

WebSphere IBM WebSphere Process Server for Multiplatforms
Version 7.0.0

Common Event Infrastructure



WebSphere IBM WebSphere Process Server for Multiplatforms
Version 7.0.0

Common Event Infrastructure



April 2010

Diese Veröffentlichung ist eine Übersetzung des Handbuchs
IBM WebSphere Process Server for Multiplatforms Version 7.0.0, Common Event Infrastructure,
herausgegeben von International Business Machines Corporation, USA

© Copyright International Business Machines Corporation 2007, 2010
© Copyright IBM Deutschland GmbH 2010

Informationen, die nur für bestimmte Länder Gültigkeit haben und für Deutschland, Österreich und die Schweiz nicht zutreffen, wurden in dieser Veröffentlichung im Originaltext übernommen.

Möglicherweise sind nicht alle in dieser Übersetzung aufgeführten Produkte in Deutschland angekündigt und verfügbar; vor Entscheidungen empfiehlt sich der Kontakt mit der zuständigen IBM Geschäftsstelle.

Änderung des Textes bleibt vorbehalten.

Herausgegeben von:
SW TSC Germany
Kst. 2877
April 2010

Inhaltsverzeichnis

Kapitel 1. Common Event Infrastructure. 1

Common Base Event-Modell 3

Kapitel 2. Common Event Infrastructure konfigurieren 5

Komponenten von Common Event Infrastructure (CEI) 5

Common Event Infrastructure mit der Administrationskonsole konfigurieren 7

Common Event Infrastructure-Anwendung implementieren 9

 Common Event Infrastructure in einem Cluster implementieren 10

Ereignis-Messaging konfigurieren 13

 Zusätzliche JMS-Warteschlangen konfigurieren 13

 Ereignis-Messaging über einen externen JMS-Provider konfigurieren 13

 JMS-Authentifizierungsaliasnamen konfigurieren 15

Ereignisdatenbank konfigurieren 16

 Einschränkungen für die Ereignisdatenbank 16

 Derby-Ereignisdatenbank konfigurieren 17

 DB2-Ereignisdatenbank konfigurieren (Linux-, UNIX- und Windows-Systeme) 19

 DB2-Datenbank auf einem z/OS-System konfigurieren 20

 DB2-Datenbank auf einem iSeries-System konfigurieren 22

 Informix-Ereignisdatenbank konfigurieren 24

 Oracle-Ereignisdatenbank konfigurieren 26

 SQL Server-Ereignisdatenbank konfigurieren 27

 Manuelle Ausführung der Scripts zur Datenbankkonfiguration 29

 Upgrade einer Ereignisdatenbank aus einer früheren Version durchführen 36

Zellübergreifende Common Event Infrastructure-Konfiguration für WebSphere Business Monitor 41

Common Event Infrastructure-Konfiguration entfernen 41

 Ereignisdatenbank entfernen 42

 Common Event Infrastructure-Anwendung entfernen 42

 Ereignis-Messaging aus dem Common Event Infrastructure-Server entfernen 43

Kapitel 3. Common Event Infrastructure verwalten 45

Events Service mit der Administrationskonsole verwalten 45

 Events Service mit der Administrationskonsole aktivieren und inaktivieren 45

 Ereignisemitter-Factory mit der Administrationskonsole erstellen 46

 Ereignisgruppe mit der Administrationskonsole erstellen 46

 Ereignisfilter mit der Administrationskonsole erstellen 47

Events Service über Scripting verwalten 48

 Events Service über Scripting aktivieren 48

 Events Service über Scripting inaktivieren 49

 Emitter-Factory über Scripting erstellen 50

 Ereignisgruppe über Scripting erstellen 52

 Ereignisfilter über Scripting erstellen 54

Protokollierung und Traceerstellung für Common Event Infrastructure-Komponenten 55

Ereignisdatenbank verwalten 56

 Statistiken der DB2-Ereignisdatenbank aktualisieren 56

 Tabellen der DB2-Ereignisdatenbank reorganisieren 57

 Ereignisse aus der Ereignisdatenbank löschen 58

Kapitel 4. Sicherer Zugriff auf Common Event Infrastructure-Funktionen 65

Kapitel 5. Fehlerbehebung für Common Event Infrastructure 69

Fehler beim Start 69

 Events Service startet nicht (Nachricht CEIDS0058E) 69

Fehler beim Senden von Ereignissen 69

 Fehler beim Senden eines Ereignisses (Nachricht CEIDS0060E) 69

 Fehler beim Senden eines Ereignisses (ServiceUnavailableException) 70

 Fehler beim Senden eines Ereignisses (NameNotFoundException) 71

 Fehler beim Senden eines Ereignisses (Nachricht CEIEM0025E) 72

 Fehler beim Senden eines Ereignisses (Nachricht CEIEM0034E) 72

 Ereignis ist ungültig (Nachricht CEIEM0027E) 73

 Synchronisationsmodus wird nicht unterstützt (Nachricht CEIEM0015E) 73

 Transaktionsmodus wird nicht unterstützt (Nachricht CEIEM0016E) 75

Fehler beim Empfangen oder Abfragen von Ereignissen 75

 Fehler beim Abfragen von Ereignissen (Nachricht CEIDS0060E) 75

 Ereignisse werden nicht im persistenten Datenspeicher gespeichert 76

 Ereignisse werden von Konsumenten nicht empfangen (keine Fehlernachricht) 77

 Ereignisse werden von Konsumenten nicht empfangen (NameNotFoundException) 79

 Ereignisgruppe mit erweiterten Datenelementen enthält keine Ereignisse 79

 Fehler beim Abfragen einer Ereignisgruppe (Nachricht CEIES0048E) 80

Sonstige Fehler	81
Abfrage für Ereigniskatalogmuster schlägt auf Windows-System fehl	81

Kapitel 6. Common Event Infrastructure-Befehle 83

Befehl 'configEventServiceDB2DB'	83
Befehl 'configEventServiceDB2iSeriesDB'	86
Befehl 'configEventServiceDB2ZOSDB'	88
Befehl 'configEventServiceDerbyDB'	91
Befehl 'configEventServiceInformixDB'	93
Befehl 'configEventServiceOracleDB'	95
Befehl 'configEventServiceSQLServerDB'	98
Befehl 'deployEventService'	101
Befehl 'deployEventServiceMdb'	102

Befehl 'setEventServiceJmsAuthAlias'	104
Befehl 'enableEventService'	105
Befehl 'disableEventService'	106
Befehl 'showEventServiceStatus'	107
Befehl 'removeEventService'	109
Befehl 'removeEventServiceMdb'	110
Befehl 'removeEventServiceDB2DB'	111
Befehl 'removeEventServiceDB2iSeriesDB'	113
Befehl 'removeEventServiceDB2ZOSDB'	114
Befehl 'removeEventServiceDerbyDB'	115
Befehl 'removeEventServiceInformixDB'	117
Befehl 'removeEventServiceOracleDB'	118
Befehl 'removeEventServiceSQLServerDB'	119
Befehlszeilendienstprogramm 'eventbucket'	121
Befehlszeilendienstprogramm 'eventpurge'	122

Kapitel 1. Common Event Infrastructure.

Common Event Infrastructure ist eine integrierbare Technologie, die grundlegende Services zur Ereignisverwaltung für Anwendungen bereitstellt, die solche Services benötigen.

Diese Ereignisinfrastuktur dient als Integrationspunkt für die Konsolidierung und Persistenz unformatierter Ereignisse von mehreren heterogenen Quellen und für die Verteilung dieser Ereignisse an Ereigniskonsumenten. Dabei werden Ereignisse anhand des Common Base Event-Modells dargestellt, eines standardisierten, XML-basierten Formats, das die Struktur eines Ereignisses definiert. Weitere Informationen finden Sie im Unterabschnitt über das Common Base Event-Modell.

Bei Verwendung dieser einheitlichen Infrastruktur können unterschiedliche Produkte, die nicht eng miteinander verknüpft sind, ihre Ereignisverwaltung integrieren und eine umfassende Übersicht der Unternehmensressourcen und der korrelierenden Ereignisse über Domänengrenzen hinweg bereitstellen. Beispielsweise können Ereignisse, die von einer Netzüberwachungsanwendung generiert wurden, mit Ereignissen korreliert werden, die von einer Sicherheitsanwendung generiert wurden. Eine solche Korrelation ist nur schwer zu bewerkstelligen, wenn jedes Produkt ein eigenes Konzept zur Ereignisverwaltung verwendet.

Common Event Infrastructure stellt Funktionen für die Generierung, Weitergabe, Persistenz und Nutzung von Ereignissen bereit, ohne die Ereignisse an sich zu definieren. Statt dessen definieren die Anwendungsentwickler und Administratoren Ereignistypen, Ereignisgruppen, Filterung und Korrelation.

Komponenten von Common Event Infrastructure (CEI)

Common Event Infrastructure besteht aus den folgenden Hauptkomponenten:

Common Base Event

Diese Komponente unterstützt das Erstellen von Ereignissen und den Zugriff auf die zugehörigen Eigenschaftsdaten. Ereignisquellen nutzen die Common Base Event-APIs zum Erstellen neuer Ereignisse in Übereinstimmung mit dem Common Base Event-Modell; Ereigniskonsumenten verwenden die APIs zum Lesen von Eigenschaftsdaten aus empfangenen Ereignissen. Zusätzlich können Anwendungen Ereignisse aus dem und in das XML-Textformat konvertieren, um den Austausch mit anderen Tools zu unterstützen. Die Komponente Common Base Event ist Bestandteil der Eclipse-Plattform für Test- und Leistungstools (Test and Performance Tools Platform, TPTP).

Emitter

Die Emitterkomponente unterstützt das Senden von Ereignissen. Nachdem ein Ereignis von einer Ereignisquelle erstellt und mit Daten gefüllt wurde, übergibt die Ereignisquelle das Ereignis an einen Emitter. Der Emitter vervollständigt den Inhalt automatisch (optionaler Vorgang) und prüft das Ereignis, um sicherzustellen, dass es mit der Common Base Event-Spezifikation übereinstimmt. Außerdem vergleicht der Emitter das Ereignis mit konfigurierbaren Filterkriterien.

Wenn das Ereignis gültig ist und die Filterkriterien erfüllt, sendet der Emittter das Ereignis zum Ereignisservice. Ein Emittter kann Ereignisse entweder synchron zum Ereignisservice senden (mit Hilfe von Enterprise JavaBeans™-Aufrufen) oder asynchron (mit Hilfe einer Java™ Message Service-Warteschlange).

Ereignisservice

Der Ereignisservice ist der Übertragungsweg zwischen Ereignisquellen und Ereigniskonsumenten. Der Ereignisservice empfängt Ereignisse, die von Ereignisquellen an Emittter übergeben werden. Er speichert Ereignisse in einem permanenten Datenspeicher und verteilt sie asynchron an subskribierte Ereigniskonsumenten. Außerdem unterstützt der Ereignisservice synchrone Abfragen für aufgezeichnete Ereignisse im permanenten Speicher.

Ereigniskatalog

Der Ereigniskatalog ist ein Repository für Ereignismetadaten. Anwendungen nutzen den Ereigniskatalog zum Abrufen von Informationen zu Ereignisklassen und des zulässigen Inhalts dieser Klassen.

Außerdem kann eine Anwendung oder Lösung, die Common Event Infrastructure verwendet, die folgenden Komponenten beinhalten, die nicht Teil der Infrastruktur sind:

Ereignisquelle

Eine Ereignisquelle ist eine beliebige Anwendung, die mit Hilfe eines Emitters Ereignisse zum Ereignisservice sendet.

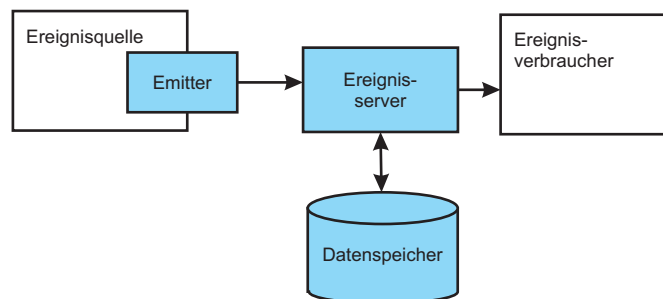
Ereigniskonsument

Ein Ereigniskonsument ist eine Anwendung, die Ereignisse vom Ereignisservice empfängt.

Ereigniskataloganwendung

Eine Ereigniskataloganwendung ist eine Anwendung, die Ereignismetadaten im Ereigniskatalog speichert oder von dort abruft. Dies kann ein Verwaltungs- oder Entwicklungstool sein bzw. eine Ereignisquelle oder ein Ereigniskonsument.

Das folgende Diagramm zeigt den allgemeinen Ereignisfluss von der Ereignisquelle zum Ereigniskonsumenten unter Verwendung der Common Event Infrastructure.



Common Base Event-Modell

Das Common Base Event-Modell ist ein Standard, der eine einheitliche Darstellung von Ereignissen für Unternehmensverwaltungs- und Geschäftsanwendungen definiert. Dieser vom IBM[®] Autonomic Computing Architecture Board entwickelte Standard unterstützt die Codierung von Protokollierungs-, Tracereinstellungs-, Verwaltungs- und Business-Ereignissen unter Verwendung eines einheitlichen XML-basierten Formats, wodurch eine Korrelierung verschiedener Ereignistypen aus unterschiedlichen Anwendungen ermöglicht wird. Das Common Base Event-Modell ist Bestandteil des IBM Autonomic Computing Toolkit. Weitere Informationen hierzu finden Sie unter <http://www.ibm.com/autonomic>.

Common Event Infrastructure unterstützt derzeit Version 1.0.1 der Spezifikation.

Das grundlegende Konzept hinter dem Common Base Event-Modell ist die *Situation*. Eine Situation ist ein Vorfall, der sich irgendwo innerhalb der IT-Infrastruktur ereignet, z. B. das Herunterfahren eines Servers, ein Plattenlaufwerkfehler oder eine fehlgeschlagene Benutzeranmeldung. Das Common Base Event-Modell definiert eine Reihe von Standardsituationstypen, die einen Großteil der möglichen Situationen abdecken (z. B. StartSituation und CreateSituation).

Ein *Ereignis* ist eine strukturierte Benachrichtigung mit situationsbezogenen Informationen. Ein Ereignis dokumentiert drei Arten von Informationen:

- Informationen zur Situation selbst (welcher Vorfall eingetreten ist)
- Informationen zur Identität der betroffenen Komponente (z. B. der Server, der heruntergefahren wurde)
- Informationen zur Identität der Komponente, von der die Situation gemeldet wird (kann mit der betroffenen Komponente identisch sein)

Die Common Base Event-Spezifikation definiert ein Ereignis als XML-Element mit Merkmalen, die alle drei Arten von Informationen bereitstellen. Diese Merkmale sind als Attribute und Unterelemente des Stammelements (CommonBaseEvent) codiert.

Das Common Base Event-Format ist erweiterbar. Zusätzlich zu den standardmäßig definierten Ereignismerkmalen kann ein Ereignis auch erweiterte Datenelemente enthalten. Bei diesen Elementen handelt es sich um anwendungsspezifische Elemente, die beliebige situationsrelevante Informationen enthalten können. Das Attribut *extensionName* bezeichnet ein Ereignis mit einem optionalen Klassifikationsnamen (Ereignisklasse), das Anwendungen darüber informiert, welche Arten von erweiterten Datenelementen zu erwarten sind. Im Ereigniskatalog werden Ereignisdefinitionen gespeichert, die diese Ereignisklassen und die zulässigen Inhalte beschreiben.

Eine umfassende Beschreibung des Common Base Event-Formats enthält das Spezifikationsdokument und das XSD-Schema, die im Lieferumfang des IBM Autonomic Computing Toolkit enthalten sind.

Kapitel 2. Common Event Infrastructure konfigurieren

Common Event Infrastructure-Ressourcen können mit dem Objekt 'AdminTask' für Server konfiguriert werden.

Informationen zu diesem Vorgang

Common Event Infrastructure (CEI) kann mit einer Standardkonfiguration installiert werden, die auf einer eigenständigen Serverkonfiguration den vollen Funktionsumfang bietet. Sie würden diese Konfiguration jedoch nur anwenden, um ein eigenständiges Serverprofil mit dem Profile Management Tool erstellen. In allen übrigen Fällen sollten Sie die Administrationskonsole zum Konfigurieren von CEI verwenden (z. B. wenn die Installation in einer Netzimplementierungsumgebung oder in einem Cluster erfolgt), um sicherzustellen, dass die Konfiguration Ihrem System entspricht.

Sie können auch den Befehl 'wsadmin' zum Konfigurieren von CEI oder zum Ändern einer vorhandenen CEI-Konfiguration verwenden. In beiden Fällen würden Sie die Konfiguration von CEI unter Verwendung des Objekts 'AdminTask' für Server zum Ausführen von Verwaltungsbefehlen ändern.

Nachdem Sie die CEI-Konfiguration geändert haben, müssen Sie einen Neustart des Servers oder Clusters durchführen.

Komponenten von Common Event Infrastructure (CEI)

Die Komponenten von Common Event Infrastructure (CEI) werden als eine Gruppe von Anwendungen, Services und Ressourcen auf dem Server installiert.

Beim Konfigurieren von Common Event Infrastructure werden eine Anzahl von Komponenten erstellt und auf Ihrem Server implementiert.

Common Event Infrastructure-Service

Hierbei handelt es sich um einen im Server installierten Service, der Anwendungen und Clients die Verwendung von Common Event Infrastructure ermöglicht. Sie können die Konfiguration des Common Event Infrastructure-Service wie folgt in der Administrationskonsole anzeigen:

- Wenn es sich um die Konfiguration eines Servers handelt, wählen Sie die Optionen **Server > Anwendungsserver > *servername* > Business Integration > Common Event Infrastructure > Common Event Infrastructure Service** aus.
- Wenn es sich um die Konfiguration eines Clusters handelt, wählen Sie die Optionen **Server > Cluster > *clustername* > Business Integration > Common Event Infrastructure > Common Event Infrastructure Service** aus.

Wenn das Markierungsfeld mit der Beschriftung 'Event Infrastructure-Server aktivieren' ausgewählt ist, so ist der Service installiert und aktiv oder wird nach dem Neustart des Servers oder Clusters gestartet. Wenn das Markierungsfeld nicht ausgewählt ist, dann ist der Service nicht installiert oder wird nach dem Neustart des Servers oder des Clusters deinstalliert.

Events Service-Einstellungen

Hier handelt es sich um eine Gruppe von durch Events Service verwendeten Merkmalen, die die Ereignisverteilung sowie die Persistenz unter Verwendung des Datenspeichers ermöglichen. Für diese Ressource ist im Normalfall keine Konfiguration erforderlich. Unter Umständen müssen Sie jedoch zusätzliche Events Service-Einstellungen erstellen, wenn Sie mehrere Events Services in ein und derselben Zelle konfigurieren möchten. Um die Einstellungen für Events Service anzuzeigen, klicken sie auf die Optionen **Serviceintegration > Events Service > Events Service-Einstellungen**.

Konfiguration für Ereignis-Messaging

Hierbei handelt es sich um die Ressourcen, die die asynchrone Nachrichtenübertragung an Events Service mit Java Messaging Service (JMS) unterstützen. Die Standard-Messaging-Konfiguration verwendet das in den Server integrierte Messaging. Optional können Sie einen externen JMS-Provider für Ereignis-Messaging konfigurieren.

Ereignisdatenbank

Die Ereignisdatenbank dient dem persistenten Speichern von Ereignissen, die der Events Service empfängt. Die Derby-Datenbank ist als Bestandteil des Servers enthalten. Von ihrer Verwendung in Produktionsumgebungen wird jedoch abgeraten. Stattdessen können Sie eine externe Ereignisdatenbank auf den folgenden Produkten konfigurieren: DB2, Oracle, SQL Server und Informix.

Ereignisfilter-Plug-in

Ein Filter-Plug-in wird zum Filtern von Ereignissen an der Quelle anhand von XPath-Ereignisselektoren verwendet. Klicken Sie zum Konfigurieren der Filtermerkmale auf die Optionen **Serviceintegration > Common Event Infrastructure > Ereignis-Emitter-Factories > Einstellungen für Ereignisfilter**.

Emitter-Factory

Eine Emitter-Factory ist ein Objekt, das Ereignisquellen zum Erstellen von Emittlern verwenden. Ein Emitter wird zum Senden von Ereignissen an den Events Service eingesetzt. Die Merkmale einer Emitter-Factory wirken sich auf das Verhalten aller Emittler aus, die mit dieser Emitter-Factory erstellt werden. Klicken Sie zum Anzeigen der verfügbaren Emitter-Factories auf die Optionen **Serviceintegration > Common Event Infrastructure > Ereignis-Emitter-Factories**.

Events Service-Übertragung

Eine Events Service-Übertragung ist ein Objekt, mit dem Merkmale definiert werden, die festlegen, wie Emittler synchron unter Verwendung von EJB-Aufrufen auf den Events Service zugreifen. Emitter-Factories verwenden diese Merkmale bei der Erstellung neuer Emittler. Sie können die verfügbaren Events Service-Übertragungen über die Emitter-Factory-Einstellungen anzeigen oder ändern.

JMS-Übertragung

Eine JMS-Übertragung ist ein Objekt, mit dem Merkmale definiert werden, die festlegen, wie Emittler asynchron unter Verwendung einer JMS-Warteschlange auf den Events Service zugreifen. Emitter-Factories verwenden diese Merkmale bei der Erstellung neuer Emittler. Sie können die verfügbaren JMS-Übertragungen über die Emitter-Factory-Einstellungen anzeigen oder ändern.

Ereignisgruppe

Eine Ereignisgruppe ist eine logische Sammlung von Ereignissen, mit denen Ereignisse anhand ihres Inhalts kategorisiert werden. Beim Abfragen von Ereignissen aus dem Events Service oder beim Abonnieren der Ereignisverteilung kann ein Ereignisnutzer angeben, dass eine Ereignisgruppe nur die Ereignisse in dieser Gruppe abrufen soll. Mit Ereignisgruppen kann auch festgelegt werden, welche Ereignisse im persistenten Datenspeicher gespeichert werden sollen. Um die verfügbaren Ereignisgruppen in der Administrationskonsole anzuzeigen, klicken Sie auf die Optionen **Serviceintegration > Common Event Infrastructure > Events Service > Ereignisservices > *ereignisservice* > Ereignisgruppen**.

Common Event Infrastructure mit der Administrationskonsole konfigurieren

Common Event Infrastructure mit der Server-Administrationskonsole konfigurieren.

Informationen zu diesem Vorgang

Öffnen Sie in der Administrationskonsole die Anzeige für Common Event Infrastructure Server. Gehen Sie hierzu wie folgt vor:

Wenn Sie einen Server konfigurieren, wählen Sie **Server > Servertypen > WebSphere-Anwendungsserver > *servername* > Business Integration > Common Event Infrastructure > Server für Common Event Infrastructure** aus.

Wenn Sie einen Cluster konfigurieren, klicken Sie auf **Server > Cluster > WebSphere-Anwendungsservercluster > *clustername* > Business Integration > Common Event Infrastructure > Server für Common Event Infrastructure**.

Vorgehensweise

1. Aktivieren Sie die Implementierung der Common Event Infrastructure-Enterprise-Anwendung, indem Sie das Markierungsfeld mit der Beschriftung **Event Infrastructure-Server aktivieren** auswählen. Wenn der Server bereits konfiguriert wurde, können Sie ihn durch Aus- oder Abwählen des Markierungsfelds aktivieren bzw. inaktivieren. Wenn das Markierungsfeld für die Aktivierung nicht ausgewählt ist, wurde Common Event Infrastructure nicht konfiguriert oder eine vorherige Konfiguration wurde inaktiviert und es wurde kein Neustart des Servers durchgeführt. Eine Informationsnachricht gibt an, ob für das Implementierungsziel Common Event Infrastructure konfiguriert ist. Wenn der Server bereits konfiguriert wurde, können Sie bei Bedarf die Einstellungen für die Datenquelle für die Ereignisdatenbank und/oder den Nachrichtenspeicher ändern.

Anmerkung: Wenn Sie das Markierungsfeld zum Aktivieren des Common Event Infrastructure-Servers auswählen, der Server jedoch noch nicht konfiguriert ist, werden die angezeigten Parameter für die Serverkonfiguration verwendet, sofern Sie diese nicht ändern.

- Wenn Sie die Konfiguration erstmals durchführen, werden die Tabellen für die Ereignisdatenquelle auf der Common-Datenbank erstellt. Wenn bereits eine Common Event Infrastructure-Serverkonfiguration vorhanden ist, müssen Sie eine Datenbank erstellen.
- Der Messaging-Service wird unter einem eindeutigen Schema unter der Common-Datenbank erstellt.

Nachdem ein Neustart des Servers bzw. des Clusters, auf dem Common Event Infrastructure konfiguriert wurde, ausgeführt worden ist, treten die neuen Änderungen in Kraft.

2. Konfigurieren Sie die Ereignisdatenbank oder ändern Sie die gegenwärtigen Einstellungen für eine vorhandene Konfiguration der Ereignisdatenbank, indem Sie eine folgenden Methoden verwenden, um die Felder mit den geeigneten Einstellungen füllen zu lassen.
 - Klicken Sie in einer Datenbankkonfigurationsanzeige, deren Liste von Optionen länger als die angezeigte Optionsliste ist, auf **Bearbeiten**.
 - Geben Sie über die Felder in der Anzeige die entsprechenden Informationen ein:
 - a. **Datenbankname** – Dies ist der Name der Datenbank, die Sie zum Speichern von Ereignissen verwenden.
 - b. **Tabellen erstellen** – Wählen Sie dieses Markierungsfeld aus, wenn die Datenbanktabellen auf der Ereignisdatenbank erstellt werden sollen.

Anmerkung: Wenn Sie Common Event Infrastructure für die Verwendung einer Datenbank auf einem anderen Server konfigurieren, kann die Erstellung der Tabellen nicht über dieses Steuerelement erfolgen. Stattdessen müssen Sie die Datenbankscripts verwenden, die nach Abschluss der übrigen Schritte dieser Konfiguration generiert werden. In diesem Fall können Sie die Anzeige für die Datenquelle öffnen, indem Sie auf **Bearbeiten** klicken, um zu erfahren, an welcher Position sich die Datenbankerstellungsscripts befinden.

- c. **Benutzername** und **Kennwort** – Angaben für die Authentifizierung bei der Ereignisdatenbank.
- d. **Server** – Name des Servers, auf dem sich die Ereignisdatenbank befindet.
- e. **Provider** – Wählen Sie im Menü einen Provider für Ihre Datenbank aus.

Anmerkung: Das Feld **Schema** ist nur aktiviert, wenn die Datenbank mit DB2 auf einer iSeries- oder z/OS-Plattform erstellt wird. In allen anderen Fällen ist das Schemafeld inaktiviert.

Wichtig: Wenn die Tabellen auf der Zieldatenbank vorhanden sind, schlägt die Konfiguration unter Umständen fehl.

3. Wählen Sie aus, ob sich der Common Event Infrastructure-Bus auf dem lokalen Server befindet (Option **Lokal**) oder auf einem anderen Server (Option **Fern**). Wählen Sie bei Auswahl der Option 'Fern' im Menü die ferne Position aus oder klicken Sie auf **Neu**, um einen neuen fernen Bus zu erstellen.
4. Konfigurieren Sie die Common Event Infrastructure-Unterstützung für Messaging.
 - Klicken Sie in einer Datenbankkonfigurationsanzeige, deren Liste von Optionen länger als die angezeigte Optionsliste ist, auf **Bearbeiten**.
 - Geben Sie über die Felder in der Anzeige die entsprechenden Informationen ein:
 - a. **Datenbankname** – Geben Sie den Namen der Datenbank ein, die Sie zum Speichern von Nachrichten verwenden werden.
 - b. **Schema** – Geben Sie einen Namen für das Schema ein oder übernehmen Sie den angegebenen Standardnamen.
 - c. **Benutzername** und **Kennwort** – Angaben für die Authentifizierung bei der Messaging-Datenbank.
 - d. **Server** – Der Name des Servers, auf dem sich die Messaging-Datenbank befindet.
 - e. **Provider** – Wählen Sie im Menü einen Provider für Ihre Datenbank aus.

5. Erstellen Sie für den Common Event Infrastructure-Bus einen Aliasnamen für die Messaging-Authentifizierung.
 - a. Wählen Sie die Optionen **Weitere Merkmale > JMS-Authentifizierungsaliasname** aus.
 - b. Geben Sie die Kombination aus Benutzer-ID und Kennwort ein, die Sie für die sichere Kommunikation im gesamten System Integration Bus verwenden werden. Wenn die Sicherheit nicht aktiviert ist, können Sie die standardmäßig konfigurierten Werte 'CEI' für die Benutzer-ID und das Kennwort übernehmen. Wenn die Sicherheit jedoch aktiviert ist, wird die Kombination aus Benutzer-ID und Kennwort zur Busauthentifizierung verwendet. In einer Produktionsumgebung wählen Sie eine eigene Benutzer-ID und ein eigenes Kennwort aus, um das System zu schützen.
 - c. Klicken Sie auf **OK**.
6. Klicken Sie auf **OK** oder **Anwenden**.
7. Führen Sie einen Neustart des Servers oder Clusters durch.

Ergebnisse

Alle übergeordneten Komponenten von Common Event Infrastructure werden jetzt konfiguriert und auf Ihrem Server oder Cluster ausgeführt. Hierzu zählen unter anderem der Ereignisdatenspeicher, die Messaging-Engine und die Ereignisanwendung. Sie können diese Einzelanzeige an Stelle einer Vielzahl von Befehlen und Schritten verwenden, die Sie andernfalls zum Konfigurieren von Common Event Infrastructure ausführen würden.

Nächste Schritte

Nach dem Neustart des Servers oder Clusters sind Sie in der Lage, Servicekomponentenereignisse zu speichern, die von Ihrer Anwendung ausgegeben werden. Sie können die Laufzeitmerkmale für den des Common Event Infrastructure-Server ändern, indem Sie die Anzeige **Common Event Infrastructure-Ziel** auswählen. Außerdem können Sie auswählen, ob der Common Event Infrastructure-Server beim Systemstart gestartet werden soll, und Sie können den JNDI-Namen für die Emitter-Factory angeben, an die die Ereignisse gesendet werden.

Common Event Infrastructure-Anwendung implementieren

Bevor Sie Common Event Infrastructure verwenden können, müssen Sie zunächst Events Service und die zugeordneten Ressourcen in der Serverlaufzeitumgebung implementieren.

Informationen zu diesem Vorgang

Die Common Event Infrastructure-Enterprise-Anwendung enthält die Laufzeitkomponenten von Events Service und die Standard-Messaging-Konfiguration, die für die Übergabe asynchroner Ereignisse verwendet wird.

Gehen Sie wie folgt vor, um Events Service zu implementieren:

Vorgehensweise

Führen Sie vom Tool 'wsadmin' den Verwaltungsbefehl **deployEventService** im Stapelverarbeitungsmodus oder im interaktiven Modus aus.

Für den Verwaltungsbefehl **deployEventService** gibt es folgende Parameter:

nodeName

Der Name des Knotens, auf dem der Events Service implementiert werden soll. Dieser Parameter ist optional. Wenn kein Knotenname angegeben wird, wird standardmäßig der Name des aktuellen Knotens verwendet. Bei Angabe eines Knotennamens muss außerdem auch der Servername mit dem Parameter **serverName** angegeben werden. Dieser Parameter ist nicht gültig, wenn Sie Events Service in einem Cluster implementieren.

serverName

Der Name des Server, auf dem der Events Service implementiert werden soll. Dieser Parameter ist nur bei Angabe eines Knotens erforderlich. Er ist nicht gültig, wenn Sie Events Service in einem Cluster implementieren.

clusterName

Der Name des Clusters, in dem der Events Service implementiert werden soll. Dieser Parameter ist optional und darf nicht angegeben werden, wenn die Implementierung für den Knoten- oder Servergeltungsbereich erfolgt.

enable

Gibt an, ob der Events Service automatisch gestartet werden soll, sobald der Server startet. Der Standardwert ist true (wahr).

Ergebnisse

Nachdem der Verwaltungsbefehl abgeschlossen worden ist, werden Events Service von Common Event Infrastructure und die Standard-Messaging-Konfiguration für den angegebenen Geltungsbereich implementiert.

Nächste Schritte

Wenn die WebSphere-Sicherheit aktiviert ist, müssen Sie unter Verwendung des Verwaltungsbefehls **setEventServiceJmsAuthAlias** auch den Aliasnamen und das Kennwort für die JMS-Authentifizierung konfigurieren.

Wenn Events Service in einem Cluster implementiert werden soll, müssen Sie darüber hinaus die Ereignisdatenbank manuell konfigurieren.

Common Event Infrastructure in einem Cluster implementieren

Es gibt mehrere Möglichkeiten für die Implementierung von Common Event Infrastructure-Ressourcen in einer Clusterumgebung.

Common Event Infrastructure in einem vorhandenen Cluster implementieren

Sie können die Events Service-Anwendung in einem vorhandenen Cluster implementieren.

Informationen zu diesem Vorgang

Bei der Implementierung der Events Service-Anwendung handelt es sich im Wesentlichen um dasselbe wie bei der Implementierung der Anwendung auf einem Standalone-Server. In einer Clusterumgebung wird jedoch keine Standardereignisdatenbank konfiguriert.

Gehen Sie wie folgt vor, um Common Event Infrastructure in einer Clusterumgebung zu implementieren und konfigurieren:

Vorgehensweise

1. Führen Sie den Befehl **deployEventService** so aus, wie Sie dies auch für einen Standalone-Server tun würden, jedoch unter Angabe des Clusternamens. Geben Sie den Cluster mit dem Parameter 'clusterName' an.
2. Führen Sie auf dem System mit Deployment Manager den Verwaltungsbefehl für die Datenbankkonfiguration aus. Geben Sie den Clusternamen mit dem Parameter 'clusterName' an. Durch diesen Befehl wird das Datenbankkonfigurationsscript generiert.
3. Kopieren Sie das Datenbankkonfigurationsscript in das Datenbanksystem.
4. Führen Sie das Datenbankkonfigurationsscript auf dem Datenbanksystem aus, damit die Ereignisdatenbank erstellt wird.
5. Führen Sie auf dem System mit Deployment Manager den Befehl **enableEventService** aus, damit Events Service aktiviert wird. Geben Sie den Namen des Clusters mit dem Parameter 'clusterName' an.

Cluster durch Konvertieren eines vorhandenen Common Event Infrastructure-Servers erstellen

Durch Konvertieren eines vorhandenen eigenständigen Servers, der bereits mit Common Event Infrastructure konfiguriert ist, können Sie einen Cluster erstellen.

Vorbereitende Schritte

Bevor Sie den vorhandenen Server konvertieren können, sollten Sie sicherstellen, dass er vollständig für Common Event Infrastructure konfiguriert ist. Die Konfiguration schließt auch die Implementierung der Events Service-Anwendung und die Konfiguration der Ereignisdatenbank ein.

Informationen zu diesem Vorgang

Gehen Sie wie folgt vor, um den Cluster zu erstellen:

Vorgehensweise

1. Gehen Sie anhand des typischen WebSphere-Prozesses zum Konvertieren eines eigenständigen Servers in das erste Member eines Clusters vor. Beim Konvertieren des Servers finden die folgenden Schritte statt:
 - Die für den Geltungsbereich des Servers verfügbaren Common Event Infrastructure-Ressourcen werden zum neuen Clustergeltungsbereich verschoben.

Standarddatenbank: Wenn der vorhandene Server mit der Standard-Derby-Datenbank konfiguriert ist, werden die Datenbankressourcen nicht zum Clustergeltungsbereich verschoben. Diese Ressourcen werden stattdessen entfernt. Die Standarddatenbankkonfiguration wird in einem Cluster nicht unterstützt. In diesem Fall ist Events Service im Cluster standardmäßig inaktiviert.
 - Die Liste der implementierten Ziele der Events Service-Anwendung wird geändert, indem der konvertierte Server entfernt und der neue Cluster hinzugefügt wird.
2. Optional: Wenn der konvertierte Server mit der Standard-Derby-Datenbank konfiguriert war, müssen Sie eine neue Ereignisdatenbank für den Cluster konfigurieren und dann Events Service aktivieren. Gehen Sie hierzu wie folgt vor:
 - a. Führen Sie auf dem System mit Deployment Manager den Verwaltungsbefehl für die Datenbankkonfiguration aus. Geben Sie den Clusternamen mit dem Parameter 'clusterName' an. Durch diesen Befehl wird das Datenbankkonfigurationsscript generiert.

- b. Kopieren Sie das Datenbankkonfigurationsscript in das Datenbanksystem.
- c. Führen Sie das Datenbankkonfigurationsscript auf dem Datenbanksystem aus, damit die Ereignisdatenbank erstellt wird.
- d. Führen Sie auf dem System mit Deployment Manager den Befehl **enableEventService** aus, damit Events Service aktiviert wird. Geben Sie den Namen des Clusters mit dem Parameter 'clusterName' an.

Cluster durch Verwenden eines vorhandenen Common Event Infrastructure-Servers als Schablone erstellen

Sie können einen Cluster erstellen, indem Sie einen vorhandenen Common Event Infrastructure-Server als Schablone angeben.

Vorbereitende Schritte

Sie können einen Cluster nur dann mit dieser Methode erstellen, wenn bereits ein Server vorhanden ist, der für Common Event Infrastructure vollständig konfiguriert ist. Die Konfiguration schließt auch die Implementierung der Events Service-Anwendung und die Konfiguration der Ereignisdatenbank ein.

Informationen zu diesem Vorgang

Gehen Sie wie folgt vor, um den Cluster zu erstellen:

Vorgehensweise

1. Gehen Sie anhand des typischen WebSphere-Prozesses zum Erstellen eines neuen Clusters vor und verwenden Sie hierbei den vorhandenen Common Event Infrastructure-Server als Schablone für das erste Member des Clusters. Beim Erstellen des ersten Members finden Sie folgenden Schritte statt:
 - Die für den Geltungsbereich des vorhandenen Servers verfügbaren Common Event Infrastructure-Ressourcen werden zum neuen Clustergeltungsbereich kopiert.

Standarddatenbank: Wenn der vorhandene Server mit der Standard-Derby-Datenbank konfiguriert ist, werden die Datenbankressourcen nicht zum Clustergeltungsbereich kopiert. Die Standarddatenbankkonfiguration wird in einem Cluster nicht unterstützt. In diesem Fall ist Events Service im Cluster standardmäßig inaktiviert.
 - Die Liste der implementierten Ziele der Events Service-Anwendung wird so geändert, dass der neue Cluster enthalten ist.
2. Optional: Wenn der vorhandene Server mit der Standard-Derby-Datenbank konfiguriert war, müssen Sie eine neue Ereignisdatenbank für den Cluster konfigurieren und dann Events Service aktivieren. Gehen Sie hierzu wie folgt vor:
 - a. Führen Sie auf dem System mit Deployment Manager den Verwaltungsbeefehl für die Datenbankkonfiguration aus. Geben Sie den Clusternamen mit dem Parameter 'clusterName' an. Durch diesen Befehl wird das Datenbankkonfigurationsscript generiert.
 - b. Kopieren Sie das Datenbankkonfigurationsscript in das Datenbanksystem.
 - c. Führen Sie das Datenbankkonfigurationsscript auf dem Datenbanksystem aus, damit die Ereignisdatenbank erstellt wird.
 - d. Führen Sie auf dem System mit Deployment Manager den Befehl **enableEventService** aus, damit Events Service aktiviert wird. Geben Sie den Namen des Clusters mit dem Parameter 'clusterName' an.

Ereignis-Messaging konfigurieren

Sie können die Messaging-Konfiguration ändern, die für die JMS-Übertragung von Ereignissen an Events Service verwendet wird.

