

Netcool/OMNIbus Gateway for Siebel  
Communications  
5.0

*Reference Guide*  
*November 30, 2012*



**Notice**

Before using this information and the product it supports, read the information in [Appendix A, “Notices and Trademarks,”](#) on page 37.

**Edition notice**

This (SC23-6394-03) edition applies to version 5.0 of Gateway for Siebel Communications and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SC23-6394-02.

© **Copyright International Business Machines Corporation 2006, 2012.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

---

# Contents

- Document control page..... V**
  
- Chapter 1. Gateway for Siebel Communications..... 1**
  - Summary..... 1
  - Gateway overview..... 1
  - Gateway for Siebel Communications architecture.....2
    - Gateway operation..... 2
    - Gateway components.....2
    - Service request generation ..... 3
    - Service request modification .....4
  - Installing the gateway.....4
    - Installing the gateway on Tivoli Netcool/OMNIbus V7.4.0..... 4
  - Setting up a unidirectional Gateway for Siebel Communications..... 5
  - Siebel system configuration for a unidirectional gateway..... 6
    - Step 1: Configuring integration objects and business services using Siebel Tools..... 6
    - Step 2: Setting up workflows..... 13
    - Step 3: Configuring the Siebel Web Engine..... 15
    - Step 4: Creating a Siebel account details..... 16
  - Configuring a unidirectional gateway..... 16
  - Creating the gateway conversions table..... 17
    - Manually editing the conversion table..... 17
  - Setting up a bidirectional Gateway for Siebel Communications..... 18
  - Siebel system configuration for a bidirectional gateway..... 18
    - Creating integration objects using Siebel tools..... 18
    - Setting up workflows.....19
    - Setting up the Siebel workflow policies and actions ..... 20
  - Configuring a bidirectional gateway..... 24
    - Configuring action SQL..... 25
  - Gateway configuration mappings..... 26
    - Mapping syntax.....26
    - Example configuration file..... 26
  - Writer attributes.....28
  - Frequently asked questions..... 33
    - Why can't I check out projects in Siebel Tools?..... 33
    - How do I change GW\_HOST and/or GW\_PORT?..... 34
  
- Appendix A. Notices and Trademarks..... 37**
  - Notices..... 37
  - Trademarks..... 38



## Document control page

---

Use this information to track changes between versions of this guide.

The IBM Tivoli® Gateway for Siebel Communications documentation is provided in softcopy format only. To obtain the most recent version, please visit the IBM Tivoli Netcool® Information Center:

[http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/index.jsp?topic=/com.ibm.tivoli.nam.doc/welcome\\_ptsm.htm](http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/index.jsp?topic=/com.ibm.tivoli.nam.doc/welcome_ptsm.htm)

Document Version	Publication Date	Comments
SC23-6394-00	April 27, 2007	First IBM® publication.
SC23-6394-01	July 25, 2008	<p>“<a href="#">Summary</a>” on page 1 section updated.</p> <p>Sybase support for Netcool/OMNIbus V7.2 has been added.</p> <p>All references to the nco_gwconv tool have been removed.</p> <p>Example configuration file section updated.</p> <p>SEPERATOR_CHAR attribute has been added.</p>
SC23-6394-02	September 30, 2010	<p>“<a href="#">Summary</a>” on page 1 section updated.</p> <p>All references to Sybase support tool have been removed.</p> <p>Descriptions for the following attributes added to “<a href="#">Writer attributes</a>” on page 28:</p> <ul style="list-style-type: none"><li>• Filter_condition</li><li>• Rebuild_Cache</li></ul>
SC23-6394-03	November 30, 2012	<p>Guide updated for Netcool/OMNIbus V7.4 release.</p> <p>“<a href="#">Installing the gateway</a>” on page 4 updated.</p>



---

# Chapter 1. Gateway for Siebel Communications

This guide describes the integration of IBM Tivoli Netcool/OMNIBus and Siebel Communications version 7.8.2.0 through the Gateway for Siebel Communications.

Siebel is the e-Business component of a complete solution for the communications industry. It is based on Siebel's highly configurable, scalable, and open Web-based architecture. It supports large communication providers with tens of thousands of users, as well as smaller and emerging service providers with fewer than ten users.

This guide contains the following sections:

- [“Summary” on page 1](#)
- [“Gateway overview” on page 1](#)
- [“Gateway for Siebel Communications architecture” on page 2](#)
- [“Installing the gateway” on page 4](#)
- [“Setting up a unidirectional Gateway for Siebel Communications” on page 5](#)
- [“Siebel system configuration for a unidirectional gateway” on page 6](#)
- [“Configuring a unidirectional gateway” on page 16](#)
- [“Creating the gateway conversions table” on page 17](#)
- [“Setting up a bidirectional Gateway for Siebel Communications” on page 18](#)

## Summary

---

Use this summary information to learn about the gateway.

The following table provides a summary of the Gateway for Siebel Communications.

Gateway target	Siebel Communications version 7.8.2.0
Gateway executable name	nco_g_siebel
Package version	2.0
Gateway supported on	For details of supported operating systems, see the following Release Notice on the IBM Software Support website: <a href="http://www-01.ibm.com/support/docview.wss?uid=swg21613138">http://www-01.ibm.com/support/docview.wss?uid=swg21613138</a>
Configuration files	\$OMNIHOME/gates/siebel/G_SIEBEL.conf
Requirements	A currently supported version of Tivoli Netcool/OMNIBus
Multicultural support	Available

## Gateway overview

---

Siebel Communications provides multi-channel access to users over dedicated, HTML, mobile/handheld, and wireless clients. It supports the widely-accepted International Telecommunications Union (ITU) Telecommunications Managed Network (TMN) framework and models.

Siebel's architecture consists of:

- Applet Manager for data presentation

- Object Manager for managing the business logic
- Data Manager for managing the database interaction
- Database for data storage

The Gateway for Siebel Communications allows IBM Tivoli Netcool/OMNIbus users to automatically create, update, and close service requests and their corresponding activities in Siebel. The gateway receives notification when alerts in the ObjectServer are created, modified, or deleted. The gateway also receives notification when a journal entry has been added to an alert. Filters can be used to determine which alerts are sent to the gateway.

