用します。**clusterName** パラメーターを指定する場合は、このパラメーターを指定しないでください。

- clusterName

イベント・サービス・データ・ソースを作成する必要があるクラスターの名前。このパラメーターを指定する場合は、**serverName** パラメーターと **nodeName** パラメーターを設定しないでください。**serverName** パラメーターと **nodeName** パラメーターを指定する場合は、このパラメーターを指定しないでください。

- **jdbcClassPath**
JDBC ドライバーへのパス。ドライバー・ファイルのパスのみを指定します。パスにファイル名を含めないでください。このパラメーターは必須です。
- **oracleHome**
ORACLE_HOME ディレクトリー。このパラメーターは、**createDB** パラメーターを true に設定したときに指定する必要があります。
- **dbHostName**
Oracle データベースがインストールされているサーバーのホスト名。指定しない場合、デフォルト値は localhost です。
- **dbPort**
Oracle インスタンス・ポート。指定しない場合、デフォルト値は 1521 です。
- **dbName**
Oracle システム ID (SID)。イベント・サービス・コマンドでテーブルを作成し、そのテーブルにデータを設定するには、SID が既に存在していて、使用可能である必要があります。指定しない場合、デフォルト値は orcl です。
- **dbUser**
イベント・サービス Oracle テーブルを所有する Oracle スキーマ・ユーザー ID。データベース作成時にユーザー ID が作成されます。WebSphere データ・ソースは、このユーザー ID を使用して Oracle データベース接続を認証します。指定しない場合、デフォルト値は ceiuser です。
- **dbPassword**
スキーマ・ユーザー ID のパスワード。データベース作成時にパスワードが作成されます。WebSphere データ・ソースは、このパスワードを使用して Oracle データベース接続を認証します。このパラメーターは必須です。
- **sysUser**
Oracle システム・ユーザー ID。これは、SYSDBA 特権を持つユーザーである必要があります。指定しない場合、デフォルト値は sys です。
- **sysPassword**
sysUser パラメーターで指定したユーザーのパスワード。指定しない場合、デフォルト値は空ストリングです。
- **ceiInstancePrefix**
コマンドは、ディレクトリー内のデータベース・ファイルを固有の名前でグループ化するためにイベント・サービス・インスタンス名を使用します。指定しない場合、デフォルト値は ceiinst1 です。
- **outputScriptDir**
データベース・スクリプトの任意指定の出力ディレクトリー。このパラメーターを指定すると、コマンドは、指定されたディレクトリーにイベント・サービス・データベース・スクリプトを生成します。指定されたディレクトリーに絶対パスが含まれない場合、コマンドは、*profile_root/bin* 内に、指定されたディレクトリーを作成します。このパラメーターを指定しない場合、データベース・スクリプトのデフォルトの出力ディレクトリーは、*profile_root/databases/event/node/server/dbscripts/oracle* です。

サンプル

バッチ・モードの使用例:

- Jacl を使用する場合:

```
$AdminTask configEventServiceOracleDB {-createDB true
  -overrideDataSource true
  -nodeName nodename
  -serverName servername
  -jdbcClassPath c:¥oracle¥ora92¥jdbc¥lib
  -oracleHome c:¥oracle¥ora92
  -dbUser ceiuser
  -dbPassword ceipassword
  -dbHostName host_name
  -dbPort 1521
  -sysUser sys
  -sysPassword syspassword}
```

- Jython スtringを使用する場合:

```
AdminTask.configEventServiceOracleDB( ['-createDB true
  -overrideDataSource true
  -nodeName nodename
  -serverName servername
  -jdbcClassPath c:¥oracle¥ora92¥jdbc¥lib
  -oracleHome c:¥oracle¥ora92
  -dbUser ceiuser
  -dbPassword ceipassword
  -dbHostName host_name
  -dbPort 1521
  -sysUser sys
  -sysPassword syspassword] ' )
```

- Jython リストを使用する場合:

```
AdminTask.configEventServiceOracleDB(['-createDB', 'true',
  '-overrideDataSource', 'true',
  '-nodeName', 'nodename',
  '-serverName', 'servername',
  '-jdbcClassPath', 'c:¥oracle¥ora92¥jdbc¥lib',
  '-oracleHome', 'c:¥oracle¥ora92',
  '-dbUser', 'ceiuser',
  '-dbPassword', 'ceipassword',
  '-dbHostName', 'host_name',
  '-dbPort', '1521',
  '-sysUser', 'sys',
  '-sysPassword', 'syspassword'])
```

対話モードの使用例:

- Jacl を使用する場合:

```
$AdminTask configEventServiceOracleDB -interactive
```

- Jython スtringを使用する場合:

```
AdminTask.configEventServiceOracleDB(['-interactive'] ')
```

- Jython リストを使用する場合:

```
AdminTask.configEventServiceOracleDB(['-interactive'])
```

configEventServiceSQLServerDB コマンド

configEventServiceSQLServerDB コマンドを使用して、SQL Server データベースを使用する Common Event Infrastructure を構成します。

目的

configEventServiceSQLServerDB コマンドは、AdminTask オブジェクトに使用可能な Common Event Infrastructure 管理コマンドです。このコマンドは、サーバーまたはクラスター上に SQL Server 用のイベント・サービス・データベースとデータ・ソースを作成するために使用します。AdminTask オブジェクトの詳細については、WebSphere Application Server Network Deployment バージョン 6.1 の資料を参照してください。

注: **Vista** **Windows 7** この製品は、Microsoft Windows 2003、Windows Vista および Windows 7 オペレーティング・システムをサポートしない Jython バージョンを使用します。

パラメーター

- createDB

このパラメーターを `true` に設定すると、コマンドは DDL データベース・スクリプトとデータベースを作成します。このパラメーターを `false` に設定すると、コマンドは DDL データベース・スクリプトのみを作成します。データベースを作成するには、現在のサーバーが、データベース・コマンドを実行するように既に構成済みである必要があります。指定しない場合、デフォルト値は `false` です。

- overrideDataSource

データベース・スクリプトの任意指定の出力ディレクトリー。このパラメーターを指定すると、コマンドは、指定されたディレクトリーにイベント・サービス・データベース・スクリプトを生成します。指定されたディレクトリーに絶対パスが含まれない場合、コマンドは、*profile_root/bin* 内に、指定されたディレクトリーを作成します。このパラメーターを指定しない場合、データベース・スクリプトのデフォルトの出力ディレクトリーは、*profile_root/databases/event/node/server/dbscripts/dbtype* です。

- nodeName

イベント・サービス・データ・ソースを作成する必要があるサーバーを含むノードの名前。このパラメーターを指定する場合は、**serverName** パラメーターも設定する必要があります。**clusterName** パラメーターを指定する場合は、このパラメーターを指定しないでください。

- serverName

イベント・サービス・データ・ソースを作成する必要があるサーバーの名前。**nodeName** パラメーターを指定せずにこのパラメーターを指定した場合は、コマンドは、現在の WebSphere プロファイルのノード名を使用します。**clusterName** パラメーターを指定する場合は、このパラメーターを指定しないでください。

- clusterName

イベント・サービス・データ・ソースを作成する必要があるクラスターの名前。このパラメーターを指定する場合は、**serverName** パラメーターと **nodeName** パラメーターを設定しないでください。**serverName** パラメーターと **nodeName** パラメーターを指定する場合は、このパラメーターを指定しないでください。

