

IBM Cúram Social Program Management



事件适配器指南

V 6.0.5

IBM Cúram Social Program Management



事件适配器指南

V 6.0.5

修改时间: **2013 年 5 月**

此修订版适用于 IBM Cúram Social Program Management V6.0 5 及其所有后续发行版, 直到在新版本中另有说明为止。

Licensed Materials - Property of IBM.

© **Copyright IBM Corporation 2013, 2013.**

目录

图	v	3.1 概述	11
表	vii	3.2 什么是 WebSphere Message Broker?	11
		3.3 WebSphere Message Broker 针对 Curam 的设置	11
第 1 章 简介	1	第 4 章 管理事件适配器	15
1.1 目的	1	4.1 概述	15
1.2 目标受众	1	4.2 启用事件适配器	15
1.3 先决条件	1	4.3 启用和禁用外部事件类型	15
1.4 什么是 Intelligent Operations Center?	1	4.4 配置目标系统	16
1.5 什么是事件适配器?	1	第 5 章 附录	17
1.6 使用事件适配器	2	5.1 外部链接	17
第 2 章 开发用于使用“事件适配器”的代码	5	声明	19
2.1 概述	5	商标	21
2.2 配置外部事件类型	5		
2.3 执行服务层 API 调用	7		
第 3 章 WebSphere Message Broker			
for Curam	11		



1. 样本 CAP 事件.	2	5. 使用 CAPPublisher.	9
2. 添加具有 CTX 数据的事件类型代码表项目	5	6. 使用 XMLPublisher	10
3. 使用 DMX 数据配置外部事件类型	6	7. 具有 SOAP 包络的样本 CAP 事件.	12
4. 使用 SimplePublisher	8	8. 样本 XSL 变换	13

表

1. 缺省 CAP 事件值 7

第 1 章 简介

1.1 目的

本指南介绍 Cúram 事件适配器所提供的功能。描述如何开发用于使用“事件适配器”的代码，并提供管理员如何对其进行配置的信息。

1.2 目标受众

本文档的目标受众是要使用 Cúram 事件适配器向 Intelligent Operations Center 发送事件的开发人员和管理员。

1.3 先决条件

本文档假定读者熟悉 Cúram 应用程序、Intelligent Operations Center 和 IBM WebSphere Message Broker 的开发和管理。

1.4 什么是 Intelligent Operations Center?

Intelligent Operations Center for Smarter Cities 提供了可帮助城市领导人洞察城市各方面信息的执行仪表盘。该仪表盘可跨越多个机构，并且能向下钻取至每个底层机构，如应急管理、公共安全、社会服务、运输或水利。Intelligent Operations Center 让城市领导人能够：

- 利用整个城市中所有机构和部门的信息来作出明智的决定
- 预料各种问题，以便最大程度降低破坏对城市服务和运作的影响
- 协调各机构间的资源以便快速高效地解决问题

Intelligent Operations Center 旨在：

- 监视机构和全市运作
- 使市民和企业参与偶发事件报告和解决
- 通过社交媒体收集和分析市民的反馈
- 管理政府和商业运作的方方面面
- 使用最少的 IT 资源进行快速部署

有关更多信息可在 Intelligent Operations Center 产品网站上找到 - 请参见“附录”获取详细信息。

1.5 什么是事件适配器?

事件适配器

Cúram 包含“事件适配器”。“事件适配器”为 Cúram 提供一种机制，通过 Intelligent Operations Center 的 WebSphere Message Broker 实例借由 Web Service 将事件发布到 Intelligent Operations Center 中。

事件适配器能做什么?