Informationen zu diesem Vorgang

Die Messaging-Infrastruktur für Common Event Infrastructure wird erstellt, wenn Sie über die Anzeige der Administrationskonsole Common Event Infrastructure auf einem Server konfigurieren. Im Allgemeinen wird für die Messaging-Konfiguration der Standard-Messaging-Provider verwendet und eine einzelne JMS-Warteschlange für die asynchrone Übertragung von Ereignissen an Events Service erstellt. Sie können diese Messaging-Konfiguration bei Bedarf ändern.

Zusätzliche JMS-Warteschlangen konfigurieren

Wenn Sie die Standardkonfiguration für Ereignis-Messaging verwenden, können Sie zusätzliche JMS-Warteschlangen für die Übertragung von Ereignissen an Events Service hinzufügen.

Informationen zu diesem Vorgang

Wenn Sie zusätzliche JMS-Warteschlangen hinzufügen möchten und die Standardkonfiguration für Ereignis-Messaging verwenden, können Sie mehrere JMS-Warteschlangen mit Routing zum SIB-Warteschlangenziel (SIB = Service Integration Bus) konfigurieren. Das Warteschlangenziel für Common Event Infrastructure Service Integration Bus hängt von dem Geltungsbereich ab, für den Events Service implementiert wird:

Geltungsbereich	SIB-Warteschlangenziel
Server	<i>knoten.server.CommonEventInfrastructureQueueDestination</i>
Cluster	<i>cluster.CommonEventInfrastructureQueueDestination</i>

Weitere Informationen zur Konfiguration von Service Integration Bus enthält die Dokumentation.

Ereignis-Messaging über einen externen JMS-Provider konfigurieren

Wenn Sie nicht die standardmäßig eingebettete Messaging-Konfiguration für die Ereignisübertragung verwenden möchten, können Sie das asynchrone Messaging-Protokoll für die Verwendung eines externen Java Messaging Service-Providers (JMS) konfigurieren.

Vorbereitende Schritte

Bevor Sie Ereignis-Messaging über einen externen JMS-Providers konfigurieren können, müssen Sie zunächst mit den entsprechenden Schnittstellen für Ihren JMS-Provider eine JMS-Warteschlange und -Verbindungsfactory erstellen. Außerdem müssen Sie einen Listener-Port oder eine Aktivierungsspezifikation erstellen.

Informationen zu diesem Vorgang

Gehen Sie wie folgt vor, um Ereignis-Messaging über einen externen JMS-Provider zu konfigurieren:

Vorgehensweise

Führen Sie vom Tool 'wsadmin' den Verwaltungsbefehl **deployEventServiceMdb** im Stapelverarbeitungsmodus oder im interaktiven Modus aus. Für den Befehl **deployEventServiceMdb** gibt es die folgenden Parameter:

applicationName

Der Anwendungsname der von Events Service gesteuerten Message-driven Bean, die implementiert werden soll. Dieser Parameter ist erforderlich.

nodeName

Der Name des Knotens, auf dem die von Events Service gesteuerte Message-driven Bean implementiert werden soll. Bei Angabe eines Knotennamens muss auch ein Servername angegeben werden. Der Knotenname ist ein optionaler Parameter. Als Standardwert wird der aktuelle Knoten verwendet. Geben Sie diesen Parameter nicht an, wenn Sie die Anwendung in einem Cluster implementieren.

serverName

Der Name des Servers, auf dem die von Events Service gesteuerte Message-driven Bean implementiert werden soll. Dieser Parameter ist erforderlich, wenn Sie die Anwendung für den Servergeltungsbereich implementieren. Andernfalls ist der Parameter optional. Geben Sie keinen Servernamen an, wenn Sie die Anwendung in einem Cluster implementieren.

clusterName

Der Name des Clusters, in dem die von Events Service gesteuerte Message-driven Bean implementiert werden soll. Geben Sie diesen Parameter nur an, wenn Sie die Anwendung in einem Cluster implementieren.

listenerPort

Der Name des Listener-Ports, den die von Events Service gesteuerte Message-driven Bean zum Veröffentlichen von Ereignissen verwendet. Der angegebene Listener-Port muss vorhanden sein. Sie müssen entweder einen Listener-Port oder eine Aktivierungsspezifikation angeben, nicht jedoch beides.

activationSpec

Der JNDI-Name der Aktivierungsspezifikation, den die von Events Service gesteuerte Message-driven Bean zum Veröffentlichen von Ereignissen verwendet. Die angegebene Aktivierungsspezifikation muss vorhanden sein. Sie müssen entweder einen Listener-Port oder eine Aktivierungsspezifikation angeben, nicht jedoch beides.

qcfJndiName

Der JNDI-Name der Verbindungs-Factory für JMS-Warteschlangen, die die von Events Service gesteuerte Message-driven Bean verwenden soll. Dieser Parameter ist erforderlich, wenn Sie eine Aktivierungsspezifikation angeben. Andernfalls ist der Parameter optional. Wenn Sie eine Warteschlangenverbindungsfactory und einen Listener-Port angeben, muss die Warteschlangenverbindungsfactory der für den Warteschlangenverbindungsfactory konfigurierten Verbindungsfactory entsprechen.

Ergebnisse

Der Verwaltungsbefehl **deployEventServiceMdb** implementiert die Message-driven Bean für Events Service mit der Konfiguration für den angegebenen Listener-Port oder die angegebene Aktivierungsspezifikation. Außerdem werden eine Emitter-Factory und eine JMS-Übertragung über die externe JMS-Konfiguration erstellt.

Anwendungen können entweder die Standard-Emitter-Factory verwenden, die für die Verwendung der Standard-Messaging-Konfiguration konfiguriert ist, oder die neue Emitter-Factory, die den externen JMS-Provider verwenden.

Nächste Schritte

Wenn Sie mehr als eine JMS-Warteschlange für Events Service einrichten möchten, können Sie diesen Befehl mehrmals ausführen und dabei jeweils andere Enterprise-Anwendungsnamen und JMS-Warteschlangen angeben. Bei jeder Ausführung bewirkt das Script die Implementierung einer zusätzlichen Message-driven Bean und konfiguriert neue Ressourcen für die Verwendung der angegebenen JMS-Warteschlange.

JMS-Authentifizierungsaliasnamen konfigurieren

Wenn die WebSphere-Sicherheit aktiviert ist und Sie asynchrones JMS-Messaging zum Übergeben von Ereignissen an Events Service verwenden möchten, müssen Sie den JMS-Authentifizierungsaliasnamen konfigurieren.

Informationen zu diesem Vorgang

Gehen Sie wie folgt vor, um den JMS-Authentifizierungsaliasnamen zu konfigurieren:

Vorgehensweise

Führen Sie vom Tool 'wsadmin' den Verwaltungsbefehl **setEventServiceJmsAuthAlias** im Stapelverarbeitungsmodus oder im interaktiven Modus aus. Für den Befehl **setEventServiceJmsAuthAlias** gibt es die folgenden Parameter:

userName

Der Name des Benutzers, der für den JMS-Authentifizierungsaliasnamen verwendet werden soll. Dieser Parameter ist erforderlich.

password

Das Kennwort des Benutzers, der für den JMS-Authentifizierungsaliasnamen verwendet werden soll. Dieser Parameter ist erforderlich.

nodeName

Der Name des Knotens, für den Sie den JMS-Authentifizierungsaliasnamen aktualisieren oder erstellen möchten. Bei Angabe eines Knotennamens muss auch ein Servername angegeben werden. Geben Sie keinen Knotennamen an, wenn Sie den Authentifizierungsaliasnamen in einem Cluster konfigurieren.

serverName

Der Name des Servers, für den Sie den JMS-Authentifizierungsaliasnamen aktualisieren oder erstellen möchten. Dieser Parameter ist nur bei Angabe eines Knotens erforderlich. Er ist nicht gültig, wenn Sie den Authentifizierungsaliasnamen in einem Cluster konfigurieren.

clusterName

Der Name des Clusters, für den Sie den JMS-Authentifizierungsaliasnamen aktualisieren oder erstellen möchten. Geben Sie diesen Parameter nur an, wenn Sie den Authentifizierungsaliasnamen in einem Cluster konfigurieren. Geben Sie keinen Knoten oder Servernamen an.

Ergebnisse

Der JMS-Authentifizierungsaliasname, den die Objekte von Events Service verwenden, wird für den angegebenen Geltungsbereich aktualisiert. Sollte die Authentifizierung nicht existieren, wird sie unter Verwendung der angegebenen Werte erstellt.

Ereignisdatenbank konfigurieren

Sie können die Ereignisdatenbank mit Befehlen konfigurieren, die abhängig von dem jeweiligen unterstützten Datenbankprodukt variieren.

Informationen zu diesem Vorgang

Die Ereignisdatenbank ist zur Unterstützung der Persistenz von Ereignissen erforderlich. Falls Sie nicht die Konfigurationsanzeige für Common Event Infrastructure in der Administrationskonsole verwendet haben, können Sie die Ereignisdatenbank noch mit den hier beschriebenen Befehlen erstellen.

Einschränkungen für die Ereignisdatenbank

Für Konfigurationen der Ereignisdatenbank gelten bei Verwendung bestimmter Datenbanksoftware einige Einschränkungen.

Prüfen Sie den Inhalt der folgenden Tabelle, um zu erfahren, welche Einschränkungen möglicherweise für Ihre Umgebung gelten.

Tabelle 1. Einschränkungen für die Ereignisdatenbank

Datenbanktyp	Einschränkungen
Oracle	<ul style="list-style-type: none">• Durch den JDBC-Thin-Treiber für Oracle 11 ergeben sich bei Verwendung eines Unicode-Zeichensatzes einige Größenbeschränkungen für Zeichenfolgewerte. Dies kann zu einem Oracle-Fehler des Typs ORA-01461 führen, wenn Ereignisse, die große Werte enthalten (wie zum Beispiel ein langes Nachrichtenattribut), in der Ereignisdatenbank gespeichert werden. Weitere Informationen zu dieser Einschränkung enthält die Dokumentation für Oracle 11. Um dieses Problem zu vermeiden, verwenden Sie den OCI-Treiber für Oracle 11 oder den Thin-Treiber für Oracle 11.• Die Oracle-Datenbanksoftware behandelt eine leere Zeichenfolge als NULL-Wert. Bei Angabe einer leeren Zeichenfolge als Wert für ein Ereignisattribut wird diese Zeichenfolge beim Speichern in einer Oracle-Ereignisdatenbank in NULL umgewandelt.

Tabelle 1. Einschränkungen für die Ereignisdatenbank (Forts.)

Datenbanktyp	Einschränkungen
Informix	<ul style="list-style-type: none"> • Der JDBC 3.0-Treiber (oder höher) ist erforderlich. Ältere Versionen des JDBC-Treibers liefern keine volle Unterstützung für die erforderlichen XA-Transaktionen. • Für die mit dem Verwaltungsbefehl configEventServiceInformixDB generierten Scripts zum Konfigurieren und Entfernen der Datenbank ist der Befehl dbaccess zum Ausführen von SQL-Scripts erforderlich. Unter Umständen ist dieser Befehl nur auf dem Informix-Server verfügbar. Wenn sich der Informix-Server nicht auf dem WebSphere-Server, sondern auf einem anderen System befindet, kann es daher erforderlich sein, die Datenbankkonfigurationsscripts auf den Informix-Server zu kopieren und lokal auszuführen.
SQL Server	<ul style="list-style-type: none"> • Die SQL Server-Datenbank muss für die Verwendung des gemischten Authentifizierungsmodus konfiguriert sein. Die Verwendung vertrauenswürdiger Verbindungen wird nicht unterstützt. • Die gespeicherten XA-Prozeduren müssen installiert sein. Diese gespeicherten Prozeduren werden mit dem JDBC-Treiber der Microsoft® Corporation bereitgestellt. • Die Datei 'sqljdbc.dll' muss in einem Verzeichnis zur Verfügung stehen, das in der PATH-Anweisung angegeben wurde. Diese Datei wird mit dem JDBC-Treiber der Microsoft Corporation bereitgestellt. • Der Distributed Transaction Coordinator (DTC), ein Koordinationsservice für verteilte Transaktionen, muss gestartet werden.

Derby-Ereignisdatenbank konfigurieren

Sie können eine Derby-Ereignisdatenbank für den Server- oder Clustergeltungsbereich auf einem Linux®, UNIX®- oder Windows®-System konfigurieren.

Informationen zu diesem Vorgang

Für die Ereignisdatenbank können Sie zwei Typen von Derby-Datenbanken verwenden: Derby Embedded und Derby Network. Beide Datenbanktypen werden mit WebSphere Application Server bereitgestellt, besitzen jedoch nur eine eingeschränkte Funktionalität, die nicht für Produktionsumgebungen geeignet ist. Aus diesem Grund sollten Sie Derby nur zu Entwicklungs- oder Testzwecken als Ereignisdatenbank verwenden. Weitere Informationen zu Derby-Datenbanken enthält die Dokumentation für WebSphere Application Server, die am Ende dieser Seite verlinkt ist.

Derby Embedded kann nur mit einem eigenständigen Server verwendet werden. Falls Sie den eigenständigen Server irgendwann in eine Cluster- oder Network Deployment-Umgebung einbinden, müssen Sie demzufolge die Ereignisdatenquelle mit einem anderen Datenbankprodukt von Grund auf neu konfigurieren. Sie wird dann beim Start des Servers automatisch gestartet.

Derby Network kann in einer Cluster- oder einer Network Deployment-Umgebung verwendet werden, sein Einsatz in echten Produktionssystemen sollte jedoch vermieden werden. Zur Verwendung der Datenbank mit dem Server müssen Sie die Datenbank manuell starten.

Gehen Sie wie folgt vor, um eine Derby-Ereignisdatenbank zu konfigurieren:

Vorgehensweise

1. Starten Sie das Tool 'wsadmin'.
2. Verwenden Sie das Objekt 'AdminTask', um den Verwaltungsbefehl **configEventServiceDerbyDB** im Stapelverarbeitungsmodus oder im interaktiven Modus auszuführen. Für den Befehl **configEventServerDerbyDB** sind mindestens folgende Parameter erforderlich:

createDB

Gibt an, ob der Verwaltungsbefehl die Konfigurationsskripts für die Ereignisdatenbank erstellt und ausführt. Geben Sie die Werte `true` oder `false` an. Wenn für diesen Parameter der Wert `false` definiert ist, werden die Skripts zwar erstellt, jedoch nicht ausgeführt. In diesem Fall müssen Sie die Datenbankkonfigurationsskripts ausführen, um die Datenbankkonfiguration abzuschließen.

nodeName

Der Name des Knotens, der den Server enthält, auf dem die Datenquelle für Events Service erstellt wird. Bei Angabe eines Knotennamens muss auch ein Servername angegeben werden. Einer der folgenden Werte muss angegeben werden:

- Knotenname und Servername
- Clusternamen

serverName

Der Name des Servers, auf dem die Datenquelle für Events Service erstellt wird.

clusterName

Der Name des Clusters, in dem die Datenquelle für Events Service erstellt wird. Bei Angabe eines Clusternamens sollten Sie keine Knoten- und Servernamen angeben.

Unter Umständen sind für Ihre Umgebung weitere Parameter erforderlich. Eine vollständige Liste der Parameter mit Syntaxinformationen enthält die Hilfe für den Verwaltungsbefehl **configEventServiceDerbyDB**.

Ergebnisse

Mit dem Verwaltungsbefehl wird die erforderliche Datenquelle für den angegebenen Geltungsbereich erstellt. Wenn Sie für den Parameter 'createDB' den Wert `true` angegeben haben, führt der Befehl auch das generierte Datenbankkonfigurationsskript zur Erstellung der Datenbank aus.

Die generierten Datenbankkonfigurationsskripts werden standardmäßig im Verzeichnis '*profilstammverzeichnis*/databases/event/knoten/server/dbscripts/derby' gespeichert. (In einer Network Deployment-Umgebung werden diese Skripts im Verzeichnis des Deployment Manager-Profiles gespeichert.) Wenn Sie für den optio-

nalen Parameter 'outputScriptDir' einen Wert angegeben haben, werden die Scripts stattdessen an der hierdurch festgelegten Position gespeichert. Mit diesen Scripts können Sie die Ereignisdatenbank zu jedem beliebigen Zeitpunkt manuell konfigurieren.

DB2-Ereignisdatenbank konfigurieren (Linux-, UNIX- und Windows-Systeme)

Sie können eine externe Ereignisdatenbank, die DB2 Universal Database verwendet, auf einem Linux-, UNIX- oder Windows-System konfigurieren.

Informationen zu diesem Vorgang

Gehen Sie wie folgt vor, um eine DB2-Ereignisdatenbank auf einem Linux-, UNIX- oder Windows-System zu konfigurieren:

Vorgehensweise

1. Starten Sie das Tool 'wsadmin'.
2. Verwenden Sie das Objekt 'AdminTask', um den Verwaltungsbefehl **configEventServiceDB2DB** im Stapelverarbeitungsmodus oder im interaktiven Modus auszuführen. Für den Befehl **configEventServiceDB2DB** sind mindestens folgende Parameter erforderlich:

createDB

Gibt an, ob der Verwaltungsbefehl die Konfigurationsscripts für die Ereignisdatenbank erstellt und ausführt. Geben Sie die Werte `true` oder `false` an. Wenn für diesen Parameter der Wert `false` definiert ist, werden die Scripts zwar erstellt, jedoch nicht ausgeführt. In diesem Fall müssen Sie die Datenbankkonfigurationsscripts ausführen, um die Datenbankkonfiguration abzuschließen.

nodeName

Der Name des Knotens, der den Server enthält, auf dem die Datenquelle für Events Service erstellt wird. Bei Angabe eines Knotennamens muss auch ein Servername angegeben werden. Einer der folgenden Werte muss angegeben werden:

- Knotenname und Servername
- Clustername

serverName

Der Name des Servers, auf dem die Datenquelle für Events Service erstellt wird.

clusterName

Der Name des Clusters, in dem die Datenquelle für Events Service erstellt wird. Bei Angabe eines Clusternamens sollten Sie keine Knoten- und Servernamen angeben.

jdbcClassPath

Der Pfad zum JDBC-Treiber. Geben Sie nur den Pfad zur Treiberdatei, jedoch nicht den Dateinamen an.

dbHostName

Der Hostname des Servers, auf dem die Datenbank installiert ist.

dbUser

Die DB2-Benutzer-ID, die bei der Erstellung der Ereignisdatenbank verwendet werden soll. Die angegebene Benutzer-ID muss über ausreichende Berechtigungen zum Erstellen und Löschen von Datenbanken verfügen.

dbPassword

Das DB2-Kennwort, das verwendet werden soll.

Unter Umständen sind für Ihre Umgebung weitere Parameter erforderlich. Eine vollständige Liste der Parameter mit Syntaxinformationen enthält die Hilfe für den Verwaltungsbefehl `configEventServiceDB2DB`.

Ergebnisse

Mit dem Verwaltungsbefehl wird die erforderliche Datenquelle für den angegebenen Geltungsbereich erstellt. Wenn Sie für den Parameter 'createDB' den Wert `true` angegeben haben, führt der Befehl auch das generierte Datenbankkonfigurationsscript zur Erstellung der Datenbank aus.

Die generierten Datenbankkonfigurationsscripts werden standardmäßig im Verzeichnis `'profilstammverzeichnis/databases/event/knoten/server/dbscripts/db2'` gespeichert. (In einer Network Deployment-Umgebung werden diese Scripts im Verzeichnis des Deployment Manager-Profiles gespeichert.) Wenn Sie für den optionalen Parameter 'outputScriptDir' einen Wert angegeben haben, werden die Scripts stattdessen an der hierdurch festgelegten Position gespeichert. Mit diesen Scripts können Sie die Ereignisdatenbank zu jedem beliebigen Zeitpunkt manuell konfigurieren.

DB2-Datenbank auf einem z/OS-System konfigurieren

Sie können eine Ereignisdatenbank auf einem z/OS-System unter Verwendung von DB2-Datenbanksoftware konfigurieren.

Vorbereitende Schritte

Um die DB2-Datenbank von einem fernen Client konfigurieren zu können, muss das Produkt DB2 Connect mit den aktuellsten Fixpacks installiert sein.

Informationen zu diesem Vorgang

Gehen Sie wie folgt vor, um die Ereignisdatenbank zu konfigurieren:

Vorgehensweise

1. Linux UNIX Windows Wenn Sie die z/OS-Ereignisdatenbank von einem Linux-, UNIX- oder Windows-Clientsystem aus konfigurieren, führen Sie die folgenden Schritte zu Erstellen und Katalogisieren der Datenbank aus:
 - a. Erstellen Sie auf dem z/OS-System mit dem DB2-Administrationsmenü ein Subsystem.
 - b. Optional: Erstellen Sie die Speichergruppe, die Sie für die Ereignisdatenbank verwenden möchten. Sie können auch eine vorhandene Speichergruppe (zum Beispiel `sysdeflt`) verwenden.
 - c. Aktivieren Sie die 4 K-, 8 K- und 16 K-Pufferpools, die Sie für die Ereignisdatenbank verwenden möchten.
 - d. Erteilen Sie der Benutzer-ID, die die Datenquelle verwenden soll, die notwendigen Berechtigungen. Diese Benutzer-ID muss Berechtigungen zum Zugreifen auf die von Ihnen erstellte Datenbank und Speichergruppe besitzen. Darüber hinaus muss sie auch über die Berechtigung zum Erstellen neuer Tabellen, Tabellenbereiche und Indizes für die Datenbank verfügen.

- e. Katalogisieren Sie die ferne Datenbank. Führen Sie die folgenden Befehle wahlweise mit einem Script oder von einem DB2-Befehlszeilenfenster aus:
- ```
catalog tcpip node zosnode remote hostname server port-IP
 system db-subsystem
catalog database db-name as db-name at node zosnode authentication DCS
```
- Weitere Informationen dazu, wie Sie einen Knoten und die zugehörigen Datenbanken katalogisieren, enthält die Dokumentation für DB2 Connect.
- f. Überprüfen Sie, ob Sie eine Verbindung zum fernen Subsystem herstellen können. Führen Sie dazu den folgenden Befehl aus:
- ```
db2 connect to subsystem user benutzer-id using kennwort
```
- g. Erzeugen Sie eine Bindung zur Hostdatenbank. Führen Sie die folgenden Befehle aus:
- ```
db2 connect to db-name user benutzer-id using kennwort
db2 bind db2-stammverzeichnis/bnd/@ddcsmvs.lst blocking all sqlerror
continue message mvs.msg grant public
db2 connect reset
```
- Weitere Informationen dazu, wie Sie einen Client an eine Hostdatenbank binden, enthält die Dokumentation für DB2 Connect.
2. Starten Sie auf dem WebSphere-System das Tool 'wsadmin'.
  3. Verwenden Sie das Objekt 'AdminTask', um den Verwaltungsbefehl **configEventServiceDB2ZOSDB** im Stapelverarbeitungsmodus oder im interaktiven Modus auszuführen. Für den Befehl **configEventServiceDB2ZOSDB** sind mindestens folgende Parameter erforderlich:

#### createDB

**Linux** **UNIX** **Windows** Gibt an, ob der Verwaltungsbefehl die Konfigurationsscripts für die Ereignisdatenbank erstellt und ausführt. Dieser Parameter gilt nur, wenn Sie den Verwaltungsbefehl von einem Linux-, UNIX- oder Windows-Clientsystem ausführen. Geben Sie die Werte true oder false an.

Wenn für diesen Parameter der Wert false definiert ist oder Sie den Befehl auf einem z/OS-System ausführen, werden die Scripts erstellt, jedoch nicht ausgeführt. In diesem Fall müssen Sie die Datenbankkonfigurationsscripts ausführen, um die Datenbankkonfiguration abzuschließen.

#### nodeName

Der Name des Knotens, der den Server enthält, auf dem die Datenquelle für Events Service erstellt wird. Bei Angabe eines Knotennamens muss auch ein Servername angegeben werden. Einer der folgenden Werte muss angegeben werden:

- Knotenname und Servername
- Clustername

#### serverName

Der Name des Servers, auf dem die Datenquelle für Events Service erstellt wird.

#### clusterName

Der Name des Clusters, in dem die Datenquelle für Events Service erstellt wird. Bei Angabe eines Clusternamens sollten Sie keine Knoten- und Servernamen angeben.

#### jdbcClassPath

Der Pfad zum JDBC-Treiber. Geben Sie nur den Pfad zur Treiberdatei, jedoch nicht den Dateinamen an.

**dbHostName**

Der Hostname des Servers, auf dem die Datenbank installiert ist.

**dbUser**

Die DB2-Benutzer-ID, die bei der Erstellung der Ereignisdatenbank verwendet werden soll. Die angegebene Benutzer-ID muss über ausreichende Berechtigungen zum Erstellen und Löschen von Datenbanken verfügen.

**dbPassword**

Das DB2-Kennwort, das verwendet werden soll.

Unter Umständen sind für Ihre Umgebung weitere Parameter erforderlich. Eine vollständige Liste der Parameter mit Syntaxinformationen enthält die Hilfe für den Verwaltungsbefehl **configEventServiceDB2ZOSDB**.

## Ergebnisse

Mit dem Verwaltungsbefehl wird die erforderliche Datenquelle für den angegebenen Geltungsbereich erstellt. Wenn Sie den Befehl auf einem Linux, UNIX- oder Windows DB2-Client ausführen und für den Parameter 'createDB' den Wert true angegeben haben, führt der Befehl auch das generierte Datenbankkonfigurationscript zur Erstellung der Datenbank aus. Auf einem z/OS-System müssen Sie die generierten DLL-Dateien unter Verwendung der SPUFI-Funktion (SPUFI = SQL Processor Using File Input) ausführen. Die DLL-Dateien werden im Verzeichnis *'profilstammverzeichnis/databases/event/knoten/server/db2zos/ddl'* gespeichert.

Die generierten Datenbankkonfigurationsscripts werden standardmäßig im Verzeichnis *'profilstammverzeichnis/databases/event/knoten/server/dbscripts/db2zos'* gespeichert. (In einer Network Deployment-Umgebung werden diese Scripts im Verzeichnis des Deployment Manager-Profiles gespeichert.) Wenn Sie für den optionalen Parameter 'outputScriptDir' einen Wert angegeben haben, werden die Scripts stattdessen an der hierdurch festgelegten Position gespeichert. Mit diesen Scripts können Sie die Ereignisdatenbank zu jedem beliebigen Zeitpunkt manuell konfigurieren.

## Nächste Schritte

Nachdem Sie die Konfiguration der Datenbank abgeschlossen haben, können Sie die Datenbankkonfiguration mit der Administrationskonsole des Servers testen. Navigieren Sie hierzu zu der entsprechenden JDBC-Datenquelle und wählen Sie die Option **Verbindung testen** aus.

## DB2-Datenbank auf einem iSeries-System konfigurieren

Sie können eine Ereignisdatenbank auf einem iSeries-System unter Verwendung von DB2-Datenbanksoftware konfigurieren.

### Informationen zu diesem Vorgang

Wenn Sie einen lokalen iSeries-Server zum Konfigurieren eines fernen iSeries-Servers verwenden, müssen Sie auf Ihrem lokalen Server einen Eintrag für die ferne Datenbank als Aliasnamen für die Zieldatenbank angeben. Gehen Sie wie folgt vor, um die Ereignisdatenbank zu konfigurieren:

### Vorgehensweise

1. Starten Sie das Tool 'wsadmin'.
2. Verwenden Sie das Objekt 'AdminTask', um den Verwaltungsbefehl **configEventServiceDB2iSeriesDB** im Stapelverarbeitungsmodus oder im inter-

aktiven Modus auszuführen. Für den Befehl **configEventServiceDB2iSeriesDB** sind mindestens folgende Parameter erforderlich:

**createDB**

Gibt an, ob der Verwaltungsbefehl die Konfigurationsscripts für die Ereignisdatenbank erstellt und ausführt. Geben Sie die Werte `true` oder `false` an. Wenn für diesen Parameter der Wert `false` definiert ist, werden die Scripts zwar erstellt, jedoch nicht ausgeführt. In diesem Fall müssen Sie die Datenbankkonfigurationsscripts ausführen, um die Datenbankkonfiguration abzuschließen.

**Einschränkung:** Mit dem Verwaltungsbefehl kann das Datenbankkonfigurationsscript nur auf dem iSeries-System automatisch ausgeführt werden. Wenn Sie den Befehl auf einem Clientsystem ausführen, wird ein Fehler zurückgegeben.

**nodeName**

Der Name des Knotens, der den Server enthält, auf dem die Datenquelle für Events Service erstellt wird. Bei Angabe eines Knotennamens muss auch ein Servername angegeben werden. Einer der folgenden Werte muss angegeben werden:

- Knotenname und Servername
- Clustername

**serverName**

Der Name des Servers, auf dem die Datenquelle für Events Service erstellt wird.

**clusterName**

Der Name des Clusters, in dem die Datenquelle für Events Service erstellt wird. Bei Angabe eines Clusternamens sollten Sie keine Knoten- und Servernamen angeben.

**toolboxJdbcClassPath**

Der Pfad zum JDBC-Treiber für IBM Toolbox for Java DB2. Verwenden Sie diesen Parameter nur, wenn Sie anstelle des nativen JDBC-Treibers den Toolbox for Java-Treiber verwenden wollen. Geben Sie nur den Pfad zur Treiberdatei, jedoch nicht den Dateinamen an.

**nativeJdbcClassPath**

Der Pfad zum nativen JDBC-Treiber für DB2 für iSeries. Verwenden Sie diesen Parameter nur, wenn Sie anstelle des Treibers von Toolbox for Java den nativen JDBC-Treiber verwenden wollen. Geben Sie nur den Pfad zur Treiberdatei, jedoch nicht den Dateinamen an.

**dbHostName**

Der Hostname des Servers, auf dem die Datenbank installiert ist. Dieser Parameter ist erforderlich, wenn Sie den JDBC-Treiber von Toolbox for Java verwenden.

**dbUser**

Die DB2-Benutzer-ID, die bei der Erstellung der Ereignisdatenbank verwendet werden soll. Die angegebene Benutzer-ID muss über ausreichende Berechtigungen zum Erstellen und Löschen von Datenbanken verfügen.

**dbPassword**

Das DB2-Kennwort, das verwendet werden soll.

Unter Umständen sind für Ihre Umgebung weitere Parameter erforderlich. Eine vollständige Liste der Parameter mit Syntaxinformationen enthält die Hilfe für den Verwaltungsbefehl **configEventServiceDB2iSeriesDB**.

## Ergebnisse

Mit dem Verwaltungsbefehl werden Scripts zum Erstellen der erforderlichen Datenbank und Datenquelle für den angegebenen Geltungsbereich generiert. Die Scripts werden standardmäßig im Verzeichnis *'profilstammverzeichnis/databases/event/knoten/server/dbscripts/db2iseries'* gespeichert. Wenn Sie für den optionalen Parameter *'outputScriptDir'* einen Wert angegeben haben, werden die Scripts stattdessen an der hierdurch festgelegten Position gespeichert. Mit diesen Scripts können Sie die Ereignisdatenbank zu jedem beliebigen Zeitpunkt manuell konfigurieren.

## Nächste Schritte

Wenn Sie den Verwaltungsbefehl für die Datenbankkonfiguration auf einem Client-System ausgeführt haben, müssen Sie die generierten Scripts auf das iSeries-System übertragen und dort ausführen, um die erforderlichen Ressourcen zu erstellen.

Nachdem Sie die Konfiguration der Datenbank abgeschlossen haben, können Sie die Datenbankkonfiguration mit der Administrationskonsole des Servers testen. Navigieren Sie hierzu zu der entsprechenden JDBC-Datenquelle und wählen Sie die Option **Verbindung testen** aus.

## Informix-Ereignisdatenbank konfigurieren

Sie können eine externe Ereignisdatenbank, die IBM Informix Dynamic Server verwendet, auf einem Linux-, UNIX- oder Windows-System konfigurieren.

### Informationen zu diesem Vorgang

Gehen Sie wie folgt vor, um eine Informix-Ereignisdatenbank zu konfigurieren:

### Vorgehensweise

1. Starten Sie das Tool *'wsadmin'*.
2. Verwenden Sie das Objekt *'AdminTask'*, um den Verwaltungsbefehl **configEventServiceInformixDB** im Stapelverarbeitungsmodus oder im interaktiven Modus auszuführen. Für den Befehl **configEventServiceInformixDB** sind mindestens folgende Parameter erforderlich:

#### **createDB**

Gibt an, ob der Verwaltungsbefehl die Konfigurationsscripts für die Ereignisdatenbank erstellen und ausführen soll. Geben Sie die Werte *true* oder *false* an. Wenn für diesen Parameter der Wert *false* definiert ist, werden die Scripts zwar erstellt, jedoch nicht ausgeführt. In diesem Fall müssen Sie die Datenbankkonfigurationsscripts ausführen, um die Datenbankkonfiguration abzuschließen.

**Berechtigungen:** Wenn Sie für diesen Parameter den Wert *true* angeben, müssen Sie sicherstellen, dass die von Ihnen verwendete Benutzer-ID über ausreichend Berechtigungen zum Erstellen von Informix-Datenbanken, Datenbankbereichen, Tabellen, Sichten, Indizes und gespeicherten Prozeduren verfügt.

#### **nodeName**

Der Name des Knotens, der den Server enthält, auf dem die Datenquelle für Events Service erstellt werden soll. Bei Angabe eines Knotennamens muss auch ein Servername angegeben werden.

Einer der folgenden Werte muss angegeben werden:

- Knotenname und Servername
- Clustername

**serverName**

Der Name des Servers, auf dem die Datenquelle für Events Service erstellt werden soll.

**clusterName**

Der Name des Clusters, in dem die Datenquelle für Events Service erstellt werden soll. Bei Angabe eines Clusternamens sollten Sie keine Knoten- und Servernamen angeben.

**jdbcClassPath**

Der Pfad zum JDBC-Treiber. Geben Sie nur den Pfad zur Treiberdatei, jedoch nicht den Dateinamen an.

**dbInformixDir**

Das Verzeichnis, in dem die Informix-Datenbanksoftware installiert ist. Dieser Parameter ist nur erforderlich, wenn Sie für den Parameter 'createDB' den Wert true angegeben haben.

**dbHostName**

Der Hostname des Systems, auf dem der Datenbankserver installiert ist.

**dbServerName**

Der Name des Informix-Servers (zum Beispiel `ol_servername`).

**dbUser**

Die Benutzer-ID für das Informix-Datenbankschema, die als Eigner der Ereignisdatenbanktabellen definiert wird. Diese Benutzer-ID muss über ausreichend Berechtigungen zum Erstellen von Datenbanken und Datenbankbereichen verfügen. Die WebSphere-Datenquelle verwendet diese Benutzer-ID zur Authentifizierung der Verbindung der Informix-Datenbank.

**dbPassword**

Das Kennwort für die angegebene Schemabeanutzer-ID.

Unter Umständen sind für Ihre Umgebung weitere Parameter erforderlich. Eine vollständige Liste der Parameter mit Syntaxinformationen enthält die Hilfe für den Verwaltungsbefehl **configEventServiceInformixDB**.

## Ergebnisse

Mit dem Verwaltungsbefehl wird die erforderliche Datenquelle für den angegebenen Geltungsbereich erstellt. Wenn Sie für den Parameter 'createDB' den Wert true angegeben haben, führt der Befehl auch das generierte Datenbankkonfigurationsscript zur Erstellung der Datenbank aus.

Die generierten Datenbankkonfigurationsscripts werden standardmäßig im Verzeichnis '*profilstammverzeichnis*/databases/event/knoten/server/dbscripts/informix' gespeichert. (In einer Network Deployment-Umgebung werden diese Scripts im Verzeichnis des Deployment Manager-Profiles gespeichert.) Wenn Sie für den optionalen Parameter 'outputScriptDir' einen Wert angegeben haben, werden die Scripts stattdessen an der hierdurch festgelegten Position gespeichert. Mit diesen Scripts können Sie die Ereignisdatenbank zu jedem beliebigen Zeitpunkt manuell konfigurieren.

**Ausführen der Scripts:** Für die mit dem Verwaltungsbefehl **configEventServiceInformixDB** generierten Scripts zum Konfigurieren und Entfernen der Datenbank ist

der Befehl **dbaccess** zum Ausführen von SQL-Skripts erforderlich. Unter Umständen ist dieser Befehl nur auf dem Informix-Server verfügbar. Wenn sich der Informix-Server nicht auf dem Server, sondern auf einem anderen System befindet, kann es daher erforderlich sein, die Datenbankkonfigurationsskripts auf den Informix-Server zu kopieren und lokal auszuführen.

## Oracle-Ereignisdatenbank konfigurieren

Sie können eine externe Ereignisdatenbank, die Oracle Database verwendet, auf einem Linux-, UNIX- oder Windows-System konfigurieren.

### Vorbereitende Schritte

Bevor Sie eine Oracle-Ereignisdatenbank konfigurieren können, müssen Sie zuerst die Datenbank erstellen. Die Oracle-System-ID (SID) muss bereits vorhanden sein, bevor Sie den Befehl für die Konfiguration der Ereignisdatenbank ausführen. Die Standard-System-ID für die Ereignisdatenbank lautet event.

### Informationen zu diesem Vorgang

Gehen Sie wie folgt vor, um eine Oracle-Ereignisdatenbank zu konfigurieren:

### Vorgehensweise

1. Starten Sie das Tool 'wsadmin'.
2. Verwenden Sie das Objekt 'AdminTask', um den Verwaltungsbefehl **config-EventServiceOracleDB** im Stapelverarbeitungsmodus oder im interaktiven Modus auszuführen. Für den Befehl **configEventServiceOracleDB** sind mindestens folgende Parameter erforderlich:

#### **createDB**

Gibt an, ob der Verwaltungsbefehl die Konfigurationsskripts für die Ereignisdatenbank erstellen und ausführen soll. Geben Sie die Werte **true** oder **false** an. Wenn für diesen Parameter der Wert **false** definiert ist, werden die Skripts zwar erstellt, jedoch nicht ausgeführt. In diesem Fall müssen Sie die Datenbankkonfigurationsskripts ausführen, um die Datenbankkonfiguration abzuschließen.

#### **nodeName**

Der Name des Knotens, der den Server enthält, auf dem die Datenquelle für Events Service erstellt werden soll. Bei Angabe eines Knotennamens muss auch ein Servername angegeben werden. Einer der folgenden Werte muss angegeben werden:

- Knotenname und Servername
- Clustername

#### **serverName**

Der Name des Servers, auf dem die Datenquelle für Events Service erstellt werden soll.

#### **clusterName**

Der Name des Clusters, in dem die Datenquelle für Events Service erstellt werden soll. Bei Angabe eines Clusternamens sollten Sie keine Knoten- und Servernamen angeben.

#### **jdbcClassPath**

Der Pfad zum JDBC-Treiber. Geben Sie nur den Pfad zur Treiberdatei, jedoch nicht den Dateinamen an.



**oracleHome**

Das Verzeichnis ORACLE\_AUSGANGSVERZEICHNIS. Dieser Parameter ist nur erforderlich, wenn Sie für den Parameter 'createDB' den Wert true angegeben haben.

**dbPassword**

Das Kennwort, das für die Schemabenzutzer-ID verwendet werden soll, die während der Datenbankkonfiguration erstellt wurde. Die Standardbenutzer-ID lautet ceiuser. Dieses Kennwort wird zur Authentifizierung der Oracle-Datenbankverbindung verwendet.

**sysUser**

Die Oracle-Benutzer-ID für SYSUSER. Diese Benutzer-ID muss über SYSDBA-Berechtigungen verfügen.

**sysPassword**

Das Kennwort für die angegebene SYSUSER-Benutzer-ID.

Unter Umständen sind für Ihre Umgebung weitere Parameter erforderlich. Eine vollständige Liste der Parameter mit Syntaxinformationen enthält die Hilfe für den Verwaltungsbefehl **configEventServiceOracleDB**.