- **dbServerName**
SQL サーバー・データベースのサーバー名。このパラメーターは、**createDB** パラメーターを **true** に設定したときに指定する必要があります。
- **dbHostName**
SQL サーバー・データベースが稼働しているサーバーのホスト名。
- **dbPort**
SQL サーバー・ポート。指定しない場合、デフォルト値は 1433 です。
- **dbName**
作成するデータベースの名前。指定しない場合、デフォルト値は **event** です。
- **dbUser**
イベント・サービス・テーブルを所有する SQL サーバー・ユーザー ID。指定しない場合、デフォルト値は **ceiuser** です。
- **dbPassword**
dbUser パラメーターで指定した SQL サーバー・ユーザー ID のパスワード。このパラメーターは必須です。
- **saUser**
データベースとユーザーを作成および除去する特権を持つユーザー ID。このパラメーターは、**createDB** パラメーターを **true** に設定した場合は、必須です。指定しない場合、デフォルト値は **sa** です。
- **saPassword**
sa パスワード。**sa** ユーザー ID にパスワードがない場合は、このパラメーターを指定しないでください。
- **ceiInstancePrefix**
コマンドは、ディレクトリー内のデータベース・ファイルを固有の名前でグループ化するためにイベント・サービス・インスタンス名を使用します。指定しない場合、デフォルト値は **ceiinst1** です。
- **outputScriptDir**
データベース・スクリプトの任意指定の出力ディレクトリー。このパラメーターを指定すると、コマンドは、指定されたディレクトリーにイベント・サービス・データベース・スクリプトを生成します。指定されたディレクトリーに絶対パスが含まれない場合、コマンドは、*profile_root/bin* 内に、指定されたディレクトリーを作成します。このパラメーターを指定しない場合、データベース・スクリプトのデフォルトの出力ディレクトリーは、*profile_root/databases/event/node/server/dbscripts/sqlserver* です。

サンプル

バッチ・モードの使用例:

- Jacl を使用する場合:

```
$AdminTask configEventServiceSQLServerDB {-createDB true
-overrideDataSource true
-nodeName nodename
-serverName servername
-dbUser ceiuser
-dbPassword ceipassword
-dbServerName sqlservername}
```

```
-dbHostName host_name
-dbPort 1433
-saUser sa
-saPassword sapassword}
```

- Jython スtringを使用する場合:

```
AdminTask.configEventServiceSQLServerDB(['-createDB true
-overrideDataSource true
-nodeName nodename
-serverName servername
-dbUser ceiuser
-dbPassword ceipassword
-dbServerName sqlservername
-dbHostName host_name
-dbPort 1433
-saUser sa
-saPassword sapassword'])
```

- Jython リストを使用する場合:

```
AdminTask.configEventServiceSQLServerDB(['-createDB', 'true',
'-overrideDataSource', 'true',
'-nodeName', 'nodename',
'-serverName', 'servername',
'-dbUser', 'ceiuser',
'-dbPassword', 'ceipassword',
'-dbServerName', 'sqlservername',
'-dbHostName', 'host_name',
'-dbPort', '1433',
'-saUser', 'sa',
'-saPassword', 'sapassword'])
```

対話モードの使用例:

- Jacl を使用する場合:

```
$AdminTask configEventServiceSQLServerDB -interactive
```

- Jython Stringを使用する場合:

```
AdminTask.configEventServiceSQLServerDB(['-interactive'])
```

- Jython リストを使用する場合:

```
AdminTask.configEventServiceSQLServerDB(['-interactive'])
```

deployEventService コマンド

deployEventService コマンドを使用して、ご使用のサーバーにイベント・サービス・アプリケーションをデプロイします。

目的

deployEventService コマンドは、AdminTask オブジェクトに使用可能な Common Event Infrastructure 管理コマンドです。このコマンドは、サーバーまたはクラスターにイベント・サービスをデプロイし、構成するために使用します。AdminTask オブジェクトの詳細については、WebSphere Application Server Network Deployment バージョン 6.1 の資料を参照してください。

注: **Vista** **Windows 7** この製品は、Microsoft Windows 2003、Windows Vista および Windows 7 オペレーティング・システムをサポートしない Jython バージョンを使用します。

パラメーター

- nodeName

イベント・サービスをデプロイする必要があるノードの名前。このパラメーターを指定する場合は、`serverName` パラメーターを指定する必要があります。`clusterName` パラメーターを指定する場合は、このパラメーターを指定しないでください。

- serverName

イベント・サービスをデプロイする必要があるサーバーの名前。`nodeName` パラメーターを指定するときは、このパラメーターを指定する必要があります。`clusterName` パラメーターを指定する場合は、このパラメーターを指定しないでください。

- clusterName

イベント・サービスをデプロイする必要があるクラスターの名前。`nodeName` パラメーターまたは `serverName` パラメーターを指定する場合は、このパラメーターを指定しないでください。

- enable

サーバーの次の再始動後にイベント・サービスを始動するには、このパラメーターを `true` に設定します。デフォルト値は `true` です。

サンプル

バッチ・モードの使用例:

- Jacl を使用する場合:

```
$AdminTask deployEventService {-nodeName nodename
-serverName servername}
$AdminTask deployEventService {-clusterName clustername
-enable false}
```

- Jython ストリングを使用する場合:

```
AdminTask.deployEventService(['-nodeName nodename
-serverName servername'])

AdminTask.deployEventService(['-clusterName clustername
-enable false'])
```

- Jython リストを使用する場合:

```
AdminTask.deployEventService(['-nodeName', 'nodename',
'-serverName', '-servername'])
AdminTask.deployEventService(['-clusterName', 'clustername',
'-enable', 'false'])
```

対話モードの使用例:

- Jacl を使用する場合:

```
$AdminTask deployEventService {-interactive}
```

- Jython ストリングを使用する場合:

```
AdminTask.deployEventService(['-interactive'])
```

- Jython リストを使用する場合:

```
AdminTask.deployEventService(['-interactive'])
```

deployEventServiceMdb コマンド

deployEventServiceMdb コマンドを使用して、イベント・サービス・メッセージ駆動型の Bean をサーバーにデプロイします。

目的

deployEventServiceMdb コマンドは、AdminTask オブジェクトに使用可能な Common Event Infrastructure 管理コマンドです。このコマンドは、サーバーまたはクラスターにイベント・サービス MDB をデプロイするために使用します。AdminTask オブジェクトの詳細については、WebSphere Application Server Network Deployment バージョン 6.1 の資料を参照してください。

注: **Vista** **Windows 7** この製品は、Microsoft Windows 2003、Windows Vista および Windows 7 オペレーティング・システムをサポートしない Jython バージョンを使用します。

パラメーター

- nodeName

イベント・サービス MDB をデプロイする必要があるノードの名前。このパラメーターを指定する場合は、**serverName** パラメーターを指定する必要があります。**clusterName** パラメーターを指定する場合は、このパラメーターを指定しないでください。

- serverName

イベント・サービス MDB をデプロイする必要があるサーバーの名前。**nodeName** パラメーターを指定するときは、このパラメーターを指定する必要があります。**clusterName** パラメーターを指定する場合は、このパラメーターを指定しないでください。

- clusterName

イベント・サービス MDB をデプロイする必要があるクラスターの名前。**nodeName** パラメーターと **serverName** パラメーターを指定する場合は、このパラメーターを指定しないでください。

- applicationName

サーバーまたはクラスターにデプロイするイベント・サービス MDB アプリケーションの名前。

- listenerPort

イベント・サービス MDB がイベントを公開する必要があるリスナー・ポートの名前。リスナー・ポートは、既に作成済みである必要があります。**activationSpec** パラメーターを指定する場合は、このパラメーターを指定しないでください。

- activationSpec

イベント・サービス MDB がイベントを公開する必要があるアクティベーション・スペックの JNDI 名前。アクティベーション・スペックは、既に作成済みである必要があります。**listenerPort** パラメーターを指定する場合は、このパラメーターを指定しないでください。