“事件适配器”呈现了可将数据提供给 Intelligent Operations Center 的 API。调用该 API 时，将会创建一个 CAP V1.2（通用警报协议）事件，并通过 WebService 将此事件发送给 WebSphere Message Broker 的 Intelligent

Operations Center 实例。然后 WebSphere Message Broker 可将此事件重新定向到 Intelligent Operations Center 事件流。

通用警报协议 V1.2

通用警报协议是警报和通知的 OASIS 标准格式。有关 CAP V1.2 的更多信息可在 OASIS 网站上找到 - 请参见“附录”获取详细信息。CAP 事件示例如下所示:

```
<cap:alert xmlns:cap="urn:oasis:names:tc:emergency:cap:1.2" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="urn:oasis:names:tc:emergency:cap:1.2-CAP-v1.2-os.xsd">
  <cap:identifier>d2a42209-33a5-4fcc-8658-0fc2c08c95da</cap:identifier>
  <cap:sender>IBMCuramSample</cap:sender>
  <cap:sent>2012-07-12T15:11:14+01:00</cap:sent>
  <cap:status>Actual</cap:status>
  <cap:msgType>Alert</cap:msgType>
  <cap:scope>Public</cap:scope>
  <cap:code>KPI</cap:code>
  <cap:info>
    <cap:category>Other</cap:category>
    <cap:event>ea.ipa.pd</cap:event>
    <cap:urgency>Unknown</cap:urgency>
    <cap:severity>Moderate</cap:severity>
    <cap:certainty>Observed</cap:certainty>
    <cap:headline>Sample CAP Event</cap:headline>
    <cap:description>Sample CAP Event Description</cap:description>
    <cap:onset>2012-07-12T15:11:14+01:00</cap:onset>
    <cap:senderName>IBMCuramSample</cap:senderName>
    <cap:parameter>
      <cap:valueName>DisposalDateTime</cap:valueName>
      <cap:value>2012-07-12 15:11:14</cap:value>
    </cap:parameter>
    <cap:parameter>
      <cap:valueName>DisposalDuration</cap:valueName>
      <cap:value>0</cap:value>
    </cap:parameter>
    <cap:parameter>
      <cap:valueName>ProgramType</cap:valueName>
      <cap:value>reference</cap:value>
    </cap:parameter>
    <cap:parameter>
      <cap:valueName>DisposalResult</cap:valueName>
      <cap:value>Deny</cap:value>
    </cap:parameter>
  </cap:info>
</cap:alert>
```

图 1. 样本 CAP 事件

1.6 使用事件适配器

使用“事件适配器”将事件发布到 Intelligent Operations Center 涉及以下任务:

1. 开发代码以通过“事件适配器”API 将 CAP 事件发布到 Intelligent Operations Center 上。“开发用于使用事件适配器的代码”一节中对此进行了描述。
2. 配置 Intelligent Operations Center WebSphere Message Broker 实例，以显示用于从 Cúram Intelligent Operations Center 事件适配器接收消息并将这些消息重新定向到 Intelligent Operations Center CAP 事件流的 Web Service 端点。“配置 Intelligent Operations Center WebSphere Message Broker”一节中对此进行了描述。

3. 启用“事件适配器”组件和事件类型，配置“事件适配器”以指向 Intelligent Operations Center WebSphere Message Broker Web Service 端点。“管理事件适配器”一节中对此进行了描述。

第 2 章 开发用于使用“事件适配器”的代码

2.1 概述

本章概述如何开发使用“事件适配器”通过 Intelligent Operations Center 的 WebSphere Message Broker Web Service 端点将 CAP 消息发布到 Intelligent Operations Center 的代码。具体包含以下操作：

1. 为要发送到 Intelligent Operations Center 的事件创建代码表项和事件类型。
2. 开发要在“事件适配器”上调用的代码。

2.2 配置外部事件类型

将需要配置“外部事件类型”以便能够管理哪些事件会使 CAP 消息发送给 Message Broker。

此外，需要为每个外部事件类型添加代码表项目，从而使每个事件类型都有一个与之关联的代码表值。

有关代码表的更多信息，请参阅“系统配置指南”中的“应用程序配置”一节。

添加具有 CTX 的事件类型代码表项目

可以使用 CTX 数据文件来添加外部事件类型，如下例所示。

```
<codetables package="curam.codetable">
  <codetable
    java_identifier="EXTERNALEVENTTYPE"
    name="ExternalEventType"
  >
    <displaynames>
      <locale language="en">External Event Type</locale>
    </displaynames>
    <code
      default="false"
      java_identifier="EVENTADAPTOR_SAMPLE"
      status="ENABLED"
      value="ET2000"
    >
      <locale
        language="en"
        sort_order="0"
      >
        <description>Sample External Event</description>
        <annotation/>
      </locale>
    </code>
  </codetable>
</codetables>
```