## Ergebnisse

Mit dem Verwaltungsbefehl wird die erforderliche Datenquelle für den angegebenen Geltungsbereich erstellt. Wenn Sie für den Parameter 'createDB' den Wert true angegeben haben, führt der Befehl auch das generierte Datenbankkonfigurationsscript zur Erstellung der Datenbank aus.

Die generierten Datenbankkonfigurationsscripts werden standardmäßig im Verzeichnis '*profilstammverzeichnis*/databases/event/knoten/server/dbscripts/oracle' gespeichert. (In einer Network Deployment-Umgebung werden diese Scripts im Verzeichnis des Deployment Manager-Profiles gespeichert.) Wenn Sie für den optionalen Parameter 'outputScriptDir' einen Wert angegeben haben, werden die Scripts stattdessen an der hierdurch festgelegten Position gespeichert. Mit diesen Scripts können Sie die Ereignisdatenbank zu jedem beliebigen Zeitpunkt manuell konfigurieren.

## SQL Server-Ereignisdatenbank konfigurieren

Sie können eine externe Ereignisdatenbank, die Microsoft SQL Server Enterprise verwendet, auf einem Windows-System konfigurieren.

### Informationen zu diesem Vorgang

Gehen Sie wie folgt vor, um eine SQL Server-Ereignisdatenbank zu konfigurieren:

#### Vorgehensweise

1. Erstellen Sie auf dem Serversystem mit der SQL Server-Datenbank das Verzeichnis, das später die Datenbankdateien enthalten soll. Standardmäßig werden die Dateien in das Verzeichnis 'C:\Programme\ibm\event\ceiinst1\sqlserverdaten' geschrieben. Wenn Sie eine andere Position angeben wollen, müssen Sie das generierte Datenbankkonfigurationsscript bearbeiten, indem Sie den Wert für den Parameter 'ceiInstancePrefix' ändern. Führen Sie das Script anschließend manuell aus.
2. Starten Sie auf dem Serversystem das Tool 'wsadmin'.

3. Verwenden Sie das Objekt 'AdminTask', um den Verwaltungsbefehl **configEventServiceSQLServerDB** im Stapelverarbeitungsmodus oder im interaktiven Modus auszuführen. Für den Befehl **configEventServiceSQLServerDB** sind mindestens folgende Parameter erforderlich:

**createDB**

Gibt an, ob der Verwaltungsbefehl die Konfigurationsskripts für die Ereignisdatenbank erstellen und ausführen soll. Geben Sie die Werte true oder false an. Wenn für diesen Parameter der Wert false definiert ist, werden die Skripts zwar erstellt, jedoch nicht ausgeführt. In diesem Fall müssen Sie die Datenbankkonfigurationsskripts ausführen, um die Datenbankkonfiguration abzuschließen.

**nodeName**

Der Name des Knotens, der den Server enthält, auf dem die Datenquelle für Events Service erstellt werden soll. Bei Angabe eines Knotennamens muss auch ein Servername angegeben werden. Einer der folgenden Werte muss angegeben werden:

- Knotenname und Servername
- Clustername

**serverName**

Der Name des Servers, auf dem die Datenquelle für Events Service erstellt werden soll. Bei Angabe eines Servernamens muss auch ein Knotenname angegeben werden.

**clusterName**

Der Name des Clusters, in dem die Datenquelle für Events Service erstellt werden soll. Bei Angabe eines Clusternamens sollten Sie keine Knoten- und Servernamen angeben.

**dbServerName**

Der Servername der SQL Server-Datenbank. Dieser Parameter ist nur erforderlich, wenn Sie für den Parameter 'createDB' den Wert true angegeben haben.

**dbHostName**

Der Hostname des Servers, auf dem die SQL Server-Datenbank ausgeführt wird.

**dbPassword**

Das Kennwort, das für die Benutzer-ID verwendet werden soll, die als Eigentümer der Ereignisdatenbanktabellen erstellt wurde. Die Standardbenutzer-ID lautet 'ceiuser'. Die WebSphere-Datenquelle verwendet dieses Kennwort zur Authentifizierung der SQL Server-Datenbankverbindung.

**saUser**

Eine Benutzer-ID mit den Berechtigungen zum Erstellen und Löschen von Datenbanken und Benutzern. Dieser Parameter ist nur erforderlich, wenn Sie für den Parameter 'createDB' den Wert true angegeben haben.

**saPassword**

Das Kennwort für den angegebenen SA-Benutzer.

Unter Umständen sind für Ihre Umgebung weitere Parameter erforderlich. Eine vollständige Liste der Parameter mit Syntaxinformationen enthält die Hilfe für den Verwaltungsbefehl **configEventServiceSQLServerDB**.

## Ergebnisse

Mit dem Verwaltungsbefehl wird die erforderliche Datenquelle für den angegebenen Geltungsbereich erstellt. Wenn Sie für den Parameter 'createDB' den Wert true angegeben haben, führt der Befehl auch das generierte Datenbankkonfigurationsscript zur Erstellung der Datenbank aus.

Die generierten Datenbankkonfigurationsscripts werden standardmäßig im Verzeichnis *profilstammverzeichnis/databases/event/knoten/server/dbscripts/dbscripts/sqlserver* gespeichert. (In einer Network Deployment-Umgebung werden diese Scripts im Verzeichnis des Deployment Manager-Profiles gespeichert.) Wenn Sie für den optionalen Parameter 'outputScriptDir' einen Wert angegeben haben, werden die Scripts stattdessen an der hierdurch festgelegten Position gespeichert. Mit diesen Scripts können Sie die Ereignisdatenbank zu jedem beliebigen Zeitpunkt manuell konfigurieren.

## Manuelle Ausführung der Scripts zur Datenbankkonfiguration

Die mit den Verwaltungsbefehlen für die Datenbankkonfiguration generierten Scripts können jederzeit manuell ausgeführt werden.

### Informationen zu diesem Vorgang

Der Prozess der Datenbankkonfiguration erfolgt in zwei Schritten. Mit dem Verwaltungsbefehl für die Datenbankkonfiguration wird zunächst ein datenbankspezifisches Script für Ihre Umgebung erstellt. Dieses generierte Script konfiguriert dann die Ereignisdatenbank und Datenquellen. Wenn Sie beim Ausführen des Verwaltungsbefehls für den Parameter 'createDB' den Wert true angeben, werden beide Schritte automatisch ausgeführt.

Wenn Sie für den Parameter 'createDB' jedoch false angeben, müssen Sie die Datenbankkonfiguration abschließen, indem Sie das generierte Script manuell auf dem Zielsystem ausführen. In jedem der folgenden Fälle muss das Script gegebenenfalls manuell ausgeführt werden:

- Die Ereignisdatenbank muss auf einem anderen System konfiguriert werden als dem, auf dem der Verwaltungsbefehl ausgeführt wurde.
- Die Ereignisdatenbank muss zu einem späteren Zeitpunkt erneut erstellt werden.
- Die Standardoptionen, die das generierte Script verwendet, müssen geändert werden.

### Derby-Ereignisdatenbank manuell erstellen

Verwenden Sie den Befehl `cr_event_derby`, um ein Datenbankkonfigurationsscript für eine Derby-Ereignisdatenbank manuell zu generieren.

### Informationen zu diesem Vorgang

Gehen Sie wie folgt vor, um das generierte Datenbankkonfigurationsscript für eine Derby-Ereignisdatenbank manuell auszuführen:

#### Vorgehensweise

1. Rufen Sie auf dem Serversystem das Verzeichnis auf, in dem sich das generierte Script befindet. Das Standardverzeichnis ist *installationsstammverzeichnis/profiles/profilname/dbscripts/CEI\_cei-datenbankname*. Wenn Sie für den Parameter 'outputScriptDir' des Verwaltungsbefehls für die Datenbankkonfiguration einen Wert angegeben haben, werden die Scripts stattdessen an der hierdurch festgelegten Position gespeichert.

2. Öffnen Sie das Script in einem ASCII-Texteditor und nehmen Sie die erforderlichen Änderungen vor. Der Name des Scripts hängt jeweils von dem verwendeten Betriebssystem ab:

- **Windows** `cr_event_derby.bat`
- **Linux** **UNIX** `cr_event_derby.sh`
- **Auf i5/OS-Plattformen:** `cr_event_derby`

3. Optional: Wenn Sie die Datenbank auf einem iSeries-System konfigurieren, starten Sie Qshell Interpreter.
4. Führen Sie das Datenbankerstellungsscript unter Verwendung der folgenden Syntax aus und vergessen Sie nicht, gegebenenfalls die entsprechende Dateierweiterung anzugeben:

```
cr_event_derby -p profilpfad [-s servername|-c clustername]
```

Die folgenden Parameter werden verwendet:

**-p** *profilpfad*

Der Pfad zum WebSphere-Profilverzeichnis. Dieser Parameter ist erforderlich.

**-s** *servername*

Der Name des Servers. Dieser Parameter ist erforderlich, wenn Sie die Datenbank für den Servergeltungsbereich konfigurieren.

**-c** *clustername*

Der Name des Clusters. Dieser Parameter ist erforderlich, wenn Sie die Datenbank für den Clustergeltungsbereich konfigurieren.

Mit dem folgenden Befehl würde beispielsweise die Derby-Datenbank für den Geltungsbereich des Servers 'server1' mit dem Profil 'profile1' erstellt:

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

5. Führen Sie einen Neustart des Servers durch. Bei einem eingebundenen Knoten müssen Sie außerdem den Knotenagenten mit dem Befehl **stopNode** stoppen und anschließend mit **startNode** erneut starten.

## Nächste Schritte

Nachdem Sie die Konfiguration der Datenbank abgeschlossen haben, können Sie die Datenbankkonfiguration mit der Administrationskonsole testen. Navigieren Sie hierzu zu der entsprechenden JDBC-Datenquelle und wählen Sie die Option **Verbindung testen** aus.

## DB2-Ereignisdatenbank auf einem Linux-, UNIX- oder Windows-System manuell erstellen

Verwenden Sie den Befehl `cr_event_db2`, um ein Datenbankkonfigurationsscript für eine DB2-Ereignisdatenbank auf einem Linux-, UNIX- oder Windows-System manuell zu generieren.

## Informationen zu diesem Vorgang




Gehen Sie wie folgt vor, um das generierte Datenbankkonfigurationsscript für eine DB2-Ereignisdatenbank auf einem Linux-, UNIX- oder Windows-System manuell auszuführen:

### Vorgehensweise

1. Rufen Sie auf dem Serversystem das Verzeichnis auf, in dem sich das generierte Script befindet. Das Standardverzeichnis ist *installationsstammverzeichnis/*

`profiles/profilname/dbscripts/CEI_cei-datenbankname`. Wenn Sie für den Parameter 'outputScriptDir' des Verwaltungsbefehls für die Datenbankkonfiguration einen Wert angegeben haben, werden die Scripts stattdessen an der hierdurch festgelegten Position gespeichert.

- Öffnen Sie das Script in einem ASCII-Texteditor und nehmen Sie die erforderlichen Änderungen vor. Der Name des Scripts hängt jeweils von dem verwendeten Betriebssystem ab:

-  Windows-Systeme: `cr_event_db2.bat`
-   Linux- und UNIX-Systeme: `cr_event_db2.sh`

- Führen Sie das Datenbankerstellungsscript unter Verwendung der folgenden Syntax aus und vergessen Sie nicht, gegebenenfalls die entsprechende Dateierweiterung anzugeben:

```
cr_event_db2 [client|server] datenbankbenutzer [datenbankkennwort]
```

Die folgenden Parameter werden verwendet:

#### **client|server**

Hiermit wird angegeben, ob es sich bei der Datenbank um einen Client oder einen Server handelt. Sie müssen entweder **client** oder den Wert **server** angeben.

#### *datenbankbenutzer*

Die ID des Datenbankbenutzers. Dieser Parameter ist erforderlich.

#### *datenbankkennwort*

Das Kennwort für den Datenbankbenutzer. Wenn Sie kein Kennwort für eine Clientdatenbank angeben, werden Sie zur Eingabe des Kennworts aufgefordert.

Mit dem folgenden Befehl würde beispielsweise die DB2-Ereignisdatenbank für eine Clientdatenbank mit der Benutzer-ID `db2admin` und dem Kennwort `meinkennwort` erstellt:

```
cr_event_db2 client db2admin meinkennwort
```

- Führen Sie einen Neustart des Servers durch. Bei einem eingebundenen Knoten müssen Sie außerdem den Knotenagenten mit dem Befehl **stopNode** stoppen und anschließend mit **startNode** erneut starten.

## **Nächste Schritte**

Nachdem Sie die Konfiguration der Datenbank abgeschlossen haben, können Sie die Datenbankkonfiguration mit der Administrationskonsole testen. Navigieren Sie hierzu zu der entsprechenden JDBC-Datenquelle und wählen Sie die Option **Verbindung testen** aus.

## **DB2-Ereignisdatenbank auf einem z/OS-System manuell erstellen**

Verwenden Sie den Befehl `cr_event_db2zos`, um ein Datenbankkonfigurationsscript für eine DB2-Ereignisdatenbank auf einem z/OS-System mit einem Linux-, UNIX- oder Windows-Clientsystem manuell zu generieren.

## **Informationen zu diesem Vorgang**

Gehen Sie wie folgt vor, um das generierte Datenbankkonfigurationsscript für eine DB2-Ereignisdatenbank auf einem z/OS-System mit einem Linux-, UNIX- oder Windows-Clientsystem manuell auszuführen:

## Vorgehensweise

1. Rufen Sie auf dem Serversystem das Verzeichnis auf, in dem sich das generierte Script befindet. Das Standardverzeichnis ist *installationsstammverzeichnis/profiles/profilname/dbscripts/CEI\_cei-datenbankname*. Wenn Sie für den Parameter 'outputScriptDir' des Verwaltungsbefehls für die Datenbankkonfiguration einen Wert angegeben haben, werden die Scripts stattdessen an der hierdurch festgelegten Position gespeichert.
2. Öffnen Sie das Script in einem ASCII-Texteditor und nehmen Sie die erforderlichen Änderungen vor. Der Name des Scripts hängt jeweils von dem verwendeten Betriebssystem ab:
  - Windows-Systeme: *cr\_event\_db2zos.bat*
  - Linux- und UNIX-Systeme: *cr\_event\_db2zos.sh*
3. Führen Sie das Datenbankerstellungsscript unter Verwendung der folgenden Syntax aus und vergessen Sie nicht, gegebenenfalls die entsprechende Dateierweiterung anzugeben:

```
cr_event_db2zos [dbName=datenbankname] datenbankbenutzer [datenbankkennwort]
```

Die folgenden Parameter werden verwendet:

### *datenbankname*

Der Datenbankname, der verwendet werden soll. Dieser Parameter ist optional. Wenn kein Datenbankname angegeben wird, so wird ein Name generiert.

### *datenbankbenutzer*

Die Datenbankbenutzer-ID, die verwendet werden soll. Dieser Parameter ist erforderlich.

### *datenbankkennwort*

Das Kennwort für den Datenbankbenutzer. Wenn Sie das Kennwort nicht angeben, werden Sie von der DB2-Datenbank zur Eingabe des Kennworts aufgefordert.

Mit dem folgenden Befehl würde beispielsweise eine DB2-Ereignisdatenbank namens *event* mit der Benutzer-ID *db2admin* und dem Kennwort *meinkennwort* erstellt:

```
cr_event_db2zos dbName=client db2admin meinkennwort
```

4. Führen Sie einen Neustart des Servers durch. Bei einem eingebundenen Knoten müssen Sie außerdem den Knotenagenten mit dem Befehl **stopNode** stoppen und anschließend mit **startNode** erneut starten.

## Nächste Schritte

Nachdem Sie die Konfiguration der Datenbank abgeschlossen haben, können Sie die Datenbankkonfiguration mit der Administrationskonsole testen. Navigieren Sie hierzu zu der entsprechenden JDBC-Datenquelle und wählen Sie die Option **Verbindung testen** aus.

## DB2-Ereignisdatenbank auf einem iSeries-System manuell erstellen

Verwenden Sie den Befehl *cr\_event\_db2iseries*, um ein Datenbankkonfigurations-script für eine DB2-Ereignisdatenbank auf einem iSeries-System manuell zu generieren.

## Informationen zu diesem Vorgang

Gehen Sie wie folgt vor, um das generierte Datenbankkonfigurationsscript für eine DB2-Ereignisdatenbank auf einem iSeries-System manuell auszuführen:

### Vorgehensweise

1. Rufen Sie auf dem Serversystem das Verzeichnis auf, in dem sich das generierte Script befindet. Das Standardverzeichnis ist *installationsstammverzeichnis/profiles/profilname/dbscripts/CEI\_cei-datenbankname*. Wenn Sie für den Parameter 'outputScriptDir' des Verwaltungsbefehls für die Datenbankkonfiguration einen Wert angegeben haben, werden die Scripts stattdessen an der hierdurch festgelegten Position gespeichert.
2. Öffnen Sie das Script 'cr\_event\_db2iseries' in einem ASCII-Texteditor und nehmen Sie die erforderlichen Änderungen vor.
3. Starten Sie Qshell Interpreter.
4. Führen Sie das Script für die Datenbankeinstellung unter Verwendung der folgenden Syntax aus:

```
cr_event_db2iseries datenbankbenutzer datenbankkennwort
```

Die folgenden Parameter werden verwendet:

*datenbankbenutzer*

Die ID des Datenbankbenutzers. Dieser Parameter ist erforderlich.

*datenbankkennwort*

Das Kennwort für den Datenbankbenutzer. Dieser Parameter ist erforderlich.

Mit dem folgenden Befehl würde beispielsweise die DB2-Ereignisdatenbank mit der Benutzer-ID db2admin und dem Kennwort meinkennwort erstellt:

```
cr_event_db2iseries db2admin meinkennwort
```

5. Führen Sie einen Neustart des Servers durch. Bei einem eingebundenen Knoten müssen Sie außerdem den Knotenagenten mit dem Befehl **stopNode** stoppen und anschließend mit **startNode** erneut starten.

### Nächste Schritte

Nachdem Sie die Konfiguration der Datenbank abgeschlossen haben, können Sie die Datenbankkonfiguration mit der Administrationskonsole testen. Navigieren Sie hierzu zu der entsprechenden JDBC-Datenquelle und wählen Sie die Option **Verbindung testen** aus.

### Informix-Ereignisdatenbank manuell erstellen

Die mit den Verwaltungsbefehlen für die Datenbankkonfiguration generierten Scripts können jederzeit manuell ausgeführt werden.

## Informationen zu diesem Vorgang

Gehen Sie wie folgt vor, um die generierten Datenbankkonfigurationsscripts für eine Informix-Ereignisdatenbank manuell auszuführen:

### Vorgehensweise

1. Rufen Sie auf dem Serversystem das Verzeichnis auf, in dem sich das generierte Script befindet. Das Standardverzeichnis ist *installationsstammverzeichnis/profiles/profilname/dbscripts/CEI\_cei-datenbankname*.

Wenn Sie für den Parameter 'outputScriptDir' des Verwaltungsbefehls für die Datenbankkonfiguration einen Wert angegeben haben, werden die Scripts stattdessen an der hierdurch festgelegten Position gespeichert.

- Öffnen Sie das Script in einem ASCII-Texteditor und nehmen Sie die erforderlichen Änderungen vor. Der Name des Scripts hängt jeweils von dem verwendeten Betriebssystem ab:
  - Windows** Windows-Systeme: cr\_event\_informix.bat
  - Linux** **UNIX** Linux- und UNIX-Systeme: cr\_event\_informix.sh
- Führen Sie das Script für die Datenbankerstellung ohne Angabe von Parametern aus.
- Führen Sie einen Neustart des Servers durch. Bei einem eingebundenen Knoten müssen Sie außerdem den Knotenagenten mit dem Befehl **stopNode** stoppen und anschließend mit **startNode** erneut starten.

## Nächste Schritte

Nachdem Sie die Konfiguration der Datenbank abgeschlossen haben, können Sie die Datenbankkonfiguration mit der Administrationskonsole testen. Navigieren Sie hierzu zu der entsprechenden JDBC-Datenquelle und wählen Sie die Option **Verbindung testen** aus.

## Oracle-Ereignisdatenbank manuell erstellen

Verwenden Sie den Befehl `cr_event_oracle`, um ein Datenbankkonfigurationsscript für eine Oracle-Ereignisdatenbank manuell zu generieren.

## Informationen zu diesem Vorgang

Gehen Sie wie folgt vor, um das generierte Datenbankkonfigurationsscript für eine Oracle-Ereignisdatenbank manuell auszuführen:

### Vorgehensweise

- Rufen Sie auf dem Serversystem das Verzeichnis auf, in dem sich das generierte Script befindet. Das Standardverzeichnis ist `installationsstammverzeichnis/profiles/profilname/dbscripts/CEI_cei-datenbankname`. Wenn Sie für den Parameter 'outputScriptDir' des Verwaltungsbefehls für die Datenbankkonfiguration einen Wert angegeben haben, werden die Scripts stattdessen an der hierdurch festgelegten Position gespeichert.
- Öffnen Sie das Script in einem ASCII-Texteditor und nehmen Sie die erforderlichen Änderungen vor. Der Name des Scripts hängt jeweils von dem verwendeten Betriebssystem ab:
  - Windows** Windows-Systeme: cr\_event\_oracle.bat
  - Linux** **UNIX** Linux- und UNIX-Systeme: cr\_event\_oracle.sh
- Führen Sie das Datenbankerstellungsscript unter Verwendung der folgenden Syntax aus und vergessen Sie nicht, gegebenenfalls die entsprechende Dateierweiterung anzugeben:

```
cr_event_oracle kennwort sys-benutzer
 sys-kennwort [sid=system-id]
 [oracleHome=oracle-ausgangsverzeichnis]
```

Die folgenden Parameter werden verwendet:

*kennwort*

Das Kennwort für die Schemabeneutzer-ID. Dieser Parameter ist erforderlich.



#### *sys-benutzer*

Die Benutzer-ID, die über SYSDBA-Berechtigungen in der Oracle-Datenbank verfügt. (Hierbei handelt es sich in der Regel um den Benutzer 'sys'.) Dieser Parameter ist erforderlich.

#### *sys-kennwort*

Das Kennwort für die angegebene Benutzer-ID für den Benutzer 'sys'. Wenn diese Benutzer-ID kein Kennwort verwendet, geben Sie none ein.

#### **sid=system-id**

Die Oracle-System-ID (SID). Dieser Parameter ist optional.

#### **oracleHome=oracle-ausgangsverzeichnis**

Das Oracle-Ausgangsverzeichnis. Dieser Parameter ist optional. Wenn Sie keinen Wert angeben, wird ein generierter Pfad verwendet.

Mit dem folgenden Befehl würde beispielsweise die Oracle-Ereignisdatenbank mit der Schemabenutzer-ID auser und der sys-Benutzer-ID sys erstellt:

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

4. Führen Sie einen Neustart des Servers durch. Bei einem eingebundenen Knoten müssen Sie außerdem den Knotenagenten mit dem Befehl **stopNode** stoppen und anschließend mit **startNode** erneut starten.

## Nächste Schritte

Nachdem Sie die Konfiguration der Datenbank abgeschlossen haben, können Sie die Datenbankkonfiguration mit der Administrationskonsole testen. Navigieren Sie hierzu zu der entsprechenden JDBC-Datenquelle und wählen Sie die Option **Verbindung testen** aus.

## SQL Server-Ereignisdatenbank manuell erstellen

Verwenden Sie den Befehl `cr_event_mssql`, um ein Datenbankkonfigurationsscript für eine SQL Server-Ereignisdatenbank manuell zu generieren.

## Informationen zu diesem Vorgang

Gehen Sie wie folgt vor, um das generierte Datenbankkonfigurationsscript für eine SQL Server-Ereignisdatenbank manuell auszuführen:

### Vorgehensweise

1. Rufen Sie auf dem Serversystem das Verzeichnis auf, in dem sich das generierte Script befindet. Das Standardverzeichnis ist `installationsstammverzeichnis/profiles/profilname/dbscripts/CEI_cei-datenbankname`. Wenn Sie für den Parameter 'outputScriptDir' des Verwaltungsbefehls für die Datenbankkonfiguration einen Wert angegeben haben, werden die Scripts stattdessen an der hierdurch festgelegten Position gespeichert.
2. Öffnen Sie das Script 'cr\_event\_mssql.bat' in einem ASCII-Texteditor und nehmen Sie die erforderlichen Änderungen vor.
3. Führen Sie das Script für die Datenbankerstellung unter Verwendung der folgenden Syntax aus:

```
cr_event_mssql benutzer-id kennwort [server=server] sauser=sa-benutzer
sapassword=sa-kennwort
```

Die folgenden Parameter werden verwendet:

#### *benutzer-id*

Die Anmeldebenutzer-ID für SQL Server, die als Eigner der erstellten Tabellen definiert wird. Diese Benutzer-ID muss in SQL Server erstellt werden,

sodass eine JDBC-Verbindung zur Datenbank hergestellt werden kann. (Die JDBC-Treiber unterstützen keine vertrauenswürdigen Verbindungen.)

*kennwort*

Das Kennwort für die neue Anmelde-Benutzer-ID, die erstellt wird.

**server=***server*

Der Name des Servers, der die SQL Server-Datenbank enthält. Dieser Parameter ist optional. Als Standardwert wird der lokale Host verwendet.

**sauser=***sa-benutzer*

Die Benutzer-ID für den Benutzer 'sa'. Diese Benutzer-ID muss über ausreichende Berechtigungen zum Erstellen von Datenbanken und Benutzeranmeldungen verfügen.

**sapassword=***sa-kennwort*

Das Kennwort für 'sa', wenn der gemischte Authentifizierungsmodus verwendet wird. Wenn für die Benutzer-ID 'sa' kein Kennwort festgelegt wurde, geben Sie `sapassword=` ohne Angabe eines Werts an. Übergehen Sie diesen Parameter, wenn Sie eine vertrauenswürdige Verbindung verwenden.

Mit dem folgenden Befehl würde beispielsweise die SQL Server-Ereignisdatenbank mit der Anmelde-Benutzer-ID `userid` erstellt:

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

4. Führen Sie einen Neustart des Servers durch. Bei einem eingebundenen Knoten müssen Sie außerdem den Knotenagenten mit dem Befehl **stopNode** stoppen und anschließend mit **startNode** erneut starten.

## Nächste Schritte

Nachdem Sie die Konfiguration der Datenbank abgeschlossen haben, können Sie die Datenbankkonfiguration mit der Administrationskonsole testen. Navigieren Sie hierzu zu der entsprechenden JDBC-Datenquelle und wählen Sie die Option **Verbindung testen** aus.

## Upgrade einer Ereignisdatenbank aus einer früheren Version durchführen

Wenn Sie eine Migration von einer früheren Version von Common Event Infrastructure durchgeführt haben und Ereignispersistenz verwenden, müssen Sie unter Umständen ein Upgrade auf eine vorhandene Ereignisdatenbank durchführen.

### Informationen zu diesem Vorgang

Ein Upgrade der Ereignisdatenbank ist erforderlich, wenn Sie eine Migration von Common Event Infrastructure Version 5.1 oder früher durchführen.

Beim Upgradeprozess für die Datenbank wird ein Upgrade des Schemas und der Metadaten der vorhandenen Datenbank auf die aktuelle Version unter Beibehaltung der vorhandenen Ereignisdaten durchgeführt.

Das Datenbankupgrade-Script führt ein Upgrade des Schemas und der Metadaten der vorhandenen Ereignisdatenbank auf die aktuelle Version durch.

**Nicht unterstützte Versionen:** Falls Ihre Ereignisdatenbank Datenbanksoftware mit einem Versionsstand verwendet, der nicht mehr von Common Event Infrastructure 6.0 unterstützt wird, müssen Sie zuerst unter Verwendung des geeigneten Verfahrens für die Datenbanksoftware eine Migration der Datenbank auf eine unter-

stützte Version durchführen. Danach können Sie den Upgradeprozess für die Ereignisdatenbank ausführen und ein Upgrade der Datenbank durchführen.

## Upgrade einer DB2-Ereignisdatenbank von einer früheren Version durchführen

Wenn Sie über eine DB2-Ereignisdatenbank von Common Event Infrastructure Version 5.1 auf einem Linux-, UNIX- oder Windows-System verfügen, müssen Sie ein Upgrade auf die aktuelle Version durchführen.

### Informationen zu diesem Vorgang

Gehen Sie wie folgt vor, um ein Upgrade für eine DB2-Ereignisdatenbank auf einem Linux- oder UNIX-System durchzuführen:

### Vorgehensweise

1. Erstellen Sie eine Sicherungskopie der vorhandenen Ereignisdatenbank.
2. Wechseln Sie in das Verzeichnis *profilstammverzeichnis/bin*.
3. Führen Sie für Ihr jeweiliges Betriebssystem das Upgrade-Script für DB2 aus:

- **Windows** Windows-Systeme:  

```
eventUpgradeDB2 runUpgrade=[true|false] dbUser=benutzer
[dbName=name] [dbPassword=kennwort]
[dbNode=knoten] [scriptDir=verz]
```
- **Linux** **UNIX** Linux- und UNIX-Systeme:  

```
eventUpgradeDB2.sh runUpgrade=[true|false] dbUser=benutzer
[dbName=name] [dbPassword=kennwort]
[dbNode=knoten] [scriptDir=verz]
```

Die normalerweise erforderlichen Parameter lauten wie folgt:

#### runUpgrade

Legt fest, ob das Upgrade-Script die generierten DLL-Scripts automatisch für die Fertigstellung des Datenbankupgrades ausführen soll. Dieser Parameter ist erforderlich. Geben Sie den Wert *false* an, wenn Sie das Upgrade der Datenbank zu einem späteren Zeitpunkt oder auf einem anderen System manuell durchführen möchten.

#### dbUser

Gibt die DB2-Benutzer-ID an, die verwendet werden soll. Dieser Parameter ist erforderlich.

#### dbName

Gibt den Namen der DB2-Datenbank an. Der Standardname der Ereignisdatenbank lautet *event*. Dieser Parameter ist erforderlich, wenn Sie *runUpgrade=true* angegeben haben.

#### dbPassword

Gibt das Kennwort für die angegebene DB2-Benutzer-ID an. Dieser Parameter ist optional. Wenn Sie kein Kennwort angeben, werden Sie von DB2 zur Eingabe des Kennworts aufgefordert.

#### dbNode

Gibt den Namen des Datenbankknotens an. Dieser Parameter ist erforderlich, wenn Sie das Upgrade-Scripts von einem DB2-Clientsystem ausführen.

#### scriptDir

Gibt das Verzeichnis an, das die generierten DLL-Scripts enthalten soll. Dieser Parameter ist optional. Wenn kein Verzeichnis angegeben wird, werden die Scripts im Verzeichnis *.\eventDBUpgrade\db2'* gespeichert.

Eine vollständige Liste der Parameter mit Syntaxinformationen wird angezeigt, wenn Sie das Script **eventUpgradeDB2** ohne Parameter ausführen.

## Ergebnisse

Mit dem Upgrade-Script werden die DLL-Scripts generiert, die für das Upgrade der Ereignisdatenbank erforderlich sind. Wenn Sie `runUpgrade=true` angegeben haben, werden die DDL-Scripts automatisch ausgeführt und schließen das Upgrade ab.

## Beispiel

Im folgenden Beispiel wird das Upgrade für eine vorhandene DB2-Datenbank auf einem Windows-System durchgeführt:

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

## Nächste Schritte

Wenn Sie `runUpgrade=false` angegeben haben, müssen Sie die DDL-Scripts manuell auf dem Datenbanksystem ausführen, um das Datenbank-Upgrade abzuschließen.

## Upgrade einer DB2 for z/OS-Ereignisdatenbank von einer früheren Version durchführen

Wenn Sie über eine DB2-Ereignisdatenbank von Common Event Infrastructure Version 5.1 auf einem z/OS-System verfügen, müssen Sie ein Upgrade auf die aktuelle Version durchführen.

## Informationen zu diesem Vorgang

Gehen Sie wie folgt vor, um ein Upgrade für eine DB2-Ereignisdatenbank auf einem z/OS-System durchzuführen:

## Vorgehensweise

1. Erstellen Sie eine Sicherungskopie der vorhandenen Ereignisdatenbank.
2. Wechseln Sie in das Verzeichnis `profilstammverzeichnis/bin`.
3. Führen Sie das Upgrade-Script für DB2 for z/OS für das Betriebssystem Ihres Clients aus:

- **Windows** Windows-Systeme:  

```
eventUpgradeDB2ZOS runUpgrade=[true|false] dbUser=benutzer
[dbName=name] [dbPassword=kennwort]
[scriptDir=verz] storageGroup=gruppe
bufferPool14K=4k-pufferpool bufferPool18k=8k-pufferpool
bufferPool16K=16k-pufferpool
```

- **Linux** **UNIX** Linux- und UNIX-Systeme:  

```
eventUpgradeDB2ZOS.sh runUpgrade=[true|false] dbUser=benutzer
[dbName=name] [dbPassword=kennwort]
[scriptDir=verz] storageGroup=gruppe
bufferPool14K=4k-pufferpool bufferPool18k=8k-pufferpool
bufferPool16K=16k-pufferpool
```

Die normalerweise erforderlichen Parameter lauten wie folgt:

## runUpgrade

Legt fest, ob das Upgrade-Script die generierten DLL-Scripts automatisch für die Fertigstellung des Datenbankupgrades ausführen soll. Dieser Para-

meter ist erforderlich. Geben Sie den Wert 'false' an, wenn Sie das Upgrade der Datenbank zu einem späteren Zeitpunkt oder auf einem anderen System manuell durchführen möchten.

**z/OS-Systeme:** Dieser Parameter wird auf einem nativen z/OS-System ignoriert. Die automatische Ausführung der generierten DLL-Scripts wird nur auf einem Clientsystem unterstützt.

#### **dbUser**

Gibt die DB2-Benutzer-ID an, die verwendet werden soll. Dieser Parameter ist erforderlich.

#### **dbName**

Gibt den Namen der DB2-Datenbank an. Der Standardname der Ereignisdatenbank lautet event. Dieser Parameter ist erforderlich, wenn Sie `runUpgrade=true` angegeben haben.

#### **dbPassword**

Gibt das Kennwort für die angegebene DB2-Benutzer-ID an. Dieser Parameter ist optional. Wenn Sie kein Kennwort angeben, werden Sie von DB2 zur Eingabe des Kennworts aufgefordert.

#### **scriptDir**

Gibt das Verzeichnis an, das die generierten DLL-Scripts enthalten soll. Dieser Parameter ist optional. Wenn kein Verzeichnis angegeben wird, werden die Scripts im Verzeichnis '`.\eventDBUpgrade\db2zos`' gespeichert.

#### **storageGroup**

Gibt den Namen der Speichergruppe an. Dieser Parameter ist erforderlich.

#### **bufferPool4K**

Gibt den Namen des 4K-Pufferpools an. Dieser Parameter ist erforderlich.

#### **bufferPool8K**

Gibt den Namen des 8K-Pufferpools an. Dieser Parameter ist erforderlich.

#### **bufferPool16K**

Gibt den Namen des 16K-Pufferpools an. Dieser Parameter ist erforderlich.

Eine vollständige Liste der Parameter mit Syntaxinformationen wird angezeigt, wenn Sie das Script **eventUpgradeDB2ZOS** ohne Parameter ausführen.

## **Ergebnisse**

Mit dem Upgrade-Script werden die DLL-Scripts generiert, die für das Upgrade der Ereignisdatenbank erforderlich sind. Wenn Sie `runUpgrade=true` auf einem Clientsystem angegeben haben, werden die DDL-Scripts automatisch ausgeführt und schließen das Upgrade ab.

## **Beispiel**

Im folgenden Beispiel wird ein Upgrade einer a DB2 for z/OS-Ereignisdatenbank von einem Windows-System durchgeführt:

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

## **Nächste Schritte**

Wenn Sie `runUpgrade=false` angegeben haben oder das Upgrade-Script auf dem z/OS-System ausgeführt haben, müssen Sie die generierten DLL-Scripts manuell unter Verwendung der SPUFI-Funktion (SPUFI = SQL Processor Using File Input)

auf dem z/OS-System durchführen. Dieser Schritt schließt das Datenbankupgrade ab.

## Upgrade einer Oracle-Ereignisdatenbank von Version 5 durchführen

Wenn Sie über eine Oracle-Ereignisdatenbank von Common Event Infrastructure Version 5.1 verfügen, müssen Sie ein Upgrade auf die aktuelle Version durchführen.

### Informationen zu diesem Vorgang

Gehen Sie wie folgt vor, um ein Upgrade für eine Oracle-Ereignisdatenbank durchzuführen:

### Vorgehensweise

1. Erstellen Sie eine Sicherungskopie der vorhandenen Ereignisdatenbank.
2. Wechseln Sie in das Verzeichnis *profilstammverzeichnis/bin*.
3. Führen Sie für Ihr jeweiliges Betriebssystem das Upgrade-Script für Oracle aus:

- Windows-Systeme:

```
eventUpgradeOracle runUpgrade=[true|false] schemaUser=schemabnutzer
[oracleHome=verz] [dbName=name]
[dbUser=sysuser] [dbPassword=kennwort]
[scriptDir=verz]
```

- Linux- und UNIX-Systeme:

```
eventUpgradeOracle.sh runUpgrade=[true|false] schemaUser=schemabnutzer
[oracleHome=verz] [dbName=name]
[dbUser=sysuser] [dbPassword=kennwort]
[scriptDir=verz]
```

Die normalerweise erforderlichen Parameter lauten wie folgt:

#### runUpgrade

Legt fest, ob das Upgrade-Script die generierten DLL-Scripts automatisch für die Fertigstellung des Datenbankupgrades ausführen soll. Dieser Parameter ist erforderlich. Geben Sie den Wert *false* an, wenn Sie das Upgrade der Datenbank zu einem späteren Zeitpunkt oder auf einem anderen System manuell durchführen möchten.

#### schemaUser

Gibt die Oracle-Benutzer-ID an, die Eigner der Datenbanktabellen ist. Dieser Parameter ist erforderlich.

#### oracleHome

Gibt das Oracle-Ausgangsverzeichnis an. Dieser Parameter ist erforderlich, wenn Sie *runUpgrade=true* angegeben haben.

#### dbName

Gibt den Namen der Oracle-Datenbank an. Der Standardname der Ereignisdatenbank lautet *event*. Dieser Parameter ist erforderlich, wenn Sie *runUpgrade=true* angegeben haben.

#### dbUser

Gibt die Benutzer-ID des Oracle-Benutzers 'sys' an. Dieser Parameter ist erforderlich, wenn Sie *runUpgrade=true* angegeben haben.

#### dbPassword

Gibt das Kennwort für die Benutzer-ID des Benutzers 'sys' an. Geben Sie diesen Parameter nicht an, wenn die Systembenutzer-ID kein Kennwort besitzt.

### **scriptDir**

Gibt das Verzeichnis an, das die generierten DLL-Skripts enthalten soll. Dieser Parameter ist optional. Wenn kein Verzeichnis angegeben wird, werden die Skripts im Verzeichnis '.\eventDBUpgrade\oracle' gespeichert.

Eine vollständige Liste der Parameter mit Syntaxinformationen wird angezeigt, wenn Sie das Skript **eventUpgradeOracle** ohne Parameter ausführen.

### **Ergebnisse**

Mit dem Upgrade-Skript werden die DLL-Skripts generiert, die für das Upgrade der Ereignisdatenbank erforderlich sind. Wenn Sie `runUpgrade=true` angegeben haben, werden die DDL-Skripts automatisch ausgeführt und schließen das Upgrade ab.

### **Beispiel**

Im folgenden Beispiel wird das Upgrade für eine vorhandene Oracle-Datenbank auf einem Windows-System durchgeführt:

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

### **Nächste Schritte**

Wenn Sie `runUpgrade=false` angegeben haben, müssen Sie die DDL-Skripts manuell auf dem Datenbanksystem ausführen, um das Datenbank-Upgrade abzuschließen.