- **qcfJndiName**

イベント・サービス MDB が使用する JMS キュー接続ファクトリー・オブジェクトの JNDI 名。 **activationSpec** パラメーターを指定する場合は、このパラメーターを指定する必要があります。

サンプル

バッチ・モードの使用例:

- Jacl を使用する場合:

```
$AdminTask deployEventServiceMdb {-applicationName appname -nodeName nodename
-serverName servername -listenerPort lpname}$AdminTask deployEventServiceMdb
{-applicationName appname -clusterName clustername -activationSpec asjndiname
-qcfJndiName qcfjndiname}
```

- Jython スtringを使用する場合:

```
AdminTask.deployEventServiceMdb(['-applicationName appname -nodeName nodename
-serverName servername -listenerPort lpname'])AdminTask.deployEventServiceMdb
(['-applicationName appname -clusterName clustername -activationSpec asjndiname
-qcfJndiName qcfjndiname'])
```

- Jython リストを使用する場合:

```
AdminTask.deployEventServiceMdb(['-applicationName', 'appname', '-nodeName',
'nodename', '-serverName', 'servername', '-listenerPort', 'lpname'])
AdminTask.deployEventServiceMdb(['-applicationName', 'appname',
'-clusterName', 'clustername', '-activationSpec', 'asjndiname',
'-qcfJndiName', 'qcfjndiname'])
```

対話モードの使用例:

- Jacl を使用する場合:

```
$AdminTask deployEventServiceMdb {-interactive}
```

- Jython スtringを使用する場合:

```
AdminTask.deployEventServiceMdb(['-interactive'])
```

- Jython リストを使用する場合:

```
AdminTask.deployEventServiceMdb(['-interactive'])
```

setEventServiceJmsAuthAlias コマンド

setEventServiceJmsAuthAlias コマンドを使用して、サーバー上でイベント・サービスに関連付けられた JMS 認証別名を設定または更新します。

目的

setEventServiceJmsAuthAlias コマンドは、AdminTask オブジェクトに使用可能な Common Event Infrastructure 管理コマンドです。このコマンドを使用して、サーバーまたはクラスター上のイベント・サービス JMS オブジェクトが使用する認証別名を更新します。JMS 認証別名が存在しない場合は作成されます。AdminTask オブジェクトの詳細については、WebSphere Application Server Network Deployment バージョン 6.1 の資料を参照してください。

注: **Vista** **Windows 7** この製品は、Microsoft Windows 2003、Windows Vista および Windows 7 オペレーティング・システムをサポートしない Jython バージョンを使用します。

パラメーター

- nodeName

イベント・サービス JMS 認証別名を更新する必要があるノードの名前。このパラメーターを指定する場合は、**serverName** パラメーターを指定する必要があります。**clusterName** パラメーターを指定する場合は、このパラメーターを指定しないでください。

- serverName

イベント・サービス JMS 認証別名を更新する必要があるサーバーの名前。このパラメーターを指定する場合は、**serverName** パラメーターを指定する必要があります。**clusterName** パラメーターを指定する場合は、このパラメーターを指定しないでください。

- clusterName

イベント・サービス JMS 認証別名を更新する必要があるクラスターの名前。**nodeName** パラメーターと **serverName** パラメーターを指定する場合は、このパラメーターを指定しないでください。

- userName

サーバーまたはクラスター上のイベント・サービス JMS 認証別名を更新するときに使用するユーザーの名前。

重要: 有効なユーザー ID を指定する必要があります。このフィールドを空にすることはできません。

- password

サーバーまたはクラスター上のイベント・サービス JMS 認証別名を更新するときに使用するユーザーのパスワード。

重要: 有効なパスワードを指定する必要があります。このフィールドを空にすることはできません。

サンプル

バッチ・モードの使用例:

- Jacl を使用する場合:

```
$AdminTask setEventServiceJmsAuthAlias{-nodeName nodename  
-serverName servername username -password pwd}  
$AdminTask setEventServiceJmsAuthAlias {-clusterName clustername  
-userName username -password pwd}
```

- Jython スtringを使用する場合:

```
AdminTask.setEventServiceJmsAuthAlias(['-nodeName nodename  
-serverName servername -userName username -password pwd'])  
AdminTask.setEventServiceJmsAuthAlias(['-clusterName clustername  
-userName username -password pwd'])
```

- Jython リストを使用する場合:

```
AdminTask.setEventServiceJmsAuthAlias(['-nodeName', 'nodename',  
'-serverName', '-servername', '-userName', 'username', '-password', 'pwd'])  
AdminTask.setEventServiceJmsAuthAlias(['-clusterName',  
'clustername', '-userName', 'username', '-password', 'pwd'])
```

対話モードの使用例:

- Jacl を使用する場合:

```
$AdminTask setEventServiceJmsAuthAlias {-interactive}
```

- Jython スtringを使用する場合:

```
AdminTask.setEventServiceJmsAuthAlias( '[-interactive]')
```

- Jython リストを使用する場合:

```
AdminTask.setEventServiceJmsAuthAlias(['-interactive'])
```

enableEventService コマンド

enableEventService コマンドを使用して、ご使用のサーバーでイベント・サービスを使用可能に設定します。

目的

enableEventService コマンドは、AdminTask オブジェクトに使用可能な Common Event Infrastructure 管理コマンドです。このコマンドは、nodeName、serverName、または clusterName パラメーターで指定したサーバーの次の再始動後に、イベント・サービスを開始できるようにするために使用します。AdminTask オブジェクトの詳細については、WebSphere Application Server Network Deployment バージョン 6.1 の資料を参照してください。

注: **Vista** **Windows 7** この製品は、Microsoft Windows 2003、Windows Vista および Windows 7 オペレーティング・システムをサポートしない Jython バージョンを使用します。

DITA

パラメーター

- nodeName

イベント・サービスを使用可能にする必要のあるノードの名前。このパラメーターを指定する場合は、**serverName** パラメーターを指定する必要があります。

clusterName パラメーターを指定する場合は、このパラメーターを指定しないでください。

- serverName

イベント・サービスを使用可能にする必要のあるサーバーの名前。**nodeName** パラメーターを指定するときは、このパラメーターを指定する必要があります。

clusterName パラメーターを指定する場合は、このパラメーターを指定しないでください。

- clusterName

イベント・サービスを使用可能にする必要のあるクラスターの名前。

nodeName パラメーターと **serverName** パラメーターを指定する場合は、このパラメーターを指定しないでください。

サンプル

バッチ・モードの使用例:

- Jacl を使用する場合:

```
$AdminTask enableEventService {-nodeName nodename  
-serverName servername}
```

```
$AdminTask enableEventService {-clusterName clustername}
```

- Jython スtringを使用する場合:

```
AdminTask.enableEventService('[-nodeName nodename  
-serverName servername]')
```

```
AdminTask.enableEventService('[-clusterName clustername]')
```

- Jython リストを使用する場合:

```
AdminTask.enableEventService(['-nodeName', 'nodename',  
'-serverName', '-servername'])
```

```
AdminTask.enableEventService(['-clusterName', 'clustername'])
```

対話モードの使用例:

- Jacl を使用する場合:

```
$AdminTask enableEventService {-interactive}
```

- Jython Stringを使用する場合:

```
AdminTask.enableEventService('[-interactive]')
```

- Jython リストを使用する場合:

```
AdminTask.enableEventService(['-interactive'])
```

disableEventService コマンド

disableEventService コマンドを使用して、イベント・サービスをサーバーで使用不可にします。

目的

disableEventService コマンドは、AdminTask オブジェクトに使用可能な Common Event Infrastructure 管理コマンドです。このコマンドは、nodeName、serverName、または clusterName パラメーターで指定したサーバー (単数または複数) の次の再始動後に、イベント・サービスを開始できないようにするために使用します。