The Gateway for Siebel Communications is bidirectional, which means that alerts pass out of the ObjectServer into Siebel, and if they are modified in Siebel, those changes can be sent back to the ObjectServer.

For more information about how to use Siebel Communications, refer to your Siebel documentation.

## Gateway for Siebel Communications architecture

---

The Gateway for Siebel Communications contains various components that work together to maintain alerts and Siebel Service Requests.

The architecture of the gateway is described in the following topics:

- [“Gateway operation” on page 2](#)
- [“Gateway components” on page 2](#)
- [“Service request generation ” on page 3](#)
- [“Service request modification ” on page 4](#)

### Gateway operation

The role of the gateway is to allow the flow of communications from IBM Tivoli Netcool/OMNIbus to Siebel.

The flow of communications is as follows:

1. In the ObjectServer, an alert that matches the filter requirements is created, updated, or deleted or a journal entry is added to one of these alerts.
2. The gateway process (nco\_g\_siebel) converts the Netcool alert into an Extensible Markup Language (XML) format and posts this to the Siebel Web Engine using the Hypertext Transfer Protocol (HTTP).
3. When the message is received by the Siebel Web Engine, it activates the appropriate business service within the Siebel Server.
4. The business service initiates a workflow process, which converts the XML formatted alert into a Siebel format and then creates or updates the relevant Siebel service request or activity.

### Gateway components

The Gateway for Siebel Communications comprises various components that, when combined, allows communications between IBM Tivoli Netcool/OMNIbus and Siebel.

This section describes the following gateway components:

- [“Gateway ” on page 3](#)
- [“Siebel Web Engine” on page 3](#)
- [“Siebel server” on page 3](#)



## Gateway

The gateway passes information between the ObjectServer and Siebel.

When the gateway receives alerts from the ObjectServer, it converts them into an XML format and then sends them as the body of an HTTP POST request to the Siebel Web Engine. The gateway also sends the Siebel Service Request ID of new Service Requests back to the ObjectServer so that alerts can be linked with associated Service Requests.

Whenever an alert is updated or deleted in the ObjectServer, the changes are mapped through the gateway to modify the appropriate Siebel Service Request. Likewise, if the alert journal is updated in the ObjectServer, the changes are mapped through the gateway and reflected in the associated Siebel Service Request.

The gateway also receives notification of any relevant changes within Siebel in an XML format using HTTP. It then parses the XML alert and sends the relevant details to the gateway.

**Note :** If an alert is deleted in the IBM Tivoli Netcool/OMNIBus, the associated Service Request is closed in Siebel but is not deleted.

## Siebel Web Engine

The Siebel Web Engine works with a standard HTTP Server (such as Microsoft IIS) to provide access to the Siebel Server.

The Siebel Web Engine provides access to the Siebel Server as follows:

1. An HTTP POST request is made to the HTTP Server.
2. Depending on the request, the Siebel Web Engine contacts the Siebel Server through the Siebel Gateway Server and triggers a predefined business service within Siebel.
3. The XML-formatted alert is extracted from the body of the HTTP request and passed on to the Siebel Server.
4. The Siebel Web Engine receives the result of its request to invoke the business service, which is passed back to the Siebel Writer Module in the form of a HTTP response to its POST request.

## Siebel server

The Siebel Server stores information required to manage and operate the Siebel suite of applications.

The Siebel Server processes a request from the Siebel Web Engine as follows:

1. The Siebel Server receives a request from the Siebel Web Engine to invoke a business service, which runs a predefined workflow process within Siebel.
2. This workflow process takes the XML-formatted alert from the Siebel Web Engine and converts it into a Siebel format.
3. The alert is then upserted (updated or inserted) into Siebel as a service request (alert) or Activity (journal entry).
4. Upon completion, the business service reports the outcome of the request back to the Siebel Web Engine.

When service requests that have been created by the gateway are modified or deleted within Siebel, or when an activity is associated with one of those service requests, the Siebel Server converts the details of these changes into an XML format and posts them using HTTP to the gateway.

## Service request generation

A service request is generated when an alert is passed through the gateway.

When this occurs, the following sequence takes place:

1. An alert is read from the ObjectServer.
2. The alert is mapped into open Service Request fields and used to create a Service Request.

3. The gateway then sends the Service Request fields to Siebel.
4. Siebel creates a Service Request for the alert in the Siebel repository and returns the Service Request Row ID to the gateway.
5. The gateway reads the Service Request Number (SR#) and adds it to the associated alert in the Feedback field of the ObjectServer.

The ObjectServer now holds the alert and Siebel holds its associated SR#. The alert and the Service Request are related by the common SR#. The serial and SR# are both stored in a gateway cache.

## Service request modification

When an alert is changed in the ObjectServer, a service request modification takes place.

When this occurs, the following sequence takes place:

1. An alert is read from the ObjectServer.
2. The alert is mapped into service request fields.
3. The gateway uses the service request fields to create a service request update request and sends the request to Siebel.
4. Siebel receives the service request update request and makes the update.

The change that was made to the alert in the ObjectServer is now reflected in the associated Siebel service request.

## Installing the gateway

---

There are separate procedures for installing the gateway on each version of Tivoli Netcool/OMNIbus.

Follow the procedure for the version of Tivoli Netcool/OMNIbus that your site uses.

### Installing the gateway on Tivoli Netcool/OMNIbus V7.4.0

For Tivoli Netcool/OMNIbus V7.4.0, all gateways are installed using the Tivoli Netcool/OMNIbus installer.

You can install the gateway using any of the following:

- “The installation wizard” on page 5
- A text-based installer (“Console mode” on page 5)
- Settings predefined in a text file (“Silent mode” on page 5)

The installation package and patches for the gateway are supplied as archives. The archive management application that you use to extract the files must be able to preserve the directory structure contained in the archive on extraction.

**Note :** If you are installing a 32-bit gateway on a system that runs a 64-bit UNIX or Linux operating system, you will need to install additional, 32-bit operating system libraries. See the *IBM Tivoli Netcool/OMNIbus Installation and Deployment Guide* for more information.