---

## **Zellübergreifende Common Event Infrastructure-Konfiguration für WebSphere Business Monitor**

Sie müssen zwischen einem fernen Server, der Common Event Infrastructure-Ereignisse (CEI-Ereignisse) erzeugt, und dem WebSphere Business Monitor-Server die Konnektivität konfigurieren.

### **Informationen zu diesem Vorgang**

Detailinformationen dazu, wie Sie CEI zellübergreifend in einer Umgebung mit mehreren Servern konfigurieren, enthält der Artikel 'Fernes CEI-Server zur Verwendung von WebSphere Business Monitor konfigurieren' im Information Center von IBM WebSphere Business Monitor.

---

## **Common Event Infrastructure-Konfiguration entfernen**

Wenn die Konfiguration für Common Event Infrastructure entfernt werden soll, müssen Sie als Vorbereitung zur Deinstallation des Servers zuerst die implementierten Enterprise-Anwendungen und die Datenbankkonfiguration entfernen.

### **Informationen zu diesem Vorgang**

Common Event Infrastructure ist in der Basisinstallation des Servers installiert, aber nur dann aktiv, wenn die Instanz bereits konfiguriert wurde. In diesem Abschnitt wird lediglich beschrieben, wie Sie eine zuvor konfigurierte Instanz von Common Event Infrastructure entfernen können. Führen Sie die folgenden Schritte aus, um die Konfiguration für Common Event Infrastructure zu entfernen:

## Ereignisdatenbank entfernen

Verwenden Sie zum Entfernen der Ereignisdatenbank den entsprechenden Verwaltungsbefehl für den jeweiligen Datenbanktyp.

### Informationen zu diesem Vorgang

Gehen Sie wie folgt vor, um die Ereignisdatenbank zu entfernen:

### Vorgehensweise

1. Starten Sie das Tool 'wsadmin'.
2. Verwenden Sie das Objekt 'AdminTask', um den entsprechenden Verwaltungsbefehl für die jeweilige Ereignisdatenbank auszuführen:

| Datenbanktyp                               | Befehl                                                  |
|--------------------------------------------|---------------------------------------------------------|
| Derby                                      | „Befehl 'removeEventServiceDerbyDB'“ auf Seite 115      |
| DB2 auf Linux-, UNIX- und Windows-Systemen | „Befehl 'removeEventServiceDB2DB'“ auf Seite 111        |
| DB2 auf z/OS-Systemen                      | „Befehl 'removeEventServiceDB2ZOSDB'“ auf Seite 114     |
| DB2 auf iSeries-Systemen                   | „Befehl 'removeEventServiceDB2iSeriesDB'“ auf Seite 113 |
| Informix                                   | „Befehl 'removeEventServiceInformixDB'“ auf Seite 117   |
| Oracle                                     | „Befehl 'removeEventServiceOracleDB'“ auf Seite 118     |
| SQL Server                                 | „Befehl 'removeEventServiceSQLServerDB'“ auf Seite 119  |

### Datenbankspezifische Hinweise:

- Auf einem z/OS-System entfernt der Verwaltungsbefehl nur die JDBC-Datenquelle. Zum Entfernen der Datenbank müssen Sie mithilfe von SPUFI das Script zum Entfernen von Datenbanken ausführen, das während der Datenbankerstellung generiert wurde. Dieses Script befindet sich standardmäßig im Verzeichnis *profilstammverzeichnis/databases/event/knoten/server/dbscripts/db2zos*.
- Auf einem iSeries-System entfernt der Verwaltungsbefehl nur die JDBC-Datenquelle. Löschen Sie zum Entfernen der Datenbank auf dem iSeries-System die Sammlung, die für die Datenbank erstellt wurde.

Die erforderlichen Parameter (z. B. Benutzer-IDs und Kennwörter) sind je nach Datenbanktyp unterschiedlich. Eine vollständige Liste der Parameter mit Syntaxinformationen finden Sie in der Hilfe für den Verwaltungsbefehl.

## Common Event Infrastructure-Anwendung entfernen

Mit dem Verwaltungsbefehl **removeEventService** können Sie die Events Service-Enterprise-Anwendung und -Ressourcen manuell aus dem Server entfernen.

### Informationen zu diesem Vorgang

Gehen Sie wie folgt vor, um die Events Service-Enterprise-Anwendung zu entfernen:



## Vorgehensweise

1. Starten Sie das Tool 'wsadmin'.
2. Verwenden Sie das Objekt 'AdminTask', um den Verwaltungsbefehl **remove-EventService** im Stapelverarbeitungsmodus oder im interaktiven Modus auszuführen. Für den Verwaltungsbefehl **removeEventService** gibt es folgende Parameter:

### **nodeName**

Der Name des Knotens, auf dem der Events Service implementiert wird. Dieser Parameter ist optional. Wenn kein Knotenname angegeben wird, wird standardmäßig der Name des aktuellen Knotens verwendet. Bei Angabe eines Knotennamens muss außerdem auch der Servername mit dem Parameter **serverName** angegeben werden. Dieser Parameter ist nicht gültig, wenn Sie den Events Service aus einem Cluster entfernen.

### **serverName**

Der Name des Servers, auf dem der Events Service implementiert wird. Dieser Parameter ist nur bei Angabe eines Knotens erforderlich. Er ist nicht gültig, wenn Sie den Events Service aus einem Cluster entfernen.

### **clusterName**

Der Name des Clusters, in dem der Events Service implementiert wird. Dieser Parameter ist optional und darf nicht angegeben werden, wenn Sie den Events Service aus einem Server entfernen.

## Ereignis-Messaging aus dem Common Event Infrastructure-Server entfernen

Mit dem Verwaltungsbefehl 'removeEventServiceMdb' können Sie die Events Service-Messaging-Konfiguration für einen externen JMS-Provider entfernen.

### Informationen zu diesem Vorgang

Mit diesem Befehl wird die für die JMS-Konfiguration implementierte Message-driven Bean entfernt. Gehen Sie wie folgt vor, um die vom Events Service gesteuerte Message-driven Bean zu entfernen:

## Vorgehensweise

1. Starten Sie das Tool 'wsadmin'.
2. Verwenden Sie das Objekt 'AdminTask', um den Verwaltungsbefehl **remove-EventServiceMdb** im Stapelverarbeitungsmodus oder im interaktiven Modus auszuführen. Für den Verwaltungsbefehl **removeEventServiceMdb** gibt es folgende Parameter:

### **applicationName**

Der Anwendungsname der implementierten vom Events Service gesteuerten Message-driven Bean.

### **nodeName**

Der Name des Knotens, auf dem die vom Events Service gesteuerte Message-driven Bean implementiert wird. Bei Angabe eines Knotennamens muss auch ein Servername angegeben werden. Der Knotenname ist ein optionaler Parameter. Als Standardwert wird der aktuelle Knoten verwendet. Geben Sie diesen Parameter nicht an, wenn Sie die Anwendung aus einem Cluster entfernen.

**serverName**

Der Name des Servers, auf dem die vom Events Service gesteuerte Message-driven Bean implementiert wird. Dieser Parameter ist erforderlich, wenn Sie die Anwendung aus einem Server entfernen. Andernfalls ist der Parameter optional. Geben Sie keinen Servernamen an, wenn Sie die Anwendung aus einem Cluster entfernen.

**clusterName**

Der Name des Clusters, in dem die vom Events Service gesteuerte Message-driven Bean implementiert wird. Geben Sie diesen Parameter nur an, wenn Sie die Anwendung aus einem Cluster entfernen.

---

## Kapitel 3. Common Event Infrastructure verwalten

In den folgenden Abschnitten werden einige Verwaltungstasks beschrieben, mit deren Hilfe die Verarbeitungsoperationen der Common Event Infrastructure-Komponenten während der Ausführung gesteuert werden können.

---

### Events Service mit der Administrationskonsole verwalten

Der Events Service kann mithilfe der webbasierten Administrationskonsole verwaltet werden.

#### Events Service mit der Administrationskonsole aktivieren und inaktivieren

Sie können den Events Service aktivieren und inaktivieren, indem Sie die Merkmale für den Events Service in der Administrationskonsole des Servers ändern.

##### Informationen zu diesem Vorgang

Wenn der Events Service aktiviert ist, wird er beim Starten des Servers automatisch gestartet.

Gehen Sie wie folgt vor, um den Events Service über die Administrationskonsole zu aktivieren oder zu inaktivieren:

##### Vorgehensweise

1. Wählen Sie eine der folgenden Methoden aus, um die entsprechende Anzeige der Administrationskonsole aufzurufen:
  - Öffnen Sie die Anzeige für den Common Event Infrastructure-Server in der Administrationskonsole.
    - Wählen Sie für Server **Server** → **Servertypen** → **WebSphere-Anwendungsserver** → *servername* → **Business Integration** → **Common Event Infrastructure** → **Server für Common Event Infrastructure** aus.
    - Wählen Sie für Cluster **Server** → **Cluster** → **WebSphere-Anwendungsservercluster** → *clustername* → **Business Integration** → **Common Event Infrastructure** → **Server für Common Event Infrastructure** aus.
  - Alternativ können Sie die Containerservices öffnen, um die folgende Task auszuführen:
    - Klicken Sie für Server auf **Server** → **Servertypen** → **WebSphere-Anwendungsserver** → *servername* → **Containerservices** → **Common Event Infrastructure Service**.
    - Klicken Sie für Cluster auf **Server** → **Cluster** → **WebSphere-Anwendungsservercluster** → *clustername* → **Cluster-Member** → *server* → **Containerservices** → **Common Event Infrastructure Service**.
2. Wählen Sie das Merkmal **Service beim Serverstart aktivieren** aus bzw. ab. Wenn das Markierungsfeld ausgewählt ist, wird der Common Event Infrastructure-Service beim Starten des Servers gestartet.
3. Speichern Sie die Konfigurationsänderungen.
4. Synchronisieren Sie den Knoten (nur in einer Network Deployment-Umgebung).
5. Führen Sie einen Neustart für den Server oder Cluster durch.

## Ereignisemitter-Factory mit der Administrationskonsole erstellen

Eine Emitter-Factory wird von Ereignisquellen zum Erstellen von Emittlern verwendet.

### Informationen zu diesem Vorgang

Die Merkmale einer Emitter-Factory wirken sich auf das Verhalten aller Emittler aus, die mit dieser Emitter-Factory erstellt werden. Sie können die Standard-Emitter-Factory verwenden oder zusätzliche Emitter-Factorys für die Ereignisquellen erstellen. Sie können auch eine zusätzliche Emitter-Factory erstellen, um einen anderen Transaktionsmodus oder eine andere Ereignisübertragung zu definieren. Gehen Sie wie folgt vor, um den Ereignisemitter, der nach dem Konfigurieren von Common Event Infrastructure (CEI) erstellt wurde, im Fenster 'Common Event Infrastructure - Ziel' anzuzeigen:

1.
  - Wählen Sie für einen einzelnen Server **Server > Servertypen > WebSphere-Anwendungsserver > *servername*** aus.
  - Wählen Sie für einen Cluster **Server > Cluster > WebSphere-Anwendungsservercluster *clustername*** aus.
2. Wählen Sie auf der Registerkarte **Konfiguration** die Optionen **Business Integration > Common Event Infrastructure > Common Event Infrastructure - Ziel** aus.
3. Sie können eine vorhandene Ereignis-Emitter-Factory aus dem Menü mit den JNDI-Namen auswählen oder im Textfeld eine angeben.

Die JNDI-Namen müssen in einen fernen Server aufgelöst werden, wenn der CEI-Server kein lokaler Server ist. Weitere Informationen zur Benennung in einer Network Deployment-Umgebung finden Sie in der WebSphere Application Server-Dokumentation. Gehen Sie wie folgt vor, um eine Emitter-Factory zu erstellen:

### Vorgehensweise

1. Klicken Sie in der Administrationskonsole des Servers auf **Serviceintegration > Common Event Infrastructure > Ereignis-Emitter-Factorys > Neu**.
2. Geben Sie die Merkmale der neuen Emitter-Factory an. Detaillierte Informationen zu diesen Merkmalen finden Sie auf der Seite für die Emitter-Factory-Einstellungen in der Onlinehilfe.
3. Speichern Sie die Konfigurationsänderungen.
4. Synchronisieren Sie den Knoten (nur in einer Network Deployment-Umgebung).
5. Führen Sie einen Neustart des Servers durch.

### Ergebnisse

Die Ereignisquellen können die konfigurierte Emitter-Factory nun zum Erstellen von Emittlern verwenden.

## Ereignisgruppe mit der Administrationskonsole erstellen

Eine Ereignisgruppe definiert eine logische Sammlung von Ereignissen, und zwar basierend auf dem Inhalt der zugehörigen Merkmaldaten. Eine Ereignisgruppe kann bei der Abfrage von Ereignissen aus dem Events Service verwendet werden; außerdem kann sie zur asynchronen Ereignisverteilung optional einem JMS-Ziel zugeordnet werden.

## Informationen zu diesem Vorgang

Gehen Sie wie folgt vor, um eine Ereignisgruppe zu erstellen:

### Vorgehensweise

1. Optional: Definieren Sie für die Ereignisgruppe mindestens ein JMS-Ziel. Eine Ereignisgruppe kann einem JMS-Thema und mindestens einer JMS-Warteschlange zugeordnet werden. Informationen dazu, wie JMS-Ziele und Verbindungsfactorys erstellt und in einen JNDI-Namensbereich eingebunden werden, finden Sie in der Dokumentation zum jeweiligen JMS-Provider.

**Sicherheit:** Wenn die WebSphere-Sicherheit aktiviert ist, muss die Konfiguration für das JMS-Ziel einen Authentifizierungsaliasnamen angeben.

2. Erstellen Sie die neue Ereignisgruppe. Klicken Sie in der Administrationskonsole des Servers auf **Serviceintegration > Common Event Infrastructure > Events Service > Ereignisservices > *ereignisservice* > Ereignisgruppen > Neu**.
3. Geben Sie die Merkmale der neuen Ereignisgruppe einschließlich des Ereigniselektors und optionaler JMS-Ziele an.
4. Speichern Sie die Konfigurationsänderungen.
5. Synchronisieren Sie den Knoten (nur in einer Network Deployment-Umgebung).
6. Führen Sie einen Neustart des Servers durch.

### Ergebnisse

Ereigniskonsumenten können die Ereignisgruppe nun beim Abfragen von Ereignissen angeben. Wenn die Ereignisverteilung in den Events Service-Einstellungen aktiviert ist, werden die zur Ereignisgruppe gehörenden Ereignisse auch für jedes in der Ereignisgruppe angegebene JMS-Ziel veröffentlicht. Ereigniskonsumenten können dann Ereignisse asynchron empfangen, indem sie die entsprechenden Ziele abonnieren.

## Ereignisfilter mit der Administrationskonsole erstellen

Ein Ereignisfilter definiert die vom Standardfilter-Plug-in verwendeten Merkmale. Dieses Plug-in wird von den Emittlern zum Filtern von Ereignissen an der Quelle verwendet.

## Informationen zu diesem Vorgang

Ein Ereignisfilter kann im Rahmen der Konfiguration einer Emitter-Factory angegeben werden. Ein mit dieser Emitter-Factory erstellter Emitter verwendet dann diesen Filter, um festzulegen, welches Ereignis an den Events Service gesendet wird.

Gehen Sie wie folgt vor, um einen Ereignisfilter zu erstellen:

### Vorgehensweise

1. Klicken Sie in der Administrationskonsole auf **Serviceintegration > Common Event Infrastructure > Ereignis-Emitter-Factorys > *emitter-factory* > Ereignisfilter > Neu**.

2. Geben Sie die Merkmale des neuen Ereignisfilters an. Detaillierte Informationen zu diesen Merkmalen finden Sie auf der Seite mit den Einstellungen für Ereignisfilter in der Onlinehilfe.
3. Speichern Sie die Konfigurationsänderungen.
4. Synchronisieren Sie den Knoten (nur in einer Network Deployment-Umgebung).
5. Führen Sie einen Neustart des Servers durch.

## Ergebnisse

Ereignisemitter können nun mithilfe des konfigurierten Filters festlegen, welche Ereignisse an den Events Service gesendet werden.

---

## Events Service über Scripting verwalten

Der Events Service kann über die Scripting-Schnittstellen verwaltet werden.

### Events Service über Scripting aktivieren

Sie können den Events Service aktivieren, indem Sie einen Verwaltungsbefehl 'AdminTask' mit dem Tool 'wsadmin' ausführen.

#### Informationen zu diesem Vorgang

Wenn der Events Service aktiviert ist, wird er beim Starten des Servers automatisch gestartet.

Gehen Sie wie folgt vor, um den Events Service mit dem Tool 'wsadmin' zu aktivieren:

#### Vorgehensweise

1. Starten Sie das Tool 'wsadmin'.
2. Verwenden Sie das Objekt 'AdminTask', um den Verwaltungsbefehl **enableEventService** auszuführen.
  - Verwendung von JACL:
 

```
enable event service at server scope
$AdminTask enableEventService { -nodeName knoten1 -serverName server1 }

enable event service at cluster scope
$AdminTask enableEventService { -clusterName cluster1 }
```
  - Verwendung von Jython:
 

```
enable event service at server scope
AdminTask.enableEventService([' -nodeName knoten1 -serverName server1]')

enable event service at cluster scope
AdminTask.enableEventService([' -clusterName cluster1]')
```

Für den Befehl **enableEventService** gibt es die folgenden Parameter:

#### nodeName

Der Name des Knotens, auf dem Events Service aktiviert werden soll. Dieser Parameter ist optional. Als Standardwert wird der aktuelle Knoten verwendet. Bei Angabe eines Knotennamens muss auch ein Servername angegeben werden. Geben Sie keinen Knoten an, wenn Sie den Events Service in einem Cluster aktivieren.

**serverName**

Der Name des Servers, auf dem Events Service aktiviert werden soll. Dieser Parameter ist erforderlich, wenn Sie einen Knotennamen angeben. Geben Sie keinen Servernamen an, wenn Sie den Events Service in einem Cluster aktivieren.

**clusterName**

Der Name des Clusters, auf dem Events Service aktiviert werden soll. Dieser Parameter ist erforderlich, wenn Sie den Events Service in einem Cluster aktivieren.

3. Führen Sie einen Neustart des Servers durch.

## Events Service über Scripting inaktivieren

Sie können den Events Service inaktivieren, indem Sie einen Verwaltungsbefehl 'AdminTask' mit dem Tool 'wsadmin' ausführen.

### Informationen zu diesem Vorgang

Wenn der Events Service inaktiviert ist, wird er beim Starten des Servers nicht automatisch gestartet.

Gehen Sie wie folgt vor, um den Events Service mit dem Tool 'wsadmin' zu inaktivieren:

### Vorgehensweise

1. Starten Sie das Tool 'wsadmin'.
2. Verwenden Sie das Objekt 'AdminTask', um den Verwaltungsbefehl **disableEventService** auszuführen.

- Verwendung von JACL:

```
disable event service at server scope
$AdminTask disableEventService { -nodeName knoten1 -serverName server1 }
```

```
disable event service at cluster scope
$AdminTask disableEventService { -clusterName cluster1 }
```

- Verwendung von Jython:

```
disable event service at server scope
AdminTask.disableEventService([' -nodeName knoten1 -serverName server1 '])
```

```
disable event service at cluster scope
AdminTask.disableEventService([' -clusterName cluster1 '])
```

Für den Befehl **disableEventService** gibt es die folgenden Parameter:

**nodeName**

Der Name des Knotens, auf dem Events Service inaktiviert werden soll. Dieser Parameter ist optional. Als Standardwert wird der aktuelle Knoten verwendet. Bei Angabe eines Knotennamens muss auch ein Servername angegeben werden. Geben Sie keinen Knoten an, wenn Sie den Events Service in einem Cluster inaktivieren.

**serverName**

Der Name des Servers, auf dem Events Service inaktiviert werden soll. Dieser Parameter ist erforderlich, wenn Sie einen Knotennamen angeben. Geben Sie keinen Servernamen an, wenn Sie den Events Service in einem Cluster inaktivieren.

### clusterName

Der Name des Clusters, auf dem Events Service inaktiviert werden soll. Dieser Parameter ist erforderlich, wenn Sie den Events Service in einem Cluster inaktivieren.

3. Führen Sie einen Neustart des Servers durch.

## Emitter-Factory über Scripting erstellen

Eine Emitter-Factory kann mit einem Jacl- oder Jython-Script erstellt werden.

### Vorbereitende Schritte

Vor dem Starten dieser Task muss das Tool 'wsadmin' aktiviert werden. Weitere Informationen hierzu finden Sie in der Dokumentation zu WebSphere Application Server.

### Informationen zu diesem Vorgang

Führen Sie die folgenden Schritte aus, um eine neue Emitter-Factory zu konfigurieren:

### Vorgehensweise

1. Ermitteln Sie die Common Event Infrastructure-Provider-ID.

- Verwendung von JACL:

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

- Verwendung von Jython:

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

Beispielausgabe:

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

2. Legen Sie die erforderlichen Attribute fest.

- Verwendung von JACL:

```
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"]
```

```
set TransactionMode to true to send each event in a new transaction
set TransactionMode [list preferredTransactionMode "false"]
```

```
set SynchronizationMode to true to use synchronous event transmission
as the preferred synchronization mode
set SynchronizationMode [list preferredSynchronizationMode "true"]
```

```
leave blank if synchronous transmission is not supported
set SyncJNDIName [list synchronousTransmissionProfileJNDIName \
"Put JNDI name of synchronous transmission profile here "]
```

```
leave blank if asynchronous transmission is not supported
set AsyncJNDIName [list asynchronousTransmissionProfileJNDIName \
"Put JNDI name of asynchronous transmission profile here "]
```

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

```
leave blank if filtering is not enabled
```



```

set FilterJNDIName [list filterFactoryJNDIName \
 "Put JNDI name of event filter here"]

custom properties include compatibility mode
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]]]

```

- Verwendung von Jython:

```

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

set TransactionMode to true to send each event in a new transaction
TransactionMode = ['preferredTransactionMode', 'false']

set SynchronizationMode to true to use synchronous event transmission
as the preferred synchronization mode
SynchronizationMode = ['preferredSynchronizationMode', 'true']

leave blank if synchronous transmission is not supported
SyncJNDIName = ['synchronousTransmissionProfileJNDIName', \
 'Put JNDI name of synchronous transmission profile here']

leave blank if asynchronous transmission is not supported
AsyncJNDIName = ['asynchronousTransmissionProfileJNDIName', \
 'Put JNDI name of asynchronous transmission profile here']

FilteringEnabled = ['filteringEnabled', 'false']

leave blank if filtering is not enabled
FilterJNDIName = ['filterFactoryJNDIName', 'Put JNDI name of event filter here']

custom properties include compatibility mode
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. Definieren Sie die Merkmale für die neue Emitter-Factory.

- Verwendung von JACL:

```

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

```

- Verwendung von Jython:

```

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

```

#### Beispielausgabe:

```

[['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. Erstellen Sie die Emitter-Factory.

- Verwendung von JACL:

```
set emitterProf [$AdminConfig create EmitterFactoryProfile \
$providerid $properties]
```

- Verwendung von Jython:

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

Beispielausgabe:

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

#### 5. Speichern Sie die Konfigurationsänderungen.

#### 6. Synchronisieren Sie den Knoten (nur in einer Network Deployment-Umgebung).

#### 7. Führen Sie einen Neustart des Servers durch.

## Ereignisgruppe über Scripting erstellen

Eine Ereignisgruppe kann mit einem JACL- oder Jython-Script erstellt werden.

### Vorbereitende Schritte

Vor dem Starten dieser Task muss das Tool 'wsadmin' aktiviert werden. Weitere Informationen hierzu finden Sie in der Dokumentation zu WebSphere Application Server.

### Informationen zu diesem Vorgang

Führen Sie die folgenden Schritte aus, um eine neue Ereignisgruppe zu konfigurieren:

### Vorgehensweise

#### 1. Ermitteln Sie die Common Event Infrastructure-Provider-ID.

- Verwendung von JACL:

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

```

- Verwendung von Jython:

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

Beispielausgabe:

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

#### 2. Rufen Sie die Liste mit den Ereignisgruppen ab.

- Verwendung von JACL:

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

- Verwendung von Jython:

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

### 3. Legen Sie die erforderlichen Attribute fest.

- Verwendung von JACL:

```
set name [list eventGroupName "EventGroupName"]

escape all '[' characters in the event selector string.
set selectorString [list eventSelectorString "Set event selector here"]

leave blank if events should not be published using JMS
set JNDIName [list topicJNDIName "Set topic JNDI name here"]

leave blank if events should not be published using JMS
set connectionFactoryJNDIName [list topicConnectionFactoryJNDIName \
 "Set topic connection factory JNDI name here"]

set persistEventsFlag [list persistEvents "true"]

custom properties include compatibility mode
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]]]
```

- Verwendung von Jython:

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

leave blank if events should not be published using JMS
JNDIName = ['topicJNDIName', 'Set topic JNDI name here']

leave blank if events should not be published using JMS
ConnectionFactoryJNDIName = ['topicConnectionFactoryJNDIName', \
 'Set topic connection factory JNDI name here']

PersistEventsFlag = ['persistEvents', 'true']

custom properties include compatibility mode
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. Definieren Sie die Merkmale für die neue Ereignisgruppe.

- Verwendung von JACL:

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

- Verwendung von Jython:

```
properties = [Name,SelectorString,JNDIName,ConnectionFactoryJNDIName, \
 PersistEventsFlag,customProperties]
```

Beispielausgabe:

```
[[['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. Erstellen Sie die Ereignisgruppe.

- Verwendung von JACL:

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

- Verwendung von Jython:

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

Beispielausgabe:

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

6. Speichern Sie die Konfigurationsänderungen.
7. Synchronisieren Sie den Knoten (nur in einer Network Deployment-Umgebung).
8. Führen Sie einen Neustart des Servers durch.

## Ereignisfilter über Scripting erstellen

Ein Ereignisfilter kann mit einem JACL- oder Jython-Script erstellt werden.

### Vorbereitende Schritte

Vor dem Starten dieser Task muss das Tool 'wsadmin' aktiviert werden. Weitere Informationen hierzu finden Sie in der Dokumentation zu WebSphere Application Server.

### Informationen zu diesem Vorgang

Führen Sie die folgenden Schritte aus, um einen neuen Ereignisfilter zu konfigurieren:

### Vorgehensweise

1. Ermitteln Sie die Common Event Infrastructure-Provider-ID.

- Verwendung von JACL:

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

- Verwendung von Jython:

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

Beispielausgabe:

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

2. Legen Sie die erforderlichen Attribute fest.

- Verwendung von JACL:

```

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"]

escape all '[' characters in the configuration string
(for example, "CommonBaseEvent\[severity=50]")
set filterConfigurationString [list filterConfigurationString \
 "Set filter configuration string here"]

```

- Verwendung von Jython:

```

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. Definieren Sie die Merkmale für den neuen Ereignisfilter.

- Verwendung von JACL:

```

set properties [list $name $jndiName $description $category \
 $filterConfigurationString]

```

- Verwendung von Jython:

```

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

```

Beispielausgabe:

```

[['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. Erstellen Sie den Ereignisfilter.

- Verwendung von JACL:

```

set filterProf [$AdminConfig create FilterFactoryProfile \
 $providerid $properties]

```

- Verwendung von Jython:

```

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

```

Beispielausgabe:

```

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

```

5. Speichern Sie die Konfigurationsänderungen.
6. Synchronisieren Sie den Knoten (nur in einer Network Deployment-Umgebung).
7. Führen Sie einen Neustart des Servers durch.

---

## Protokollierung und Traceerstellung für Common Event Infrastructure-Komponenten

Sie können die Protokollierung und Traceerstellung aktivieren, um Probleme mit einer Anwendung zu beheben, die Common Event Infrastructure verwendet.

### Informationen zu diesem Vorgang

Common Event Infrastructure-Komponenten verwenden das Java-Protokollierungsframework JSR47, das in Common Event Infrastructure-Server- und -Clientumgebungen zur Verfügung steht. Weitere Informationen zur Verwendung des Protokollierungsframeworks finden Sie in der Fehlerbehebungsdocumentation für WebSphere Application Server Network Deployment, Version 6.1.

In der folgenden Tabelle sind die Protokollfunktionsnamen aufgelistet, die von den Common Event Infrastructure-Komponenten verwendet werden.

Tabelle 2. Protokollfunktionsnamen

| Komponente                      | Protokollfunktionsname                                                                              |
|---------------------------------|-----------------------------------------------------------------------------------------------------|
| Stammname der Protokollfunktion | com.ibm.events                                                                                      |
| Ereigniskatalog                 | com.ibm.events.catalog                                                                              |
| Events Service-Unterkomponenten | com.ibm.events.access<br>com.ibm.events.bus<br>com.ibm.events.distribution<br>com.ibm.events.server |
| Standarddatenspeicher-Plug-in   | com.ibm.events.datastore                                                                            |
| Ereignisemitter                 | com.ibm.events.emitter                                                                              |
| Benachrichtigungshilfsprogramm  | com.ibm.events.notification                                                                         |
| Konfiguration                   | com.ibm.events.configuration<br>com.ibm.events.admintask                                            |
| Datenbankkonfiguration          | com.ibm.events.install.db                                                                           |
| Migration                       | com.ibm.events.migration                                                                            |
| Sonstige Dienstprogramme        | com.ibm.events.util                                                                                 |

## Ereignisdatenbank verwalten

Wenn Sie eine externe Ereignisdatenbank verwenden, sollten Sie regelmäßig die bereitgestellten Scripts zur Datenbankpflege ausführen.

### Statistiken der DB2-Ereignisdatenbank aktualisieren

Damit eine DB2-Datenbank unter Linux, UNIX oder Windows Abfragen optimieren und freien Speicherplatz finden kann, können Sie die Datenbankstatistiken durch Ausführung des Scripts **runstats** aktualisieren.

#### Informationen zu diesem Vorgang

Die regelmäßige Aktualisierung der DB2-Datenbankstatistiken wird empfohlen, vor allem wenn die folgenden Bedingungen eingetreten sind:

- Mit den Schnittstellen zum Löschen von Ereignissen des Events Service oder dem Dienstprogramm für schnelle Ereignisbereinigung des Standarddatenspeicher-Plug-ins wurden Ereignisse aus der Datenbank gelöscht.
- Eine große Anzahl von Ereignissen wurde in die Datenbank eingefügt.
- Mit dem Script **reorg** wurden Tabellen reorganisiert.
- Aus einer Tabelle wurden Indizes entfernt oder es wurden Indizes hinzugefügt.

Das Script **runstats** befindet sich im Verzeichnis *profilstammverzeichnis/event/knotenname/servername* oder *clustername/dbscripts/db2*.

#### Vorgehensweise

Führen Sie zum Aktualisieren der Datenbankstatistiken einen der folgenden Befehle aus:

- **Windows** Auf Windows-Systemen:  
runstats.bat datenbankbenutzer [password=datenbankkennwort]

- **Linux** **UNIX** Auf Linux- und UNIX-Systemen:  
`runstats.sh datenbankbenutzer [password=datenbankkennwort]`

Die folgenden Parameter werden verwendet:

*datenbankbenutzer*

Die Datenbankbenutzer-ID, die verwendet werden soll. Dieser Parameter ist erforderlich.

*datenbankkennwort*

Das Datenbankkennwort. Dieser Parameter ist optional. Wenn Sie in der Befehlszeile kein Kennwort angeben, werden Sie von der DB2-Datenbank zur Eingabe des Kennworts aufgefordert.

## Beispiel

Mit dem folgenden Befehl werden beispielsweise die DB2-Datenbankstatistiken auf einem Windows-System aktualisiert (Datenbankbenutzer-ID: dbadmin, Kennwort: meinkennwort):

```
runstats.bat dbadmin meinkennwort
```

## Tabellen der DB2-Ereignisdatenbank reorganisieren

Nach dem Löschen von Ereignissen aus einer DB2-Ereignisdatenbank unter Linux, UNIX oder Windows können Sie die Datenbanktabellen mit dem Script **reorg** reorganisieren.

### Informationen zu diesem Vorgang

Das Script **reorg** befindet sich im Verzeichnis *profilstammverzeichnis/event/knotenname/servername* oder *clustername/dbscripts/db2*.

### Vorgehensweise

Führen Sie zum Reorganisieren der Tabellen der Ereignisdatenbank einen der folgenden Befehle aus:

- **Windows** Auf Windows-Systemen:  
`reorg.bat datenbankaliasname datenbankbenutzer [password=datenbankkennwort]`
- **Linux** **UNIX** Auf Linux- und UNIX-Systemen:  
`reorg.sh datenbankaliasname datenbankbenutzer [password=datenbankkennwort]`

Die folgenden Parameter werden verwendet:

*datenbankaliasname*

Der Datenbankaliasname. Die Ereignisdatenbank muss auf dem DB2-Client katalogisiert sein; bei Ausführung des Scripts auf dem DB2-Server ist die Datenbank bereits katalogisiert.

*datenbankbenutzer*

Die Datenbankbenutzer-ID, die verwendet werden soll. Dieser Parameter ist erforderlich.

*datenbankkennwort*

Das Datenbankkennwort. Dieser Parameter ist optional. Wenn Sie in der Befehlszeile kein Kennwort angeben, werden Sie von der DB2-Datenbank zur Eingabe des Kennworts aufgefordert.

## Beispiel

Mit dem folgenden Befehl werden beispielsweise die Ereignisdatenbanktabellen auf einem Windows-System reorganisiert (Datenbankaliasname: eventdb, Benutzer-ID: dbadmin, Kennwort: meinkennwort):

```
reorg.bat eventdb dbadmin meinkennwort
```

## Nächste Schritte

Nach Ausführung des Scripts **reorg** sollten Sie mit dem Script **runstats** die Datenbankstatistiken aktualisieren. Weitere Informationen hierzu finden Sie unter „Statistiken der DB2-Ereignisdatenbank aktualisieren“ auf Seite 56.

## Ereignisse aus der Ereignisdatenbank löschen

Mithilfe der bereitgestellten Scripts können Sie große Mengen von Ereignissen aus der Ereignisdatenbank löschen.

### Informationen zu diesem Vorgang

Das Standarddatenspeicher-Plug-in enthält eine Reihe von Dienstprogrammen, die Sie regelmäßig ausführen können, um große Mengen von alten Ereignissen aus der Ereignisdatenbank zu löschen. Diese Dienstprogramme unterscheiden sich vom Events Service-Befehl `eventpurge`, der Ereignisse löscht, die bestimmte Kriterien erfüllen.

**Derby-Datenbanken:** Das Dienstprogramm für die Datenbankbereinigung wird für Derby-Ereignisdatenbanken nicht unterstützt.

Die Datenbankbereinigungsfunktion verwendet das Konzept von *Buckets*. Ein Bucket besteht aus einer Reihe von Tabellen, die zum Speichern von Ereignissen in der Ereignisdatenbank verwendet werden. Das Standarddatenspeicher-Plug-in verwendet zwei Buckets:

- Das aktive Bucket ist das Bucket, das die neuesten Ereignisse enthält; neue Ereignisse werden im aktiven Bucket gespeichert. Das aktive Bucket kann mit dem Dienstprogramm zur Datenbankbereinigung nicht bereinigt werden.
- Das inaktive Bucket enthält ältere Ereignisse. Im inaktiven Bucket gespeicherte Ereignisse können abgefragt, geändert oder gelöscht werden; normalerweise werden aber keine neuen Ereignisse im inaktiven Bucket gespeichert. Das inaktive Bucket kann mit dem Dienstprogramm zur Datenbankbereinigung bereinigt werden.

Jedes Ereignis wird in nur einem Bucket gespeichert. Aus der Sicht des Ereigniskonsumenten gibt es keinen Unterschied zwischen dem aktiven und dem inaktiven Bucket; ein Konsument kann ein bestimmtes Ereignis abfragen, ändern oder löschen, ohne zu wissen, in welchem Bucket das Ereignis gespeichert ist. Der Vorteil dieser Methode besteht darin, dass das inaktive Bucket mithilfe von datenbankspezifischen Schnittstellen bereinigt werden kann und das aktive Bucket davon nicht betroffen ist. Der normale Ereignisverkehr kann auch während der Bereinigungsoperation fortgesetzt werden.

Nach der Bereinigung des inaktiven Buckets können die Buckets ausgetauscht werden, so dass das aktive Bucket inaktiv und das inaktive Bucket aktiv wird. Die Buckets können nur dann ausgetauscht werden, wenn das inaktive Bucket leer ist.



Obwohl neue Ereignisse nur im aktiven Bucket gespeichert werden, kann es in bestimmten Fällen vorkommen, dass Ereignisse unmittelbar nach dem Austausch der Buckets im inaktiven Bucket gespeichert werden. Das Datenspeicher-Plug-in überprüft regelmäßig, welches Bucket momentan als aktiv gekennzeichnet ist; in der Zeit bis zur nächsten Überprüfung können einige Ereignisse jedoch weiterhin im inaktiven Bucket gespeichert werden. Gleichmaßen werden alle stapelweise gesendeten Ereignisse im selben Bucket gespeichert, auch wenn das Bucket noch während der Stapelverarbeitung inaktiviert wird.

Wenn Sie mit der Funktion für schnelle Bereinigung arbeiten möchten, müssen Sie festlegen, wie häufig ein Austausch der Buckets erforderlich ist bzw. das inaktive Bucket bereinigt werden muss, und dabei Faktoren wie Ereignisverkehr, Speicherplatz, Archivierungsanforderungen usw. berücksichtigen.

### **Status des aktiven Buckets der Ereignisdatenbank anzeigen oder ändern**

Der Status des aktiven Buckets gibt an, welches Bucket derzeit verfügbar bzw. nicht verfügbar ist.

#### **Vorgehensweise**

Mit dem Befehl `eventbucket` können Sie den Status des aktiven Buckets anzeigen oder ändern:

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

Dieser Befehl enthält die folgenden Parameter:

#### **-status**

Mit diesem Parameter werden Informationen zur gegenwärtigen Bucketkonfiguration angezeigt, und zwar einschließlich der Einstellung für das aktive Bucket und des Bucketprüfintervalls. (Das Bucketprüfintervall legt fest, wie häufig das Datenspeicher-Plug-in überprüft, welches Bucket aktiv ist.)

#### **-change**

Mit diesem Parameter können das aktive und das inaktive Bucket ausgetauscht werden. Sie können diesen Parameter erst verwenden, wenn das inaktive Bucket leer ist.

### **Inaktives Bucket der Ereignisdatenbank bereinigen**

Die Methode, die zur Bereinigung des inaktiven Buckets verwendet wird, hängt von der Datenbanksoftware ab.

#### **Informationen zu diesem Vorgang**

**Derby-Datenbanken:** Das Dienstprogramm für schnelle Bereinigung wird für Derby-Ereignisdatenbanken nicht unterstützt.

#### **Inaktives Bucket für eine DB2-Ereignisdatenbank bereinigen (Linux-, UNIX- oder Windows-Systeme):**

Auf Linux-, UNIX- und Windows-Systemen wird das Dienstprogramm zur Datenbankbereinigung für eine DB2-Datenbank als Shell-Script oder Stapeldatei implementiert.

## Vorgehensweise

Führen Sie zum Bereinigen des inaktiven Buckets einen der folgenden Befehle aus:

- Führen Sie zum Bereinigen des inaktiven Buckets den folgenden Befehl aus:  
`fastpurge datenbankaliasname datenbankbenutzer [password=datenbankkennwort]  
[copydir=kopierverz]`

Die Parameter für diesen Befehl lauten wie folgt:

### *datenbankaliasname*

Der Datenbankaliasname. Die Ereignisdatenbank muss auf dem DB2-Client katalogisiert sein; bei Ausführung des Scripts auf dem DB2-Server ist die Datenbank bereits katalogisiert.

### *datenbankbenutzer*

Die Datenbankbenutzer-ID, die zur Herstellung der Verbindung mit der Ereignisdatenbank verwendet wird.