图 2. 添加具有 CTX 数据的事件类型代码表项目

添加事件类型代码表项目

或者，还有另一种提供外部事件信息的机制，即使用管理应用程序 UI。

1. 以具有系统管理特权的用户身份登录应用程序，然后导航至“代码表”部分。
2. 搜索“ExternalEventType”。应返回“外部事件类型”代码表。

3. 在列表行菜单中单击“新建项目...”。
4. 在“新建代码表项目”模式对话框中，输入项目名称和代码表值，然后单击“保存”。根据需要重复此步骤。
5. 然后，在“代码表”选项卡菜单上单击“发布...”以使更改可用。

“外部事件类型”代码表项目已添加且即可配置。

使用 DMX 配置外部事件类型

或者，也可以使用 DMX 数据来配置外部事件类型，如下例所示。

```
<table name="EXTERNALEVENTTYPE">
  <column name="externalEventType" type="text" />
  <column name="externalEventEnabled" type="bool" />
  <column name="externalEventTypeID" type="id" />
  <column name="versionNo" type="number" />
  <row>
    <attribute name="externalEventType">
      <value>ET2000</value>
    </attribute>
    <attribute name="externalEventEnabled">
      <value>1</value>
    </attribute>
    <attribute name="externalEventTypeID">
      <value>8897986963777257472</value>
    </attribute>
    <attribute name="versionNo">
      <value>2</value>
    </attribute>
  </row>
</table>
```

图 3. 使用 DMX 数据配置外部事件类型

通过应用程序配置外部事件类型

另一种提供外部事件信息的机制是使用管理应用程序 UI。有关启用/禁用的详细信息，请参阅管理部分。

要访问“外部事件类型”管理页面，请以具有管理特权的用户身份登录，然后导航至“管理工作空间”。从快捷方式面板选择“事件适配器”。此时，“事件适配器”选项卡将打开。

添加外部事件类型

将需要把外部事件类型添加至列表，以便可以对其进行管理。

1. 从页面级菜单中选择“添加...”以打开“添加外部事件类型”模式对话框。
2. 在此模式对话框页面上选中要添加的事件的框，然后单击“是”。

选中的外部事件类型将添加至列表。

删除外部事件类型

可能需要删除外部事件类型。

1. 从要删除的外部事件的列表行菜单中选择“删除”。
2. 在“删除外部事件类型”模式对话框上单击“是”按钮将删除外部事件类型。

2.3 执行服务层 API 调用

Cúram 事件适配器上可调用三种 API 方法来创建 CAP 消息并将其发布到 Intelligent Operations Center 的 WebSphere Message Broker 实例中。

SimplePublisher

```
curam.eventadaptor.publishers.intf.SimplePublisherIntf.publish(SimpleHeader, EventAdaptorNVPairList, EXTERNALEVENTTYPEEntry)
```

SimplePublisher 接口允许发布具有多个缺省值填充值的 CAP 消息。此方法将 SimpleHeader 中的值和缺省值分配给 CAPHeader。随后此 CAPHeader 将作为参数传递给 curam.eventadaptor.publishers.intf.CAPPublisherIntf.publish 方法。

SimpleHeader 允许在 CAP 消息中设置一些值。可设置的值包括：

- 描述
- 事件
- 标题
- 发送者
- 开始

其余为缺省值，根据环境变量（在代码表中定义）进行设置。

表 1. 缺省 CAP 事件值

值名称	缺省值
标识	随机值
类别	“事件”
代码	“其他”
确定性	“已观测”
消息类型	“警报”
范围	“公共”
严重性	“中等”
状态	“实际”
紧急程度	“未知”