### Obtaining the installation package

To obtain the installation package and prepare it for installation use the following steps:

1. Download the installation package for the gateway from the Passport Advantage Online Web site:  
[http://www-306.ibm.com/software/howtobuy/passportadvantage/pao\\_customers.htm](http://www-306.ibm.com/software/howtobuy/passportadvantage/pao_customers.htm)
2. Make a backup of any existing configuration files that you want to retain.
3. Extract the contents of the installation package to a temporary directory.

Now use one of the installation methods to install your gateway. In each case, the gateway is installed in the following directory:

`$NCHOME/omnibus/gates`

## The installation wizard

To install the gateway using the installation wizard:

1. Run the installer for your operating system:

```
$NCHOME/omnibus/install/nco_install_integration
```

2. When the installation wizard starts, specify the extracted directory that contains the `README.txt` file as the location of the gateway installation files.
3. Accept the license conditions.

## Console mode

To install the gateway in console mode:

1. Run the installer for your operating system:

```
$NCHOME/omnibus/install/nco_install_integration -i console
```

2. When the text-based installer starts, specify the extracted directory that contains the `README.txt` file as the location of the gateway installation files.
3. Accept the license conditions.

## Silent mode

To install the gateway in silent mode:

1. Create a text file named `reponse.txt` and add the following entries:

```
PROBE_OR_GATE_LOCATION=README_directorypath  
LICENSE_ACCEPTED=true
```

where `README_directorypath` is the path to the directory containing the `README.txt` file in the extracted package.

2. Run the installer for your operating system:

```
$NCHOME/omnibus/install/nco_install_integration -i silent -f  
response_path/response.txt
```

where `response_path` is the full path to the `response.txt` file.

## Setting up a unidirectional Gateway for Siebel Communications

This section describes how to configure IBM Tivoli Netcool/OMNIbus and Siebel to create Siebel Service Requests from IBM Tivoli Netcool/OMNIbus alerts.

When you have completed the steps described in this section, you should have a working unidirectional implementation of the gateway.

To configure the gateway, use the following steps:

1. Configure the Siebel System for the gateway, see [“Siebel system configuration for a unidirectional gateway”](#) on page 6.
2. Set up the gateway configuration, see [“Configuring a unidirectional gateway”](#) on page 16.
3. Create the gateway conversions table, see [“Creating the gateway conversions table”](#) on page 17.
4. Create the gateway configuration file, see [“Gateway configuration mappings”](#) on page 26.
5. Install the gateway, see [“Installing the gateway”](#) on page 4
6. Run the gateway with the following binary: `nco_g_siebel`.

## Siebel system configuration for a unidirectional gateway

---

This section describes how to configure your Siebel system to work with the IBM Tivoli Netcool/OMNIBus Siebel gateway.

Netcool alerts are linked to Siebel Communications Service Requests and Netcool journal entries are linked to Activities associated with these Service Requests.

This section describes the necessary steps to configure Siebel to initiate Workflows based on Netcool alerts.

For a unidirectional gateway, you must make the following configurations:

1. Configure Integration Objects and Business Services using Siebel Tools.
2. Set up Workflows using the WorkFlow Process Designer.
3. Configure the Siebel Web Engine to invoke Business Services within Siebel.
4. Create Siebel Account Details.

### Results

This configuration is described in the following topics:

1. [“Step 1: Configuring integration objects and business services using Siebel Tools” on page 6](#)
2. [“Step 2: Setting up workflows” on page 13](#)
3. [“Step 3: Configuring the Siebel Web Engine” on page 15](#)
4. [“Step 4: Creating a Siebel account details” on page 16](#)

**Note :** The configuration described in the following sections is an example only. You may need to make different settings to suit your environment.

## Step 1: Configuring integration objects and business services using Siebel Tools

You must create integration objects within the Siebel repository to provide the definition of the mapping between the fields in the ObjectServer and Siebel. This is done using Siebel Tools.

The default configuration described in the sections that follow entails creating the following integration objects:

### Results

- Netcool SR Insert
- Netcool SR Get Id
- Netcool SR Modify
- Netcool SR Delete
- Netcool Activity Insert

To make these integration objects available to the gateway, their names must be assigned to attributes (variables) in the gateway configuration file (G\_SIEBEL.conf), as described in [“Gateway configuration mappings” on page 26](#). The integration objects require the following attributes:

- SR\_INSERT\_INT\_OBJ
- SR\_QUERY\_INT\_OBJ
- SR\_MODIFY\_INT\_OBJ
- SR\_DELETE\_INT\_OBJ
- ACTIVITY\_INSERT\_INT\_OBJ

In addition, the following associated attributes must be set in the configuration file with the XML tag values of their corresponding integration objects:

- SR\_INSERT\_XML\_TAG
- SR\_QUERY\_XML\_TAG
- SR\_MODIFY\_XML\_TAG
- SR\_DELETE\_XML\_TAG
- ACTIVITY\_INSERT\_XML\_TAG

Since integration object names and XML tags can be configured, you can specify alternatives if required.

## Creating the Netcool SR Insert integration object

These are the steps required to create the Netcool SR Insert Integration Object.

To create the Netcool SR Insert Integration Object:

1. Log in to Siebel Tools and connect to the Local database.
2. It is recommended that you create a new Netcool project at this stage, although you can simply check-out an existing one. If you create a new project, ensure that you set the Locked flag.
3. Select **File > New Object**.
4. Select the **EAI** tab from the **New Object Wizard** dialog box.
5. Select the **Integration Object** icon.
6. Set the project name to the name of the checked-out project in which you will store the Integration Objects. For example, Netcool.
7. Select **EAI Siebel Wizard** as the source system.
8. Select **Next**.
9. Select **Service Request** as the Source Object for the new integration object.
10. Select **Service Request** as the Source Root for the new integration object.
11. Set the name of the new object to Netcool SR Insert.
12. Select **Next**.

**Note :** By default, Siebel Tools give this integration object an XML tag value of ListofNetcoolSRInsert.

13. Expand the Service Request list by clicking + next to the item. A list of components appears. All of these components are selected.
14. Deselect the Service Request box. This will also deselect all the components in the list.
15. Select the Service Request box. The rest of the list stays deselected.

**Note :** Before continuing with this procedure, make sure only the **Service Request** box is selected. The rest of the list should be deselected.