### *datenbankkennwort*

Das Kennwort für die angegebene Benutzer-ID. Dieser Parameter ist optional. Wenn Sie kein Kennwort angeben, werden Sie von der DB2-Datenbank zur Eingabe des Kennworts aufgefordert.

### *kopierverz*

Der Pfad zu einem Verzeichnis, das für die vom Ladedienstprogramm generierten Dateien verwendet wird. Dieser Parameter ist nur dann erforderlich, wenn Sie für die Ereignisdatenbank die aktualisierende Wiederherstellung aktiviert haben (Datenbankkonfigurationseinstellungen für LOGRETAIN oder USEREXIT sind aktiviert). Standardmäßig wird die aktualisierende Wiederherstellung von der Ereignisdatenbank nicht verwendet.

## Inaktives Bucket für eine DB2-Ereignisdatenbank bereinigen:

Das Dienstprogramm zur Datenbankbereinigung für eine DB2-Ereignisdatenbank wird mit dem DB2-Ladedienstprogramm implementiert.

## Informationen zu diesem Vorgang

Gehen Sie wie folgt vor, um das inaktive Bucket zu bereinigen:

### Vorgehensweise

1. Führen Sie den Befehl `eventbucket` aus, um das inaktive Bucket zu identifizieren (Bucket 0 oder Bucket 1).
2. Laden Sie die entsprechende Dienstprogrammsteuerdatei hoch. Diese Dateien werden während der Datenbankkonfiguration generiert und befinden sich im Verzeichnis `/event/dbscripts/db2zos`. Laden Sie eine der folgenden Dateien hoch:
  - Wenn Bucket 0 inaktiv ist, laden Sie die Datei `'fastpurge00.ctl'` hoch.
  - Wenn Bucket 1 inaktiv ist, laden Sie die Datei `'fastpurge01.ctl'` hoch.

**Hochladeformat:** Die Steuerdatei muss mit einem festen Satzlängenformat und einer logischen Satzlänge von 80 hochgeladen werden.

3. Wechseln Sie in das ISPF-DB2I-Hauptmenü und wählen Sie die Option **Utilities** aus.
4. Geben Sie die folgenden Informationen an:

| Feld               | Wert                                                        |
|--------------------|-------------------------------------------------------------|
| Function           | EDITJCL                                                     |
| Utility            | LOAD                                                        |
| Statement Data Set | Der Name des Datensatzes mit der hochgeladenen Steuerdatei. |
| LISTDEF            | NO                                                          |
| Template           | NO                                                          |

5. Drücken Sie die Eingabetaste, um mit der nächsten Anzeige fortzufahren.
6. Geben Sie in das Eingabefeld 'recdsn' den Namen des Datensatzes mit der hochgeladenen Steuerdatei ein.
7. Drücken Sie die Eingabetaste. Daraufhin wird das JCL-Script zur Bereinigung des inaktiven Buckets generiert.
8. Drücken Sie die Eingabetaste, um die Ausgabenachrichten zu löschen.
9. Bearbeiten Sie das generierte JCL-Script nach Bedarf.
10. Übergeben Sie das JCL-Script.

### Ergebnisse

Das generierte JCL-Script kann wiederverwendet werden. Nachdem Sie die Scripts zur Bereinigung der beiden Buckets (0 und 1) erstellt haben, müssen Sie diese Prozedur nicht mehr wiederholen.

### Inaktives Bucket für eine DB2-Ereignisdatenbank bereinigen (iSeries-Systeme):

Auf iSeries-Systemen wird das Dienstprogramm zur Datenbankbereinigung für eine DB2-Ereignisdatenbank als gespeicherte Prozedur implementiert.

### Informationen zu diesem Vorgang

Gehen Sie wie folgt vor, um das inaktive Bucket zu bereinigen:

#### Vorgehensweise

1. Stellen Sie mit einer Terminalanwendung, die ein 5250-Terminal emuliert (z. B. IBM Personal Communications), eine Verbindung zum iSeries-System her.
2. Melden Sie sich mit einer Benutzer-ID an, die über ausreichende Berechtigungen zur Ausführung der gespeicherten Prozedur für schnelle Bereinigung verfügt. Es wird empfohlen, die zur Erstellung der Ereignisdatenbank verwendete Benutzer-ID anzugeben.
3. Führen Sie den Befehl **strsql** aus.
4. Geben Sie die folgende SQL-Anweisung ein:

```
call sammlung/fast_purge()
```

Dabei ist *sammlung* der Name der Sammlung, die die Ereignisdatenbank enthält. Der Name der Standardsammlung lautet event:

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

### Inaktives Bucket für eine Oracle-Ereignisdatenbank bereinigen:

Das Dienstprogramm zur Datenbankbereinigung für eine Oracle-Ereignisdatenbank wird als gespeicherte Prozedur implementiert.

## Vorbereitende Schritte

Bevor Sie das Dienstprogramm zur Datenbankbereinigung für eine Oracle-Datenbank verwenden können, muss SQL\*Plus auf dem Oracle-Client installiert werden. Außerdem muss der Client für die Kommunikation mit der Oracle-Datenbank konfiguriert sein (die Datei 'tnsnames.ora' muss ordnungsgemäß konfiguriert sein). Weitere Informationen hierzu finden Sie in der Oracle-Dokumentation.

## Vorgehensweise

Führen Sie zum Bereinigen des inaktiven Buckets die gespeicherte Prozedur mit SQL\*Plus aus:

```
sqlplus verbindungszeichenfolge @fastpurge.sql
```

Die Datei 'fastpurge.sql' wird im selben Verzeichnis wie die anderen Scripts gespeichert, die während der Datenbankkonfiguration generiert wurden (Standardverzeichnis: *profilstammverzeichnis/databases/event/knotenname/servername* oder *clustername/dbscripts/oracle*).

Der Parameter *verbindungszeichenfolge* ist die Oracle-Verbindungszeichenfolge. Verwenden Sie dieselbe Datenbankbenutzer-ID wie zur Erstellung der Ereignisdatenbank.

## Inaktives Bucket für eine Informix-Ereignisdatenbank bereinigen:

Das Dienstprogramm zur Datenbankbereinigung für eine Informix-Ereignisdatenbank wird als gespeicherte Prozedur implementiert.

## Vorbereitende Schritte

Zur Verwendung des Dienstprogramms für eine Informix-Datenbank müssen Sie den Informix-Befehl 'dbaccess' in einer Umgebung ausführen, die mithilfe einer Source-Operation ordnungsgemäß für die Informix-Umgebungsvariablen vorbereitet wurde. Weitere Informationen hierzu finden Sie in der Informix-Dokumentation.

## Vorgehensweise

1. Führen Sie zum Bereinigen des inaktiven Buckets die gespeicherte Prozedur mit 'dbaccess' aus:

```
dbaccess - fastpurge.sql
```

Die Datei 'fastpurge.sql' wird im selben Verzeichnis wie die anderen Scripts gespeichert, die während der Datenbankkonfiguration generiert wurden (Standardverzeichnis: *profilstammverzeichnis/databases/event/knoten/server/dbscripts/informix*).

2. Führen Sie zum Bereinigen des Verbindungspools das Script 'eventpurgepool' aus:

- **Windows** Auf Windows-Systemen: eventpurgepool.bat
- **Linux** **UNIX** Auf Linux- oder UNIX-Systemen: eventpurgepool.sh

## Inaktives Bucket für eine SQL Server-Ereignisdatenbank bereinigen:

Das Dienstprogramm zur Datenbankbereinigung für eine SQL Server-Ereignisdatenbank wird als gespeicherte Prozedur implementiert.

## Vorbereitende Schritte

Zur Verwendung des Dienstprogramms für eine SQL Server-Datenbank muss das Dienstprogramm 'osql' auf dem SQL Server-Client installiert sein. Weitere Informationen hierzu finden Sie in der SQL Server-Dokumentation.

**Anmerkung:** Zur Ausführung der gespeicherten Prozedur für die Datenbankbereinigung darf das Dienstprogramm 'isql' nicht verwendet werden.

## Vorgehensweise

Führen Sie zum Bereinigen des inaktiven Buckets die gespeicherte Prozedur mit dem Dienstprogramm 'osql' aus:

```
osql -Sservername -Udatenbankbenutzer -Pdatenbankkennwort
-dereignisdatenbank -Q"fast_purge"
```

Die Datei 'fast\_purge' wird im selben Verzeichnis wie die anderen Scripts gespeichert, die während der Datenbankkonfiguration generiert wurden (Standardverzeichnis: *profilstammverzeichnis/databases/event/knoten/server/dbscripts/sqlserver*). Die Parameter für diesen Befehl lauten wie folgt:

*servername*

Der Name des Datenbankservers.

*datenbankbenutzer*

Die Datenbankbenutzer-ID, die zur Herstellung der Verbindung mit der Ereignisdatenbank verwendet wird.

*datenbankkennwort*

Das Kennwort für die angegebene Benutzer-ID. Dieser Parameter ist optional. Wenn Sie kein Kennwort angeben, werden Sie vom Dienstprogramm 'osql' zur Eingabe des Kennworts aufgefordert.

*ereignisdatenbank*

Der Name der Ereignisdatenbank (normalerweise event).

## Prüfintervall für das Bucket der Ereignisdatenbank ändern

Das Bucketprüfintervall gibt an, wie häufig das Datenspeicher-Plug-in überprüft, welches Bucket aktiv ist. Dieser Wert ist als benutzerdefiniertes Merkmal in den Einstellungen des Datenspeichers definiert.

## Informationen zu diesem Vorgang

Das standardmäßig definierte Bucketprüfintervall beträgt 5 Minuten (300 Sekunden). Ein kürzeres Intervall verringert die Wahrscheinlichkeit, dass Ereignisse nach dem Austausch im inaktiven Bucket gespeichert werden, hat aber möglicherweise negative Auswirkungen auf die Leistung.

Gehen Sie wie folgt vor, um das Bucketprüfintervall zu ändern:

## Vorgehensweise

1. Klicken Sie in der Administrationskonsole auf **Serviceintegration > Common Event Infrastructure > Events Service > Ereignisservices > ereignisservice > Ereignisdatenspeicher**.
2. Ändern Sie den Wert des Merkmals 'BucketCheckInterval', um das Bucketprüfintervall in Sekunden anzugeben.

## **Ergebnisse**

Die Änderung wird beim nächsten Neustart des Servers wirksam.

## Kapitel 4. Sicherer Zugriff auf Common Event Infrastructure-Funktionen

Mit der deklarativen Sicherheit auf Methodenebene von WebSphere können Sie sicheren Zugriff auf die Common Event Infrastructure-Funktionen gewährleisten.

Common Event Infrastructure definiert sechs Sicherheitsaufgabenbereiche, die jeweils einer zugehörigen Gruppe von Funktionen zugeordnet sind. Diese Sicherheitsaufgabenbereiche steuern den Zugriff auf Programmierschnittstellen und Befehle.

In der folgenden Tabelle werden die Sicherheitsaufgabenbereiche und die Benutzertypen beschrieben, die den einzelnen Aufgabenbereichen zugeordnet sind.

*Tabelle 3. Sicherheitsaufgabenbereiche und Benutzertypen*

| Sicherheitsaufgabenbereich | Benutzertypen                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| eventCreator               | <p>Ereignisquellen, die Ereignisse an einen Emitter übergeben müssen und dabei synchrone EJB-Aufrufe verwenden. Dieser Aufgabenbereich bietet Zugriff die folgenden Schnittstellen:</p> <ul style="list-style-type: none"> <li>• Emitter.sendEvent()</li> <li>• Emitter.sendEvents()</li> <li>• Befehl <b>eventemit</b></li> </ul> <p>Der Aufgabenbereich 'eventCreator' schränkt den Zugriff auf die Ereignisübergabe nur dann ein, wenn der Emitter für die Ereignisübertragung synchrone EJB-Aufrufe verwendet. Wenn der Emitter für die Ereignisübertragung asynchrones JMS-Messaging verwendet, müssen Sie mithilfe der JMS-Sicherheit den Zugriff auf das zum Übergeben von Ereignissen verwendete Ziel einschränken.</p> |
| eventUpdater               | <p>Ereigniskonsumenten, die in der Ereignisdatenbank gespeicherte Ereignisse aktualisieren müssen. Dieser Aufgabenbereich bietet Zugriff die folgenden Schnittstellen:</p> <ul style="list-style-type: none"> <li>• EventAccess.updateEvents()</li> <li>• EventAccess.eventExists()</li> <li>• EventAccess.queryEventByGlobalInstanceId()</li> <li>• EventAccess.queryEventsByAssociation()</li> <li>• EventAccess.queryEventsByEventGroup()</li> <li>• Befehl <b>eventquery</b></li> </ul>                                                                                                                                                                                                                                     |
| eventConsumer              | <p>Ereigniskonsumenten, die in der Ereignisdatenbank gespeicherte Ereignisse abfragen müssen. Dieser Aufgabenbereich bietet Zugriff die folgenden Schnittstellen:</p> <ul style="list-style-type: none"> <li>• EventAccess.eventExists()</li> <li>• EventAccess.queryEventByGlobalInstanceId()</li> <li>• EventAccess.queryEventsByAssociation()</li> <li>• EventAccess.queryEventsByEventGroup()</li> <li>• Befehl <b>eventquery</b></li> </ul>                                                                                                                                                                                                                                                                                |

Tabelle 3. Sicherheitsaufgabenbereiche und Benutzertypen (Forts.)

| Sicherheitsaufgabenbereich | Benutzertypen                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| eventAdministrator         | <p>Ereigniskonsumenten, die in der Ereignisdatenbank gespeicherte Ereignisse abfragen, aktualisieren und löschen müssen. Dieser Aufgabenbereich bietet Zugriff die folgenden Schnittstellen:</p> <ul style="list-style-type: none"> <li>• EventAccess.purgeEvents()</li> <li>• EventAccess.eventExists()</li> <li>• EventAccess.queryEventByGlobalInstanceId()</li> <li>• EventAccess.queryEventsByAssociation()</li> <li>• EventAccess.queryEventsByEventGroup()</li> <li>• EventAccess.updateEvents()</li> <li>• Emitter.sendEvent()</li> <li>• Emitter.sendEvents()</li> <li>• Befehl <b>eventquery</b></li> <li>• Befehl <b>eventpurge</b></li> <li>• Befehl <b>eventemit</b></li> <li>• Befehl <b>eventbucket</b></li> </ul>                                                                                                                                                                                      |
| catalogReader              | <p>Ereigniskataloganwendungen, die Ereignisdefinitionen aus dem Ereigniskatalog abrufen müssen. Dieser Aufgabenbereich bietet Zugriff die folgenden Schnittstellen:</p> <ul style="list-style-type: none"> <li>• EventCatalog.getAncestors()</li> <li>• EventCatalog.getChildren()</li> <li>• EventCatalog.getDescendants()</li> <li>• EventCatalog.getEventDefinition()</li> <li>• EventCatalog.getEventDefinitions()</li> <li>• EventCatalog.getEventExtensionNamesForSourceCategory()</li> <li>• EventCatalog.getEventExtensionToSourceCategoryBindings()</li> <li>• EventCatalog.getParent()</li> <li>• EventCatalog.getRoot()</li> <li>• EventCatalog.getSourceCategoriesForEventExtension()</li> <li>• Befehl <b>eventcatalog</b> (Parameter -listdefinitions)</li> <li>• Befehl <b>eventcatalog</b> (Parameter -listcategories)</li> <li>• Befehl <b>eventcatalog</b> (Parameter -exportdefinitions)</li> </ul> |
| catalogAdministrator       | <p>Ereigniskataloganwendungen, die Ereignisdefinitionen im Ereigniskatalog erstellen, aktualisieren, löschen oder abrufen müssen. Dieser Aufgabenbereich bietet Zugriff auf alle Methoden der Schnittstelle 'EventCatalog' und alle Funktionen des Befehls <b>eventcatalog</b>. Da Änderungen am Ereigniskatalog die Generierung von Ereignissen zur Folge haben können, bietet dieser Aufgabenbereich auch Zugriff auf die Schnittstellen für die Ereignisübergabe.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                               |

Die vom Events Service gesteuerte Message-driven Bean wird mit der Benutzeridentität des Servers ausgeführt. Wenn Sie mit der asynchronen JMS-Übertragung Ereignisse an den Events Service übergeben und die auf Methoden basierende Sicherheit aktiviert haben, müssen Sie diese Benutzeridentität dem Aufgabenbereich 'eventCreator' zuordnen.



**Sicherheit:** Wenn die Ereignisquelle mit aktivierter Java-Sicherheit ausgeführt wird und Sie eigene GUIDs (Globally Unique Identifiers) generieren möchten, müssen Sie die Richtliniendatei ändern, um die entsprechende Verarbeitung zu aktivieren. Fügen Sie die folgenden Einträge hinzu:

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



---

## Kapitel 5. Fehlerbehebung für Common Event Infrastructure

Diese folgenden Abschnitte enthalten Fehlerbehebungsinformationen für den Events Service, und zwar basierend auf der Task oder Aktivität, die Sie bei Auftreten des Fehlers ausgeführt haben.

---

### Fehler beim Start

Fehler beim Start werden mit dem Common Event Infrastructure-Server behoben.

#### Events Service startet nicht (Nachricht CEIDS0058E)

*Der Events Service wird nicht gestartet und gibt die Nachricht CEIDS0058E in die WebSphere-Protokolldatei aus.*

##### Fehlerursache

Der Events Service verwendet SQL-Anweisungen, die mit dem Benutzernamen qualifiziert sind. Dieser Fehler weist darauf hin, dass der vom Events Service verwendete Benutzername zum Herstellen einer Verbindung zur Ereignisdatenbank nicht der Benutzer-ID entspricht, die zur Erstellung der Datenbank verwendet wurde.

##### Fehlerbehebung

Die zum Herstellen einer Verbindung zur Ereignisdatenbank verwendete Benutzer-ID muss mit der Benutzer-ID zur Erstellung der Ereignisdatenbank identisch sein. Gehen Sie wie folgt vor, um den Fehler zu beheben:

1.
  - Wählen Sie für einen einzelnen Server **Server > Anwendungsserver > servername** aus.
  - Wählen Sie für einen Cluster **Server > Cluster > clustername** aus.
2. Wählen Sie auf der Registerkarte **Konfiguration** die Optionen **Business Integration > Common Event Infrastructure > Server für Common Event Infrastructure** aus.
3. Ändern Sie die angegebene Benutzer und das zugehörige Kennwort so, dass sie mit der Benutzer-ID und dem Kennwort zur Erstellung der Datenbank identisch sind.
4. Speichern Sie die Konfigurationsänderungen.
5. Führen Sie einen Neustart des Servers durch.

---

### Fehler beim Senden von Ereignissen

Fehler beim Senden von Ereignissen werden mit dem Common Event Infrastructure-Server behoben.

#### Fehler beim Senden eines Ereignisses (Nachricht CEIDS0060E)

*Bei der Ereignisquelle tritt beim Senden eines Ereignisses ein Fehler auf. In der WebSphere-Protokolldatei wird die Nachricht CEIDS0060E angezeigt.*

## Fehlerursache

Der Events Service verwendet für die Zuordnung von Common Base Event-Elementen und -Attributen zu Datenbanktabellen und -spalten in der Ereignisdatenbank gespeicherte Metadaten. Diese Informationen werden aus der Datenbank gelesen, wenn eine Anwendung nach dem Start zum ersten Mal versucht, den Events Service zu verwenden.

Die Metadatentabellen werden bei der Erstellung der Ereignisdatenbank mit Daten gefüllt. Dieser Fehler tritt auf, wenn die Tabellen die erforderlichen Metadaten während der Ausführung nicht enthalten.











## Fehlerbehebung

Zur Behebung dieses Fehlers müssen die erforderlichen Metadaten erneut erstellt werden. Bei der Erstellung der Ereignisdatenbank generiert der Verwaltungsbefehl für die Datenbankkonfiguration ein Datenbankskript, mit dem die Metadaten zu einem späteren Zeitpunkt erneut gefüllt werden können. Der Name dieses Skripts hängt vom Datenbanktyp ab:

| Datenbanktyp        | Scriptname         |
|---------------------|--------------------|
| Derby               | ins_metadata.derby |
| DB2                 | ins_metadata.db2   |
| Informix            | ins_metadata.sql   |
| Oracle              | ins_metadata.ora   |
| SQL Server          | ins_metadata.mssql |
| DB2 UDB for iSeries | ins_metadata.db2   |

Das Skript wird standardmäßig im Verzeichnis *profilstammverzeichnis/dbscripts/CEI-datenbankname* erstellt. Sie können dieses Skript jederzeit ausführen.

Verwenden Sie zur erneuten Erstellung der Metadaten den entsprechenden SQL-Prozessor für die Ausführung des Skripts:

-    DB2: db2
-    Oracle: SQL\*Plus
-    Informix: dbaccess
-  SQL Server: osql

Führen Sie einen Neustart für den Server durch, nachdem Sie die Metadaten erneut gefüllt haben.

## Fehler beim Senden eines Ereignisses (ServiceUnavailableException)

*Bei der Ereignisquellenanwendung tritt beim Senden eines Ereignisses an den Ereignisserver ein Fehler auf. Die Protokolldatei enthält eine Ausnahmebedingung 'ServiceUnavailableException' mit der Nachricht 'A communication failure occurred while attempting to obtain an initial context with the provider URL'.*

## Fehlerursache

Dieser Fehler weist darauf hin, dass die Ereignisquellenanwendung keine Verbindung zum Ereignisserver herstellen kann. Dieser Umstand kann auf eine der folgenden Bedingungen zurückzuführen sein:

- Der Ereignisserver ist nicht aktiv.
- Die Ereignisquellenanwendung ist nicht für die korrekte JNDI-Provider-URL konfiguriert.

## Fehlerbehebung

Führen Sie die folgenden Schritte aus, um diesen Fehler zu beheben:

1. Wechseln Sie in das Verzeichnis `profilstammverzeichnis/bin` und führen Sie den Befehl **serverStatus** aus, um den Status des Ereignisserver zu überprüfen:  
`serverStatus servername`
2. Wenn der Ereignisserver nicht aktiv ist, starten Sie ihn mit dem Befehl **startServer**:  
`startServer servername`
3. Überprüfen Sie den Hostnamen und den RMI-Port (RMI - Remote Method Invocation) für den Server, der die Anwendung enthält, die keine Verbindung zum Ereignisserver herstellen kann. Stellen Sie sicher, dass dieselben Werte in der für die Ereignisquellenanwendung konfigurierten JNDI-URL angegeben sind. Wenn sich der CEI-Server auf einem anderen Server befindet, muss die JNDI mit diesem fernen Implementierungsziel aufgelöst werden.

## Fehler beim Senden eines Ereignisses (NameNotFoundException)

*Bei der Ereignisquellenanwendung tritt beim Senden eines Ereignisses an den Events Service ein Fehler auf. Die Protokolldatei enthält eine Ausnahmebedingung 'NameNotFoundException' mit einer Nachricht ähnlich der folgenden: 'Die erste Komponente im Name events/configuration/emitter/Default wurde nicht gefunden'.*

## Fehlerursache

Dieser Fehler weist darauf hin, dass der Events Service nicht verfügbar und der Fehler auf eine der folgenden Bedingungen zurückzuführen ist:

- Der Events Service wurde nicht implementiert.
- Der Events Service ist inaktiviert.

## Fehlerbehebung

Gehen Sie wie folgt vor, um Events Service zu implementieren:

1. Starten Sie das Tool 'wsadmin'.
2. Verwenden Sie das Objekt 'AdminTask', um den Verwaltungsbefehl `deployEventService` auszuführen.
3. Führen Sie einen Neustart des Servers durch.

Gehen Sie wie folgt vor, um den Events Service mit dem Tool 'wsadmin' zu aktivieren:

1. Starten Sie das Tool 'wsadmin'.
2. Verwenden Sie das Objekt 'AdminTask', um den Verwaltungsbefehl `enableEventService` auszuführen.

3. Führen Sie einen Neustart des Servers durch.

Gehen Sie wie folgt vor, um den Events Service über die Administrationskonsole zu aktivieren:

1. Klicken Sie auf **Anwendungen** → **Anwendungstypen** → **WebSphere-Enterprise-Anwendungen** → *server* → **Containerservices** → **Common Event Infrastructure Service**.
2. Wählen Sie das Merkmal **Service beim Serverstart aktivieren** aus.
3. Klicken Sie auf **OK**, um die Änderungen zu speichern.
4. Führen Sie einen Neustart des Servers durch.

## Fehler beim Senden eines Ereignisses (Nachricht CEIEM0025E)

*Bei der Ereignisquellenanwendung tritt beim Senden eines Ereignisses an den Ereignisserver ein Fehler auf. Die Protokolldatei enthält die Ausnahmebedingung 'DuplicateGlobalInstanceIdException'.*

### Fehlerursache

Dieser Fehler weist darauf hin, dass der Emmitter das Ereignis übergeben, der Events Service das Ereignis jedoch zurückgewiesen hat, weil bereits ein anderes Ereignis mit derselben globalen Instanz-ID vorhanden ist. Jedes Ereignis muss über eine eindeutige globale Instanz-ID verfügen, die über das Merkmal 'globalInstanceId' angegeben wird.

### Fehlerbehebung

Zur Behebung dieses Fehlers müssen Sie eine der folgenden Aktionen ausführen:

- Stellen Sie sicher, dass die Ereignisquellenanwendung für jedes Ereignis eine eindeutige globale Instanz-ID generiert.
- Lassen Sie das Merkmal 'globalInstanceId' für die übergebenen Ereignisse leer. Der Emmitter generiert dann automatisch eine eindeutige ID für jedes Ereignis.

## Fehler beim Senden eines Ereignisses (Nachricht CEIEM0034E)

*Bei der Ereignisquelle tritt beim Senden eines Ereignisses an den Events Service ein Fehler auf. Die Protokolldatei enthält eine Ausnahmebedingung 'EmitterException' mit der Nachricht 'The JNDI lookup of a JMS queue failed because the JNDI name defined in the emitter profile is not bound in the JNDI'.*

### Fehlerursache

Dieser Fehler weist darauf hin, dass die vom Emmitter verwendete JMS-Übertragungskonfiguration mindestens eine JMS-Ressource angibt, die nicht in der JMS-Konfiguration definiert ist.

### Fehlerbehebung

Gehen Sie wie folgt vor, um den Fehler zu beheben:

1. Klicken Sie in der Administrationskonsole auf **Serviceintegration** > **Common Event Infrastructure** > **Ereignis-Emmitter-Factories** > *emitter-factory* > **Einstellungen**

gen für JMS-Übertragung. Stellen Sie sicher, dass die JMS-Übertragung für die Emitter-Factory angezeigt wird, die von der Ereignisquellenanwendung verwendet wird.

- Überprüfen Sie die Werte, die für die Merkmale **JNDI-Name der Warteschlange** und **JNDI-Name der Verbindungs-Fabrik für Warteschlangen** angegeben wurden. Stellen Sie sicher, dass die angegebenen JNDI-Namen im JNDI-Namensbereich vorhanden und gültige JMS-Objekte sind. Gegebenenfalls müssen Sie Änderungen an diesen Merkmalen vornehmen oder die erforderlichen JMS-Ressourcen erstellen.

## Ereignis ist ungültig (Nachricht CEIEM0027E)

*Die Ereignisquelle versucht, ein Ereignis zu senden, aber der Emitter übergibt es nicht an den Events Service und gibt die Nachricht CEIEM0027E ('The emitter did not send the event to the event server because the Common Base Event is not valid') an die Protokolldatei aus.*

### Fehlerursache

Diese Nachricht weist darauf hin, dass mindestens ein Ereignismerkmal Daten enthält, die nicht mit der Common Base Event-Spezifikation übereinstimmen. Ereignisdaten können aus verschiedenen Gründen ungültig sein, z. B. wenn die folgenden Bedingungen nicht zutreffen:

- Die globale Instanz-ID muss mindestens 32 Zeichen lang sein, darf aber 64 Zeichen nicht überschreiten.
- Die Wertigkeit muss im Bereich zwischen 0 und 70 liegen.

### Fehlerbehebung

Gehen Sie wie folgt vor, um den Fehler zu beheben:

- Stellen Sie anhand der detaillierten Ausnahmebedingungsnachricht in der Protokolldatei fest, welches Ereignismerkmal nicht gültig ist. Die folgende Nachricht gibt beispielsweise an, dass die Länge der globalen Instanz-ID (ABC) nicht gültig ist:

```
Exception: org.eclipse.hyades.logging.events.cbe.ValidationException
: IWAT0206E The length of the identifier in the specified Common
Base Event property is outside the valid range of 32 to 64
characters.
Property: CommonBaseEvent.globalInstanceId
Value: ABC
```

- Korrigieren Sie den Ereignisinhalt an der Quelle so, dass er mit der Common Base Event-Spezifikation konform ist.
- Übergeben Sie das Ereignis erneut.

## Synchronisationsmodus wird nicht unterstützt (Nachricht CEIEM0015E)

*Die Ereignisquelle versucht, ein Ereignis zu senden, aber der Emitter übergibt es nicht an den Events Service und gibt die Nachricht CEIEM0015E ('The emitter does not support the specified synchronization mode') in die Protokolldatei aus.*

## Fehlerursache

Dieser Fehler weist darauf hin, dass die beim Senden des Ereignisses von der Ereignisquelle übergebenen Parameter einen Synchronisationsmodus angeben, der vom Emmitter nicht unterstützt wird. Dieser Umstand kann auf eine der folgenden Bedingungen zurückzuführen sein:

- Die Ereignisquelle gibt einen Synchronisationsmodus an, der nicht gültig ist. In diesem Fall wird eine Ausnahmebedingung 'IllegalArgumentException' mit der Nachricht 'Synchronization mode *mode* is not valid' ausgegeben.
- Die Ereignisquelle gibt einen Synchronisationsmodus an, der vom Emmitter nicht unterstützt wird. In diesen Fall wird eine Ausnahmebedingung 'SynchronizationModeNotSupportedException' mit der Nachricht 'The emitter does not support the specified synchronization mode: *mode*' ausgegeben.

## Fehlerbehebung

Wenn die Ausnahmebedingungsricht darauf hinweist, dass die Ereignisquelle einen ungültigen Synchronisationsmodus angibt (IllegalArgumentException), müssen Sie den Methodenaufruf überprüfen, der das Ereignis sendet. Stellen Sie sicher, dass die Methodenparameter einen der gültigen Synchronisationsmodi angeben:

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

Diese Konstanten werden über die Schnittstelle 'com.ibm.events.emitter.SynchronizationMode' definiert.

Wenn die Ausnahmebedingungsricht darauf hinweist, dass der angegebene Synchronisationsmodus vom Emmitter nicht unterstützt wird (SynchronizationModeNotSupportedException), müssen Sie die Konfiguration der Emmitter-Factory überprüfen:

1. Klicken Sie in der Administrationskonsole auf **Serviceintegration > Common Event Infrastructure > Ereignis-Emmitter-Factorys > emitter-factory**. Stellen Sie sicher, dass die von der Ereignisquellenanwendung verwendete Emmitter-Factory angezeigt wird.
2. Stellen Sie anhand der Emmitter-Factory-Einstellungen fest, welche Synchronisationsmodi unterstützt werden:
  - Wenn das Merkmal **Ereignisserviceübertragung unterstützen** ausgewählt ist, wird der synchrone Modus unterstützt.
  - Wenn das Merkmal **JMS-Übertragung unterstützen** ausgewählt ist, wird der asynchrone Modus unterstützt.

**Transaktionsmodi abfragen:** Eine Ereignisquelle kann die unterstützten Transaktionsmodi für einen bestimmten Emmitter mit der Methode 'isSynchronizationModeSupported()' programmgesteuert abfragen. Weitere Informationen hierzu finden Sie in der Javadoc-API-Dokumentation.

3. Wenn der Emmitter den angegebenen Synchronisationsmodus nicht unterstützt, müssen Sie entweder die Emmitter-Factory-Konfiguration oder die Ereignisquelle so ändern, dass ein unterstützter Synchronisationsmodus verwendet wird.



## Transaktionsmodus wird nicht unterstützt (Nachricht CEIEM0016E)

*Die Ereignisquelle versucht, ein Ereignis zu senden, aber der Emmitter übergibt es nicht an den Events Service und gibt die Nachricht CEIEM0016E ('The emitter does not support the specified transaction mode') in die Protokolldatei aus.*

### Fehlerursache

Dieser Fehler weist darauf hin, dass die beim Senden des Ereignisses von der Ereignisquelle übergebenen Parameter einen Transaktionsmodus angeben, der vom Emmitter nicht unterstützt wird. Dieser Umstand kann auf eine der folgenden Bedingungen zurückzuführen sein:

- Die Ereignisquelle gibt einen Transaktionsmodus an, der nicht gültig ist.
- Die Ereignisquelle gibt einen Synchronisationsmodus an, der von der Emmitterumgebung nicht unterstützt wird. Transaktionen werden nur in einem Java EE-Container unterstützt.

### Fehlerbehebung

Überprüfen Sie zur Behebung dieses Fehlers den Methodenaufruf, der das Ereignis sendet, und stellen Sie sicher, dass die Methodenparameter den richtigen Transaktionsmodus angeben:

- Wenn der Emmitter in einem Java EE-Container ausgeführt wird, müssen Sie sicherstellen, dass die Methodenparameter einen der gültigen Transaktionsmodi angeben:
  - `TransactionMode.NEW`
  - `TransactionMode.SAME`
  - `TransactionMode.DEFAULT`

Diese Konstanten werden über die Schnittstelle `com.ibm.events.emitter.TransactionMode` definiert.

- Wenn der Emmitter nicht in einem Java EE-Container ausgeführt wird, müssen Sie sicherstellen, dass die Methodenparameter `'TransactionMode.DEFAULT'` angeben.

---

## Fehler beim Empfangen oder Abfragen von Ereignissen

Fehler im Zusammenhang mit dem Empfangen und Abfragen von Ereignissen werden mit dem Common Event Infrastructure-Server behoben.

## Fehler beim Abfragen von Ereignissen (Nachricht CEIDS0060E)

*Beim Ereigniskonsumenten tritt beim Abfragen von Ereignissen aus dem Events Service ein Fehler auf. In der WebSphere-Protokolldatei wird die Nachricht CEIDS0060E angezeigt.*

### Fehlerursache

Der Events Service verwendet für die Zuordnung von Common Base Event-Elementen und -Attributen zu Datenbanktabellen und -spalten in der Ereignisdatenbank gespeicherte Metadaten. Diese Informationen werden aus der Datenbank gelesen, wenn eine Anwendung nach dem Start zum ersten Mal versucht, den Events Service zu verwenden.

Die Metadatentabellen werden bei der Erstellung der Ereignisdatenbank mit Daten gefüllt. Dieser Fehler tritt auf, wenn die Tabellen die erforderlichen Metadaten während der Ausführung nicht enthalten.

## Fehlerbehebung

Zur Behebung dieses Fehlers müssen die erforderlichen Metadaten erneut erstellt werden. Bei der Erstellung der Ereignisdatenbank generiert der Verwaltungsbefehl für die Datenbankkonfiguration ein Datenbankskript, mit dem die Metadaten zu einem späteren Zeitpunkt erneut gefüllt werden können. Der Name dieses Scripts hängt vom Datenbanktyp ab:

| Datenbanktyp        | Scriptname         |
|---------------------|--------------------|
| Derby               | ins_metadata.derby |
| DB2                 | ins_metadata.db2   |
| Informix            | ins_metadata.sql   |
| Oracle              | ins_metadata.ora   |
| SQL Server          | ins_metadata.mssql |
| DB2 UDB for iSeries | ins_metadata.db2   |

Das Script wird standardmäßig im Verzeichnis *profilstammverzeichnis/dbscripts/CEI-datenbankname* erstellt. Sie können dieses Script jederzeit ausführen.

Verwenden Sie zur erneuten Erstellung der Metadaten den entsprechenden SQL-Prozessor für die Ausführung des Scripts:

- Windows Linux UNIX DB2: db2
- Windows Linux UNIX Oracle: SQL\*Plus
- Windows Linux UNIX Informix: dbaccess
- Windows SQL Server: osql

Führen Sie einen Neustart für den Server durch, nachdem Sie die Metadaten erneut gefüllt haben.

## Ereignisse werden nicht im persistenten Datenspeicher gespeichert

*Die Ereignisquellenanwendung übergibt Ereignisse erfolgreich an den Emitter, aber wenn eine Ereignisquelle Abfragen für die Ereignisse ausführt, befinden sie sich nicht im persistenten Datenspeicher.*

### Fehlerursache

Dieser Fehler weist darauf hin, dass der Emitter keine Ereignisse an den Events Service sendet oder der Events Service die Ereignisse nicht im persistenten Datenspeicher speichert. Mögliche Ursachen:

- Der persistente Datenspeicher ist für den Events Service nicht aktiviert.
- Die Ereignisse gehören zu keiner Ereignisgruppe, die für Ereignispersistenz konfiguriert ist.
- Die Ereignisse werden vom Emitter ausgefiltert.

## Fehlerbehebung

Gehen Sie wie folgt vor, um sicherzustellen, dass der persistente Datenspeicher für den Events Service aktiviert ist:

1. Klicken Sie in der Administrationskonsole auf **Serviceintegration > Common Event Infrastructure > Events Service > Ereignisservices > ereignisservice**.
2. Stellen Sie sicher, dass das Markierungsfeld **Ereignisdatspeicher aktivieren** ausgewählt ist.
3. Klicken Sie auf **OK**, um eventuelle Änderungen zu speichern.

Gehen Sie wie folgt vor, um sicherzustellen, dass die Ereignisgruppe für Ereignispersistenz konfiguriert ist:

1. Klicken Sie in der Administrationskonsole auf **Serviceintegration > Common Event Infrastructure > Events Service > Ereignisservices > ereignisservice > Ereignisgruppen > ereignisgruppe**.
2. Stellen Sie sicher, dass das Markierungsfeld **Ereignisse im Ereignisdatspeicher permanent speichern** ausgewählt ist.
3. Klicken Sie auf **OK**, um eventuelle Änderungen zu speichern.

**Mehrere Ereignisgruppen:** Ein Ereignis kann zu mehreren Ereignisgruppen gehören. Wenn eine gültige Ereignisgruppe für Persistenz konfiguriert und der Datenspeicher aktiviert ist, wird das Ereignis im Datenspeicher gespeichert.

Gehen Sie wie folgt vor, um die Filtereinstellungen zu überprüfen:

1. Klicken Sie in der Administrationskonsole auf **Serviceintegration > Common Event Infrastructure > Ereignis-Emitter-Factorys > emitter-factory > Ereignisfilter**. (Stellen Sie sicher, dass die Einstellungen für die Emitter-Factory angezeigt werden, die von der Ereignisquellenanwendung verwendet wird.)
2. Überprüfen Sie, ob die Zeichenfolge für die Filterkonfiguration Ereignisse ausschließt, die an die Konsumenten gesendet werden sollen. Sollte dies der Fall sein, können Sie entweder die Zeichenfolge für die Filterkonfiguration oder die Ereignisdaten ändern, damit die Ereignisse nicht ausgefiltert werden.
3. Klicken Sie auf **OK**, um eventuelle Änderungen zu speichern.

## Ereignisse werden von Konsumenten nicht empfangen (keine Fehlernachricht)

*Die Ereignisquellenanwendung übergibt Ereignisse erfolgreich an den Emitter, aber die Ereignisse werden von den Konsumenten über die JMS-Schnittstelle nicht empfangen.*

### Fehlerursache

Mögliche Fehlerursachen:

- Die Ereignisverteilung ist für den Events Service nicht aktiviert.
- Die Ereignisse werden vom Emitter ausgefiltert.
- Die Ereignisse werden vom Benachrichtigungshilfsprogramm ausgefiltert.
- Der Ereigniskonsument gibt nicht die korrekte Ereignisgruppe an.
- Die JMS-Verbindung wurde nicht gestartet.

### Fehlerbehebung

Die Fehlerbehebungsmaßnahme hängt von der zugrunde liegenden Ursache ab.