AdminTask オブジェクトの詳細については、WebSphere Application Server Network Deployment バージョン 6.1 の資料を参照してください。

注: **Vista** この製品は、Microsoft Windows 2003 および Windows Vista オペレーティング・システムをサポートしない Jython バージョンを使用します。

パラメーター

- nodeName

イベント・サービスを使用不可にする必要のあるノードの名前。このパラメーターを指定する場合は、**serverName** パラメーターを指定する必要があります。

clusterName パラメーターを指定する場合は、このパラメーターを指定しないでください。

- serverName

イベント・サービスを使用不可にする必要のあるサーバーの名前。**nodeName**

パラメーターを指定するときは、このパラメーターを指定する必要があります。
clusterName パラメーターを指定する場合は、このパラメーターを指定しないでください。

- **clusterName**

イベント・サービスを使用不可にする必要のあるクラスターの名前。

nodeName パラメーターと **serverName** パラメーターを指定する場合は、このパラメーターを指定しないでください。

サンプル

バッチ・モードの使用例:

- Jacl を使用する場合:

```
$AdminTask disableEventService {-nodeName nodename  
-serverName servername}
```

```
$AdminTask disableEventService {-clusterName clustername}
```

- Jython スtringを使用する場合:

```
AdminTask.disableEventService(['-nodeName nodename  
-serverName servername'])
```

```
AdminTask.disableEventService(['-clusterName clustername'])
```

- Jython リストを使用する場合:

```
AdminTask.disableEventService(['-nodeName', 'nodename',  
'-serverName', '-servername'])
```

```
AdminTask.disableEventService(['-clusterName', 'clustername'])
```

対話モードの使用例:

- Jacl を使用する場合:

```
$AdminTask disableEventService {-interactive}
```

- Jython スtringを使用する場合:

```
AdminTask.disableEventService(['-interactive'])
```

- Jython リストを使用する場合:

```
AdminTask.disableEventService(['-interactive'])
```

showEventServiceStatus コマンド

showEventServiceStatus コマンドを使用して、イベント・サービスの状況をサーバーに表示します。

目的

showEventServiceStatus コマンドは、AdminTask オブジェクトに使用可能な Common Event Infrastructure 管理コマンドです。このコマンドは、サーバーまたはクラスター内のイベント・サービスの状況を返すために使用します。パラメーターを指定せずにタスクを実行した場合は、すべてのイベント・サービスの状況が表示されます。表示するイベント・サービスのリストをフィルターに掛けるには、nodeName、serverName、または clusterName を指定します。AdminTask オブジェクトの詳細については、WebSphere Application Server Network Deployment バージョン 6.1 の資料を参照してください。

注: **Vista** **Windows 7** この製品は、Microsoft Windows 2003、Windows Vista および Windows 7 オペレーティング・システムをサポートしない Jython バージョンを使用します。

パラメーター

- nodeName

このパラメーターは、指定したノードに属するイベント・サービスの状況のみを表示する場合に使用します。 **clusterName** パラメーターを指定する場合は、このパラメーターを指定しないでください。

- serverName

このパラメーターは、指定したサーバーに属するイベント・サービスの状況のみを表示する場合に使用します。このパラメーターを **nodeName** パラメーターと共に使用すると、指定したノードとサーバーに属するイベント・サービスの状況を表示できます。 **clusterName** パラメーターを指定する場合は、このパラメーターを指定しないでください。

- clusterName

このパラメーターは、指定したクラスターに属するイベント・サービスの状況のみを表示する場合に使用します。 **nodeName** パラメーターまたは **serverName** パラメーターを指定する場合は、このパラメーターを指定しないでください。

サンプル

バッチ・モードの使用例:

- Jacl を使用する場合:

```
$AdminTask showEventServiceStatus {-nodeName nodename  
-serverName servername}
```

```
$AdminTask showEventServiceStatus {-clusterName clustername}
```

- Jython スtringを使用する場合:

```
AdminTask.showEventServiceStatus(['-nodeName nodename  
-serverName servername'])
```

```
AdminTask.showEventServiceStatus(['-clusterName clustername'])
```

- Jython リストを使用する場合:

```
AdminTask.showEventServiceStatus(['-nodeName', 'nodename',  
'-serverName', '-servername'])
```

```
AdminTask.showEventServiceStatus(['-clusterName', 'clustername'])
```

対話モードの使用例:

- Jacl を使用する場合:

```
$AdminTask showEventServiceStatus {-interactive}
```

- Jython スtringを使用する場合:

```
AdminTask.showEventServiceStatus(['-interactive'])
```

- Jython リストを使用する場合:

```
AdminTask.showEventServiceStatus(['-interactive'])
```


removeEventService コマンド

removeEventService コマンドを使用して、イベント・サービス・アプリケーションをサーバーから除去します。

目的

removeEventService コマンドは、AdminTask オブジェクトに使用可能な Common Event Infrastructure 管理コマンドです。このコマンドは、サーバーまたはクラスターからイベント・サービスを除去するために使用します。AdminTask オブジェクトの詳細については、WebSphere Application Server Network Deployment バージョン 6.1 の資料を参照してください。

注: **Vista** **Windows 7** この製品は、Microsoft Windows 2003、Windows Vista および Windows 7 オペレーティング・システムをサポートしない Jython バージョンを使用します。

パラメーター

- nodeName

イベント・サービスを除去する必要があるノードの名前。このパラメーターを指定する場合は、**serverName** パラメーターを指定する必要があります。

clusterName パラメーターを指定する場合は、このパラメーターを指定しないでください。

- serverName

イベント・サービスを除去する必要があるサーバーの名前。**nodeName** パラメーターを指定するときは、このパラメーターを指定する必要があります。

clusterName パラメーターを指定する場合は、このパラメーターを指定しないでください。

- clusterName

イベント・サービスを除去する必要があるクラスターの名前。**nodeName** パラメーターと **serverName** パラメーターを指定する場合は、このパラメーターを指定しないでください。

サンプル

バッチ・モードの使用例:

- Jacl を使用する場合:

```
$AdminTask removeEventService {-nodeName nodename  
-serverName servername}
```

```
$AdminTask removeEventService {-clusterName clustername}
```

- Jython スtringを使用する場合:

```
AdminTask.removeEventService('[-nodeName nodename  
-serverName servername]')
```

```
AdminTask.removeEventService('[-clusterName clustername]')
```

- Jython リストを使用する場合:

```
AdminTask.removeEventService(['-nodeName', 'nodename',  
'-serverName', '-servername'])
```

```
AdminTask.removeEventService(['-clusterName', 'clustername'])
```

対話モードの使用例:

- Jacl を使用する場合:
`$AdminTask removeEventService {-interactive}`
- Jython スtringを使用する場合:
`AdminTask.removeEventService(['-interactive'])`
- Jython リストを使用する場合:
`AdminTask.removeEventService(['-interactive'])`

removeEventServiceMdb コマンド

removeEventServiceMdb コマンドを使用して、イベント・サービス・メッセージ駆動型の Bean をサーバーから除去します。

目的

removeEventServiceMdb コマンドは、AdminTask オブジェクトに使用可能な Common Event Infrastructure 管理コマンドです。このコマンドは、サーバーまたはクラスターからイベント・サービス MDB を除去するために使用します。AdminTask オブジェクトの詳細については、WebSphere Application Server Network Deployment バージョン 6.1 の資料を参照してください。

注: **Vista** **Windows 7** この製品は、Microsoft Windows 2003、Windows Vista および Windows 7 オペレーティング・システムをサポートしない Jython バージョンを使用します。