以下代码段说明如何使用 SimplePublisherIntf。

```

@Inject
private TargetSystemDAO targetSystemDAO;

@Inject
SimplePublisher simplePublisher;

public ClassConstructor(string arg0) {
    super(arg0);
    GuiceWrapper.getInjector().injectMembers(this);
}

public simplePublisherSample() {
    // How to set up the variables that are passed to the publish method

    SimpleHeader simpleHeader = new SimpleHeader();

    simpleHeader.description = "Sample description";
    simpleHeader.event = "Sample event title";
    simpleHeader.headline = "Sample headline";
    simpleHeader.sender = "Sample sender";
    simpleHeader.onset = header. DateTime.getCurrentDateTime();

    EventAdaptorNVPair nvPair1 = new EventAdaptorNVPair();
    nvPair1.name = "parameter1 name";
    nvPair1.value = "parameter1 value";

    EventAdaptorNVPair nvPair2 = new EventAdaptorNVPair();
    nvPair2.name = "parameter2 name";
    nvPair2.value = "parameter2 value";

    nvPairList.dtls.add(nvPair1);
    nvPairList.dtls.add(nvPair2);

    // Call the publish method
    simplePublisher.publish(simpleHeader, nvPairList, EXTERNALEVENTTYPEEntry.EVENTADAPTOR_SAMPLE);
}

```

图 4. 使用 *SimplePublisher*

CAPPublisher

```
curam.eventadaptor.publishers.intf.CAPPublisherIntf.publish(CAPHeader, EventAdaptorNVPairList,
EXTERNALEVENTTYPEEntry)
```

CAPPublisher 接口允许发布所有值都由 CAPHeader 参数中传递的值来填充的 CAP 消息。

此方法会验证 CAPHeader。随后会使用头内容通过 capXMLBuilder.convertToDocument() 来构建 CAP XML。然后将此变换中的输出作为参数传递给 curam.eventadaptor.publishers.intf.XMLPublisherIntf.publish 方法。

以下代码段说明如何使用 CAPPublisherIntf。

```

@Inject
private TargetSystemDAO targetSystemDAO;

@Inject
CAPPublisher capPublisher;

public ClassConstructor(string arg0) {
    super(arg0);
    GuiceWrapper.getInjector().injectMembers(this);
}

public capPublisherSample() {
    // How to set up the variables that are passed to the publish method

    CAPHeader capHeader = new CAPHeader();

    capHeader.description = "Sample description";
    capHeader.event = "Sample event name";
    capHeader.headline = "Sample headline";
    capHeader.identifier = "sampleIdentifier";
    capHeader.sender = "Sample sender";
    capHeader.category = EAALERTINFOCATEGORY.FIRE;
    capHeader.certainty = EAALERTINFOCERTAINTY.LIKELY;
    capHeader.messageType = EAALERTMESSAGETYPE.UPDATE;
    capHeader.onset = DateTime.getCurrentDateTime();
    capHeader.senderName = "the sendername";
    capHeader.scope = EAALERTSCOPE.PRIVATE;
    capHeader.severity = EAALERTINFOSEVERITY.EXTREME;
    capHeader.status = EAALERTSTATUS.TEST;
    capHeader.urgency = EAALERTINFOURGENCY.EXPECTED;
    capHeader.code = EAALERTCODE.EVENT;
    capHeader.webIdentifier = "http://www.curamssoftware.com";

    EventAdaptorNVPairList nvPairList = new EventAdaptorNVPairList();

    EventAdaptorNVPair nvPair1 = new EventAdaptorNVPair();
    nvPair1.name = "parameter1 name";
    nvPair1.value = "parameter1 value";

    EventAdaptorNVPair nvPair2 = new EventAdaptorNVPair();
    nvPair2.name = "parameter2 name";
    nvPair2.value = "parameter2 value";

    nvPairList.dtls.add(nvPair1);
    nvPairList.dtls.add(nvPair2);

    // Call the publish method
    capPublisher.publish(capHeader, nvPairList, EXTERNALEVENTTYPEEntry.EVENTADAPTOR_SAMPLE);
}

```

图 5. 使用 CAPPublisher

XMLPublisher

```
curam.eventadaptor.publishers.intf.XMLPublisherImpl.publish(Document, EXTERNALEVENTTYPEEntry)
```

XMLPublisher 接口允许发布具有文档参数中传递的 XML 文档的 SOAP 消息。由于 XML 是定制的，所以这能最大程度地控制任何 API 发布。当“事件适配器”要和与 CAP 消息配合使用的外部服务进行交互时，或者当首选项是定制工种 CAP 消息时，将使用此方法。