16. Select **Next**.
17. Select **Finish**.
18. Navigate to the Object Explorer.
19. Select **IntegrationObject :: Netcool SR Insert | Integration Component :: Service Request | Integration Component Field**.

**Note :** This notation means select the **Integration Object** from the Object Explorer (OE), next select the **Netcool SR Insert** record in the Object List Editor (OBLE), then select the **Integration Component** child object type from the OE, then select **Service Request** from the OBLE, then select the **Integration Component Field** child object type from the OE. This notation is used throughout this guide.

**Note :** Items not visible in the Object Explorer can be activated by selecting **Menu > View > OptionsObject Explorer**.

20. Set all fields not required by the integration Component to Inactive.

**Note :** This is most easily achieved by inactivating all fields (as described below) and then re-activating the following fields:

```
Abstract
Account
Created By Name
Id
Opened Date
Row Status
SR Number
Severity
Status
```

To do this:

- a. Select all fields.
- b. Select **Menu > Edit > Change Records**
- c. In the dialog box that appears, for **Change 1**, enter **Inactive** in the **Field** text box and **Y** in the **Value** text box.
- d. Select **OK**.

**Note :** The field list contains the fields required by Siebel and those additional fields required by the default Tivoli Netcool/OMNIBus configuration. You can deviate from the default configuration and amend the field list to suit your environment if required.

Make a note of the XML Tag for each field that is still active. These names are used in the left side of map expressions. If you are using the default configuration file that is provided with the gateway, these fields are already set within the SIEBEL\_OPEN\_MAP section.

21. Select **IntegrationObject :: Netcool SR Insert | Integration Component :: Service Request | Integration Component Key**.
22. Create a new User Key record with the following field-value pairs:

```
Name           User Key:3
Key Sequence Number 3
Key Type       User Key
```

23. Select **IntegrationObject :: Netcool SR Insert | Integration Component :: Service Request | Integration Component Key :: User Key:3 | Integration Component Key Field**
24. Create a new User Key Field record with the following field-value pairs:

```
Name           Abstract
Field Name     Abstract
```

**Note :** The abstract field in Siebel must be unique for each entry.

## Creating the Netcool SR Get Id integration object

This section describes the list involved in creating the Netcool SR Get Id Integration Object.

To create the Netcool SR Get Id Integration Object:

1. Log in to Siebel Tools and connect to the Local database.
2. Check out the project that you are using to store the Integration Objects, if necessary.
3. Select **FileNew Object**.
4. Select the **EAI** tab from the **New Object Wizard** dialog box.
5. Select the **Integration Object** icon.
6. Set the project name to the name of the checked-out project in which you will store the integration objects. For example, Netcool.
7. Select **EAI Siebel Wizard** as the source system.
8. Select **Next**.

9. Select **Service Request** as the Source Object for the new integration object.
10. Select **Service Request** as the Source Root for the new integration object.
11. Set the name of the new object to Netcool SR Get Id.
12. Select **Next**.
 

**Note :** By default, Siebel Tools give this integration object an XML tag value of ListOfNetcoolSRGetID.
13. Expand the **Service Request** list by clicking + next to the item. A list of components appears. All of these components are selected.
14. Deselect the **Service Request** box. This will also deselect all the components in the list.
15. Select the **Service Request** box. The rest of the list stays deselected.
 

**Note :** Before continuing with this procedure, make sure only the Service Request box is selected. The rest of the list should be deselected.
16. Select **Next**.
17. Select **Finish**.

## Creating the Netcool SR Modify and Netcool SR Delete integration objects

This section describes the list involved in creating the Netcool SR Modify and Netcool SR Delete Integration Objects.

To create the Netcool SR Modify and Netcool SR Delete Integration Objects:

1. Log in to Siebel Tools and connect to the Local database.
2. Check out the project that you are using to store the Integration Objects, if necessary.
3. Navigate to the Object Explorer.
4. Select **Integration > Object :: Netcool SR Insert**.
5. Right-click the mouse and select **Copy Record** from the popup menu.
6. Set the name of the copy to the name of the new integration object that you are creating (either Netcool SR Modify or Netcool SR Delete) and set the name of the XML Tag to a unique value.

**Note :**

For Netcool SR Modify, use the following field-value pairs:

**Note :**

Name	Netcool SR Modify
XML Tag	ListOfNetcoolSrModify

**Note :**

For Netcool SR Delete, use the following field-value pairs:

**Note :**

Name	Netcool SR Delete
XML Tag	ListOfNetcoolSrDelete

**Note :**

Make a note of both sets of values.

7. Select **IntegrationObject :: *integration\_component* | Integration Component :: Service Request | Integration Component Field**, where *integration\_component* is either **Netcool SR Modify** or **Netcool SR Delete**.
8. Set all fields not required by the Integration Component to Inactive.

9. If you are using the default configuration file that is provided with the gateway, the following fields must be activated:

**Note :**

For the Netcool SR Modify integration object:

**Note :**

```
Abstract
Id
Opened Date
Row Status
SR Number
Severity
Status
```

**Note :**

For the Netcool SR Delete integration object:

**Note :**

```
Id
Opened Date
Row Status
SR Number
Status
```

10. Select **IntegrationObject :: *integration\_component* | Integration Component :: Service Request | Integration Component Key**, where *integration\_component* is either **Netcool SR Modify** or **Netcool SR Delete**.

11. Create a new User Key record with the following field-value pairs:

**Note :**

```
Name                User Key:3
Key Sequence Number 3
Key Type            User Key
```

12. Select **IntegrationObject :: *integration\_component* | Integration Component :: Service Request | Integration Component Key::User:3|Integration Component Key Field**, where *integration\_component* is either **Netcool SR Modify** or **Netcool SR Delete**.

13. Delete the Integration Component Key Field Abstract.

14. Create a new User Key Field record with the following field-value pairs:

**Note :**

```
Name                SR Number
Field Name          SR Number
```

## Creating the Netcool Activity Insert integration object

These are the lis involved in creating the Netcool Activity Insert Integration Object.