パラメーター

- nodeName

イベント・サービス MDB を除去する必要があるノードの名前。このパラメーターを指定する場合は、**serverName** パラメーターを指定する必要があります。**clusterName** パラメーターを指定する場合は、このパラメーターを指定しないでください。

- serverName

イベント・サービス MDB を除去する必要があるサーバーの名前。このパラメーターを指定する場合は、**serverName** パラメーターを指定する必要があります。**clusterName** パラメーターを指定する場合は、このパラメーターを指定しないでください。

- clusterName

イベント・サービス MDB を除去する必要があるクラスターの名前。**nodeName** パラメーターと **serverName** パラメーターを指定する場合は、このパラメーターを指定しないでください。

- applicationName

サーバーまたはクラスターから除去するイベント・サービス MDB アプリケーションの名前。

サンプル

バッチ・モードの使用例:

- Jacl を使用する場合:

```
$AdminTask removeEventServiceMdb {-applicationName appname  
-nodeName nodename -serverName servername}  
$AdminTask removeEventServiceMdb {-applicationName appname  
-clusterName clustername}
```

- Jython スtringを使用する場合:

```
AdminTask.removeEventServiceMdb('[-applicationName appname  
-nodeName nodename -serverName servername]')  
AdminTask.removeEventServiceMdb('[-applicationName appname  
-clusterName clustername]')
```

- Jython リストを使用する場合:

```
AdminTask.removeEventServiceMdb (['-applicationName',  
'appname', '-nodeName', 'nodename', '-serverName', 'servername'])  
AdminTask.removeEventServiceMdb (['-applicationName',  
'appname', '-clusterName', 'clustername'])
```

対話モードの使用例:

- Jacl を使用する場合:

```
$AdminTask removeEventServiceMdb {-interactive}
```

- Jython スtringを使用する場合:

```
AdminTask.removeEventServiceMdb('[-interactive]')
```

- Jython リストを使用する場合:



```
AdminTask.removeEventServiceMdb(['-interactive'])
```

removeEventServiceDB2DB コマンド

removeEventServiceDB2DB コマンドを使用して、イベント・サービスを除去します。また、関連付けられた DB2 イベント・データベースをオプションで除去します。

目的

removeEventServiceDB2DB コマンドは、AdminTask オブジェクトに使用可能な Common Event Infrastructure 管理コマンドです。DB2 のイベント・サービス・データベースおよびデータ・ソースをサーバーまたはクラスターから除去するには、このコマンドを使用します。AdminTask オブジェクトの詳細については、WebSphere Application Server Network Deployment バージョン 6.1 の資料を参照してください。

注:   この製品は、Microsoft Windows 2003、Windows Vista および Windows 7 オペレーティング・システムをサポートしない Jython バージョンを使用します。

パラメーター

- removeDB

このパラメーターを true に設定すると、データベースが除去されます。 false

に設定すると、データベースは除去されません。データベースを除去するには、現在のサーバーが、データベース・コマンドを実行するように既に構成済みである必要があります。

- **nodeName**

イベント・サービス・データ・ソースを除去する必要があるサーバーを含むノードの名前。このパラメーターを指定する場合は、**serverName** パラメーターも設定する必要があります。**clusterName** パラメーターを指定する場合は、このパラメーターを指定しないでください。

- **serverName**

イベント・サービス・データ・ソースを除去する必要があるサーバーの名前。**nodeName** パラメーターを指定せずにこのパラメーターを指定した場合、コマンドは、現在の WebSphere プロファイルのノード名を使用します。**clusterName** パラメーターを指定する場合は、このパラメーターを指定しないでください。

- **clusterName**

イベント・サービス・データ・ソースを除去する必要があるクラスターの名前。このパラメーターを指定する場合は、**serverName** パラメーターと **nodeName** パラメーターを設定しないでください。**serverName** パラメーターと **nodeName** パラメーターを指定する場合は、このパラメーターを指定しないでください。

- **dbUser**

データベースを作成および除去する特権を持つ DB2 ユーザー ID。このパラメーターは、**removeDB** パラメーターを **true** に設定した場合、指定する必要があります。指定しない場合、デフォルト値は **db2inst1** です。

- **dbPassword**

DB2 パスワード。このパラメーターは、**removeDB** パラメーターを **true** に設定した場合、指定する必要があります。

- **dbScriptDir**

イベント・サービス・データベースの構成コマンドで生成されたデータベース・スクリプトを含むディレクトリー。指定した場合、コマンドは、このディレクトリー内のスクリプトを実行して、イベント・サービス・データベースを除去します。データベース・スクリプトのデフォルトの出力ディレクトリーは、*profile_root/databases/event/node/server/dbscripts/db2* です。

サンプル

バッチ・モードの使用例:

- Jacl を使用する場合:

```
$AdminTask removeEventServiceDB2DB {-removeDB true
  -nodeName nodename
  -serverName servername
  -dbUser db2inst1
  -dbPassword dbpassword }
```

- Jython ストリングを使用する場合:

```
AdminTask.removeEventServiceDB2DB(['-removedB true
-nodeName nodename
-serverName servername
-dbUser db2inst1
-dbPassword dbpassword'])
```

- Jython リストを使用する場合:

```
AdminTask.removeEventServiceDB2DB(['-removedB', 'true',
'-nodeName', 'nodename',
'-serverName', 'servername',
'-dbUser', 'db2inst1',
'-dbPassword', 'dbpassword'])
```

対話モードの使用例:

- Jacl を使用する場合:

```
$AdminTask removeEventServiceDB2DB -interactive
```

- Jython スtringを使用する場合:

```
AdminTask.removeEventServiceDB2DB(['-interactive'])
```

- Jython リストを使用する場合:

```
AdminTask.removeEventServiceDB2DB(['-interactive'])
```

removeEventServiceDB2iSeriesDB コマンド

removeEventServiceDB2iSeriesDB コマンドを使用して、イベント・サービスを除去します。また、関連付けられた DB2 for iSeries イベント・データベースをオプションで除去します。

目的

removeEventServiceDB2iSeriesDB コマンドは、AdminTask オブジェクトで使用できる Common Event Infrastructure 管理コマンドです。このコマンドは、サーバーまたはクラスターから DB2 for iSeries データ・ソースを除去するために使用します。AdminTask オブジェクトの詳細については、WebSphere Application Server Network Deployment バージョン 6.1 の資料を参照してください。

注: **Vista** **Windows 7** この製品は、Microsoft Windows 2003、Windows Vista および Windows 7 オペレーティング・システムをサポートしない Jython バージョンを使用します。

パラメーター

- nodeName

イベント・サービス・データ・ソースを除去する必要があるサーバーを含むノードの名前。このパラメーターを指定する場合は、**serverName** パラメーターも設定する必要があります。**clusterName** パラメーターを指定する場合は、このパラメーターを指定しないでください。

- serverName

イベント・サービス・データ・ソースを除去する必要があるサーバーの名前。**nodeName** パラメーターを指定せずにこのパラメーターを指定した場合、コマ

ンドは、現在の WebSphere プロファイルのノード名を使用します。
clusterName パラメーターを指定する場合は、このパラメーターを指定しないでください。

- **clusterName**

イベント・サービス・データ・ソースを除去する必要があるクラスターの名前。
このパラメーターを指定する場合は、**serverName** パラメーターと **nodeName** パラメーターを設定しないでください。 **serverName** パラメーターと **nodeName** パラメーターを指定する場合は、このパラメーターを指定しないでください。

サンプル

バッチ・モードの使用例:

- Jacl を使用する場合:

```
$AdminTask removeEventServiceDB2iSeriesDB {-nodeName nodename  
-serverName servername }
```

- Jython スtringを使用する場合:

```
AdminTask.removeEventServiceDB2iSeriesDB(['-nodeName nodename  
-serverName servername'])
```