- **Gehen Sie wie folgt vor, um sicherzustellen, dass die Ereignisverteilung für den Events Service aktiviert ist:**
  1. Klicken Sie in der Administrationskonsole auf **Serviceintegration > Common Event Infrastructure > Events Service > Ereignisservices > ereignisservice**.
  2. Wenn das Merkmal **Ereignisverteilung aktivieren** nicht ausgewählt ist, wählen Sie das Markierungsfeld aus.
  3. Klicken Sie auf **OK**, um eventuelle Änderungen zu speichern.
- **Gehen Sie wie folgt vor, um die Ereignisfiltereinstellungen für den Emittter zu überprüfen:**
  1. Klicken Sie in der Administrationskonsole auf **Serviceintegration > Common Event Infrastructure > Ereignis-Emittter-Factorys > emitter-factory > Ereignisfilter**. (Stellen Sie sicher, dass die Einstellungen für die Emittter-Factory angezeigt werden, die von der Ereignisquellenanwendung verwendet wird.)
  2. Überprüfen Sie, ob die Zeichenfolge für die Filterkonfiguration Ereignisse ausschließt, die an die Konsumenten gesendet werden sollen. Sollte dies der Fall sein, können Sie entweder die Zeichenfolge für die Filterkonfiguration oder die Ereignisdaten ändern, damit die Ereignisse nicht ausgefiltert werden.
  3. Klicken Sie auf **OK**, um eventuelle Änderungen zu speichern.
- **Gehen Sie wie folgt vor, um die Ereignisfiltereinstellungen für das Benachrichtigungshilfsprogramm zu überprüfen:**
  1. Überprüfen Sie die Ereigniskonsumentenanwendung, um festzustellen, ob mithilfe der Methode 'NotificationHelper.setEventSelector' ein Ereignisselektor für das Benachrichtigungshilfsprogramm angegeben wurde.
  2. Wurde ein Ereignisselektor angegeben, müssen Sie sicherstellen, dass das zu empfangende Ereignis nicht ausgeschlossen wurde. (Ein auf null gesetzter Ereignisselektor übergibt alle Ereignisse.)
- **Gehen Sie wie folgt vor, um die vom Ereigniskonsumenten angegebene Ereignisgruppe zu überprüfen:**
  1. Klicken Sie in der Administrationskonsole auf **Serviceintegration > Common Event Infrastructure > Events Service > Ereignisservices > ereignisservice > Ereignisgruppen**. Die Tabelle zeigt eine Liste aller für den Events Service definierten Ereignisgruppen an.
  2. Wählen Sie die Ereignisgruppe aus, die der Ereigniskonsument subskribiert.
  3. Suchen Sie das Merkmal **Zeichenfolge für Ereignisselektor**.
  4. Überprüfen Sie, ob der angegebene Ereignisselektor mit dem Inhalt des Ereignisses übereinstimmt, das empfangen werden soll. Sollte das nicht der Fall sein, können Sie eine der folgenden Änderungen vornehmen:
    - Ändern Sie den Ereignisselektor so, dass das Ereignis in die Ereignisgruppe eingefügt wird.
    - Ändern Sie die Ereignisdaten so, dass das Ereignis mit der Ereignisgruppe übereinstimmt.
    - Ändern Sie den Ereigniskonsumenten so, dass er eine andere Ereignisgruppe subskribiert, die das Ereignis enthält.
- **Gehen Sie wie folgt vor, um die JMS-Verbindung zu starten:**

Verwenden Sie im Ereigniskonsumenten die Methode 'QueueConnection.start()' oder die Methode 'TopicConnection.start()', bevor Sie versuchen, Ereignisse zu empfangen.

## Ereignisse werden von Konsumenten nicht empfangen (NameNotFoundException)

Die Ereignisquellenanwendung übergibt Ereignisse erfolgreich an den Emittent, aber die Ereignisse werden über die JMS-Schnittstelle nicht in den Konsumenten veröffentlicht. Die Protokolldatei enthält eine Ausnahmebedingung 'NameNotFoundException'.

### Fehlerursache

Dieser Fehler weist darauf hin, dass die Ereignisgruppenkonfiguration mindestens eine JMS-Ressource angibt, die nicht vorhanden ist.

### Fehlerbehebung

Gehen Sie wie folgt vor, um den Fehler zu beheben:

1. Klicken Sie in der Administrationskonsole auf **Serviceintegration > Common Event Infrastructure > Events Service > Ereignisservices > ereignisservice > Ereignisgruppen > ereignisgruppe**.

**Mehrere Ereignisgruppen:** Ein Ereignis kann zu mehreren Ereignisgruppen gehören.

2. Überprüfen Sie die Werte der Merkmale **JNDI-Name des Topic** und **JNDI-Name der Verbindungs-Factory für Topics**. Überprüfen Sie, ob die angegebenen JMS-Ressourcen vorhanden sind. Gegebenenfalls müssen Sie die erforderlichen Ressourcen mit der Konfigurationsschnittstelle des jeweiligen JMS-Providers erstellen.

## Ereignisgruppe mit erweiterten Datenelementen enthält keine Ereignisse

Der Benutzer hat eine Ereignisgruppe definiert, die Prädikate für erweiterte Datenelemente angibt. Abfragen für diese Ereignisgruppe geben jedoch nicht die erwarteten Ereignisse zurück.

### Fehlerursache

Die Ereignisdaten können gültige XML-Daten sein, die aber nicht mit der Common Base Event-Spezifikation übereinstimmen. Dies kann zu unerwarteten Ergebnissen ohne Fehlernachrichten führen.

Nehmen wir ein Ereignis mit folgendem Inhalt:

```
<?xml version="1.0" encoding="ASCII"?>
<!-- Event that will match the XPath expression CommonBaseEvent
 [@globalInstanceId] -->
<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"
/>
<extendedDataElement name="color" type="string">
 <values>red</values>
</extendedDataElement>
</CommonBaseEvent>

```

Dieses Ereignis enthält ein einzelnes erweitertes Datenelement mit einem einzigen untergeordneten Element.

Nehmen wir ferner eine Ereignisgruppendefinition, die mit der folgenden XPath-Ereignisselektorzeichenfolge konfiguriert ist:

```

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

```

Dieser Ereignisselektor stimmt nicht mit dem Ereignis überein, weil die XML-Definition des Ereignisses einen Rechtschreibfehler enthält. In den Ereignisdaten wird das Element `extendedDataElements` irrtümlicherweise als `extendedDataElement` angegeben. Da das XML-Format korrekt ist, kommt es zu keinem Fehler; das Element wird stattdessen als Element any behandelt, das nicht durchsucht werden kann.

## Fehlerbehebung

Stellen Sie sicher, dass die XML-Daten für übergebene Ereignisse mit der Common Base Event-Spezifikation übereinstimmen.

## Fehler beim Abfragen einer Ereignisgruppe (Nachricht CEIES0048E)

*Bei der Ereigniskonsumentenanzwendung tritt beim Abfragen von Ereignissen aus einer Ereignisgruppe ein Fehler auf. Die Protokolldatei enthält eine Ausnahmebedingung (EventGroupNotDefinedException) mit der Nachricht CEIES0048E ('The event group is not defined in the event group list that the event server instance is using').*

## Fehlerursache

Dieser Fehler weist darauf hin, dass die Ereigniskonsumentenanzwendung eine Abfrage mit der EventAccess-Bean ausgeführt hat. Der Konsument hat jedoch einen Ereignisgruppennamen angegeben, der keiner vorhandenen Ereignisgruppe entspricht.

## Fehlerbehebung

Gehen Sie wie folgt vor, um den Fehler zu beheben:

1. Klicken Sie in der Administrationskonsole auf **Serviceintegration > Common Event Infrastructure > Events Service > Ereignisservices > ereignisservice > Ereignisgruppen**. Die Tabelle zeigt eine Liste aller für den Events Service definierten Ereignisgruppen an.
2. Stellen Sie sicher, dass die Ereignisquelle in den Parametern des Methodenaufrufs der Abfrage einen definierten Ereignisgruppennamen angibt.

---

## Sonstige Fehler

Sonstige Fehler werden mit dem Common Event Infrastructure-Server behoben.

### Abfrage für Ereigniskatalogmuster schlägt auf Windows-System fehl

*Mit dem Befehl `eventcatalog` wird versucht, eine Musterabfrage für Ereignisdefinitionen auf einem Windows-System auszuführen. Beispiel: `eventcatalog -listdefinitions -name EVENT% -pattern`. Die erwarteten Ergebnisse werden nicht zurückgegeben.*

#### Fehlerursache

Das Prozentzeichen (%) ist in der Befehlszeilenschnittstelle von Windows ein reserviertes Zeichen und wird nicht ordnungsgemäß an den Befehl `eventcatalog` übergeben.

#### Fehlerbehebung

Auf Windows-Systemen muss als Escape-Zeichen %% eingegeben werden:

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





---

## Kapitel 6. Common Event Infrastructure-Befehle

Von Common Event Infrastructure verwendete Befehle zum Erstellen und Entfernen von Ereignisdatenbanken und -services.

Befehle zur Erstellung von Ereignisdatenquellen auf bestimmten Datenbanken:

- DB2 Multiplatform - „Befehl 'configEventServiceDB2DB'“
- DB2 iSeries - „Befehl 'configEventServiceDB2iSeriesDB'“ auf Seite 86
- DB2 z/OS - „Befehl 'configEventServiceDB2ZOSDB'“ auf Seite 88
- Derby - „Befehl 'configEventServiceDerbyDB'“ auf Seite 91
- Informix - „Befehl 'configEventServiceInformixDB'“ auf Seite 93
- Oracle - „Befehl 'configEventServiceOracleDB'“ auf Seite 95
- SQL Server - „Befehl 'configEventServiceSQLServerDB'“ auf Seite 98

Vom Server verwendete generische Common Event Infrastructure-Befehle für den Events Service:

- Events Service-Anwendung implementieren - „Befehl 'deployEventService'“ auf Seite 101
- Message-driven Bean der Events Service-Anwendung implementieren - „Befehl 'deployEventServiceMdb'“ auf Seite 102
- Von Common Event Infrastructure für das Messaging verwendeten Authentifizierungsaliasnamen erstellen - „Befehl 'setEventServiceJmsAuthAlias'“ auf Seite 104
- Events Service beim Neustart des Servers aktivieren - „Befehl 'enableEventService'“ auf Seite 105
- Events Service beim Neustart des Servers inaktivieren - „Befehl 'disableEventService'“ auf Seite 106
- Events Service-Status zurückgeben - „Befehl 'showEventServiceStatus'“ auf Seite 107
- Events Service-Anwendung entfernen - „Befehl 'removeEventService'“ auf Seite 109
- Message-driven Bean der Events Service-Anwendung entfernen - „Befehl 'removeEventServiceMdb'“ auf Seite 110

Befehle zum Entfernen von Ereignisdatenquellen auf bestimmten Datenbanken:

- DB2 Multiplatform — „Befehl 'removeEventServiceDB2DB'“ auf Seite 111
- DB2 iSeries - „Befehl 'removeEventServiceDB2iSeriesDB'“ auf Seite 113
- DB2 z/OS - „Befehl 'removeEventServiceDB2ZOSDB'“ auf Seite 114
- Derby - „Befehl 'removeEventServiceDerbyDB'“ auf Seite 115
- Informix - „Befehl 'removeEventServiceInformixDB'“ auf Seite 117
- Oracle - „Befehl 'removeEventServiceOracleDB'“ auf Seite 118
- SQL Server - „Befehl 'removeEventServiceSQLServerDB'“ auf Seite 119

---

### Befehl 'configEventServiceDB2DB'

Verwenden Sie den Befehl `configEventServiceDB2DB`, um Common Event Infrastructure bei Verwendung einer DB2-Datenbank zu konfigurieren.

## Zweck

Der Befehl `configEventServiceDB2DB` ist ein Verwaltungsbefehl von Common Event Infrastructure und für das Objekt 'AdminTask' verfügbar. Verwenden Sie diesen Befehl, um die Events Service-Datenbank und Datenquellen für DB2 auf einem Server oder in einem Cluster zu erstellen. Weitere Informationen zu dem Objekt 'AdminTask' enthält die Dokumentation für WebSphere Application Server Network Deployment, Version 6.1.

## Parameter

### **-createDB**

Der Befehl generiert die DDL-Datenbankscrippts und erstellt die Datenbank, wenn für diesen Parameter der Wert `true` definiert ist. Mit dem Befehl werden nur die DLL-Datenbankscrippts generiert, wenn für diesen Parameter der Wert `false` definiert ist. Zum Erstellen der Datenbank muss der aktuelle Server bereits für die Ausführung der Datenbankbefehle konfiguriert sein. Der Standardwert lautet `false`, sofern keine Angabe gemacht wurde.

### **-overrideDataSource**

Optionales Ausgabeverzeichnis für das Datenbankscrippts. Wenn dieser Parameter angegeben ist, generiert der Befehl die Scrippts für die Events Service-Datenbank in dem angegebenen Verzeichnis. Wenn das Verzeichnis nicht mit einem vollständigen Pfad angegeben wurde, erstellt der Befehl das angegebene Verzeichnis unter '*profilstammverzeichnis/bin*'. Wenn dieser Parameter nicht angegeben ist, wird das Standardausgabeverzeichnis für Datenbankscrippts *profilstammverzeichnis/databases/event/knoten/server/dbscrippts/datenbanktyp* verwendet.

### **-nodeName**

Der Name des Knotens, der den Server enthält, auf dem die Datenquelle für Events Service erstellt werden soll. Bei Angabe dieses Parameters muss auch der Parameter `serverName` angegeben werden. Dieser Parameter darf nicht angegeben werden, wenn der Parameter `clusterName` angegeben wird.

### **-serverName**

Der Name des Servers, auf dem die Datenquelle für Events Service erstellt werden soll. Wenn dieser Parameter ohne Angabe des Parameters `nodeName` definiert wird, verwendet der Befehl den Knotennamen des aktuellen WebSphere-Profiles. Dieser Parameter darf nicht angegeben werden, wenn der Parameter `clusterName` angegeben wird.

### **-clusterName**

Der Name des Clusters, in dem die Datenquelle für Events Service erstellt werden soll. Bei Angabe dieses Parameters dürfen die Parameter `serverName` und `nodeName` nicht definiert werden. Dieser Parameter darf nicht angegeben werden, wenn die Parameter `serverName` und `nodeName` angegeben werden.

### **-jdbcClassPath**

Der Pfad zum JDBC-Treiber. Geben Sie nur den Pfad zur Treiberdatei an. Der Pfadname darf den Dateinamen selbst nicht enthalten. Dieser Parameter ist erforderlich.

### **-dbName**

Der DB2-Knotenname (maximal 8 Zeichen). Dieser Knoten muss bereits katalogisiert und für die Kommunikation mit dem DB2-Server konfiguriert sein. Dieser Parameter muss definiert werden, wenn der aktuelle Server als DB2-Client konfiguriert ist und für den Parameter `createDB` der Wert `true` definiert ist.

**-dbHostName**

Der Hostname des Servers, auf dem der Datenbankserver installiert ist. Dieser Parameter ist erforderlich.

**-dbPort**

Der Port der DB2-Instanz. Als Standardwert wird '50000' verwendet, sofern keine Angabe gemacht wurde.

**-dbName**

Der Name der zu erstellenden Datenbank. Als Standardname wird event verwendet, sofern keine Angabe gemacht wurde.

**-dbUser**

Die DB2-Benutzer-ID mit der Berechtigung zum Erstellen und Löschen der Datenbanken. Als Standardwert wird db2inst1 verwendet, sofern keine Angabe gemacht wurde.

**-dbPassword**

Das DB2-Kennwort. Dieser Parameter ist erforderlich.

**-outputScriptDir**

Optionales Ausgabeverzeichnis für das Datenbanksript. Wenn dieser Parameter angegeben ist, generiert der Befehl die Scripts für die Events Service-Datenbank in dem angegebenen Verzeichnis. Wenn das Verzeichnis nicht mit einem vollständigen Pfad angegeben wurde, erstellt der Befehl das angegebene Verzeichnis unter '*profilstammverzeichnis/bin*'. Wenn dieser Parameter nicht angegeben ist, wird das Standardausgabeverzeichnis für Datenbanksripts '*profilstammverzeichnis/databases/event/knoten/server/dbscripts/db2*' verwendet.

## Muster

**Syntaxbeispiel für Stapelmodus:**

- Verwendung von JACL:

```
$AdminTask configEventServiceDB2DB {-createdB true
 -overrideDataSource true
 -nodeName knotenname
 -serverName servername
 -jdbcClassPath c:\sqllib\java
 -dbUser db2inst1
 -dbPassword datenbankkennwort
 -dbHostName hostname
 -dbPort 50000 }
```

- Verwendung von Jython-Zeichenfolge:

```
AdminTask.configEventServiceDB2DB(['-createdB true
 -overrideDataSource true
 -nodeName knotenname
 -serverName servername
 -jdbcClassPath c:\sqllib\java
 -dbUser db2inst1
 -dbPassword datenbankkennwort
 -dbHostName hostname
 -dbPort 50000]')
```

- Verwendung von Jython-Liste:

```
AdminTask.configEventServiceDB2DB(['-createdB', 'true',
 '-overrideDataSource', 'true',
 '-nodeName', 'knotenname',
 '-serverName', 'servername',
 '-jdbcClassPath', 'c:\sqllib\java',
```

```
'-dbUser', 'db2inst1',
'-dbPassword', 'datenbankkennwort',
'-dbHostName', 'hostname',
'-dbPort', '50000 '])
```

### Syntaxbeispiel für interaktiven Modus:

- Verwendung von JACL:  
`$AdminTask configEventServiceDB2DB -interactive`
- Verwendung von Jython-Zeichenfolge:

**Anmerkung:** **Vista** **Windows 7** Das Produkt verwendet eine Jython-Version, die die Betriebssysteme Microsoft Windows 2003, Windows Vista und Windows 7 nicht unterstützt.

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

- Verwendung von Jython-Liste:  
`AdminTask.configEventServiceDB2DB(['-interactive'])`

---

## Befehl 'configEventServiceDB2iSeriesDB'

Verwenden Sie den Befehl `configEventServiceDB2iSeriesDB`, um Common Event Infrastructure bei Verwendung einer DB2 für iSeries-Datenbank zu konfigurieren.

### Zweck

Der Befehl `configEventServiceDB2iSeriesDB` ist ein Verwaltungsbefehl von Common Event Infrastructure und für das Objekt 'AdminTask' verfügbar. Verwenden Sie diesen Befehl zum Generieren der DDL-Datenbankskripts, die dann auf fernen Datenbankservern verwendet werden, zum Erstellen der Events Service-Datenbank für DB2 iSeries auf einem lokalen Server und zum Erstellen von Datenquellen auf einem Server oder in einem Cluster. Weitere Informationen zu dem Objekt 'AdminTask' enthält die Dokumentation für WebSphere Application Server Network Deployment, Version 6.1.

**Anmerkung:** **Vista** **Windows 7** Das Produkt verwendet eine Jython-Version, die die Betriebssysteme Microsoft Windows 2003, Windows Vista und Windows 7 nicht unterstützt.

### Parameter

#### **-createDB**

Der Befehl generiert die DDL-Datenbankskripts und erstellt die Datenbank, wenn für diesen Parameter der Wert `true` definiert ist. Mit dem Befehl werden nur die DLL-Datenbankskripts generiert, wenn für diesen Parameter der Wert `false` definiert ist. Zum Erstellen der Datenbank muss der aktuelle Server bereits für die Ausführung der Datenbankbefehle konfiguriert sein. Der Standardwert lautet `false`, sofern keine Angabe gemacht wurde.

#### **-overrideDataSource**

Wenn für diesen Parameter der Wert `true` definiert ist, entfernt der Befehl alle vorhandenen Datenquellen für Events Service für den angegebenen Geltungsbereich, bevor eine neue Datenquelle erstellt wird. Wenn für diesen Parameter der Wert `false` definiert ist, erstellt der Befehl keine Events Service-Datenquelle für den angegebenen Geltungsbereich, wenn eine andere Events Service-Datenquelle für den gleichen Geltungsbereich gefunden wird. Der Standardwert lautet `false`, sofern keine Angabe gemacht wurde.

**-nodeName**

Der Name des Knotens, der den Server enthält, auf dem die Datenquelle für Events Service erstellt werden soll. Bei Angabe dieses Parameters muss auch der Parameter **serverName** angegeben werden. Dieser Parameter darf nicht angegeben werden, wenn der Parameter **clusterName** angegeben wird.

**-serverName**

Der Name des Servers, auf dem die Datenquelle für Events Service erstellt werden soll. Wenn dieser Parameter ohne Angabe des Parameters **nodeName** definiert wird, verwendet der Befehl den Knotennamen des aktuellen WebSphere-Profils. Dieser Parameter darf nicht angegeben werden, wenn der Parameter **clusterName** angegeben wird.

**-clusterName**

Der Name des Clusters, in dem die Datenquelle für Events Service erstellt werden soll. Bei Angabe dieses Parameters dürfen die Parameter **serverName** und **nodeName** nicht definiert werden. Dieser Parameter darf nicht angegeben werden, wenn die Parameter **serverName** und **nodeName** angegeben werden.

**-toolboxJdbcClassPath**

Der Pfad zum JDBC-Treiber für IBM Toolbox for Java DB2. Geben Sie nur den Pfad zur Treiberdatei an. Der Pfadname darf den Dateinamen selbst nicht enthalten. Sie müssen entweder diesen Parameter oder den Parameter **jdbcClassPath** angeben.

**Anmerkung:** Bei Erstellung der Datenbank auf einem iSeries-Server muss der Pfadparameter **toolboxJdbcClassPath** angegeben werden.

**-nativeJdbcClassPath**

Der Pfad zum nativen JDBC-Treiber für DB2 für iSeries. Geben Sie nur den Pfad zur Treiberdatei an. Der Pfadname darf den Dateinamen selbst nicht enthalten. Sie müssen entweder diesen Parameter oder den Parameter **toolboxJdbcClassPath** angeben.

**Anmerkung:** Bei Verwendung eines iSeries-Servers zum Erstellen der Datenbank auf einem Nicht-iSeries-Server muss der Parameter **nativeJdbcClassPath** angegeben werden. Geben Sie als Treiber DB2 Universal Driver an, wenn Sie die Datenbank auf DB2 Universal Database for Multiplatforms oder DB2 Universal Database für z/OS erstellen. Geben Sie als Treiber einen Treiber des Typs 4 an, wenn Sie die Datenbank auf Informix, Oracle oder Microsoft SQL Server erstellen.

**-dbHostName**

Der Hostname des Servers, auf dem der DB2 für iSeries-Datenbankserver installiert ist. Dieser Parameter ist erforderlich, wenn Sie den JDBC-Treiber für IBM Toolbox for Java DB2 verwenden.

**-dbName**

Der Name der DB2 für iSeries-Datenbank. Als Standardname wird \*LOCAL verwendet, sofern keine Angabe gemacht wurde.

**-collection**

Der Name der SQL-Datensammlung für die DB2 für iSeries-Bibliothek. Die maximale Länge für den Namen der Datensammlung beträgt 10 Zeichen. Der Standardwert ist eine leere Zeichenfolge, sofern keine Angabe gemacht wurde.

**-dbUser**

Die DB2-Benutzer-ID mit der Berechtigung zum Erstellen und Löschen der Datenbanken. Dieser Parameter ist erforderlich.

### **-dbPassword**

Das Kennwort für die Datenbankbenutzer-ID. Dieser Parameter ist erforderlich.

### **-outputScriptDir**

Optionales Ausgabeverzeichnis für das Datenbankskript. Wenn dieser Parameter angegeben ist, generiert der Befehl die Skripts für die Events Service-Datenbank in dem angegebenen Verzeichnis. Wenn das Verzeichnis nicht mit einem vollständigen Pfad angegeben wurde, erstellt der Befehl das angegebene Verzeichnis unter '*profilstammverzeichnis/bin*'. Wenn dieser Parameter nicht angegeben ist, wird das Standardausgabeverzeichnis für Datenbankskripts '*profilstammverzeichnis/databases/event/knoten/server/dbscripts/db2iseries*' verwendet.

## **Muster**

### **Syntaxbeispiel für Stapelmodus:**

- Verwendung von JACL:

```
$AdminTask configEventServiceDB2iSeriesDB {createDB true
 -overrideDataSource true
 -nodeName knotenname
 -serverName servername
 -dbUser db2user
 -dbPassword datenbankkennwort
 -nativeJdbcClassPath /mein_DB2-klassenpfad
 -collection event}
```

- Verwendung von Jython-Zeichenfolge:

```
AdminTask.configEventServiceDB2iSeriesDB(['-createDB true
 -overrideDataSource true
 -nodeName knotenname
 -serverName servername
 -nativeJdbcClassPath /mein_DB2-klassenpfad
 -collection event'])
```

- Verwendung von Jython-Liste:

```
AdminTask.configEventServiceDB2iSeriesDB(['-createDB', 'true',
 '-overrideDataSource', 'true',
 '-nodeName', 'knotenname',
 '-serverName', 'servername',
 '-nativeJdbcClassPath', '/mein_DB2-klassenpfad',
 '-collection', 'event'])
```

### **Syntaxbeispiel für interaktiven Modus:**

- Verwendung von JACL:

```
$AdminTask configEventServiceDB2iSeriesDB -interactive
```

- Verwendung von Jython-Zeichenfolge:

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

- Verwendung von Jython-Liste:

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

---

## **Befehl 'configEventServiceDB2ZOSDB'**

Verwenden Sie den Befehl `configEventServiceDB2ZOSDB`, um Common Event Infrastructure bei Verwendung einer DB2 für z/OS-Datenbank zu konfigurieren.

### **Zweck**

Der Befehl `configEventServiceDB2ZOSDB` ist ein Verwaltungsbefehl von Common Event Infrastructure und für das Objekt 'AdminTask' verfügbar. Verwenden Sie die

sen Befehl, um die Events Service-Datenbank und Datenquellen für DB2 z/OS auf einem Server oder in einem Cluster zu erstellen. Weitere Informationen zu dem Objekt 'AdminTask' enthält die Dokumentation für WebSphere Application Server Network Deployment, Version 6.1.

**Anmerkung:** **Vista** **Windows 7** Das Produkt verwendet eine Jython-Version, die die Betriebssysteme Microsoft Windows 2003, Windows Vista und Windows 7 nicht unterstützt.

## Parameter

### **-createDB**

Der Befehl generiert die DDL-Datenbankskripts und erstellt die Datenbank, wenn für diesen Parameter der Wert `true` definiert ist. Mit dem Befehl werden nur die DLL-Datenbankskripts generiert, wenn für diesen Parameter der Wert `false` definiert ist. Zum Erstellen der Datenbank muss der aktuelle Server bereits für die Ausführung der Datenbankbefehle konfiguriert sein. Der Standardwert lautet `false`, sofern keine Angabe gemacht wurde.

### **-overrideDataSource**

Wenn für diesen Parameter der Wert `true` definiert ist, entfernt der Befehl alle vorhandenen Datenquellen für Events Service für den angegebenen Geltungsbereich, bevor eine neue Datenquelle erstellt wird. Wenn für diesen Parameter der Wert `false` definiert ist, erstellt der Befehl keine Events Service-Datenquelle für den angegebenen Geltungsbereich, wenn eine andere Events Service-Datenquelle für den gleichen Geltungsbereich gefunden wird. Der Standardwert lautet `false`, sofern keine Angabe gemacht wurde.

### **-nodeName**

Der Name des Knotens, der den Server enthält, auf dem die Datenquelle für Events Service erstellt werden soll. Bei Angabe dieses Parameters muss auch der Parameter **serverName** angegeben werden. Dieser Parameter darf nicht angegeben werden, wenn der Parameter **clusterName** angegeben wird.

### **-serverName**

Der Name des Servers, auf dem die Datenquelle für Events Service erstellt werden soll. Wenn dieser Parameter ohne Angabe des Parameters **nodeName** definiert wird, verwendet der Befehl den Knotennamen des aktuellen WebSphere-Profiles. Dieser Parameter darf nicht angegeben werden, wenn der Parameter **clusterName** angegeben wird.

### **-clusterName**

Der Name des Clusters, in dem die Datenquelle für Events Service erstellt werden soll. Bei Angabe dieses Parameters dürfen die Parameter **serverName** und **nodeName** nicht definiert werden. Dieser Parameter darf nicht angegeben werden, wenn die Parameter **serverName** und **nodeName** angegeben werden.

### **-jdbcClassPath**

Der Pfad zum JDBC-Treiber. Geben Sie nur den Pfad zur Treiberdatei an. Der Pfadname darf den Dateinamen selbst nicht enthalten. Dieser Parameter ist erforderlich.

### **-dbHostName**

Der Hostname des Servers, auf dem die Datenbank installiert ist. Dieser Parameter ist erforderlich.

### **-dbPort**

Der Port für die DB2 für z/OS-Instanz. Es wird der Standardwert '5027' verwendet, sofern keine Angabe gemacht wurde.

**-dbName**

Der Name der DB2-Datenbank. Auf dem DB2-Client ist dies der Name der katalogisierten Datenbank. Auf dem nativen z/OS-Server ist dies der Name des Datenbanksystems. Der Standardwert lautet 'event', wenn keine Angabe gemacht wurde.

**-dbDiskSizeInMB**

Die Datenträgergröße der Events Service-Datenquelle (in MB). Geben Sie einen Wert von mindestens 10 MB an. Es wird der Standardwert von 100 MB verwendet, sofern keine Angabe gemacht wurde.

**-dbUser**

Die DB2-Benutzer-ID mit der Berechtigung zum Erstellen und Löschen der Datenbanken. Dieser Parameter ist erforderlich.

**-dbPassword**

Das Kennwort für die Datenbankbenutzer-ID. Dieser Parameter ist erforderlich.

**-storageGroup**

Die Speichergruppe für die Ereignisdatenbank und die Ereigniskatalogdatenbank. Die Speichergruppe muss bereits erstellt worden und aktiv sein.

**-bufferPool4K**

Der Name des 4K-Pufferpools. Dieser Pufferpool muss aktiv sein, damit die DDL-Scripts der Datenbank ausgeführt werden können.

**-bufferPool8K**

Der Name des 8K-Pufferpools. Dieser Pufferpool muss aktiv sein, damit die DDL-Scripts der Datenbank ausgeführt werden können.

**-bufferPool16K**

Der Name des 16K-Pufferpools. Dieser Pufferpool muss aktiv sein, damit die DDL-Scripts der Datenbank ausgeführt werden können.

**-outputScriptDir**

Optionales Ausgabeverzeichnis für das Datenbankskript. Wenn dieser Parameter angegeben ist, generiert der Befehl die Scripts für die Events Service-Datenbank in dem angegebenen Verzeichnis. Wenn das Verzeichnis nicht mit einem vollständigen Pfad angegeben wurde, erstellt der Befehl das angegebene Verzeichnis unter '*profilstammverzeichnis*/bin'. Wenn dieser Parameter nicht angegeben ist, wird das Standardausgabeverzeichnis für Datenbankskripts '*profilstammverzeichnis*/databases/event/knoten/server/dbscripts/db2zos' verwendet.

## Muster

### Syntaxbeispiel für Stapelmodus:

- Verwendung von JACL:

```
$AdminTask configEventServiceDB2ZOSDB {-createDB true -overrideDataSource true
-nodeName knotename -serverName servername -jdbcClassPath c:\sqllib\java
-dbUser db2-benutzer -dbPassword db-kennwort -dbHostName hostname -dbPort 5027
-storageGroup sysdeflt -bufferPool4K BP9 -bufferPool8K BP8K9
-bufferPool16K BP16K9}
```

- Verwendung von Jython-Zeichenfolge:

```
AdminTask.configEventServiceDB2ZOSDB('[-createDB true -overrideDataSource true
-nodeName knotename -serverName servername -jdbcClassPath c:\sqllib\java
-dbUser db2-benutzer -dbPassword db-kennwort -dbHostName hostname -dbPort 5027
-storageGroup sysdeflt -bufferPool4K BP9 -bufferPool8K BP8K9
-bufferPool16K BP16K9]')
```

- Verwendung von Jython-Liste:



```
AdminTask.configEventServiceDB2ZOSDB(['-createDB', 'true', '-overrideDataSource',
'true', '-nodeName', 'knotenname', '-serverName', 'servername', '-jdbcClassPath',
'c:\sql\lib\java', '-dbUser', 'db2-benutzer', '-dbPassword', 'db-kennwort',
'-dbHostName', 'hostname', '-dbPort', '5027', '-storageGroup', 'sysdeflt',
'-bufferPool4K', 'BP9', '-bufferPool8K', 'BP8K9', '-bufferPool16K', 'BP16K9'])
```

### Syntaxbeispiel für interaktiven Modus:

- Verwendung von JACL:  
`$AdminTask configEventServiceDB2ZOSDB -interactive`
- Verwendung von Jython-Zeichenfolge:  
`AdminTask.configEventServiceDB2ZOSDB(['-interactive'])`
- Verwendung von Jython-Liste:  
`AdminTask.configEventServiceDB2ZOSDB ['-interactive'])`

---

## Befehl 'configEventServiceDerbyDB'

Verwenden Sie den Befehl `configEventServiceDerbyDB`, um Common Event Infrastructure bei Verwendung einer Derby-Datenbank zu konfigurieren.

### Zweck

Der Befehl `configEventServiceDerbyDB` ist ein Verwaltungsbefehl von Common Event Infrastructure und für das Objekt 'AdminTask' verfügbar. Verwenden Sie diesen Befehl, um die Events Service-Datenbank und Datenquellen für Derby auf einem Server oder in einem Cluster zu erstellen. Weitere Informationen zu dem Objekt 'AdminTask' enthält die Dokumentation für WebSphere Application Server Network Deployment, Version 6.1.

**Anmerkung:**   Das Produkt verwendet eine Jython-Version, die die Betriebssysteme Microsoft Windows 2003, Windows Vista und Windows 7 nicht unterstützt.

### Parameter

#### **-createDB**

Der Befehl generiert die DDL-Datenbankskripts und erstellt die Datenbank, wenn für diesen Parameter der Wert `true` definiert ist. Mit dem Befehl werden nur die DLL-Datenbankskripts generiert, wenn für diesen Parameter der Wert `false` definiert ist. Zum Erstellen der Datenbank muss der aktuelle Server bereits für die Ausführung der Datenbankbefehle konfiguriert sein. Der Standardwert lautet `false`, sofern keine Angabe gemacht wurde.

#### **-overrideDataSource**

Wenn für diesen Parameter der Wert `true` definiert ist, entfernt der Befehl alle vorhandenen Datenquellen für Events Service für den angegebenen Geltungsbereich, bevor eine neue Datenquelle erstellt wird. Wenn für diesen Parameter der Wert `false` definiert ist, erstellt der Befehl keine Events Service-Datenquelle für den angegebenen Geltungsbereich, wenn eine andere Events Service-Datenquelle für den gleichen Geltungsbereich gefunden wird. Der Standardwert lautet `false`, sofern keine Angabe gemacht wurde.

#### **-nodeName**

Der Name des Knotens, der den Server enthält, auf dem die Datenquelle für Events Service erstellt werden soll. Bei Angabe dieses Parameters muss auch der Parameter `serverName` angegeben werden. Dieser Parameter darf nicht angegeben werden, wenn der Parameter `clusterName` angegeben wird.

**-serverName**

Der Name des Servers, auf dem die Datenquelle für Events Service erstellt werden soll. Wenn dieser Parameter ohne Angabe des Parameters **nodeName** definiert wird, verwendet der Befehl den Knotennamen des aktuellen WebSphere-Profiles. Dieser Parameter darf nicht angegeben werden, wenn der Parameter **clusterName** angegeben wird.

**-clusterName**

Der Name des Clusters, in dem die Datenquelle für Events Service erstellt werden soll. Bei Angabe dieses Parameters dürfen die Parameter **serverName** und **nodeName** nicht definiert werden. Dieser Parameter darf nicht angegeben werden, wenn die Parameter **serverName** und **nodeName** angegeben werden.

**-dbHostName**

Der Hostname der Derby-Netzdatenbank. Geben Sie zum Erstellen der Derby Network-Datenquelle diesen Parameter sowie den Parameter **dbPort** an. Geben Sie diesen Parameter und den Parameter **dbPort** nicht bei Erstellung der lokalen Derby-Datenquelle an.

**-dbPort**

Die Portnummer der Derby-Netzdatenbank. Geben Sie zum Erstellen der Derby Network-Datenquelle diesen Parameter sowie den Parameter **dbHostName** an. Geben Sie diesen Parameter und den Parameter **dbHostName** nicht bei Erstellung der lokalen Derby-Datenquelle an.

**-dbName**

Der Name der zu erstellenden Datenbank. Als Standardname wird event verwendet, sofern keine Angabe gemacht wurde.

**-dbUser**

Die Benutzer-ID, die von der Datenquelle für die Authentifizierung bei der Derby-Datenbank verwendet wird. Dieser Parameter ist optional, wenn die WebSphere-Domänensicherheit inaktiviert ist. Wenn Sie diesen Parameter angeben, müssen Sie auch den Parameter **dbPassword** angeben. Dieser Parameter ist erforderlich, wenn die WebSphere-Domänensicherheit aktiviert ist.

**-dbPassword**

Das Kennwort, das von der Datenquelle für die Authentifizierung bei der Derby-Datenbank verwendet wird. Dieser Parameter ist optional, wenn die WebSphere-Domänensicherheit inaktiviert ist. Wenn Sie diesen Parameter angeben, müssen Sie auch den Parameter **dbUser** angeben. Dieser Parameter ist erforderlich, wenn die WebSphere-Domänensicherheit aktiviert ist.

**-outputScriptDir**

Optionales Ausgabeverzeichnis für das Datenbankskript. Wenn dieser Parameter angegeben ist, generiert der Befehl die Skripts für die Events Service-Datenbank in dem angegebenen Verzeichnis. Wenn das Verzeichnis nicht mit einem vollständigen Pfad angegeben wurde, erstellt der Befehl das angegebene Verzeichnis unter '*profilstammverzeichnis/bin*'. Wenn dieser Parameter nicht angegeben ist, wird das Standardausgabeverzeichnis für Datenbankskripts '*profilstammverzeichnis/databases/event/knoten/server/dbscripts/derby*' verwendet.

## Muster

**Syntaxbeispiel für Stapelmodus:**

- Verwendung von JACL:

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

- Verwendung von Jython-Zeichenfolge:  

```
AdminTask.configEventServiceDerbyDB(['-createDB true -overrideDataSource true -nodeName knotenname -serverName servername'])
```
- Verwendung von Jython-Liste:  

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

#### Syntaxbeispiel für interaktiven Modus:

- Verwendung von JACL:  

```
$AdminTask configEventServiceDerbyDB -interactive
```
- Verwendung von Jython-Zeichenfolge:  

```
AdminTask.configEventServiceDerbyDB(['-interactive'])
```
- Verwendung von Jython-Liste:  

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

---

## Befehl 'configEventServiceInformixDB'

Verwenden Sie den Befehl `configEventServiceInformixDB`, um Common Event Infrastructure bei Verwendung einer Informix-Datenbank zu konfigurieren.

### Zweck

Der Befehl `configEventServiceInformixDB` ist ein Verwaltungsbefehl von Common Event Infrastructure und für das Objekt 'AdminTask' verfügbar. Verwenden Sie diesen Befehl, um die Events Service-Datenbank und Datenquellen für Informix auf einem Server oder in einem Cluster zu erstellen. Weitere Informationen zu dem Objekt 'AdminTask' enthält die Dokumentation für WebSphere Application Server Network Deployment, Version 6.1.

**Anmerkung:**   Das Produkt verwendet eine Jython-Version, die die Betriebssysteme Microsoft Windows 2003, Windows Vista und Windows 7 nicht unterstützt.