To create the Netcool Activity Insert Integration Object:

1. Log in to Siebel Tools and connect to the Local database.
2. Check out the project that you are using to store the Integration Objects.
3. Select **File > New Object**.
4. Select the **EAI** tab from the **New Objects Wizard** dialog box.
5. Select the **Integration Object** icon.
6. Set the project name to the name of the checked-out project in which you store the Integration Objects. For example, Netcool.
7. Select **EAI Siebel Wizard** as the source system.



8. Select **Next**.
9. Select **Action** as the Source Object for the new integration object.
10. Select **Action** as the Source Root for the new integration object.
11. Set the name of the new object to Netcool Activity Insert.
12. Select **Next**.

**Note :** By default, Siebel Tools give this integration object an XML tag value of ListofNetcoolSRActivityInsert.

13. Expand the **Action** list by clicking + next to the item. A list of components appears. All of these components are selected.
14. Deselect the **Action** box. This deselects all the components in the list.
15. Select the **Action** box. The rest of the list stays deselected.

**Note :** Before continuing with this procedure, make sure only the **Action** box is selected. The rest of the list should be deselected.

16. Select **Next**.
17. Select **Finish**.
18. Navigate to the Object Explorer.
19. Select **IntegrationObject :: Netcool Activity Insert | Integration Component :: Action | Integration Component Field**.
20. Set all fields not required by the Integration Component to Inactive.
21. If you are using the default configuration file that is provided with the gateway, the following fields will be activated:

Activity SR Id
Description

**Note :** Make a note of the XML Tag for each field that is still active. If you are using the default configuration file that comes with the gateway, these fields are already set within the SIEBEL\_JOURNAL\_MAP section.

22. Select **IntegrationObject :: Netcool Activity Insert | Integration Component :: Action | Integration Component Key**.
23. Create a new user key record with the following field-value pairs:

Name	User Key:9
Key Sequence Number	9
Key Type	User Key

24. Select **IntegrationObject :: Netcool Activity Insert | Integration Component :: Request | Integration Component Key :: User Key:9 | Integration Component Key Field**
25. Create new user key field records with the following field-value pairs:

Name	SR Number
Field Name	SR Number
Name	Description
Field Name	Description

26. Set the status of all other user keys to Inactive.

**Note :**

Check in the project within which you created the integration objects as follows lis:

**Note :** The project will be compiled into a new Siebel repository file (SRF file) after creating new business services as described in [“Creating business services using Siebel Tools” on page 12.](#)

## Checking in the integration objects

Perform these lis to check in the integration objects.

After creating the Netcool SR Insert, Netcool SR Get Id, Netcool SR Modify, Netcool SR Delete, and Netcool Activity Insert integration objects, you must check in the project using the following lis:

1. Select **Tools > Check In**.
2. Select your Netcool project from the project list (and not the five new Integration Objects just created, otherwise the Check In button is disabled).
3. Select **Check In**.
4. Then check in the five new Integration Objects.

## Creating business services using Siebel Tools

This section describes the lis involved in creating business services using Siebel Tools.

You need to add the following Business Services for the Service Requests and Activities (journal entries) for Siebel Workflow:

- EAI Netcool Insert SR (http)
- EAI Netcool Get Service Request Id (http)
- EAI Netcool Insert Activity (http)

These business services are based on the existing Workflow Process Manager within Siebel. To add these business services, use the following lis:

1. Log in to Siebel Tools and connect to the Local database.
2. Check out the project EAI Business Services as follows: Select **Tools > Check Out** Select **Tools > Check Out** Select **EAI Business Services** Select **Check Out**.
3. Navigate to the Object Explorer.
4. Select **Business Service**.
5. To create the EAI Netcool Insert SR (http) business service, copy the Workflow Process Manager business service, and specify the following field-value pairs:

### Note :

Name	EAI Netcool Insert SR (http)
Project	EAI Business Services
Class	CSSwfEngine
Display Name	Business Integration Manager

6. Select **Business Service::EAI Netcool Insert SR (http)|Business Service User Prop**.
7. Create a new Business Service User Prop record with the following field-value pairs:

### Note :

Name	ProcessName
Value	EAI Insert Service Request (http)

8. To create the EAI Netcool Get Service Request Id (http) business service, copy the Workflow Process Manager business service, and specify the following field-value pairs:

### Note :

Name	EAI Netcool Get Service Request Id (http)
Project	EAI Business Services
Class	CSSwfEngine
Display Name	Business Integration Manager

9. Select **Business Service::EAI Netcool Get Service Request Id (http)|Business Service User Prop**.
10. Create a new Business Service User Prop record with the following field-value pairs:

**Note :**

Name	ProcessName
Value	EAI Get Service Request Id (http)

11. To create the EAI Netcool Insert Activity (http) business service, copy the Workflow Process Manager business service and specify the following field-value pairs:

**Note :**

Name	EAI Netcool Insert Activity (http)
Project	EAI Business Services
Class	CSSwfEngine
Display Name	Business Integration Manager

12. Select **Business Service::EAI Netcool Insert Activity (http)|Business Service User Prop.**
13. Create a new business service user prop record with the following field-value pairs:

**Note :**

Name	ProcessName
Value	EAI Insert Activity (http)

14. Check in the project within which you created the integration objects as follows: Select **Tools > Check In** Select **EAI Business Services** from the project list. Select **Check In**.

## Compiling the integration objects and business services

Before the integration objects and business services can be used, they must be compiled.

To do this:

1. Log in to Siebel Tools and connect to the Local database.
2. Select **Tools > Compile Projects**.
3. Select **All projects** to be compiled.
4. Select a temporary Siebel Repository file; for example, `c:\sea\clients\objects\siebel_sia.srf` to save the compiled changes.
5. Select **Compile**.

## Activating the configuration

This section describes the lis involved in activating the configuration.

To activate the configuration created in [“Compiling the integration objects and business services”](#) on page 13, use the following lis:

1. Stop the Siebel server.
2. Copy the `siebel_sia.srf` to the `Siebsrvr\OBJECTS\ENU` directory.
3. Restart the Siebel server.

**Note :** Make a copy of the existing `siebel_sia.srf` file before replacing it with the new file.

## Step 2: Setting up workflows

This section describes how to set up work flows for the Gateway for Siebel Communications.