如果“外部事件类型”已启用，那么它将验证 XML，然后将文档发布到 Web Service 端点。如果“事件类型”未启用，那么它将不会发布文档。

以下代码段说明如何使用 XMLPublisherIntf。

```
@Inject
private TargetSystemDAO targetSystemDAO;

@Inject
XMLPublisher xmlPublisher;

public ClassConstructor(string arg0) {
    super(arg0);
    GuiceWrapper.getInjector().injectMembers(this);
}

public xmlPublisherSample(Document document) {
    // Call the publish method
    XmlPublisher.publish(document, EXTERNALEVENTTYPEEntry.EVENTADAPTOR_SAMPLE);
}
```

图 6. 使用 XMLPublisher

第 3 章 WebSphere Message Broker for Curam

3.1 概述

本章描述配置 WebSphere Message Broker 以启用接收、处理和发送 Curam 事件适配器创建的 CAP 事件所包含的步骤。

3.2 什么是 WebSphere Message Broker?

WebSphere Message Broker 是一个强大的消息代理程序，它能够使业务数据和信息按消息格式在不同的应用程序之间、跨多个软硬件平台进行传递。业务规则适用于通过消息代理程序传递的数据，以便重新定向、存储、检索和变换信息。

3.3 WebSphere Message Broker 针对 Curam 的设置

过程概述

Curam 事件适配器将事件发布到 SOAP 端点。作为 Intelligent Operations Center 组成部分的 WebSphere Message Broker 安装可用于显示 SOAP 端点引用，以接收由 Curam 事件适配器发布的消息。

Curam 事件适配器发布到 SOAP 端点的消息的格式与 Intelligent Operations Center 要求的格式略有不同。例如，事件包装在 SOAP 包络中。下文提供了 Curam 事件适配器发布的样本 CAP 事件。WebSphere Message Broker 也可用于从 SOAP 包络中抽取 CAP 消息并除去其他多余元素，从而使其严格遵守 Intelligent Operations Center 所要求的 CAP 消息格式。最后，WebSphere Message Broker 可用于将变换后的 CAP 消息转发给 Intelligent Operations Center，其方式是将消息重新定向到适当的 Intelligent Operations Center 输入队列。

样本 CAP 事件

下面是 Curam 事件适配器发送的具有 SOAP 包络的样本 CAP 事件:

```

<?xml version='1.0' encoding='UTF-8'?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope">
  <soapenv:Body>
    <ns1:publishEvent xmlns:ns1="http://remote.bs.publiclayer.ws.ea.curam">
      <xmlMessage>
        <cap:alert xmlns:cap="urn:oasis:names:tc:emergency:cap:1.2" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:oasis:names:tc:emergency:cap:1.2-CAP-v1.2-os.xsd">
          <cap:identifier>d2a42209-33a5-4fcc-8658-0fc2c08c95da</cap:identifier>
          <cap:sender>IBMCuramSample</cap:sender>
          <cap:sent>2012-07-12T15:11:14+01:00</cap:sent>
          <cap:status>Actual</cap:status>
          <cap:msgType>Alert</cap:msgType>
          <cap:scope>Public</cap:scope>
          <cap:code>KPI</cap:code>
          <cap:info>
            <cap:category>Other</cap:category>
            <cap:event>ea.ipa.pd</cap:event>
            <cap:urgency>Unknown</cap:urgency>
            <cap:severity>Moderate</cap:severity>
            <cap:certainity>Observed</cap:certainity>
            <cap:headline>Sample CAP Event</cap:headline>
            <cap:description>Sample CAP Event Description</cap:description>
            <cap:onset>2012-07-12T15:11:14+01:00</cap:onset>
            <cap:senderName>IBMCuramSample</cap:senderName>
            <cap:parameter>
              <cap:valueName>DisposalDateTime</cap:valueName>
              <cap:value>2012-07-12 15:11:14</cap:value>
            </cap:parameter>
            <cap:parameter>
              <cap:valueName>DisposalDuration</cap:valueName>
              <cap:value>0</cap:value>
            </cap:parameter>
            <cap:parameter>
              <cap:valueName>ProgramType</cap:valueName>
              <cap:value>areference</cap:value>
            </cap:parameter>
            <cap:parameter>
              <cap:valueName>DisposalResult</cap:valueName>
              <cap:value>Deny</cap:value>
            </cap:parameter>
          </cap:info>
        </cap:alert>
      </xmlMessage>
    </ns1:publishEvent>
  </soapenv:Body>
</soapenv:Envelope>