### Parameter

#### **-createDB**

Der Befehl generiert die DDL-Datenbankskripts und erstellt die Datenbank, wenn für diesen Parameter der Wert `true` definiert ist. Mit dem Befehl werden nur die DLL-Datenbankskripts generiert, wenn für diesen Parameter der Wert `false` definiert ist. Zum Erstellen der Datenbank muss der aktuelle Server bereits für die Ausführung der Datenbankbefehle konfiguriert sein. Der Standardwert lautet `false`, sofern keine Angabe gemacht wurde.

#### **-overrideDataSource**

Optionales Ausgabeverzeichnis für das Datenbankskript. Wenn dieser Parameter angegeben ist, generiert der Befehl die Skripts für die Events Service-Datenbank in dem angegebenen Verzeichnis. Wenn das Verzeichnis nicht mit einem vollständigen Pfad angegeben wurde, erstellt der Befehl das angegebene Verzeichnis unter '*profilstammverzeichnis/bin*'. Wenn dieser Parameter nicht angegeben ist, wird das Standardausgabeverzeichnis für Datenbankskripts *profilstammverzeichnis/databases/event/knoten/server/dbscripts/datenbanktyp* verwendet.

#### **-nodeName**

Der Name des Knotens, der den Server enthält, auf dem die Datenquelle für Events Service erstellt werden soll. Bei Angabe dieses Parameters muss auch

der Parameter **serverName** angegeben werden. Dieser Parameter darf nicht angegeben werden, wenn der Parameter **clusterName** angegeben wird.

**-serverName**

Der Name des Servers, auf dem die Datenquelle für Events Service erstellt werden soll. Wenn dieser Parameter ohne Angabe des Parameters **nodeName** definiert wird, verwendet der Befehl den Knotennamen des aktuellen WebSphere-Profils. Dieser Parameter darf nicht angegeben werden, wenn der Parameter **clusterName** angegeben wird.

**-clusterName**

Der Name des Clusters, in dem die Datenquelle für Events Service erstellt werden soll. Bei Angabe dieses Parameters dürfen die Parameter **serverName** und **nodeName** nicht definiert werden. Dieser Parameter darf nicht angegeben werden, wenn die Parameter **serverName** und **nodeName** angegeben werden.

**-jdbcClassPath**

Der Pfad zum JDBC-Treiber. Geben Sie nur den Pfad zur Treiberdatei an. Der Pfadname darf den Dateinamen selbst nicht enthalten. Dieser Parameter ist erforderlich.

**-dbInformixDir**

Das Verzeichnis, in dem die Informix-Datenbank installiert ist. Dieser Parameter muss angegeben werden, wenn für den Parameter **createDB** der Wert **true** definiert ist. Dieser Parameter ist erforderlich.

**-dbHostName**

Der Hostname des Servers, auf dem die Datenbank installiert ist. Dieser Parameter ist erforderlich.

**-dbServerName**

Der Name des Informix-Servers (zum Beispiel `ol_servername`). Dieser Parameter ist erforderlich.

**-dbPort**

Der Port der Informix-Instanz. Es wird der Standardwert '1526' verwendet, sofern keine Angabe gemacht wurde.

**-dbName**

Der Name der zu erstellenden Datenbank. Als Standardname wird `event` verwendet, sofern keine Angabe gemacht wurde.

**-dbUser**

Die Benutzer-ID für das Informix-Datenbankschema, die als Eigner der Datenbanktabellen von Events Service definiert wird. Die WebSphere-Datenquelle verwendet diese Benutzer-ID zur Authentifizierung der Verbindung der Informix-Datenbank. Dieser Parameter ist erforderlich.

**-dbPassword**

Das Kennwort für die Schemabeanutzer-ID, die als Eigner der Informix-Tabellen von Events Service definiert ist. Die WebSphere-Datenquelle verwendet dieses Kennwort zur Authentifizierung der Verbindung der Informix-Datenbank. Dieser Parameter ist erforderlich.

**-ceiInstancePrefix**

Der Befehl verwendet den Namen der Events Service-Instanz, um die Datenbankdateien in einem Verzeichnis mit eindeutigen Namen zusammenzufassen. Der Standardwert lautet `ceiinst1`, sofern keine Angabe gemacht wurde.

**-outputScriptDir**

Optionales Ausgabeverzeichnis für das Datenbankskript. Wenn dieser Parameter angegeben ist, generiert der Befehl die Skripts für die Events Service-Daten-

bank in dem angegebenen Verzeichnis. Wenn das Verzeichnis nicht mit einem vollständigen Pfad angegeben wurde, erstellt der Befehl das angegebene Verzeichnis unter *'profilstammverzeichnis/bin'*. Wenn dieser Parameter nicht angegeben ist, wird das Standardausgabeverzeichnis für Datenbankskripts *'profilstammverzeichnis/databases/event/knoten/server/dbscripts/informix'* verwendet.

## Muster

### Syntaxbeispiel für Stapelmodus:

- Verwendung von JACL:

```
$AdminTask configEventServiceInformixDB {-createdB true
-overrideDataSource true -nodeName knotenname -serverName servername
-jdbcClassPath "c:\programme\ibm\informix\jdbc\lib"
-dbInformixDir "c:\programme\ibm\informix" -dbUser informix
-dbPassword db-kennwort -dbHostName hostname -dbPort 1526
-dbServerName ol_server }
```

- Verwendung von Jython-Zeichenfolge:

```
AdminTask.configEventServiceInformixDB(['-createdB true
-overrideDataSource true
-nodename knotenname
-serverName servername
-jdbcClassPath "c:\programme\ibm\informix\jdbc\lib"
-dbInformixDir "c:\programme\ibm\informix"
-dbUser informix
-dbPassword datenbankkennwort
-dbHostName hostname
-dbPort 1526
-dbServerName ol_server'])
```

- Verwendung von Jython-Liste:

```
AdminTask.configEventServiceInformixDB(['-createdB', 'true',
'-overrideDataSource', 'true', '-nodeName', 'knotenname',
'-serverName', 'servername', '-jdbcClassPath',
'c:\programme\ibm\informix\jdbc\lib', '-dbInformixDir',
'c:\programme\ibm\informix', '-dbUser', 'informix ',
'-dbPassword', 'db-kennwort', '-dbHostName', 'hostname',
'-dbPort', '1526', '-dbServerName', 'ol_server'])
```

### Syntaxbeispiel für interaktiven Modus:

- Verwendung von JACL:

```
$AdminTask configEventServiceInformixDB -interactive
```

- Verwendung von Jython-Zeichenfolge:

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

- Verwendung von Jython-Liste:

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

---

## Befehl 'configEventServiceOracleDB'

Verwenden Sie den Befehl `configEventServiceOracleDB`, um Common Event Infrastructure bei Verwendung einer Oracle-Datenbank zu konfigurieren.

### Zweck

Der Befehl `configEventServiceOracleDB` ist ein Verwaltungsbefehl von Common Event Infrastructure und für das Objekt 'AdminTask' verfügbar. Verwenden Sie diesen Befehl, um die Events Service-Tabellen und die Datenquellen für Oracle auf einem Server oder in einem Cluster zu erstellen. Mit diesem Befehl wird die Daten-

bank nicht erstellt; die Oracle-System-ID (SID) muss bereits vorhanden sein. Weitere Informationen zu dem Objekt 'AdminTask' enthält die Dokumentation für WebSphere Application Server Network Deployment, Version 6.1.

**Anmerkung:**   Das Produkt verwendet eine Jython-Version, die die Betriebssysteme Microsoft Windows 2003, Windows Vista und Windows 7 nicht unterstützt.

## Parameter

### -createDB

Der Befehl generiert die DDL-Datenbankscrippts und erstellt die Datenbank, wenn für diesen Parameter der Wert `true` definiert ist. Mit dem Befehl werden nur die DLL-Datenbankscrippts generiert, wenn für diesen Parameter der Wert `false` definiert ist. Zum Erstellen der Datenbank muss der aktuelle Server bereits für die Ausführung der Datenbankbefehle konfiguriert sein. Der Standardwert lautet `false`, sofern keine Angabe gemacht wurde.

### -overrideDataSource

Optionales Ausgabeverzeichnis für das Datenbanksript. Wenn dieser Parameter angegeben ist, generiert der Befehl die Scripts für die Events Service-Datenbank in dem angegebenen Verzeichnis. Wenn das Verzeichnis nicht mit einem vollständigen Pfad angegeben wurde, erstellt der Befehl das angegebene Verzeichnis unter '*profilstammverzeichnis/bin*'. Wenn dieser Parameter nicht angegeben ist, wird das Standardausgabeverzeichnis für Datenbankscrippts *profilstammverzeichnis/databases/event/knoten/server/dbscripts/datenbanktyp* verwendet.

### -nodeName

Der Name des Knotens, der den Server enthält, auf dem die Datenquelle für Events Service erstellt werden soll. Bei Angabe dieses Parameters muss auch der Parameter **serverName** angegeben werden. Dieser Parameter darf nicht angegeben werden, wenn der Parameter **clusterName** angegeben wird.

### -serverName

Der Name des Servers, auf dem die Datenquelle für Events Service erstellt werden soll. Wenn dieser Parameter ohne Angabe des Parameters **nodeName** definiert wird, verwendet der Befehl den Knotennamen des aktuellen WebSphere-Profiles. Dieser Parameter darf nicht angegeben werden, wenn der Parameter **clusterName** angegeben wird.

### -clusterName

Der Name des Clusters, in dem die Datenquelle für Events Service erstellt werden soll. Bei Angabe dieses Parameters dürfen die Parameter **serverName** und **nodeName** nicht definiert werden. Dieser Parameter darf nicht angegeben werden, wenn die Parameter **serverName** und **nodeName** angegeben werden.

### -jdbcClassPath

Der Pfad zum JDBC-Treiber. Geben Sie nur den Pfad zur Treiberdatei an. Der Pfadname darf den Dateinamen selbst nicht enthalten. Dieser Parameter ist erforderlich.

### -oracleHome

Das Verzeichnis ORACLE\_AUSGANGSVERZEICHNIS. Dieser Parameter muss festgelegt werden, wenn für den Parameter **createDB** der Wert `true` definiert ist.

**-dbHostName**

Der Hostname des Servers, auf dem die Oracle-Datenbank installiert ist. Als Standardname wird localhost verwendet, sofern keine Angabe gemacht wurde.

**-dbPort**

Der Port der Oracle-Instanz. Es wird der Standardwert '1521' verwendet, sofern keine Angabe gemacht wurde.

**-dbName**

Die Oracle-System-ID (SID). Die Oracle-System-ID (SID) muss bereits vorhanden und für den Events Service-Befehl verfügbar sein, um die Tabellen zu erstellen und die Tabellen mit Daten zu füllen. Der Standardwert lautet orcl, wenn keine Angabe gemacht wurde.

**-dbUser**

Die Benutzer-ID für das Oracle-Schema, die als Eigner der Oracle-Tabellen von Events Service definiert wird. Die Benutzer-ID wird während der Datenbankerstellung erstellt. Die WebSphere-Datenquelle verwendet diese Benutzer-ID zum Authentifizieren der Oracle-Datenbankverbindung. Der Standardwert lautet ceiuser, wenn keine andere Angabe gemacht wurde.

**-dbPassword**

Das Kennwort für die Schemabeanutzer-ID. Das Kennwort wird während der Datenbankerstellung erstellt. Die WebSphere-Datenquelle verwendet dieses Kennwort zum Authentifizieren der Oracle-Datenbankverbindung. Dieser Parameter ist erforderlich.

**-sysUser**

Die Benutzer-ID des Oracle-Benutzers 'sys'. Dieser Benutzer muss über SYSDBA-Berechtigungen verfügen. Der Standardwert lautet sys, sofern keine Angabe gemacht wurde.

**-sysPassword**

Das Kennwort für den Benutzer, der mit dem **sysUser** angegeben wurde. Der Standardwert ist eine leere Zeichenfolge, sofern keine Angabe gemacht wurde.

**-ceiInstancePrefix**

Der Befehl verwendet den Namen der Events Service-Instanz, um die Datenbankdateien in einem Verzeichnis mit eindeutigen Namen zusammenzufassen. Der Standardwert lautet ceinst1, sofern keine Angabe gemacht wurde.

**-outputScriptDir**

Optionales Ausgabeverzeichnis für das Datenbankskript. Wenn dieser Parameter angegeben ist, generiert der Befehl die Skripts für die Events Service-Datenbank in dem angegebenen Verzeichnis. Wenn das Verzeichnis nicht mit einem vollständigen Pfad angegeben wurde, erstellt der Befehl das angegebene Verzeichnis unter '*profilstammverzeichnis*/bin'. Wenn dieser Parameter nicht angegeben ist, wird das Standardausgabeverzeichnis für Datenbankskripts '*profilstammverzeichnis*/databases/event/knoten/server/dbscripts/oracle' verwendet.

## Muster

**Syntaxbeispiel für Stapelmodus:**

- Verwendung von JACL:

```
$AdminTask configEventServiceOracleDB {-createDB true
 -overrideDataSource true
 -nodeName knotenname
 -serverName servername
```

```
-jdbcClassPath c:\oracle\ora92\jdbc\lib
-oracleHome c:\oracle\ora92
-dbUser ceiuser
-dbPassword cei-kennwort
-dbHostName hostname
-dbPort 1521
-sysUser sys
-sysPassword sys-kennwort}
```

- Verwendung von Jython-Zeichenfolge:

```
AdminTask.configEventServiceOracleDB(['-createDB true
-overrideDataSource true
-nodeName knotenname
-serverName servername
-jdbcClassPath c:\oracle\ora92\jdbc\lib
-oracleHome c:\oracle\ora92
-dbUser ceiuser
-dbPassword cei-kennwort
-dbHostName hostname
-dbPort 1521
-sysUser sys
-sysPassword sys-kennwort'])
```

- Verwendung von Jython-Liste:

```
AdminTask.configEventServiceOracleDB(['-createDB', 'true',
'-overrideDataSource', 'true',
'-nodeName', 'knotenname',
'-serverName', 'servername',
'-jdbcClassPath', 'c:\oracle\ora92\jdbc\lib',
'-oracleHome', 'c:\oracle\ora92',
'-dbUser', 'ceiuser',
'-dbPassword', 'cei-kennwort',
'-dbHostName', 'hostname',
'-dbPort', '1521',
'-sysUser', 'sys',
'-sysPassword', 'sys-kennwort'])
```

#### Syntaxbeispiel für interaktiven Modus:

- Verwendung von JACL:

```
$AdminTask configEventServiceOracleDB -interactive
```

- Verwendung von Jython-Zeichenfolge:

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

- Verwendung von Jython-Liste:

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

---

## Befehl 'configEventServiceSQLServerDB'

Verwenden Sie den Befehl `configEventServiceSQLServerDB`, um Common Event Infrastructure bei Verwendung einer SQL Server-Datenbank zu konfigurieren.

### Zweck

Der Befehl `configEventServiceSQLServerDB` ist ein Verwaltungsbefehl von Common Event Infrastructure und für das Objekt 'AdminTask' verfügbar. Verwenden Sie diesen Befehl, um die Events Service-Datenbank und Datenquellen für SQL Server auf einem Server oder in einem Cluster zu erstellen. Weitere Informationen zu dem Objekt 'AdminTask' enthält die Dokumentation für WebSphere Application Server Network Deployment, Version 6.1.



**Anmerkung:** **Vista** **Windows 7** Das Produkt verwendet eine Jython-Version, die die Betriebssysteme Microsoft Windows 2003, Windows Vista und Windows 7 nicht unterstützt.

## Parameter

### **-createDB**

Der Befehl generiert die DDL-Datenbankscrippts und erstellt die Datenbank, wenn für diesen Parameter der Wert `true` definiert ist. Mit dem Befehl werden nur die DLL-Datenbankscrippts generiert, wenn für diesen Parameter der Wert `false` definiert ist. Zum Erstellen der Datenbank muss der aktuelle Server bereits für die Ausführung der Datenbankbefehle konfiguriert sein. Der Standardwert lautet `false`, sofern keine Angabe gemacht wurde.

### **-overrideDataSource**

Optionales Ausgabeverzeichnis für das Datenbanksript. Wenn dieser Parameter angegeben ist, generiert der Befehl die Scripts für die Events Service-Datenbank in dem angegebenen Verzeichnis. Wenn das Verzeichnis nicht mit einem vollständigen Pfad angegeben wurde, erstellt der Befehl das angegebene Verzeichnis unter `'profilstammverzeichnis/bin'`. Wenn dieser Parameter nicht angegeben ist, wird das Standardausgabeverzeichnis für Datenbankscrippts `profilstammverzeichnis/databases/event/knoten/server/dbscripts/datenbanktyp` verwendet.

### **-nodeName**

Der Name des Knotens, der den Server enthält, auf dem die Datenquelle für Events Service erstellt werden soll. Bei Angabe dieses Parameters muss auch der Parameter **serverName** angegeben werden. Dieser Parameter darf nicht angegeben werden, wenn der Parameter **clusterName** angegeben wird.

### **-serverName**

Der Name des Servers, auf dem die Datenquelle für Events Service erstellt werden soll. Wenn dieser Parameter ohne Angabe des Parameters **nodeName** definiert wird, verwendet der Befehl den Knotennamen des aktuellen WebSphere-Profiles. Dieser Parameter darf nicht angegeben werden, wenn der Parameter **clusterName** angegeben wird.

### **-clusterName**

Der Name des Clusters, in dem die Datenquelle für Events Service erstellt werden soll. Bei Angabe dieses Parameters dürfen die Parameter **serverName** und **nodeName** nicht definiert werden. Dieser Parameter darf nicht angegeben werden, wenn die Parameter **serverName** und **nodeName** angegeben werden.

### **-dbServerName**

Der Servername der SQL Server-Datenbank. Dieser Parameter muss festgelegt werden, wenn für den Parameter **createDB** der Wert `true` definiert ist.

### **-dbHostName**

Der Hostname des Servers, auf dem die SQL Server-Datenbank ausgeführt wird.

### **-dbPort**

Der Port für SQL Server. Es wird der Standardwert `'1433'` verwendet, sofern keine Angabe gemacht wurde.

### **-dbName**

Der Name der zu erstellenden Datenbank. Als Standardname wird `event` verwendet, sofern keine Angabe gemacht wurde.

**-dbUser**

Die Benutzer-ID für SQL Server, die als Eigner der Events Service-Tabellen definiert wird. Der Standardwert lautet `ceiuser`, wenn keine andere Angabe gemacht wurde.

**-dbPassword**

Das Kennwort für die Benutzer-ID für SQL Server, die durch den Parameter `dbUser` angegeben wurde. Dieser Parameter ist erforderlich.

**-saUser**

Eine Benutzer-ID mit den Berechtigungen zum Erstellen und Löschen von Datenbanken und Benutzern. Dieser Parameter ist erforderlich, wenn für den Parameter `createDB` der Wert `true` definiert wurde. Als Standardwert wird `sa` verwendet, sofern keine Angabe gemacht wurde.

**-saPassword**

Das Kennwort für 'sa'. Dieser Parameter darf nicht angegeben werden, wenn für die Benutzer-ID 'sa' kein Kennwort angegeben wurde.

**-ceiInstancePrefix**

Der Befehl verwendet den Namen der Events Service-Instanz, um die Datenbankdateien in einem Verzeichnis mit eindeutigen Namen zusammenzufassen. Der Standardwert lautet `ceiinst1`, sofern keine Angabe gemacht wurde.

**-outputScriptDir**

Optionales Ausgabeverzeichnis für das Datenbankskript. Wenn dieser Parameter angegeben ist, generiert der Befehl die Scripts für die Events Service-Datenbank in dem angegebenen Verzeichnis. Wenn das Verzeichnis nicht mit einem vollständigen Pfad angegeben wurde, erstellt der Befehl das angegebene Verzeichnis unter `'profilstammverzeichnis/bin'`. Wenn dieser Parameter nicht angegeben ist, wird das Standardausgabeverzeichnis für Datenbankskripts `'profilstammverzeichnis/databases/event/knoten/server/dbscripts/sqlserver'` verwendet.

## Muster

**Syntaxbeispiel für Stapelmodus:**

- Verwendung von JACL:

```
$AdminTask configEventServiceSQLServerDB {-createDB true
 -overrideDataSource true
 -nodeName knotenname
 -serverName servername
 -dbUser ceiuser
 -dbPassword cei-kennwort
 -dbServerName sqlserver-name
 -dbHostName hostname
 -dbPort 1433
 -saUser sa
 -saPassword sa-kennwort}
```

- Verwendung von Jython-Zeichenfolge:

```
AdminTask.configEventServiceSQLServerDB(['-createDB true
 -overrideDataSource true
 -nodeName knotenname
 -serverName servername
 -dbUser ceiuser
 -dbPassword cei-kennwort
 -dbServerName sqlserver-name
 -dbHostName hostname
 -dbPort 1433
 -saUser sa
 -saPassword sa-kennwort'])
```

- Verwendung von Jython-Liste:

```
AdminTask.configEventServiceSQLServerDB(['-createDB', 'true',
'-overrideDataSource', 'true',
'-nodeName', 'knotenname',
'-serverName', 'servername',
'-dbUser', 'ceiuser',
'-dbPassword', 'cei-kennwort',
'-dbServerName', 'sqlserver-name',
'-dbHostName', 'hostname',
'-dbPort', '1433',
'-saUser', 'sa',
'-saPassword', 'sa-kennwort'])
```

#### Syntaxbeispiel für interaktiven Modus:

- Verwendung von JACL:

```
$AdminTask configEventServiceSQLServerDB -interactive
```

- Verwendung von Jython-Zeichenfolge:

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

- Verwendung von Jython-Liste:

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

---

## Befehl 'deployEventService'

Verwenden Sie den Befehl `deployEventService`, um die Events Service-Anwendung auf Ihrem Server zu implementieren.

### Zweck

Der Befehl `deployEventService` ist ein Verwaltungsbefehl von Common Event Infrastructure und für das Objekt 'AdminTask' verfügbar. Verwenden Sie diesen Befehl, um Events Service auf einem Server oder in einem Cluster zu implementieren und konfigurieren. Weitere Informationen zu dem Objekt 'AdminTask' enthält die Dokumentation für WebSphere Application Server Network Deployment, Version 6.1.

**Anmerkung:**   Das Produkt verwendet eine Jython-Version, die die Betriebssysteme Microsoft Windows 2003, Windows Vista und Windows 7 nicht unterstützt.

### Parameter

#### -nodeName

Der Name des Knotens, auf dem Events Service implementiert werden soll. Bei Angabe dieses Parameters muss auch der Parameter `serverName` angegeben werden. Dieser Parameter darf nicht angegeben werden, wenn der Parameter `clusterName` angegeben wird.

#### -serverName

Der Name des Servers, auf dem Events Service implementiert werden soll. Dieser Parameter muss angegeben werden, wenn der Parameter `nodeName` angegeben wird. Dieser Parameter darf nicht angegeben werden, wenn der Parameter `clusterName` angegeben wird.

#### -clusterName

Der Name des Clusters, in dem Events Service implementiert werden soll. Dieser Parameter darf nicht angegeben werden, wenn die Parameter `nodeName` oder `serverName` angegeben werden.

### **-enable**

Definieren Sie für diesen Parameter den Wert `true`, wenn Events Service nach dem nächsten Neustart des Servers gestartet werden soll. Der Standardwert lautet `true`.

## **Muster**

### **Syntaxbeispiel für Stapelmodus:**

- Verwendung von JACL:  

```
$AdminTask deployEventService {-nodeName knotenname
-serverName servername}
$AdminTask deployEventService {-clusterName clustername
-enable false}
```
- Verwendung von Jython-Zeichenfolge:  

```
AdminTask.deployEventService('[-nodeName knotenname
-serverName servername]')

AdminTask.deployEventService('[-clusterName clustername
-enable false]')
```
- Verwendung von Jython-Liste:  

```
AdminTask.deployEventService(['-nodeName', 'knotenname',
'-serverName', '-servername'])
AdminTask.deployEventService(['-clusterName', 'clustername',
'-enable', 'false'])
```

### **Syntaxbeispiel für interaktiven Modus:**

- Verwendung von JACL:  

```
$AdminTask deployEventService {-interactive}
```
- Verwendung von Jython-Zeichenfolge:  

```
AdminTask.deployEventService('[-interactive]')
```
- Verwendung von Jython-Liste:  

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

---

## **Befehl 'deployEventServiceMdb'**

Verwenden Sie den Befehl `deployEventServiceMdb`, um die MDB (Message-driven Bean) von Events Service auf Ihrem Server zu implementieren.

### **Zweck**

Der Befehl `deployEventServiceMdb` ist ein Verwaltungsbefehl von Common Event Infrastructure und für das Objekt 'AdminTask' verfügbar. Verwenden Sie diesen Befehl, um die Events Service-MDB auf einem Server oder in einem Cluster zu implementieren. Weitere Informationen zu dem Objekt 'AdminTask' enthält die Dokumentation für WebSphere Application Server Network Deployment, Version 6.1.

**Anmerkung:**   Das Produkt verwendet eine Jython-Version, die die Betriebssysteme Microsoft Windows 2003, Windows Vista und Windows 7 nicht unterstützt.

### **Parameter**

#### **-nodeName**

Der Name des Knotens, auf dem die Events Service-MDB implementiert werden soll. Bei Angabe dieses Parameters muss auch der Parameter **serverName**

angegeben werden. Dieser Parameter darf nicht angegeben werden, wenn der Parameter **clusterName** angegeben wird.

**-serverName**

Der Name des Servers, auf dem die Events Service-MDB implementiert werden soll. Dieser Parameter muss angegeben werden, wenn der Parameter **nodeName** angegeben wird. Dieser Parameter darf nicht angegeben werden, wenn der Parameter **clusterName** angegeben wird.

**-clusterName**

Der Name des Clusters, auf dem die Events Service-MDB implementiert werden soll. Dieser Parameter darf nicht angegeben werden, wenn die Parameter **nodeName** und **serverName** angegeben werden.

**-applicationName**

Der Name der Events Service-MDB-Anwendung, die auf einem Server oder in einem Cluster implementiert werden soll.

**-listenerPort**

Der Name des Listener-Ports, auf dem die Events Service-MDB die Ereignisse veröffentlichen soll. Der Listener-Port muss bereits erstellt worden sein. Dieser Parameter darf nicht angegeben werden, wenn der Parameter **activationSpec** angegeben wird.

**-activationSpec**

Der JNDI-Name der Aktivierungsspezifikation, auf der die Events Service-MDB die Ereignisse veröffentlichen soll. Die Aktivierungsspezifikation muss bereits erstellt worden sein. Dieser Parameter darf nicht angegeben werden, wenn der Parameter **listenerPort** angegeben wird.

**-qcfJndiName**

Der JNDI-Name des Verbindungsfactory-Objekts für JMS-Warteschlangen, das von der Events Service-MDB verwendet werden soll. Dieser Parameter muss angegeben werden, wenn der Parameter **activationSpec** angegeben wird.

## Muster

### Syntaxbeispiel für Stapelmodus:

- Verwendung von JACL:

```
$AdminTask deployEventServiceMdb {-applicationName anwname -nodeName knotenname
-serverName servername -listenerPort lpname}$AdminTask deployEventServiceMdb
{-applicationName anwname -clusterName clustername -activationSpec asjndiname
-qcfJndiName wvfjndiname}
```

- Verwendung von Jython-Zeichenfolge:

```
AdminTask.deployEventServiceMdb(['-applicationName anwname -nodeName knotenname
-serverName servername -listenerPort lpname'])AdminTask.deployEventServiceMdb
(['-applicationName anwname -clusterName clustername -activationSpec asjndiname
-qcfJndiName wvfjndiname'])
```

- Verwendung von Jython-Liste:

```
AdminTask.deployEventServiceMdb(['-applicationName', 'anwname', '-nodeName',
'knotenname', '-serverName', '-servername', '-listenerPort', 'lpname'])
AdminTask.deployEventServiceMdb(['-applicationName', 'anwname',
'-clusterName', 'clustername', '-activationSpec', 'asjndiname',
'-qcfJndiName', 'wvfjndiname'])
```

### Syntaxbeispiel für interaktiven Modus:

- Verwendung von JACL:

```
$AdminTask deployEventServiceMdb {-interactive}
```

- Verwendung von Jython-Zeichenfolge:

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

- Verwendung von Jython-Liste:

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

---

## Befehl 'setEventServiceJmsAuthAlias'

Verwenden Sie den Befehl `setEventServiceJmsAuthAlias` zu Festlegen oder Aktualisieren des JMS-Authentifizierungsaliasnamens, der Events Service auf Ihrem Server zugeordnet ist.

### Zweck

Der Befehl `setEventServiceJmsAuthAlias` ist ein Verwaltungsbefehl von Common Event Infrastructure und für das Objekt 'AdminTask' verfügbar. Verwenden Sie diesen Befehl, um den Authentifizierungsaliasnamen zu aktualisieren, den die JMS-Objekte von Events Service auf einem Server oder in einem Cluster verwenden. Falls kein JMS-Authentifizierungsaliasname vorhanden ist, wird ein solcher erstellt. Weitere Informationen zu dem Objekt 'AdminTask' enthält die Dokumentation für WebSphere Application Server Network Deployment, Version 6.1.

**Anmerkung:**   Das Produkt verwendet eine Jython-Version, die die Betriebssysteme Microsoft Windows 2003, Windows Vista und Windows 7 nicht unterstützt.

### Parameter

#### **-nodeName**

Der Name des Knotens, auf dem der JMS-Authentifizierungsaliasname von Events Service aktualisiert werden soll. Bei Angabe dieses Parameters muss auch der Parameter **serverName** angegeben werden. Dieser Parameter darf nicht angegeben werden, wenn der Parameter **clusterName** angegeben wird.

#### **-serverName**

Der Name des Servers, auf dem der JMS-Authentifizierungsaliasname von Events Service aktualisiert werden soll. Bei Angabe dieses Parameters muss auch der Parameter **serverName** angegeben werden. Dieser Parameter darf nicht angegeben werden, wenn der Parameter **clusterName** angegeben wird.

#### **-clusterName**

Der Name des Clusters, auf dem der JMS-Authentifizierungsaliasname von Events Service aktualisiert werden soll. Dieser Parameter darf nicht angegeben werden, wenn die Parameter **nodeName** und **serverName** angegeben werden.

#### **-userName**

Der Name des Benutzers, der bei der Aktualisierung des JMS-Authentifizierungsaliasnamens von Events Service auf einem Server oder in einem Cluster verwendet werden soll.

**Wichtig:** Sie müssen eine gültige Benutzer-ID angeben. Für diesen Parameter muss eine Angabe erfolgen.

#### **-password**

Das Kennwort für den Benutzer, der bei der Aktualisierung des JMS-Authentifizierungsaliasnamens von Events Service auf einem Server oder in einem Cluster verwendet werden soll.

**Wichtig:** Sie müssen ein gültiges Kennwort angeben. Für diesen Parameter muss eine Angabe erfolgen.

## Muster

### Syntaxbeispiel für Stapelmodus:

- Verwendung von JACL:

```
$AdminTask setEventServiceJmsAuthAlias{-nodeName knotenname
-serverName servername benutzername -password kennwort}
$AdminTask setEventServiceJmsAuthAlias {-clusterName clustername
-userName benutzername -password kennwort}
```
- Verwendung von Jython-Zeichenfolge:

```
AdminTask.setEventServiceJmsAuthAlias('[-nodeName knotenname
-serverName servername -userName benutzername -password kennwort]')
AdminTask.setEventServiceJmsAuthAlias('[-clusterName clustername
-userName benutzername -password kennwort]')
```
- Verwendung von Jython-Liste:

```
AdminTask.setEventServiceJmsAuthAlias(['-nodeName', 'knotenname',
'-serverName', '-servername', '-userName', 'benutzername', '-password', 'kennwort'])
AdminTask.setEventServiceJmsAuthAlias(['-clusterName',
'clustername', '-userName', 'benutzername', '-password', 'kennwort'])
```

### Syntaxbeispiel für interaktiven Modus:

- Verwendung von JACL:

```
$AdminTask setEventServiceJmsAuthAlias {-interactive}
```
- Verwendung von Jython-Zeichenfolge:

```
AdminTask.setEventServiceJmsAuthAlias('[-interactive]')
```
- Verwendung von Jython-Liste:

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

---

## Befehl 'enableEventService'

Verwenden Sie den Befehl `enableEventService`, um Events Service auf Ihrem Server zu aktivieren.

### Zweck

Der Befehl `enableEventService` ist ein Verwaltungsbefehl von Common Event Infrastructure und für das Objekt 'AdminTask' verfügbar. Verwenden Sie diesen Befehl, damit Events Service nach dem nächsten Neustart des oder der Server gestartet wird, der bzw. die mit den Parametern für den Knotennamen, Servernamen oder Clusternamen angegeben sind. Weitere Informationen zu dem Objekt 'AdminTask' enthält die Dokumentation für WebSphere Application Server Network Deployment, Version 6.1.

**Anmerkung:**   Das Produkt verwendet eine Jython-Version, die die Betriebssysteme Microsoft Windows 2003, Windows Vista und Windows 7 nicht unterstützt.

### DITA

### Parameter

#### **-nodeName**

Der Name des Knotens, auf dem Events Service aktiviert werden soll. Bei Angabe dieses Parameters muss auch der Parameter **serverName** angegeben werden. Dieser Parameter darf nicht angegeben werden, wenn der Parameter **clusterName** angegeben wird.

### **-serverName**

Der Name des Servers, auf dem Events Service aktiviert werden soll. Dieser Parameter muss angegeben werden, wenn der Parameter **nodeName** angegeben wird. Dieser Parameter darf nicht angegeben werden, wenn der Parameter **clusterName** angegeben wird.

### **-clusterName**

Der Name des Clusters, auf dem Events Service aktiviert werden soll. Dieser Parameter darf nicht angegeben werden, wenn die Parameter **nodeName** und **serverName** angegeben werden.

## **Muster**

### **Syntaxbeispiel für Stapelmodus:**

- Verwendung von JACL:

```
$AdminTask enableEventService {-nodeName knotenname
-serverName servername}
```

```
$AdminTask enableEventService {-clusterName clustername}
```

- Verwendung von Jython-Zeichenfolge:

```
AdminTask.enableEventService(['-nodeName knotenname
-serverName servername'])
```

```
AdminTask.enableEventService(['-clusterName clustername'])
```

- Verwendung von Jython-Liste:

```
AdminTask.enableEventService(['-nodeName', 'knotenname',
'-serverName', '-servername'])
```

```
AdminTask.enableEventService(['-clusterName', 'clustername'])
```

### **Syntaxbeispiel für interaktiven Modus:**

- Verwendung von JACL:

```
$AdminTask enableEventService {-interactive}
```

- Verwendung von Jython-Zeichenfolge:

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

- Verwendung von Jython-Liste:

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

---

## **Befehl 'disableEventService'**

Verwenden Sie den Befehl `disableEventService`, um Events Service auf Ihrem Server zu inaktivieren.

### **Zweck**

Der Befehl `disableEventService` ist ein Verwaltungsbefehl von Common Event Infrastructure und für das Objekt 'AdminTask' verfügbar. Verwenden Sie diesen Befehl, um zu verhindern, dass Events Service nach dem nächsten Neustart des oder der Server gestartet wird, der bzw. die mit den Parametern für den Knotennamen, Servernamen oder Clusternamen angegeben sind. Weitere Informationen zu dem Objekt 'AdminTask' enthält die Dokumentation für WebSphere Application Server Network Deployment, Version 6.1.

**Anmerkung:** **Vista** Das Produkt verwendet eine Jython-Version, die die Betriebssysteme Microsoft Windows 2003 bzw. Windows Vista nicht unterstützt.



## Parameter

### -nodeName

Der Name des Knotens, auf dem Events Service inaktiviert werden soll. Bei Angabe dieses Parameters muss auch der Parameter **serverName** angegeben werden. Dieser Parameter darf nicht angegeben werden, wenn der Parameter **clusterName** angegeben wird.

### -serverName

Der Name des Servers, auf dem Events Service inaktiviert werden soll. Dieser Parameter muss angegeben werden, wenn der Parameter **nodeName** angegeben wird. Dieser Parameter darf nicht angegeben werden, wenn der Parameter **clusterName** angegeben wird.

### -clusterName

Der Name des Clusters, auf dem Events Service inaktiviert werden soll. Dieser Parameter darf nicht angegeben werden, wenn die Parameter **nodeName** und **serverName** angegeben werden.

## Muster

### Syntaxbeispiel für Stapelmodus:

- Verwendung von JACL:

```
$AdminTask disableEventService {-nodeName knotenname
-serverName servername}
```

```
$AdminTask disableEventService {-clusterName clustername}
```

- Verwendung von Jython-Zeichenfolge:

```
AdminTask.disableEventService(['-nodeName knotenname
-serverName servername'])
```

```
AdminTask.disableEventService(['-clusterName clustername'])
```

- Verwendung von Jython-Liste:

```
AdminTask.disableEventService(['-nodeName', 'knotenname',
'-serverName', '-servername'])
```

```
AdminTask.disableEventService(['-clusterName', 'clustername'])
```

### Syntaxbeispiel für interaktiven Modus:

- Verwendung von JACL:

```
$AdminTask disableEventService {-interactive}
```

- Verwendung von Jython-Zeichenfolge:

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

- Verwendung von Jython-Liste:

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

---

## Befehl 'showEventServiceStatus'

Verwenden Sie den Befehl `showEventServiceStatus`, um den Events Service-Status auf Ihrem Server anzuzeigen.

### Zweck

Der Befehl `showEventServiceStatus` ist ein Verwaltungsbefehl von Common Event Infrastructure und für das Objekt 'AdminTask' verfügbar. Verwenden Sie diesen Befehl, um den Events Service-Status auf einem Server oder in einem Cluster zurück-

zugeben. Wenn die Ausführung dieser Task ohne Angabe von Parametern erfolgt, wird der Status aller Events Services (Ereignisservices) angezeigt. Wenn Sie die Liste der Events Services, die angezeigt werden sollen, filtern möchten, geben Sie 'nodeName', 'serverName' oder 'clusterName' an. Weitere Informationen zu dem Objekt 'AdminTask' enthält die Dokumentation für WebSphere Application Server Network Deployment, Version 6.1.

**Anmerkung:**   Das Produkt verwendet eine Jython-Version, die die Betriebssysteme Microsoft Windows 2003, Windows Vista und Windows 7 nicht unterstützt.

## Parameter

### -nodeName

Verwenden Sie diesen Parameter, wenn nur der Status der Events Services angezeigt werden soll, die zu dem angegebenen Knoten gehören. Dieser Parameter darf nicht angegeben werden, wenn der Parameter **clusterName** angegeben wird.

### -serverName

Verwenden Sie diesen Parameter, wenn nur der Status der Events Services angezeigt werden soll, die zu dem angegebenen Server gehören. Sie können diesen Parameter in Verbindung mit dem Parameter **nodeName** verwenden, um den Status aller Events Services anzugeben, die zu dem angegebenen Knoten und Server gehören. Dieser Parameter darf nicht angegeben werden, wenn der Parameter **clusterName** angegeben wird.

### -clusterName

Verwenden Sie diesen Parameter, wenn nur der Status der Events Services angezeigt werden soll, die zu dem angegebenen Cluster gehören. Dieser Parameter darf nicht angegeben werden, wenn die Parameter **nodeName** oder **serverName** angegeben werden.

## Muster

### Syntaxbeispiel für Stapelmodus:

- Verwendung von JACL:

```
$AdminTask showEventServiceStatus {-nodeName knotenname
-serverName servername}
```

```
$AdminTask showEventServiceStatus {-clusterName clustername}
```

- Verwendung von Jython-Zeichenfolge:

```
AdminTask.showEventServiceStatus(['-nodeName knotenname
-serverName servername'])
```

```
AdminTask.showEventServiceStatus(['-clusterName clustername'])
```

- Verwendung von Jython-Liste:

```
AdminTask.showEventServiceStatus(['-nodeName', 'knotenname',
'-serverName', '-servername'])
```

```
AdminTask.showEventServiceStatus(['-clusterName', 'clustername'])
```

### Syntaxbeispiel für interaktiven Modus:

- Verwendung von JACL:

```
$AdminTask showEventServiceStatus {-interactive}
```

- Verwendung von Jython-Zeichenfolge:

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

- Verwendung von Jython-Liste:  
`AdminTask.showEventServiceStatus(['-interactive'])`

---

## Befehl 'removeEventService'

Verwenden Sie den Befehl `removeEventService`, um die Events Service-Anwendung von Ihrem Server zu entfernen.