The Gateway for Siebel Communications posts XML-formatted Netcool alerts to the Siebel system using the HTTP protocol. When received by the Siebel Web Engine, this invokes a Business Service, which runs a predefined Workflow Process within Siebel. You must create and then import the Workflows that can receive and act on the messages received from Tivoli Netcool/OMNIBus.

## Creating a new workflow group

These are the lis involved in creating a new workflow group.

To create a group:

1. Log in to Siebel Communications and connect to the Server database.
2. Select **Site Map > Administration – Data > List of Values**
3. Create a new record with the following field-value pairs:

### Note :

Type:	WF_GROUP_CD
Display Value:	WF Group
Translate:	Yes
Language Independent Code:	WF Group
Order:	1
Active:	Yes
Language Name:	English-American

## Importing workflows

This section explains the steps involved in importing the default workflows provided with the gateway.

To import the default workflows provided with the gateway:

1. Copy the following default workflows from the \$OMNIHOME/gates/siebel directory to the machine where Siebel is running:

```
EAIInsertActivity.xml
EAIGetServiceRequestId.xml
EAIInsertServiceRequest.xml
```

2. Log in to Siebel Tools and connect to the Local database.
3. Check out the **EAI Business Services** project.
4. Navigate to the Object Explorer (OE).
5. Select **Workflow Process**.
6. Right-click within the Object List Editor (OBLE) and select **Import Workflow Process** from the pop-up menu.
7. Locate and select EAIInsertActivity.xml from the **File Open** dialog box and select **OK**.
8. Select the **EAI Business Services** project from the pop-up list. Once imported, the workflow has a status of In Progress.
9. Select **Deploy** in the OBLE. The workflow status changes to Completed.
10. Repeat lis 6-9 for the EAIGetServiceRequestId.xml and EAIInsertServiceRequest.xml workflow processes.
11. Check in the **EAI Business Services** project. There is no need to compile a new SRF.
12. Activate the new workflow processes within Siebel as follows:
  - a. Log in to Siebel Communications and connect to the Server database.
  - b. Navigate to **Site Map > Administration – Business Process > Workflow Deployment**
  - c. Navigate to the **Repository Workflow Processes** applet.
  - d. Query for the new process (EAI Get Service Request Id (http), EAI Insert Service Request (http), and EAI Insert Activity (http)).
  - e. Select each process in turn and select **Activate**

**Note :** Once completed, the new processes will be visible within the **Active Workflow Processes** applet within the same view. Query for them if necessary, and check that they all have a deployment status value of Active.

## Step 3: Configuring the Siebel Web Engine

This section describes the changes that you must make to the `EAI.cfg` and `EAPPS.cfg` files to use the new workflow processes.

### Configuring the `EAI.cfg` file

Use these `lis` to configure the `EAI.cfg` file.

To update the `EAI.cfg` file in the `BIN\ENU` directory, use the following `lis`:

1. Stop the Siebel server.
2. Add the following entries to the `[HTTP Services]` section to the `EAI.cfg` file:

**Note :**

```
wf1 = Workflow1
wf2 = Workflow2
wf3 = Workflow3
```

3. Add the following sections to the `EAI.cfg` file:

**Note :**

```
[Workflow1]
Mode = Document
Service = EAI Netcool Insert SR (http)
Method = RunProcess

[Workflow2]
Mode = Document
Service = EAI Netcool Get Service Request Id (http)
Method = RunProcess

[Workflow3]
Mode = Document
Service = EAI Netcool Insert Activity (http)
Method = RunProcess
```

4. Restart the Siebel server.

### Configuring the `EAPPS.cfg` file

Use these `lis` to configure the `EAPPS.cfg` file.

To update the `EAPPS.cfg` file in the `SWEapp\BIN` directory, use the following `lis`:

1. Stop the Siebel service.
2. Check that the `EAPPS.cfg` file in the `SWEapp\BIN` directory contains the following section:

**Note :**

```
[/eai]
ConnectString =
siebel.TCPIP.none.none://host:port/server/EAIObjMgr_enu
EnableExtServiceOnly = TRUE
```

**Note :**

Where

- *host* is the name of the Siebel server host
- *port* is the SC Broker Component port of the server (the default is 2321)
- *server* is the Siebel Enterprise name

3. Add this section if necessary.
4. Restart the web server.
5. Restart the Siebel server.

## Step 4: Creating a Siebel account details

This section describes the references involved in creating Siebel account details.

This sections describe the following references that you need to perform:

- Creating a gateway contact.
- Creating a customer account.
- Making the gateway contact the prime contact for the account.

To create a gateway contact, use the following lis:

1. Log in to Siebel Communications and connect to the Server database.
2. Navigate to **Site Map > Contacts > Contacts List**.
3. Select **All Contacts** from the visibility filter.
4. Create a new record with the following field-value pairs:

**Note :**

First Name	Gateway
Last Name	Writer

To create a customer account, use the following lis:

1. Log in to Siebel Communications and connect to the Server database.
2. Navigate to **Site Map > Accounts > Accounts List**.
3. Select **All Accounts** from the visibility filter.
4. Create a new record with the following field-value pairs:

Name	Netcool Account
Account Class	Customer

To make the gateway contact the primary contact for the account, use the following lis:

1. Select **Netcool Account** in the **Accounts List** applet.
2. Navigate to the **Contacts List** applet.
3. Select Add on the **Contacts List** applet.
4. Select **Gateway Writer** from the contacts list.

## Configuring a unidirectional gateway

This section describes how to configure a unidirectional gateway.

You must set the following properties in the G\_SIEBEL . conf gateway configuration file:

1. Set SWE\_HOST to the IP address or hostname where the HTTP server and the Siebel Web Engine reside.
2. Set SWE\_PORT to the port which the HTTP server is listening. The default is port 80.
3. Set the USERNAME and PASSWORD to a valid Siebel user name and password.
4. Set the EAI\_PAGE to http://hostname:/eai\_enu/start.swe. This tells the Microsoft IIS to execute the Siebel Web Engine.
5. Set SR\_GET\_ID\_WF to the alias that has been created for the Business Service that will retrieve service request SR numbers from Siebel (wf2).
6. Set ACTIVITY\_UPSERT\_WF to the alias that has been created for the Business Service that will insert Activities into Siebel (wf3).