```

图 7. 具有 SOAP 包络的样本 CAP 事件

创建 WebSphere Message Broker 流

本节描述如何在 Intelligent Operations Center WebSphere Message Broker 安装内创建 Message Broker 流，以接收和处理 Cúram 事件适配器发送的 CAP 消息。此流将接受 Cúram SOAP 消息，将这些消息变换为 Intelligent Operations Center 所要求的精确格式，然后再将修改后的消息转发到 Intelligent Operations Center 输入队列。创建流中包含的步骤如下：

1. 在 WebSphere Message Broker Toolkit 内创建新的消息流文件 (.msgflow)。
2. **SOAP 输入：**创建 SOAPInput 节点显示 SOAP 端点，以接受并处理 Cúram 事件适配器发送的消息。必须使用 EventAdaptorService.wsdl 可部署 WSDL 文件配置节点。此文件可在 Cúram 开发安装的 EJBServer\components\EventAdaptor\axis\EventAdaptorService 中找到。
3. 创建节点，以将 Cúram 事件适配器发送的消息变换为 Intelligent Operations Center 要求的格式。其中包含除去 SOAP 包络、抽取 CAP 消息和除去多余 HTTP 头。可使用以下步骤实现此目的：

- a. **SOAP 抽取:** 创建 SOAPExtract 节点以除去 SOAP 包络, 只允许处理 SOAP 的主体。
- b. **XSL 变换:** 使用 XSL 变换抽取父 <xmlMessage> 元素内的 CAP 消息。例如, 以下 XSL 变换可用作基本 XSL 变换:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:output indent="yes"/>
  <xsl:template match="/">
    <xsl:copy-of select="//xmlMessage/node()"/>
  </xsl:template>
</xsl:stylesheet>
```

图 8. 样本 XSL 变换

- c. **HTTP 头:** 使用 HTTPHeader 节点删除输入消息中的所有 HTTP 头。
 - d. **重置内容描述符:** 使用 ResetContentDescriptor 节点请求以 XMLNSC 解析器解析消息。
4. **MQ 输出:** 最后, 需要将变换后的消息交付到 Intelligent Operations Center IOC.CAP.IN MQ 队列中。使用 MQOutput 节点将其交付到 IOC.CAP.IN MQ 队列中。

消息流创建后, 应生成 BAR 文件。然后将此 BAR 文件部署到 Intelligent Operations Center WebSphere Message Broker 实例中。

第 4 章 管理事件适配器

4.1 概述

本章提供如何配置“事件适配器”以供外部系统使用的概述。

4.2 启用事件适配器

关于此任务

要使用“事件适配器”将事件发送到 Intelligent Operations Center，首先必须启用“事件适配器”组件。可通过配置系统属性来实现此操作。要启用“事件适配器”，请：

过程

1. 以具有管理特权的用户身份登录应用程序，然后导航至“属性管理”部分。
2. 在“属性管理”选项卡上，搜索“事件适配器启用标志”。将返回“事件适配器启用标志”属性。
3. 在列表行菜单上，选择“编辑值”。此时，“编辑值”模式对话框将打开。
4. 在值文本框中输入“true”，然后单击“保存”。
5. 然后，在选项卡级菜单上单击“发布...”以使更改可用。

结果

“事件适配器”现已启用。

4.3 启用和禁用外部事件类型

“事件适配器”还提供细粒度配置，使管理员能够启用和禁用个别事件类型。本节描述如何实现此操作。

要访问“外部事件类型”管理页面，请以具有管理特权的用户身份登录，然后导航至“管理工作空间”。从快捷方式面板选择“事件适配器”。此时，“事件适配器”选项卡将打开。

启用外部事件类型

将需要启用外部事件类型以便使用此类型的所有事件可处于活动状态，并且能够访问外部系统。