- Jython リストを使用する場合:

```
AdminTask.removeEventServiceDB2iSeriesDB(['-nodeName', 'nodename',  
'-serverName', 'servername'])
```

対話モードの使用例:

- Jacl を使用する場合:

```
$AdminTask removeEventServiceDB2iSeriesDB -interactive
```

- Jython スtringを使用する場合:

```
AdminTask.removeEventServiceDB2iSeriesDB(['-interactive'])
```

- Jython リストを使用する場合:

```
AdminTask.removeEventServiceDB2iSeriesDB(['-interactive'])
```

removeEventServiceDB2ZOSDB コマンド

removeEventServiceDB2ZOSDB コマンドを使用して、イベント・サービスを除去します。また、関連付けられた DB2 for z/OS イベント・データベースをオプションで除去します。

目的

removeEventServiceDB2ZOSDB コマンドは、AdminTask オブジェクトに使用可能な Common Event Infrastructure 管理コマンドです。このコマンドは、サーバーまたはクラスターから DB2 z/OS 用のイベント・サービス・データベースとデータ・ソースを除去するために使用します。AdminTask オブジェクトの詳細については、WebSphere Application Server Network Deployment バージョン 6.1 の資料を参照してください。

注: **Vista** **Windows 7** この製品は、Microsoft Windows 2003、Windows Vista および Windows 7 オペレーティング・システムをサポートしない Jython バージョンを使用します。

パラメーター

- removeDB

このパラメーターを **true** に設定すると、データベースが除去されます。 **false** に設定すると、データベースは除去されません。データベースを除去するには、現在のサーバーが、データベース・コマンドを実行するように既に構成済みである必要があります。

- nodeName

イベント・サービス・データ・ソースを除去する必要があるサーバーを含むノードの名前。このパラメーターを指定する場合は、**serverName** パラメーターも設定する必要があります。 **clusterName** パラメーターを指定する場合は、このパラメーターを指定しないでください。

- serverName

イベント・サービス・データ・ソースを除去する必要があるサーバーの名前。
nodeName パラメーターを指定せずにこのパラメーターを指定した場合、コマンドは、現在の WebSphere プロファイルのノード名を使用します。
clusterName パラメーターを指定する場合は、このパラメーターを指定しないでください。

- clusterName

イベント・サービス・データ・ソースを除去する必要があるクラスターの名前。このパラメーターを指定する場合は、**serverName** パラメーターと **nodeName** パラメーターを設定しないでください。 **serverName** パラメーターと **nodeName** パラメーターを指定する場合は、このパラメーターを指定しないでください。

- dbName

DB2 データベース名。これは、DB2 クライアント・マシン上では、カタログされたデータベースの名前です。ネイティブ z/OS サーバー上では、データベース・サブシステムの名前です。このパラメーターは、**removeDB** パラメーターを **true** に設定した場合、指定する必要があります。指定しない場合、デフォルト値は **event** です。

- dbUser

データベースを作成および除去する特権を持つ DB2 ユーザー ID。このパラメーターは、**removeDB** パラメーターを **true** に設定した場合、指定する必要があります。

- dbPassword

DB2 パスワード。このパラメーターは、**removeDB** パラメーターを **true** に設定した場合、指定する必要があります。

- dbScriptDir

イベント・サービス・データベースの構成コマンドで生成されたデータベース・スクリプトを含むディレクトリ。指定した場合、コマンドは、このディレクトリ内のスクリプトを実行して、イベント・サービス・データベースを除去しま

す。データベース・スクリプトのデフォルトの出力ディレクトリーは、*profile_root/databases/event/node/server/dbscripts/db2zos* です。

サンプル

バッチ・モードの使用例:

- Jacl を使用する場合:

```
$AdminTask removeEventServiceDB2ZOSDB {-removeDB true -nodeName nodename  
-serverName servername -dbUser db2user -dbPassword dbpassword}
```

- Jython スtringを使用する場合:

```
AdminTask.removeEventServiceDB2ZOSDB(['-removeDB true -nodeName nodename  
-serverName servername -dbUser db2user -dbPassword dbpassword'])
```

- Jython リストを使用する場合:

```
AdminTask.removeEventServiceDB2ZOSDB(['-removeDB', 'true', '-nodeName',  
'nodename', '-serverName', 'servername', '-dbUser', 'db2user',  
'-dbPassword', 'dbpassword'])
```

対話モードの使用例:

- Jacl を使用する場合:

```
$AdminTask removeEventServiceDB2ZOSDB -interactive
```

- Jython スtringを使用する場合:

```
AdminTask.removeEventServiceDB2ZOSDB(['-interactive'])
```

- Jython リストを使用する場合:



```
AdminTask.removeEventServiceDB2ZOSDB(['-interactive'])
```

removeEventServiceDerbyDB コマンド

removeEventServiceDerbyDB コマンドを使用して、イベント・サービスを除去します。また、関連付けられた Derby イベント・データベースをオプションで除去します。

目的

removeEventServiceDerbyDB コマンドは、AdminTask オブジェクトに使用可能な Common Event Infrastructure 管理コマンドです。このコマンドは、サーバーまたはクラスターから Derby 用のイベント・サービス・データベースとデータ・ソースを除去するために使用します。AdminTask オブジェクトの詳細については、WebSphere Application Server Network Deployment バージョン 6.1 の資料を参照してください。

注:   この製品は、Microsoft Windows 2003、Windows Vista および Windows 7 オペレーティング・システムをサポートしない Jython バージョンを使用します。

パラメーター

- removeDB

このパラメーターを true に設定すると、データベースが除去されます。 false

に設定すると、データベースは除去されません。データベースを除去するには、現在のサーバーが、データベース・コマンドを実行するように既に構成済みである必要があります。

- **nodeName**

イベント・サービス・データ・ソースを除去する必要があるサーバーを含むノードの名前。このパラメーターを指定する場合は、**serverName** パラメーターも設定する必要があります。**clusterName** パラメーターを指定する場合は、このパラメーターを指定しないでください。

- **serverName**

イベント・サービス・データ・ソースを除去する必要があるサーバーの名前。**nodeName** パラメーターを指定せずにこのパラメーターを指定した場合、コマンドは、現在の WebSphere プロファイルのノード名を使用します。**clusterName** パラメーターを指定する場合は、このパラメーターを指定しないでください。

- **clusterName**

イベント・サービス・データ・ソースを除去する必要があるクラスターの名前。このパラメーターを指定する場合は、**serverName** パラメーターと **nodeName** パラメーターを設定しないでください。**serverName** パラメーターと **nodeName** パラメーターを指定する場合は、このパラメーターを指定しないでください。

- **dbScriptDir**

イベント・サービス・データベースの構成コマンドで生成されたデータベース・スクリプトを含むディレクトリ。指定した場合、コマンドは、このディレクトリ内のスクリプトを実行して、イベント・サービス・データベースを除去します。データベース・スクリプトのデフォルトの出力ディレクトリは、*profile_root/databases/event/node/server/dbscripts/derby* です。

サンプル

バッチ・モードの使用例:

- Jacl を使用する場合:

```
$AdminTask removeEventServiceDerbyDB {-removeDB true -nodeName nodename  
-serverName servername}
```

- Jython スtringを使用する場合:

```
AdminTask.removeEventServiceDerbyDB(['-removeDB true -nodeName nodename  
-serverName servername'])
```

- Jython リストを使用する場合:

```
AdminTask.removeEventServiceDerbyDB(['-removeDB', 'true', '-nodeName',  
'nodename', '-serverName', 'servername'])
```

対話モードの使用例:

- Jacl を使用する場合:

```
$AdminTask removeEventServiceDerbyDB -interactive
```

- Jython スtringを使用する場合:

```
AdminTask.removeEventServiceDerbyDB( ['-interactive'])
```

- Jython リストを使用する場合:

removeEventServiceInformixDB コマンド

removeEventServiceInformixDB コマンドを使用して、イベント・サービスを除去します。また、関連付けられた Informix イベント・データベースをオプションで除去します。

目的

removeEventServiceInformixDB コマンドは、AdminTask オブジェクトに使用可能な Common Event Infrastructure 管理コマンドです。Informix のイベント・サービス・データベースおよびデータ・ソースをサーバーまたはクラスターから除去するには、このコマンドを使用します。AdminTask オブジェクトの詳細については、WebSphere Application Server Network Deployment バージョン 6.1 の資料を参照してください。

注: **Vista** **Windows 7** この製品は、Microsoft Windows 2003、Windows Vista および Windows 7 オペレーティング・システムをサポートしない Jython バージョンを使用します。

パラメーター

- removeDB

このパラメーターを true に設定すると、データベースが除去されます。false に設定すると、データベースは除去されません。データベースを除去するには、現在のサーバーが、データベース・コマンドを実行するように既に構成済みである必要があります。

- nodeName

イベント・サービス・データ・ソースを除去する必要があるサーバーを含むノードの名前。このパラメーターを指定する場合は、**serverName** パラメーターも設定する必要があります。**clusterName** パラメーターを指定する場合は、このパラメーターを指定しないでください。

- serverName

イベント・サービス・データ・ソースを除去する必要があるサーバーの名前。**nodeName** パラメーターを指定せずにこのパラメーターを指定した場合、コマンドは、現在の WebSphere プロファイルのノード名を使用します。**clusterName** パラメーターを指定する場合は、このパラメーターを指定しないでください。

- clusterName

イベント・サービス・データ・ソースを除去する必要があるクラスターの名前。このパラメーターを指定する場合は、**serverName** パラメーターと **nodeName** パラメーターを設定しないでください。**serverName** パラメーターと **nodeName** パラメーターを指定する場合は、このパラメーターを指定しないでください。

- dbScriptDir

イベント・サービス・データベースの構成コマンドで生成されたデータベース・スクリプトを含むディレクトリ。指定した場合、コマンドは、このディレクトリ

リー内のスクリプトを実行して、イベント・サービス・データベースを除去します。データベース・スクリプトのデフォルトの出力ディレクトリーは、`profile_root/databases/event/node/server/dbscripts/informix` です。

サンプル

バッチ・モードの使用例:

- Jacl を使用する場合:

```
$AdminTask removeEventServiceInformixDB {-removeDB true -nodeName nodename  
-serverName servername}
```

- Jython スtringを使用する場合:

```
AdminTask.removeEventServiceInformixDB(['-removeDB true -nodeName nodename  
-serverName servername'])
```

- Jython リストを使用する場合:

```
AdminTask.removeEventServiceInformixDB(['-removeDB', 'true', '-nodeName',  
'nodename', '-serverName', 'servername'])
```

対話モードの使用例:

- Jacl を使用する場合:

```
$AdminTask removeEventServiceInformixDB -interactive
```

- Jython スtringを使用する場合:

```
AdminTask.removeEventServiceInformixDB(['-interactive'])
```

- Jython リストを使用する場合:



```
AdminTask.removeEventServiceInformixDB(['-interactive'])
```

removeEventServiceOracleDB コマンド

`removeEventServiceOracleDB` コマンドを使用して、イベント・サービスを除去します。また、関連付けられた Oracle イベント・データベースをオプションで除去します。

目的

`removeEventServiceOracleDB` コマンドは、AdminTask オブジェクトに使用可能な Common Event Infrastructure 管理コマンドです。Oracle のイベント・サービス・テーブルおよびデータ・ソースをサーバーまたはクラスターから除去するには、このコマンドを使用します。このコマンドは、データベースを除去しません。AdminTask オブジェクトの詳細については、WebSphere Application Server Network Deployment バージョン 6.1 の資料を参照してください。

注:   この製品は、Microsoft Windows 2003、Windows Vista および Windows 7 オペレーティング・システムをサポートしない Jython バージョンを使用します。

パラメーター

- removeDB

このパラメーターを `true` に設定すると、イベント・サービス・テーブルが除去されます。 `false` に設定すると、テーブルは除去されません。

- **nodeName**

イベント・サービス・データ・ソースを除去する必要があるサーバーを含むノードの名前。このパラメーターを指定する場合は、**serverName** パラメーターも設定する必要があります。**clusterName** パラメーターを指定する場合は、このパラメーターを指定しないでください。

- **serverName**

イベント・サービス・データ・ソースを除去する必要があるサーバーの名前。**nodeName** パラメーターを指定せずにこのパラメーターを指定した場合、コマンドは、現在の WebSphere プロファイルのノード名を使用します。**clusterName** パラメーターを指定する場合は、このパラメーターを指定しないでください。

- **clusterName**

イベント・サービス・データ・ソースを除去する必要があるクラスターの名前。このパラメーターを指定する場合は、**serverName** パラメーターと **nodeName** パラメーターを設定しないでください。**serverName** パラメーターと **nodeName** パラメーターを指定する場合は、このパラメーターを指定しないでください。

- **sysUser**

Oracle データベース・システム・ユーザー ID。指定しない場合、デフォルト値は `sys` です。

- **sysPassword**

`sysUser` パラメーターで指定したユーザーのパスワード。

- **dbScriptDir**

イベント・サービス・データベースの構成コマンドで生成されたデータベース・スクリプトを含むディレクトリー。指定した場合、コマンドは、このディレクトリー内のスクリプトを実行して、イベント・サービス・データベースを除去します。データベース・スクリプトのデフォルトの出力ディレクトリーは、`profile_root/databases/event/node/server/dbscripts/oracle` です。

サンプル

バッチ・モードの使用例:

- Jacl を使用する場合:

```
$AdminTask removeEventServiceOracleDB {-removeDB true -nodeName nodename  
-serverName servername -sysUser sys -sysPassword syspassword}
```

- Jython スtringを使用する場合:

```
AdminTask.removeEventServiceOracleDB(['-removeDB true -nodeName nodename  
-serverName servername -sysUser sys -sysPassword syspassword'])
```

- Jython リストを使用する場合:

```
AdminTask.removeEventServiceOracleDB(['-removeDB', 'true', '-nodeName',  
'nodename', '-serverName', 'servername', '-sysUser', 'sys',  
'-sysPassword', 'syspassword'])
```

対話モードの使用例:

- Jacl を使用する場合:

```
$AdminTask removeEventServiceOracleDB -interactive
```

- Jython スtringを使用する場合:

```
AdminTask.removeEventServiceOracleDB(['-interactive'])
```

- Jython リストを使用する場合:

```
AdminTask.removeEventServiceOracleDB(['-interactive'])
```

removeEventServiceSQLServerDB コマンド

removeEventServiceSQLServerDB コマンドを使用して、イベント・サービスを除去します。また、関連付けられた SQL Server イベント・データベースをオプションで除去します。

目的

removeEventServiceSQLServerDB コマンドは、AdminTask オブジェクトに使用可能な Common Event Infrastructure 管理コマンドです。SQL Server のイベント・サービス・データベースおよびデータ・ソースをサーバーまたはクラスターから除去するには、このコマンドを使用します。AdminTask オブジェクトの詳細については、WebSphere Application Server Network Deployment バージョン 6.1 の資料を参照してください。

注: **Vista** **Windows 7** この製品は、Microsoft Windows 2003、Windows Vista および Windows 7 オペレーティング・システムをサポートしない Jython バージョンを使用します。