## Creating the gateway conversions table

The conversion table facility enables you to specify data conversions to take place between particular Siebel and ObjectServer fields.

For example, Siebel users require a status table field to be alphabetic and to have a particular value. IBM Tivoli Netcool/OMNIBus may hold these as numeric values. All conversions must be supplied before the gateway can be operational.

**Note :** The ObjectServer SQL definition file must be updated to include the conversions database. For details, see [“Setting up a bidirectional Gateway for Siebel Communications”](#) on page 18.

### Manually editing the conversion table

The conversion table is ObjectServer SQL-based. This means you can make full use of the ObjectServer SQL.

The conversion expects information to be supplied in column format, enclosed in parentheses, where the columns indicate:

```
column 1 = key field name
column 2 = column name
column 3 = ObjectServer value
column 4 = conversion value
```

In the following example, the numeric values of the ObjectServer Severity field (0, 1, 2, 3, 4, and 5) are converted to one of the ServiceCenter textual values (4-Low, 3-Medium, 2-High, or 1-Critical). User name values are also converted. This can be entered by using the SQL input script, nco\_sql.

```
use database conversions;

insert into siebel values ('Severity0','Severity','0','4-Low');
insert into siebel values ('Severity1','Severity','1','4-Low');
insert into siebel values ('Severity2','Severity','2','4-Low');
insert into siebel values ('Severity3','Severity','3','3-Medium');
insert into siebel values ('Severity4','Severity','4','2-High');
insert into siebel values ('Severity5','Severity','5','1-Critical');

go;
```

### Changes to SQL definition file

On Tivoli Netcool/OMNIBus v7.x, you can make these changes using the SQL option in the Tivoli Netcool/OMNIBus Administrator (nco\_config):

```
create database conversions;
Use database conversions;
go

create table conversions.siebel persistent (
  KeyField varchar(255) primary key,
  Colname varchar(255),
  OSValue varchar(255),
  Conversion varchar(255)
);

go

insert into conversions.siebel values ('Severity0','Severity','0','4-Low');
insert into conversions.siebel values ('Severity1','Severity','1','4-Low');
insert into conversions.siebel values ('Severity2','Severity','2','4-Low');
insert into conversions.siebel values ('Severity3','Severity','3','3-Medium');
insert into conversions.siebel values ('Severity4','Severity','4','2-High');
insert into conversions.siebel values ('Severity5','Severity','5','1-Critical');
```

**Note :** You must restart the ObjectServer to apply any SQL changes you make.

## Setting up a bidirectional Gateway for Siebel Communications

---

This section describes how to configure Gateway for Siebel Communications and Siebel to create Siebel Service Requests from IBM Tivoli Netcool/OMNIbus alerts.

When you have completed the configuration described in this section, you will have a working bidirectional implementation of the gateway.

To configure the gateway:

1. Configure the Siebel System for the gateway, as described in [“Configuring a unidirectional gateway” on page 16](#).
2. Create the gateway configuration file, as described in [“Gateway configuration mappings” on page 26](#).
3. Install the gateway, as described in [“Installing the gateway” on page 4](#).
4. Run the gateway with the following command:

**Note :**

```
nco_g_siebel
```

## Siebel system configuration for a bidirectional gateway

---

This section describes how to configure your Siebel system to work with the bidirectional Gateway for Siebel Communications.

Netcool alerts are linked to Siebel Communications Service Requests and Netcool journal entries are linked to Activities associated with these Service Requests. This section describes the necessary list to configure Siebel to initiate Workflows based on Netcool alerts.

A unidirectional gateway forms the basis for a bidirectional gateway. For bidirectional functionality, you must configure a unidirectional gateway and then create additional Integration Objects, Workflow Processes, Polices, and Actions within Siebel. For details about creating a unidirectional gateway, see [“Setting up a unidirectional Gateway for Siebel Communications” on page 5](#).

## Creating integration objects using Siebel tools

Use the default integration objects to create new integration objects.

The section contains the following topics:

- [“Using the default integration objects” on page 18](#)
- [“Creating new integration objects” on page 19](#)

### Using the default integration objects

You need to add the following Integration Objects to the Siebel repository using Siebel tools:

- Modified Service Request
- Deleted Service Request
- Inserted Activity

The default bidirectional gateway is configured to use some of the same Integration Objects that were set up for the unidirectional gateway. The default unidirectional Integration Objects used for the Bidirectional gateway are:

- MODIFIED\_SR\_INT\_OBJ - Netcool SR Modify
- DELETED\_SR\_INT\_OBJ - Netcool SR Delete
- INSERTED\_JRNL\_INT\_OBJ - Netcool Activity Insert



## Creating new integration objects

If you create your own integration objects for the bidirectional flow of data, you must set these values in the gateway configuration file. The names of these integration objects are used by the gateway to determine which operation needs to be carried out by the gateway. As a result you must use a different integration object for each of the three bidirectional functions.

Once the new integration objects have been compiled and activated, then the Action SQL files for the gateway must be configured to reflect the changes. See [“Configuring action SQL”](#) on page 25.

## Setting up workflows

The Gateway for Siebel Communications sends Netcool alerts to the Siebel system in XML documents that are posted using the HTTP protocol.

When the XML files are received by the Siebel Web Engine, this initiates the appropriate Workflow. You must create the Workflows that can receive and act on the messages received from Tivoli Netcool/OMNIBus.

## Importing bidirectional workflows

To import the default workflows provided with the gateway:

1. Copy the following default Workflows from the `$OMNIHOME/gates/siebel` directory to the machine where Siebel is running:

```
ExportActivity.xml
ExportSRDelete.xml
ExportSRModification.xml
```

**Note :** Before importing these workflows, you must edit them to replace the strings `__GW_HOST__` and `__GW_PORT__` with values you assign to the `GW_HOST` and `GW_PORT` attributes in the gateway configuration file (`G_SIEBEL.conf`). If you need to change the `GW_HOST` and `GW_PORT` attribute values at a later date, see [“Frequently asked questions”](#) on page 33.