1. 从事件的列表行菜单中选择“启用...”进行启用。
2. 在“启用外部事件类型”模式对话框中单击“是”按钮以启用事件。

禁用外部事件类型

外部事件类型可能会因为不再需要访问外部系统而需要禁用。

1. 从事件的列表行菜单中选择“禁用...”进行禁用。
2. 在“禁用外部事件类型”模式对话框中单击“是”按钮将禁用事件。

启用所有外部事件类型

将需要启用外部事件类型以便使用此类型的所有事件可处于活动状态，并且能够访问外部系统。使用此选项可以通过一个操作启用所有事件。

1. 要启用列示的所有事件，请从页面级菜单中选择“全部启用”。
2. 在此模式对话框中，选择“是”启用所有事件。

禁用所有外部事件类型

外部事件类型可能会因为不再需要访问外部系统而需要禁用。使用此选项可以通过一个操作禁用所有事件。

1. 要禁用列示的所有事件，请从页面级菜单中选择“全部禁用”。
2. 在此模式对话框中，选择“是”禁用所有事件。

4.4 配置目标系统

关于此任务

需要配置 Intelligent Operations Center WebSphere Message Broker Web Service 端点的位置，以便“事件适配器”能够为其分派事件。可通过在 Cúram 目标系统屏幕上配置端点 URL 来实现此操作。本节描述如何实现此操作。有关目标系统的更多信息，请参阅 Cúram System Administration Guide 的“目标系统”一节。

过程

1. 要设置目标系统，请以“sysadmin”登录。
2. 从中选择“目标系统”快捷方式（在快捷方式面板的“应用程序数据”部分），以打开“目标系统”选项卡。
3. 单击“新建...”以打开“新建目标系统”模型。
4. 输入目标系统名称和根 URL，然后单击“保存”。
5. 从列表行菜单单击“添加服务”。
6. 在此处选择“事件适配器 EventAdaptorService 端点引用”并输入“扩展 URL”- SOAP 端点上显示的服务（请参阅“WebSphere Message Broker for Cúram”一节以获取有关如何设置端点的更多信息），此外还需在必要时输入“用户名”和“密码”。

结果

“目标系统”现已配置为启用“事件适配器”来将 CAP 事件发布到 WebSphere Message Broker 的 Intelligent Operations Center 实例中。

第 5 章 附录

5.1 外部链接

Intelligent Operations Center - <http://www.ibm.com/software/products/us/en/intelligent-operations-center/>

OASIS CAP Standard - <http://docs.oasis-open.org/emergency/cap/v1.2/CAP-v1.2-os.html>

声明

本信息是为在美国提供的产品和服务编写的。IBM 可能在其它国家或地区不提供本文中讨论的产品、服务或功能特性。有关您当前所在区域的产品和服务的信息，请向您当地的 IBM 代表咨询。任何对 IBM 产品、程序或服务的引用并非意在明示或暗示只能使用 IBM 的产品、程序或服务。只要不侵犯 IBM 的知识产权，任何同等功能的产品、程序或服务，都可以代替 IBM 产品、程序或服务。但是，评估和验证任何非 IBM 产品、程序或服务的操作，由用户自行负责。IBM 公司可能已拥有或正在申请与本文档内容有关的各项专利。提供本文档并不意味着授予用户使用这些专利的任何许可。您可以用书面形式将许可查询寄往：

IBM Director of Licensing

IBM Corporation

North Castle Drive

Armonk, NY 10504-1785

U.S.A.

有关双字节 (DBCS) 信息的许可查询，请与您所在国家或地区的 IBM 知识产权部门联系，或用书面方式将查询寄往：

Intellectual Property Licensing

Legal and Intellectual Property Law.

IBM Japan Ltd.

19-21, Nihonbashi-Hakozakicho, Chuo-ku

Tokyo 103-8510, Japan