### Zweck

Der Befehl `removeEventService` ist ein Verwaltungsbefehl von Common Event Infrastructure und für das Objekt 'AdminTask' verfügbar. Verwenden Sie diesen Befehl, um Events Service von einem Server oder aus einem Cluster zu entfernen. Weitere Informationen zu dem Objekt 'AdminTask' enthält die Dokumentation für WebSphere Application Server Network Deployment, Version 6.1.

**Anmerkung:**   Das Produkt verwendet eine Jython-Version, die die Betriebssysteme Microsoft Windows 2003, Windows Vista und Windows 7 nicht unterstützt.

### Parameter

#### -nodeName

Der Name des Knotens, von dem Events Service entfernt werden soll. Bei Angabe dieses Parameters muss auch der Parameter **serverName** angegeben werden. Dieser Parameter darf nicht angegeben werden, wenn der Parameter **clusterName** angegeben wird.

#### -serverName

Der Name des Servers, von dem Events Service entfernt werden soll. Dieser Parameter muss angegeben werden, wenn der Parameter **nodeName** angegeben wird. Dieser Parameter darf nicht angegeben werden, wenn der Parameter **clusterName** angegeben wird.

#### -clusterName

Der Name des Clusters, von dem Events Service entfernt werden soll. Dieser Parameter darf nicht angegeben werden, wenn die Parameter **nodeName** und **serverName** angegeben werden.

### Muster

#### Syntaxbeispiel für Stapelmodus:

- Verwendung von JACL:  

```
$AdminTask removeEventService {-nodeName knotenname
 -serverName servername}

$AdminTask removeEventService {-clusterName clustername}
```
- Verwendung von Jython-Zeichenfolge:  

```
AdminTask.removeEventService(['-nodeName knotenname
 -serverName servername'])

AdminTask.removeEventService(['-clusterName clustername'])
```
- Verwendung von Jython-Liste:  

```
AdminTask.removeEventService(['-nodeName', 'knotenname',
 '-serverName', '-servername'])

AdminTask.removeEventService(['-clusterName', 'clustername'])
```

### Syntaxbeispiel für interaktiven Modus:

- Verwendung von JACL:  
`$AdminTask removeEventService {-interactive}`
- Verwendung von Jython-Zeichenfolge:  
`AdminTask.removeEventService(['-interactive'])`
- Verwendung von Jython-Liste:  
`AdminTask.removeEventService(['-interactive'])`

---

## Befehl 'removeEventServiceMdb'

Verwenden Sie den Befehl `removeEventServiceMdb`, um die MDB (Message-driven Bean) von Events Service von Ihrem Server zu entfernen.

### Zweck

Der Befehl `removeEventServiceMdb` ist ein Verwaltungsbefehl von Common Event Infrastructure und für das Objekt 'AdminTask' verfügbar. Verwenden Sie diesen Befehl, um die Events Service-MDB von einem Server oder aus einem Cluster zu entfernen. Weitere Informationen zu dem Objekt 'AdminTask' enthält die Dokumentation für WebSphere Application Server Network Deployment, Version 6.1.

**Anmerkung:**   Das Produkt verwendet eine Jython-Version, die die Betriebssysteme Microsoft Windows 2003, Windows Vista und Windows 7 nicht unterstützt.

### Parameter

#### -nodeName

Der Name des Knotens, von dem die Events Service-MDB entfernt werden soll. Bei Angabe dieses Parameters muss auch der Parameter **serverName** angegeben werden. Dieser Parameter darf nicht angegeben werden, wenn der Parameter **clusterName** angegeben wird.

#### -serverName

Der Name des Servers, von dem die Events Service-MDB entfernt werden soll. Bei Angabe dieses Parameters muss auch der Parameter **serverName** angegeben werden. Dieser Parameter darf nicht angegeben werden, wenn der Parameter **clusterName** angegeben wird.

#### -clusterName

Der Name des Clusters, aus dem die Events Service-MDB entfernt werden soll. Dieser Parameter darf nicht angegeben werden, wenn die Parameter **nodeName** und **serverName** angegeben werden.

#### -applicationName

Der Name der Events Service-MDB-Anwendung, die von einem Server oder aus einem Cluster entfernt werden soll.

### Muster

#### Syntaxbeispiel für Stapelmodus:

- Verwendung von JACL:  
`$AdminTask removeEventServiceMdb {-applicationName anwendungsname  
-nodeName knotenname -serverName servername}`  
`$AdminTask removeEventServiceMdb {-applicationName anwendungsname  
-clusterName clustername}`

- Verwendung von Jython-Zeichenfolge:
 

```
AdminTask.removeEventServiceMdb('[-applicationName anwendungsname
-nodeName knotenname -serverName servername]')
AdminTask.removeEventServiceMdb('[-applicationName anwendungsname
-clusterName clustername]')
```
- Verwendung von Jython-Liste:
 

```
AdminTask.removeEventServiceMdb (['-applicationName',
'anwendungsname', '-nodeName', 'knotenname', '-serverName', 'servername'])
AdminTask.removeEventServiceMdb (['-applicationName',
'anwendungsname', '-clusterName', 'clustername'])
```

#### Syntaxbeispiel für interaktiven Modus:

- Verwendung von JAQL:
 

```
$AdminTask removeEventServiceMdb {-interactive}
```
- Verwendung von Jython-Zeichenfolge:
 

```
AdminTask.removeEventServiceMdb('[-interactive]')
```
- Verwendung von Jython-Liste:
 

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

---

## Befehl 'removeEventServiceDB2DB'

Verwenden Sie den Befehl `removeEventServiceDB2DB`, um Events Service und wahlweise auch die zugeordnete DB2-Ereignisdatenbank zu entfernen.

### Zweck

Der Befehl `removeEventServiceDB2DB` ist ein Verwaltungsbefehl von Common Event Infrastructure und für das Objekt 'AdminTask' verfügbar. Verwenden Sie diesen Befehl, um die Events Service-Datenbank und Datenquellen für DB2 von einem Server oder aus einem Cluster zu entfernen. Weitere Informationen zu dem Objekt 'AdminTask' enthält die Dokumentation für WebSphere Application Server Network Deployment, Version 6.1.

**Anmerkung:**   Das Produkt verwendet eine Jython-Version, die die Betriebssysteme Microsoft Windows 2003, Windows Vista und Windows 7 nicht unterstützt.

### Parameter

#### -removeDB

Der Befehl entfernt die Datenbank, wenn für diesen Parameter der Wert `true` angegeben ist. Die Datenbank wird mit dem Befehl nicht entfernt, wenn hierfür der Wert `false` festgelegt ist. Zum Entfernen der Datenbank muss der aktuelle Server bereits für die Ausführung der Datenbankbefehle konfiguriert sein.

#### -nodeName

Der Name des Knotens, der den Server enthält, von dem die Datenquelle für Events Service entfernt werden soll. Bei Angabe dieses Parameters muss auch der Parameter `serverName` angegeben werden. Dieser Parameter darf nicht angegeben werden, wenn der Parameter `clusterName` angegeben wird.

#### -serverName

Der Name des Servers, von dem die Datenquelle für Events Service entfernt werden soll. Wenn dieser Parameter ohne Angabe des Parameters `nodeName`

definiert wird, verwendet der Befehl den Knotennamen des aktuellen Web-Sphere-Profiles. Dieser Parameter darf nicht angegeben werden, wenn der Parameter **clusterName** angegeben wird.

#### **-clusterName**

Der Name des Clusters, aus dem die Datenquelle für Events Service entfernt werden soll. Bei Angabe dieses Parameters dürfen die Parameter **serverName** und **nodeName** nicht definiert werden. Dieser Parameter darf nicht angegeben werden, wenn die Parameter **serverName** und **nodeName** angegeben werden.

#### **-dbUser**

Die DB2-Benutzer-ID mit der Berechtigung zum Erstellen und Löschen der Datenbanken. Dieser Parameter muss angegeben werden, wenn für den Parameter **removeDB** der Wert true definiert ist. Als Standardwert wird db2inst1 verwendet, sofern keine Angabe gemacht wurde.

#### **-dbPassword**

Das DB2-Kennwort. Dieser Parameter muss angegeben werden, wenn für den Parameter **removeDB** der Wert true definiert ist.

#### **-dbScriptDir**

Das Verzeichnis, das die Datenbankskripts enthält, die durch den Konfigurationsbefehl für die Events Service-Datenbank generiert wurden. Wenn dieser Parameter angegeben ist, führt der Befehl die Skripts in diesem Verzeichnis aus und entfernt so die Events Service-Datenbank. Das Standardausgabeverzeichnis für Datenbankskripts ist *'profilstammverzeichnis/databases/event/knoten/server/dbscripts/db2'*.

## **Muster**

### **Syntaxbeispiel für Stapelmodus:**

- Verwendung von JACL:

```
$AdminTask removeEventServiceDB2DB {-removeDB true
-nodeName knotenname
-serverName servername
-dbUser db2inst1
-dbPassword dbkennwort }
```

- Verwendung von Jython-Zeichenfolge:

```
AdminTask.removeEventServiceDB2DB(['-removeDB true
-nodeName knotenname
-serverName servername
-dbUser db2inst1
-dbPassword dbkennwort'])
```

- Verwendung von Jython-Liste:

```
AdminTask.removeEventServiceDB2DB(['-removeDB', 'true',
'-nodeName', 'knotenname',
'-serverName', 'servername',
'-dbUser', 'db2inst1',
'-dbPassword', 'dbkennwort'])
```

### **Syntaxbeispiel für interaktiven Modus:**

- Verwendung von JACL:

```
$AdminTask removeEventServiceDB2DB -interactive
```

- Verwendung von Jython-Zeichenfolge:

```
AdminTask.removeEventServiceDB2DB(['-interactive'])
```

- Verwendung von Jython-Liste:

```
AdminTask.removeEventServiceDB2DB(['-interactive'])
```

---

## Befehl 'removeEventServiceDB2iSeriesDB'

Verwenden Sie den Befehl `removeEventServiceDB2iSeriesDB`, um Events Service und wahlweise auch die zugeordnete Ereignisdatenbank von DB2 für iSeries zu entfernen.

### Zweck

Der Befehl `removeEventServiceDB2iSeriesDB` ist ein Verwaltungsbefehl von Common Event Infrastructure und für das Objekt 'AdminTask' verfügbar. Verwenden Sie diesen Befehl, um Datenquellen für DB2 für iSeries von einem Server oder aus einem Cluster zu entfernen. Weitere Informationen zu dem Objekt 'AdminTask' enthält die Dokumentation für WebSphere Application Server Network Deployment, Version 6.1.

**Anmerkung:**   Das Produkt verwendet eine Jython-Version, die die Betriebssysteme Microsoft Windows 2003, Windows Vista und Windows 7 nicht unterstützt.

### Parameter

#### **-nodeName**

Der Name des Knotens, der den Server enthält, von dem die Datenquelle für Events Service entfernt werden soll. Bei Angabe dieses Parameters muss auch der Parameter **serverName** angegeben werden. Dieser Parameter darf nicht angegeben werden, wenn der Parameter **clusterName** angegeben wird.

#### **-serverName**

Der Name des Servers, von dem die Datenquelle für Events Service entfernt werden soll. Wenn dieser Parameter ohne Angabe des Parameters **nodeName** definiert wird, verwendet der Befehl den Knotennamen des aktuellen WebSphere-Profiles. Dieser Parameter darf nicht angegeben werden, wenn der Parameter **clusterName** angegeben wird.

#### **-clusterName**

Der Name des Clusters, aus dem die Datenquelle für Events Service entfernt werden soll. Bei Angabe dieses Parameters dürfen die Parameter **serverName** und **nodeName** nicht definiert werden. Dieser Parameter darf nicht angegeben werden, wenn die Parameter **serverName** und **nodeName** angegeben werden.

### Muster

#### Syntaxbeispiel für Stapelmodus:

- Verwendung von JACL:

```
$AdminTask removeEventServiceDB2iSeriesDB {-nodeName knotenname
-serverName servername }
```

- Verwendung von Jython-Zeichenfolge:

```
AdminTask.removeEventServiceDB2iSeriesDB(['-nodeName knotenname
-serverName servername'])
```

- Verwendung von Jython-Liste:

```
AdminTask.removeEventServiceDB2iSeriesDB(['-nodeName', 'knotenname',
'-serverName', 'servername'])
```

#### Syntaxbeispiel für interaktiven Modus:

- Verwendung von JACL:

```
$AdminTask removeEventServiceDB2iSeriesDB -interactive
```

- Verwendung von Jython-Zeichenfolge:  
AdminTask.removeEventServiceDB2iSeriesDB('[-interactive]')
- Verwendung von Jython-Liste:  
AdminTask.removeEventServiceDB2iSeriesDB(['-interactive'])

---

## Befehl 'removeEventServiceDB2ZOSDB'

Verwenden Sie den Befehl `removeEventServiceDB2ZOSDB`, um Events Service und wahlweise auch die zugeordnete Ereignisdatenbank von DB2 für z/OS zu entfernen.

### Zweck

Der Befehl `removeEventServiceDB2ZOSDB` ist ein Verwaltungsbefehl von Common Event Infrastructure und für das Objekt 'AdminTask' verfügbar. Verwenden Sie diesen Befehl, um die Events Service-Datenbank und Datenquellen für DB2 z/OS von einem Server oder aus einem Cluster zu entfernen. Weitere Informationen zu dem Objekt 'AdminTask' enthält die Dokumentation für WebSphere Application Server Network Deployment, Version 6.1.

**Anmerkung:**   Das Produkt verwendet eine Jython-Version, die die Betriebssysteme Microsoft Windows 2003, Windows Vista und Windows 7 nicht unterstützt.

### Parameter

#### -removeDB

Der Befehl entfernt die Datenbank, wenn für diesen Parameter der Wert `true` angegeben ist. Die Datenbank wird mit dem Befehl nicht entfernt, wenn hierfür der Wert `false` festgelegt ist. Zum Entfernen der Datenbank muss der aktuelle Server bereits für die Ausführung der Datenbankbefehle konfiguriert sein.

#### -nodeName

Der Name des Knotens, der den Server enthält, von dem die Datenquelle für Events Service entfernt werden soll. Bei Angabe dieses Parameters muss auch der Parameter `serverName` angegeben werden. Dieser Parameter darf nicht angegeben werden, wenn der Parameter `clusterName` angegeben wird.

#### -serverName

Der Name des Servers, von dem die Datenquelle für Events Service entfernt werden soll. Wenn dieser Parameter ohne Angabe des Parameters `nodeName` definiert wird, verwendet der Befehl den Knotennamen des aktuellen WebSphere-Profiles. Dieser Parameter darf nicht angegeben werden, wenn der Parameter `clusterName` angegeben wird.

#### -clusterName

Der Name des Clusters, aus dem die Datenquelle für Events Service entfernt werden soll. Bei Angabe dieses Parameters dürfen die Parameter `serverName` und `nodeName` nicht definiert werden. Dieser Parameter darf nicht angegeben werden, wenn die Parameter `serverName` und `nodeName` angegeben werden.

#### -dbName

Der Name der DB2-Datenbank. Auf dem DB2-Client ist dies der Name der katalogisierten Datenbank. Auf dem nativen z/OS-Server ist dies der Name des Datenbanksystems. Dieser Parameter muss angegeben werden, wenn für den Parameter `removeDB` der Wert `true` definiert ist. Als Standardname wird `event` verwendet, sofern keine Angabe gemacht wurde.



### **-dbUser**

Die DB2-Benutzer-ID mit der Berechtigung zum Erstellen und Löschen der Datenbanken. Dieser Parameter muss angegeben werden, wenn für den Parameter **removeDB** der Wert `true` definiert ist.

### **-dbPassword**

Das DB2-Kennwort. Dieser Parameter muss angegeben werden, wenn für den Parameter **removeDB** der Wert `true` definiert ist.

### **-dbScriptDir**

Das Verzeichnis, das die Datenbankskripts enthält, die durch den Konfigurationsbefehl für die Events Service-Datenbank generiert wurden. Wenn dieser Parameter angegeben ist, führt der Befehl die Skripts in diesem Verzeichnis aus und entfernt so die Events Service-Datenbank. Das Standardausgabeverzeichnis für Datenbankskripts ist `'profilstammverzeichnis/databases/event/knoten/server/dbscripts/db2zos'`.

## **Muster**

### **Syntaxbeispiel für Stapelmodus:**

- Verwendung von JACL:

```
$AdminTask removeEventServiceDB2ZOSDB {-removeDB true -nodeName knotenname
-serverName servername -dbUser db2benutzer -dbPassword dbkennwort}
```

- Verwendung von Jython-Zeichenfolge:

```
AdminTask.removeEventServiceDB2ZOSDB(['-removeDB true -nodeName knotenname
-serverName servername -dbUser db2benutzer -dbPassword dbkennwort'])
```

- Verwendung von Jython-Liste:

```
AdminTask.removeEventServiceDB2ZOSDB(['-removeDB', 'true', '-nodeName',
'knotenname', '-serverName', 'servername', '-dbUser', 'db2benutzer',
'-dbPassword', 'dbkennwort'])
```

### **Syntaxbeispiel für interaktiven Modus:**

- Verwendung von JACL:

```
$AdminTask removeEventServiceDB2ZOSDB -interactive
```

- Verwendung von Jython-Zeichenfolge:

```
AdminTask.removeEventServiceDB2ZOSDB(['-interactive'])
```

- Verwendung von Jython-Liste:

```
AdminTask.removeEventServiceDB2ZOSDB(['-interactive'])
```

---

## **Befehl 'removeEventServiceDerbyDB'**

Verwenden Sie den Befehl `removeEventServiceDerbyDB`, um Events Service und wahlweise auch die zugeordnete Derby-Ereignisdatenbank zu entfernen.

### **Zweck**

Der Befehl `removeEventServiceDerbyDB` ist ein Verwaltungsbefehl von Common Event Infrastructure und für das Objekt `'AdminTask'` verfügbar. Verwenden Sie diesen Befehl, um die Events Service-Datenbank und Datenquelle für Derby von einem Server oder aus einem Cluster zu entfernen. Weitere Informationen zu dem Objekt `'AdminTask'` enthält die Dokumentation für WebSphere Application Server Network Deployment, Version 6.1.

**Anmerkung:** **Vista** **Windows 7** Das Produkt verwendet eine Jython-Version, die die Betriebssysteme Microsoft Windows 2003, Windows Vista und Windows 7 nicht unterstützt.

## Parameter

### **-removeDB**

Der Befehl entfernt die Datenbank, wenn für diesen Parameter der Wert `true` angegeben ist. Die Datenbank wird mit dem Befehl nicht entfernt, wenn hierfür der Wert `false` festgelegt ist. Zum Entfernen der Datenbank muss der aktuelle Server bereits für die Ausführung der Datenbankbefehle konfiguriert sein.

### **-nodeName**

Der Name des Knotens, der den Server enthält, von dem die Datenquelle für Events Service entfernt werden soll. Bei Angabe dieses Parameters muss auch der Parameter **serverName** angegeben werden. Dieser Parameter darf nicht angegeben werden, wenn der Parameter **clusterName** angegeben wird.

### **-serverName**

Der Name des Servers, von dem die Datenquelle für Events Service entfernt werden soll. Wenn dieser Parameter ohne Angabe des Parameters **nodeName** definiert wird, verwendet der Befehl den Knotennamen des aktuellen WebSphere-Profiles. Dieser Parameter darf nicht angegeben werden, wenn der Parameter **clusterName** angegeben wird.

### **-clusterName**

Der Name des Clusters, aus dem die Datenquelle für Events Service entfernt werden soll. Bei Angabe dieses Parameters dürfen die Parameter **serverName** und **nodeName** nicht definiert werden. Dieser Parameter darf nicht angegeben werden, wenn die Parameter **serverName** und **nodeName** angegeben werden.

### **-dbScriptDir**

Das Verzeichnis, das die Datenbankskripts enthält, die durch den Konfigurationsbefehl für die Events Service-Datenbank generiert wurden. Wenn dieser Parameter angegeben ist, führt der Befehl die Skripts in diesem Verzeichnis aus und entfernt so die Events Service-Datenbank. Das Standardausgabeverzeichnis für Datenbankskripts ist `'profilstammverzeichnis/databases/event/knoten/server/dbscripts/derby'`.

## Muster

### **Syntaxbeispiel für Stapelmodus:**

- Verwendung von JACL:  

```
$AdminTask removeEventServiceDerbyDB {-removeDB true -nodeName knotenname
-serverName servername}
```
- Verwendung von Jython-Zeichenfolge:  

```
AdminTask.removeEventServiceDerbyDB(['-removeDB true -nodeName knotenname
-serverName servername'])
```
- Verwendung von Jython-Liste:  

```
AdminTask.removeEventServiceDerbyDB(['-removeDB', 'true', '-nodeName',
'knotenname', '-serverName', 'servername'])
```

### **Syntaxbeispiel für interaktiven Modus:**

- Verwendung von JACL:  

```
$AdminTask removeEventServiceDerbyDB -interactive
```
- Verwendung von Jython-Zeichenfolge:  

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

- Verwendung von Jython-Liste:  
`AdminTask.removeEventServiceDerbyDB(['-interactive'])`

---

## Befehl 'removeEventServiceInformixDB'

Verwenden Sie den Befehl `removeEventServiceInformixDB`, um Events Service und wahlweise auch die zugeordnete Informix-Ereignisdatenbank zu entfernen.

### Zweck

Der Befehl `removeEventServiceInformixDB` ist ein Verwaltungsbefehl von Common Event Infrastructure und für das Objekt 'AdminTask' verfügbar. Verwenden Sie diesen Befehl, um die Events Service-Datenbank und Datenquellen für Informix von einem Server oder aus einem Cluster zu entfernen. Weitere Informationen zu dem Objekt 'AdminTask' enthält die Dokumentation für WebSphere Application Server Network Deployment, Version 6.1.

**Anmerkung:**   Das Produkt verwendet eine Jython-Version, die die Betriebssysteme Microsoft Windows 2003, Windows Vista und Windows 7 nicht unterstützt.

### Parameter

#### -removeDB

Der Befehl entfernt die Datenbank, wenn für diesen Parameter der Wert `true` angegeben ist. Die Datenbank wird mit dem Befehl nicht entfernt, wenn hierfür der Wert `false` festgelegt ist. Zum Entfernen der Datenbank muss der aktuelle Server bereits für die Ausführung der Datenbankbefehle konfiguriert sein.

#### -nodeName

Der Name des Knotens, der den Server enthält, von dem die Datenquelle für Events Service entfernt werden soll. Bei Angabe dieses Parameters muss auch der Parameter `serverName` angegeben werden. Dieser Parameter darf nicht angegeben werden, wenn der Parameter `clusterName` angegeben wird.

#### -serverName

Der Name des Servers, von dem die Datenquelle für Events Service entfernt werden soll. Wenn dieser Parameter ohne Angabe des Parameters `nodeName` definiert wird, verwendet der Befehl den Knotennamen des aktuellen WebSphere-Profiles. Dieser Parameter darf nicht angegeben werden, wenn der Parameter `clusterName` angegeben wird.

#### -clusterName

Der Name des Clusters, aus dem die Datenquelle für Events Service entfernt werden soll. Bei Angabe dieses Parameters dürfen die Parameter `serverName` und `nodeName` nicht definiert werden. Dieser Parameter darf nicht angegeben werden, wenn die Parameter `serverName` und `nodeName` angegeben werden.

#### -dbScriptDir

Das Verzeichnis, das die Datenbankskripts enthält, die durch den Konfigurationsbefehl für die Events Service-Datenbank generiert wurden. Wenn dieser Parameter angegeben ist, führt der Befehl die Skripts in diesem Verzeichnis aus und entfernt so die Events Service-Datenbank. Das Standardausgabeverzeichnis für Datenbankskripts ist `'profilstammverzeichnis/databases/event/knoten/server/dbscripts/informix'`.

## Muster

### Syntaxbeispiel für Stapelmodus:

- Verwendung von JACL:  

```
$AdminTask removeEventServiceInformixDB {-removeDB true -nodeName knotenname
-serverName servername}
```
- Verwendung von Jython-Zeichenfolge:  

```
AdminTask.removeEventServiceInformixDB(['-removeDB true -nodeName knotenname
-serverName servername'])
```
- Verwendung von Jython-Liste:  

```
AdminTask.removeEventServiceInformixDB(['-removeDB', 'true', '-nodeName',
'knotenname', '-serverName', 'servername'])
```

### Syntaxbeispiel für interaktiven Modus:

- Verwendung von JACL:  

```
$AdminTask removeEventServiceInformixDB -interactive
```
- Verwendung von Jython-Zeichenfolge:  

```
AdminTask.removeEventServiceInformixDB(['-interactive'])
```
- Verwendung von Jython-Liste:  

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

---

## Befehl 'removeEventServiceOracleDB'

Verwenden Sie den Befehl `removeEventServiceOracleDB`, um Events Service und wahlweise auch die zugeordnete Oracle-Ereignisdatenbank zu entfernen.

### Zweck

Der Befehl `removeEventServiceOracleDB` ist ein Verwaltungsbefehl von Common Event Infrastructure und für das Objekt 'AdminTask' verfügbar. Verwenden Sie diesen Befehl, um die Events Service-Tabellen und die Datenquellen für Oracle von einem Server oder aus einem Cluster zu entfernen. Die Datenbank selbst wird mit diesem Befehl nicht entfernt. Weitere Informationen zu dem Objekt 'AdminTask' enthält die Dokumentation für WebSphere Application Server Network Deployment, Version 6.1.

**Anmerkung:**   Das Produkt verwendet eine Jython-Version, die die Betriebssysteme Microsoft Windows 2003, Windows Vista und Windows 7 nicht unterstützt.

### Parameter

#### **-removeDB**

Der Befehl entfernt die Events Service-Tabellen, wenn für diesen Parameter der Wert `true` angegeben ist. Die Tabellen werden mit dem Befehl nicht entfernt, wenn hierfür der Wert `false` festgelegt ist.

#### **-nodeName**

Der Name des Knotens, der den Server enthält, von dem die Datenquelle für Events Service entfernt werden soll. Bei Angabe dieses Parameters muss auch der Parameter **serverName** angegeben werden. Dieser Parameter darf nicht angegeben werden, wenn der Parameter **clusterName** angegeben wird.

#### **-serverName**

Der Name des Servers, von dem die Datenquelle für Events Service entfernt

werden soll. Wenn dieser Parameter ohne Angabe des Parameters **nodeName** definiert wird, verwendet der Befehl den Knotennamen des aktuellen Web-Sphere-Profiles. Dieser Parameter darf nicht angegeben werden, wenn der Parameter **clusterName** angegeben wird.

**-clusterName**

Der Name des Clusters, aus dem die Datenquelle für Events Service entfernt werden soll. Bei Angabe dieses Parameters dürfen die Parameter **serverName** und **nodeName** nicht definiert werden. Dieser Parameter darf nicht angegeben werden, wenn die Parameter **serverName** und **nodeName** angegeben werden.

**-sysUser**

Die Benutzer-ID des Oracle-Datenbankbenutzers 'sys'. Der Standardwert lautet sys, sofern keine Angabe gemacht wurde.

**-sysPassword**

Das Kennwort für den Benutzer, der mit dem 'sysUser' angegeben wurde.

**-dbScriptDir**

Das Verzeichnis, das die Datenbankskripts enthält, die durch den Konfigurationsbefehl für die Events Service-Datenbank generiert wurden. Wenn dieser Parameter angegeben ist, führt der Befehl die Skripts in diesem Verzeichnis aus und entfernt so die Events Service-Datenbank. Das Standardausgabeverzeichnis für Datenbankskripts ist *'profilstammverzeichnis/databases/event/knoten/server/dbscripts/oracle'*.

## Muster

### Syntaxbeispiel für Stapelmodus:

- Verwendung von JACL:

```
$AdminTask removeEventServiceOracleDB {-removeDB true -nodeName knotenname
-serverName servername -sysUser sys -sysPassword sys-kennwort}
```

- Verwendung von Jython-Zeichenfolge:

```
AdminTask.removeEventServiceOracleDB(['-removeDB true -nodeName knotenname
-serverName servername -sysUser sys -sysPassword sys-kennwort'])
```

- Verwendung von Jython-Liste:

```
AdminTask.removeEventServiceOracleDB(['-removeDB', 'true', '-nodeName',
'knotenname', '-serverName', 'servername', '-sysUser', 'sys',
'-sysPassword', 'sys-kennwort'])
```

### Syntaxbeispiel für interaktiven Modus:

- Verwendung von JACL:

```
$AdminTask removeEventServiceOracleDB -interactive
```

- Verwendung von Jython-Zeichenfolge:

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

- Verwendung von Jython-Liste:

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

---

## Befehl 'removeEventServiceSQLServerDB'

Verwenden Sie den Befehl `removeEventServiceSQLServerDB`, um Events Service und wahlweise auch die zugeordnete SQL Server-Ereignisdatenbank zu entfernen.

## Zweck

Der Befehl `removeEventServiceSQLServerDB` ist ein Verwaltungsbefehl von Common Event Infrastructure und für das Objekt 'AdminTask' verfügbar. Verwenden Sie diesen Befehl, um die Events Service-Datenbank und Datenquellen für SQL Server von einem Server oder aus einem Cluster zu entfernen. Weitere Informationen zu dem Objekt 'AdminTask' enthält die Dokumentation für WebSphere Application Server Network Deployment, Version 6.1.

**Anmerkung:**   Das Produkt verwendet eine Jython-Version, die die Betriebssysteme Microsoft Windows 2003, Windows Vista und Windows 7 nicht unterstützt.

## Parameter

### **-removeDB**

Der Befehl entfernt die Datenbank, wenn für diesen Parameter der Wert `true` angegeben ist. Die Datenbank wird mit dem Befehl nicht entfernt, wenn hierfür der Wert `false` festgelegt ist. Zum Entfernen der Datenbank muss der aktuelle Server bereits für die Ausführung der Datenbankbefehle konfiguriert sein.

### **-nodeName**

Der Name des Knotens, der den Server enthält, von dem die Datenquelle für Events Service entfernt werden soll. Bei Angabe dieses Parameters muss auch der Parameter `serverName` angegeben werden. Dieser Parameter darf nicht angegeben werden, wenn der Parameter `clusterName` angegeben wird.

### **-serverName**

Der Name des Servers, von dem die Datenquelle für Events Service entfernt werden soll. Wenn dieser Parameter ohne Angabe des Parameters `nodeName` definiert wird, verwendet der Befehl den Knotennamen des aktuellen WebSphere-Profiles. Dieser Parameter darf nicht angegeben werden, wenn der Parameter `clusterName` angegeben wird.

### **-clusterName**

Der Name des Clusters, aus dem die Datenquelle für Events Service entfernt werden soll. Bei Angabe dieses Parameters dürfen die Parameter `serverName` und `nodeName` nicht definiert werden. Dieser Parameter darf nicht angegeben werden, wenn die Parameter `serverName` und `nodeName` angegeben werden.

### **-dbServerName**

Der Servername der SQL Server-Datenbank. Dieser Parameter muss angegeben werden, wenn für den Parameter `removeDB` der Wert `true` definiert ist.

### **-dbUser**

Die Benutzer-ID für SQL Server, die Eigner der Events Service-Tabellen definiert ist. Der Standardwert lautet `ceiuser`, sofern keine andere Angabe gemacht wurde.

### **-saUser**

Eine Benutzer-ID mit den Berechtigungen zum Löschen der Datenbanken und Benutzer. Als Standardwert wird `sa` verwendet, sofern keine Angabe gemacht wurde.

### **-saPassword**

Das Kennwort für den Benutzer, der durch den Parameter `saUser` angegeben ist. Dieser Parameter ist erforderlich, wenn für den Parameter `removeDB` der Wert `true` definiert ist.

### **-dbScriptDir**

Das Verzeichnis, das die Datenbankskripts enthält, die durch den Konfigurationsbefehl für die Events Service-Datenbank generiert wurden. Wenn dieser Parameter angegeben ist, führt der Befehl die Skripts in diesem Verzeichnis aus und entfernt so die Events Service-Datenbank. Das Standardausgabeverzeichnis für Datenbankskripts ist `'profilstammverzeichnis/databases/event/knoten/server/dbscripts/sqlserver'`.

## **Muster**

### **Syntaxbeispiel für Stapelmodus:**

- Verwendung von JACL:  

```
$AdminTask removeEventServiceSQLServerDB {-removeDB true -nodeName knotenname
-serverName servername -dbUser cei-benutzer -saUser sa -saPassword sa-kennwort
-dbServerName sql-servername}
```
- Verwendung von Jython-Zeichenfolge:  

```
AdminTask.removeEventServiceSQLServerDB(['-removeDB true -nodeName knotenname
-serverName servername -dbUser cei-benutzer -saUser sa -saPassword sa-kennwort
-dbServerName sql-servername'])
```
- Verwendung von Jython-Liste:  

```
AdminTask.removeEventServiceSQLServerDB(['-removeDB', 'true', '-nodeName',
'knotenname', '-serverName', 'servername', '-dbUser', 'cei-benutzer', '-saUser', 'sa',
'-saPassword', 'sa-kennwort', '-dbServerName', 'sql-servername'])
```

### **Syntaxbeispiel für interaktiven Modus:**

- Verwendung von JACL:  

```
$AdminTask removeEventServiceSQLServerDB -interactive
```
- Verwendung von Jython-Zeichenfolge:  

```
AdminTask.removeEventServiceSQLServerDB(['-interactive'])
```
- Verwendung von Jython-Liste:  

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

---

## **Befehlszeilendienstprogramm 'eventbucket'**

Der Befehl **eventbucket** zeigt die Bucketkonfiguration der Ereignisdatenbank an bzw. ändert diese.

### **Zweck**

Zeigt die Bucketkonfiguration der Ereignisdatenbank an bzw. ändert diese.

**eventbucket [-status] [-change]**

### **Beschreibung**

Der Befehl **eventbucket** zeigt die Bucketkonfiguration der Ereignisdatenbank an bzw. ändert diese. Das Dienstprogramm zum schnellen Löschen verwendet Buckets, um die Daten alter Ereignisse aus der Ereignisdatenbank zu löschen. Wenn Sie diesen Befehl ausführen, können Sie die gegenwärtige Konfiguration anzeigen oder die aktiven und inaktiven Buckets austauschen.

**Sicherheit:** Wenn die WebSphere-Sicherheit aktiviert ist, muss Ihre Benutzer-ID dem Aufgabenbereich 'eventAdministrator' für die Ereignisadministration zugeordnet sein, damit Sie die Bucketkonfiguration der Ereignisdatenbank anzeigen oder ändern können.

## Parameter

### -status

Zeigt Informationen zur gegenwärtigen Bucketkonfiguration an, und zwar einschließlich der Einstellung für das aktive Bucket und dem Bucketüberprüfungsintervall, das festlegt, wie häufig das Datenspeicherplug-in überprüft, welches Bucket aktiv ist).

### -change

Tauscht die Buckets aus, sodass das aktive Bucket inaktiv wird und das inaktive Bucket aktiv wird. Sie können diese Option erst verwenden, wenn das inaktive Bucket leer ist.

## Beispiel

In diesem Beispiel wird die aktuelle Bucketkonfiguration angezeigt:

```
eventbucket -status
```

In diesem Beispiel werden das aktive und das inaktive Bucket gegeneinander ausgetauscht:

```
eventbucket -change
```

---

## Befehlszeilendienstprogramm 'eventpurge'

Der Befehl **eventpurge** löscht Ereignisse aus der Ereignisdatenbank.

### Zweck

Ereignisse werden aus der Ereignisdatenbank gelöscht.

```
eventpurge [-seconds sekunden | -end endzeit] [-group ereignisgruppe] [-severity wertigkeit] [-extensionname erweiterungsname] [-start startzeit] [-size größe]
```

### Beschreibung

Mit dem Befehl **eventpurge** werden Ereignisse aus der Ereignisdatenbank gelöscht. Sie können alle Ereignisse aus der Ereignisdatenbank löschen oder nur diejenigen Ereignisse löschen, die bestimmte Kriterien erfüllen.

**Sicherheit:** Wenn die WebSphere-Sicherheit aktiviert ist, muss Ihre Benutzer-ID dem Aufgabenbereich 'eventAdministrator' für die Ereignisadministration zugeordnet sein, damit Sie Ereignisse löschen können.

## Parameter

### -seconds *sekunden*

Das Mindestalter der Ereignisse, die gelöscht werden sollen. Als Wert für *sekunden* muss eine Ganzzahl angegeben werden. Es werden nur Ereignisse gelöscht, die älter als das in Sekunden angegebene Alter sind. Dieser Parameter ist erforderlich, wenn Sie nicht den Parameter **-end** angeben.

### -end *endzeit*



Die Endzeit der Ereignisgruppe, die Sie löschen möchten. Es werden nur diejenigen Ereignisse gelöscht, die vor der angegebenen Zeit generiert wurden. Die Angabe des Werts für *endzeit* muss im XML-Format für 'dateTime' erfolgen (JJJJ-MM-TTThh:mm:ss ). Die Zeit für 'Mittags am 1. Januar 2006 in Eastern Standard Time' wird beispielsweise als 2006-01-01T12:00:00-05:00 angegeben. Weitere Informationen zu dem Datentyp 'dateTime' finden Sie im XML-Schema unter [www.w3.org](http://www.w3.org).

Dieser Parameter ist erforderlich, wenn Sie nicht den Parameter **-seconds** angeben.

**-group** *ereignisgruppe*

Die Ereignisgruppe, aus der Ereignisse gelöscht werden sollen. Als Wert für *ereignisgruppe* muss der Name einer in der Common Event Infrastructure-Konfiguration definierten Ereignisgruppe angegeben werden. Dieser Parameter ist optional.

**-severity** *wertigkeit*

Die Wertigkeit der Ereignisse, die gelöscht werden sollen. Als Wert für *wertigkeit* muss eine Ganzzahl angegeben werden. Es werden nur Ereignisse gelöscht, deren Wertigkeit mit dem von Ihnen angegebenen Wert übereinstimmt. Dieser Parameter ist optional.

**-extensionname** *erweiterungsname*

Der Erweiterungsname der Ereignisse, die in den Löschvorgang einbezogen werden sollen. Verwenden Sie diesen Parameter, um den Löschvorgang auf bestimmte Typen einzuschränken. Es werden nur Ereignisse gelöscht, deren Wert für das Merkmal 'extensionName' mit dem für *erweiterungsname* festgelegten Wert übereinstimmt. Dieser Parameter ist optional.

**-start** *startzeit*

Die Startzeit der Ereignisgruppe, die Sie löschen möchten. Es werden nur diejenigen Ereignisse gelöscht, die nach der angegebenen Zeit generiert wurden. Die Angabe des Werts für *startzeit* muss im XML-Format für 'dateTime' erfolgen (JJJJ-MM-TTThh:mm:ss ). Dieser Parameter ist optional.

**-size** *größe*

Die Anzahl von Ereignissen, die bei einer einzigen Transaktion gelöscht werden sollen. Als Wert für *größe* muss eine Ganzzahl angegeben werden. Nachdem die angegebene Anzahl von Ereignissen gelöscht wurde, schreibt der Befehl die Transaktion fest und fährt erst im Anschluss mit einer neuen Transaktion fort. Dieser Parameter ist optional.

## Beispiel

Im folgenden Beispiel werden alle Ereignisse aus der Datenbank gelöscht, die eine Wertigkeit von 20 (harmlos) aufweisen und deren Generierung vor über 10 Minuten erfolgt ist.

```
eventpurge -group "All events" -severity 20 -seconds 600
```