パラメーター

- removeDB

このパラメーターを true に設定すると、データベースが除去されます。false に設定すると、データベースは除去されません。データベースを除去するには、現在のサーバーが、データベース・コマンドを実行するように既に構成済みである必要があります。

- nodeName

イベント・サービス・データ・ソースを除去する必要があるサーバーを含むノードの名前。このパラメーターを指定する場合は、**serverName** パラメーターも設定する必要があります。**clusterName** パラメーターを指定する場合は、このパラメーターを指定しないでください。

- serverName

イベント・サービス・データ・ソースを除去する必要があるサーバーの名前。**nodeName** パラメーターを指定せずにこのパラメーターを指定した場合、コマンドは、現在の WebSphere プロファイルのノード名を使用します。**clusterName** パラメーターを指定する場合は、このパラメーターを指定しないでください。

- clusterName

イベント・サービス・データ・ソースを除去する必要があるクラスターの名前。このパラメーターを指定する場合は、**serverName** パラメーターと **nodeName** パラメーターを設定しないでください。**serverName** パラメーターと **nodeName** パラメーターを指定する場合は、このパラメーターを指定しないでください。

- **dbServerName**
SQL サーバー・データベースのサーバー名。このパラメーターは、**removeDB** パラメーターを **true** に設定した場合、指定する必要があります。
- **dbUser**
イベント・サービス・テーブルを所有する SQL サーバー・ユーザー ID。指定しない場合、デフォルト値は **ceiuser** です。
- **saUser**
データベースとユーザーを除去する特権を持つユーザー ID。指定しない場合、デフォルト値は **sa** です。
- **saPassword**
saUser パラメーターで指定したユーザーのパスワード。このパラメーターは、**removeDB** パラメーターを **true** に設定した場合、必須です。
- **dbScriptDir**
イベント・サービス・データベースの構成コマンドで生成されたデータベース・スクリプトを含むディレクトリー。指定した場合、コマンドは、このディレクトリー内のスクリプトを実行して、イベント・サービス・データベースを除去します。データベース・スクリプトのデフォルトの出力ディレクトリーは、*profile_root/databases/event/node/server/dbscripts/sqlserver* です。

サンプル

バッチ・モードの使用例:

- Jacl を使用する場合:

```
$AdminTask removeEventServiceSQLServerDB {-removeDB true -nodeName nodename
-serverName servername -dbUser ceiuser -saUser sa -saPassword sapassword
-dbServerName sqlservername}
```
- Jython スtringを使用する場合:

```
AdminTask.removeEventServiceSQLServerDB(['-removeDB true -nodeName nodename
-serverName servername -dbUser ceiuser -saUser sa -saPassword sapassword
-dbServerName sqlservername'])
```
- Jython リストを使用する場合:

```
AdminTask.removeEventServiceSQLServerDB(['-removeDB', 'true', '-nodeName',
'nodename', '-serverName', 'servername', '-dbUser', 'ceiuser', '-saUser', 'sa',
'-saPassword', 'sapassword', '-dbServerName', 'sqlservername'])
```

対話モードの使用例:

- Jacl を使用する場合:

```
$AdminTask removeEventServiceSQLServerDB -interactive
```
- Jython スtringを使用する場合:

```
AdminTask.removeEventServiceSQLServerDB(['-interactive'])
```
- Jython リストを使用する場合:

```
AdminTask.removeEventServiceSQLServerDB(['-interactive'])
```

eventbucket コマンド行ユーティリティー

eventbucket コマンドは、イベント・データベースのバケット構成を表示または変更するコマンドです。

目的

イベント・データベースのバケット構成を表示または変更します。

eventbucket [-status] [-change]

説明

eventbucket コマンドは、イベント・データベースのバケット構成を表示または変更するコマンドです。バケットは、古いイベント・データをイベント・データベースから消去するために、急速ページ・ユーティリティーが使用します。このコマンド行を実行することにより、現在のバケット構成を確認することや、アクティブ・バケットと非アクティブ・バケットを交換することができます。

セキュリティ: WebSphere セキュリティーを有効にしている場合は、ユーザー ID を `eventAdministrator` 役割にマップして、イベント・データベースのバケット構成を表示または変更する必要があります。

パラメーター

-status

アクティブなバケット設定やバケット確認の間隔 (どのバケットがアクティブであるかを調べるためにデータ・ストア・プラグインが検査する頻度) など、現在のバケット構成に関する情報を表示します。

-change

バケットを交換して、アクティブ・バケットが非アクティブになり、非アクティブ・バケットがアクティブになるようにします。非アクティブ・バケットは空でないと、このオプションは使用できません。

例

次の例では、現在のバケット構成が表示されます。

```
eventbucket -status
```

次の例では、アクティブ・バケットと非アクティブ・バケットが交換されます。

```
eventbucket -change
```

eventpurge コマンド行ユーティリティー

eventpurge コマンドは、イベント・データベースからイベントを削除します。

目的

イベント・データベースからイベントを削除します。

eventpurge [-seconds *seconds* | -end *end_time*] [-group *event_group*] [-severity *severity*] [-extensionname *extension_name*] [-start *start_time*] [-size *size*]

説明

eventpurge コマンドは、イベント・データベースからイベントを削除します。イベント・データベースからすべてのイベントを削除することも、特定の基準を満たすイベントのみを削除することもできます。

セキュリティー: WebSphere セキュリティーが有効の場合、イベントを削除するには、ユーザー ID を eventAdministrator 役割にマップする必要があります。

パラメーター

-seconds *seconds*

この最小経過期間を過ぎたイベントが削除されます。 *seconds* 値は、整数でなければなりません。指定した秒数を越えたイベントのみ削除されます。このパラメーターは、**-end** パラメーターを指定しない場合に必要です。

-end *end_time*

削除するイベント・グループの終了時刻。指定した時刻の前に生成されたイベントのみ削除されます。 *end_time* 値は、XML dateTime 形式 (CCYY-MM-DDThh:mm:ss) で指定する必要があります。例えば、東部標準時の 2006 年 1 月 1 日の正午は、2006-01-01T12:00:00-05:00 のようになります。dateTime データ・タイプについて詳しくは、www.w3.org の XML スキーマに関する説明を参照してください。

このパラメーターは、**-seconds** パラメーターを指定しない場合に必要です。

-group *eventGroup*

イベントの削除元のイベント・グループ。 *event_group* 値は、Common Event Infrastructure 構成に定義されたイベント・グループの名前でなければなりません。このパラメーターはオプションです。

-severity *severity*

削除するイベントの重大度。 *severity* 値は、整数でなければなりません。指定した値と同じ重大度を持つイベントのみ削除されます。このパラメーターはオプションです。

-extensionname *extension_name*

削除の対象にするイベントの拡張子名。このパラメーターは、特定のタイプのイベントのみを削除する場合に使用します。 *extensionName* プロパティが *extensionName* に等しいイベントのみが削除されます。このパラメーターはオプションです。

-start *start_time*

削除するイベント・グループの開始時刻。指定した時刻の後に生成されたイベントのみ削除されます。 *start_time* 値は、XML dateTime 形式 (CCYY-MM-DDThh:mm:ss) で指定する必要があります。このパラメーターはオプションです。

-size *size*

1 回のトランザクションで削除するイベントの数。 *size* 値は、整数でなければなりません。イベントがこの個数だけ削除されると、トランザクションがコマンドによってコミットされ、その後、新しいトランザクションが実行されます。このパラメーターはオプションです。

例

以下の例では、重大度が 20 (harmless) のイベントのうち、10 分前より前の時点で生成されたイベントがデータベースからすべて削除されます。

```
eventpurge -group "All events" -severity 20 -seconds 600
```



Printed in Japan