本条款不适用英国或这样的条款与当地法律不一致的任何国家或地区：International Business Machines Corporation“按现状”提供本出版物，不附有任何种类的（无论是明示的还是暗含的）保证，包括但不限于暗含的有关非侵权、适销或适用于某种特定用途的保证。某些国家或地区在某些交易中不允许免除明示或暗含的保证。因此本条款可能不适用于您。

本信息可能包含技术方面不够准确的地方或印刷错误。本信息将定期更改；这些更改将编入本信息的新版本中。IBM 可以随时对本出版物中描述的产品和/或程序进行改进和/或更改，而不另行通知。

本信息中对任何非 IBM Web 站点的引用都只是为了方便起见才提供的，不以任何方式充当对那些 Web 站点的保证。那些 Web 站点中的资料不是 IBM 产品资料的一部分，使用那些 Web 站点带来的风险将由您自行承担。

IBM 可以按它认为适当的任何方式使用或分发您所提供的任何信息而无须对您承担任何责任。本程序的被许可方如果要了解有关程序的信息以达到如下目的：(i) 使其能够在独立创建的程序和其它程序（包括本程序）之间进行信息交换，以及 (ii) 使其能够对已经交换的信息进行相互使用，请与下列地址联系：

IBM Corporation

Dept F6, Bldg 1

294 Route 100

Somers NY 10589-3216

U.S.A.

只要遵守适当的条款和条件，包括某些情形下的一定数量的付费，就可获得这方面的信息。

本文档中描述的许可程序及其所有可用的许可资料均由 IBM 依据 IBM 客户协议、IBM 国际程序许可协议或任何同等协议中的条款提供。

此处包含的任何性能数据都是在受控环境中测得的。因此，在其他操作环境中获取的数据可能会有明显的不同。有些测量可能是在开发级的系统上进行的，因此不保证与一般可用系统上进行的测量结果相同。此外，有些测量可能是通过推算估计出来的。实际结果可能会不同。本文档的用户应当验证其特定环境的适用数据。

涉及非 IBM 产品的信息可从这些产品的供应者、其出版说明或其他可公开获得的资料中获取。

IBM 没有对这些产品进行测试，也无法确认其性能的精确性、兼容性或任何其他关于非 IBM 产品的声明。有关非 IBM 产品性能的问题应当向这些产品的供应商提出。

除其所表示的目标和主题外，所有关于 IBM 未来发展方向和意图的声明，如有更改或撤销，恕不另行通知。

所有 IBM 的价格均是 IBM 当前的建议零售价，可随时更改而不另行通知。经销商的报价可能会不同。

本信息仅用于规划的目的。在所述产品可用之前，此处的信息可能会更改。

本信息包含日常业务运营中使用的数据与报告的示例。为了尽可能完整地说明这些示例，这些示例中可能会包括个人、公司、品牌和产品的名称。所有这些名称均属虚构，若与实际企业使用的名称和地址有任何雷同，纯属巧合。

版权许可证:

本信息包含源语言形式的样本应用程序，用以阐明在不同操作平台上的编程技术。如果是按照在编写样本程序的操作平台上的应用程序编程接口 (API) 进行应用程序的开发、使用、经销或分发为目的，您可以任何形式对这些样本程序进行复制、修改、分发，而无需向 IBM 付费。这些示例尚未在所有条件下经过全面测试。因此，IBM 不能保证或暗示这些程序的可靠性、可维护性或功能。这些实例程序“按现状”提供，不附有任何种类的保证。IBM 对于使用这些样本程序所造成的损害不应承担任何责任。

凡这些样本程序的每份拷贝或其任何部分或任何衍生产品，都必须包括如下版权声明:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs.

© Copyright IBM Corp. _enter the year or years_. All rights reserved.

如果您正在查看本信息的软拷贝，图片和彩色图例可能无法显示。

商标

IBM、IBM 徽标和 [ibm.com](http://www.ibm.com) 是 International Business Machines Corp. 在全球许多管辖区域注册的商标或注册商标。其他产品和服务名称可能是 IBM 或其他公司的商标。当前的 IBM 商标列表，可从位于 <http://www.ibm.com/legal/us/en/copytrade.shtml> 的 Web 站点上的“版权和商标信息”部分获取。

其他名称可能是它们各自的所有者的商标。其他公司、产品和服务名称可能是其他公司的商标或服务标记。



Printed in China