2. Log in to Siebel Tools and connect to the Local database.
3. Check out the **EAI Business Services** project.
4. Navigate to the Object Explorer (OE).
5. Select **Workflow Process**.
6. Right-click within the Object List Editor (OBLE) and select **Import Workflow Process** from the pop-up menu.
7. Locate and select `ExportActivity.xml` from the **File Open** dialog box and select **OK**.
8. Select the **EAI Business Services** project from the pop-up list.
9. Select **Deploy** in the OBLE.
10. Repeat steps 6-9 for the `ExportSRDelete.xml` and `ExportSRModification.xml` workflow processes.
11. Check in the **EAI Business Services** project.
12. Activate the new workflow processes within Siebel as follows:
  - a. Log in to Siebel Communications and connect to the Server database.
  - b. Navigate to **Site Map > Administration – Business Process > Workflow Deployment**
  - c. Navigate to the **Repository Workflow Processes** applet.
  - d. Select each process in turn and select **Activate**.

**Note :** Once completed, the new processes will be visible within the **Active Workflow Processes** applet within the same view. Query for them if necessary, and check that they all have a deployment status value of `Active`.

If you need to change the GW\_HOST and/or GW\_PORT values later, refer to [“Frequently asked questions”](#) on page 33.

## Setting up the Siebel workflow policies and actions

Workflow policies are a set of conditions that, when violated, triggers an associated workflow action.

The workflow action initiates a workflow process, that sends the modified data to the gateway reader module. The following policies and actions call the default workflow processes that come with the gateway.

### Defining a workflow policy group for the gateway

Create a new workflow policy group to that govern the work flow of the messages received from the Netcool/OMNIbus.

To define a workflow policy group for the gateway:

1. Start Siebel Communications and connect to the Server database.
2. Navigate to **Site Map > Administration – Business Process > Workflow Policy Groups**.
3. Navigate to the **Policy Groups List** applet.
4. Select the **New** button to create a new group with the following field-value pair:

**Note :**

Name	Netcool Group
------	---------------

### Defining a workflow action for modified service requests

When a service request is modified, the work flow action initiates a work flow process which is defined by default.

To define a workflow action for a modified service request:

1. Navigate to **Site Map > Administration – Business Process > Workflow Policy Actions**.
2. Navigate to the **Actions** applet.
3. Select the **New** button to create a new action with the following field-value pairs:

**Note :**

Name	Netcool SR Update
Program	Run Workflow Process
Workflow Object	Service Request

4. Navigate to the **Arguments** applet.
5. Select the **New** button to create a new argument with the following field-value pairs:

**Note :**

Argument Value	ProcessName Export SR Modification (http)
----------------	--

6. Select **Workflow Policies** from the link bar.
7. Navigate to the **Policies List** applet.
8. Select the **New** button to create a new policy with the following field-value pairs:

**Note :**

Name	Netcool SR Update
Workflow Object	Service Request
Policy Group	Netcool Group
Activation	Today's date
Duration	1 minute

9. Navigate to the **Conditions** applet.
10. Select the **New** button to create a new condition with the following field-value pairs:

**Note :**

Condition Field Operation	Service Request Severity IS UPDATED
---------------------------	-------------------------------------

11. Navigate to the **Actions** applet.
12. Select the **New** button to create a new action with the following field-value pairs:

**Note :**

Action Sequence	Netcool SR Update 1
-----------------	---------------------

13. Navigate to the **Arguments** applet.
14. Check that a record has been created with the following field-value pairs:

**Note :**

Argument Value	ProcessName Export SR Modification (http)
----------------	---

## Defining a workflow action for a deleted service request

The work flow action initiates a predefined work flow process when a service request is deleted.

To define a workflow action for a deleted service request:

1. Navigate to **Site Map > Administration – Business Process > Workflow Policy Actions**.
2. Navigate to the **Actions** applet.
3. Select the **New** button to create a new action with the following field-value pairs:

**Note :**

Name	Netcool SR Delete
Program	Run Workflow Process
Workflow Object	Service Request

4. Navigate to the **Arguments** applet.
5. Select the **New** button to create a new argument with the following field-value pairs:

**Note :**

Argument Value	ProcessName Export SR Delete (http)
----------------	-------------------------------------

6. Select **Workflow Policies** from the link bar.
7. Navigate to the **Policies List** applet.
8. Select the **New** button to create a new policy with the following field-value pairs:

**Note :**

Name	Netcool SR Delete
Workflow Object	Service Request
Policy Group	Netcool Group
Activation	Today's date
Duration	1 minute

9. Navigate to the **Conditions** applet.
10. Select the **New** button to create a new condition with the following field-value pairs:

**Note :**

Condition Field	Service Request Status
Operation	=
Value	Closed

11. Navigate to the **Actions** applet.
12. Select the **New** button to create a new action with the following field-value pairs:

**Note :**

Action Sequence	Netcool SR Delete 1
-----------------	------------------------

13. Navigate to the **Arguments** applet.
14. Check that a record has been created with the following values:

**Note :**

Argument Value	ProcessName Export SR Delete (http)
----------------	--

## Defining a workflow action for a new activity

The work flow action which initiates a work flow process for every new activity. This work flow action for this new activity can be defined.

To define a workflow action for a new activity:

1. Navigate to **Site Map Administration – Business ProcessWorkflow Policy Actions**.
2. Navigate to the **Actions** applet.
3. Select the **New** button to create a new action with the following field-value pairs:

**Note :**

Name	Netcool Activity Create
Program	Run Workflow Process
Workflow Object	Activity

4. Navigate to the **Arguments** applet.
5. Select the **New** button to create a new argument with the following field-value pairs:

**Note :**

Argument Value	ProcessName Export Activity (http)
----------------	---------------------------------------

6. Select **Workflow Policies** from the link bar.
7. Navigate to the **Policies List** applet.
8. Select the **New** button to create a new policy with the following field-value pairs:

**Note :**

Name	Netcool Activity Create
Workflow Object	Activity
Policy Group	Netcool Group
Activation	Today's date
Duration	1 minute

9. Navigate to the **Conditions** applet.
10. Select the **New** button to create a new condition with the following field-value pairs:

**Note :**

Condition Field	Activity Status
Operation	=
Value	In Progress

11. Navigate to the **Actions** applet.
12. Select the **New** button to create a new action with the following field-value pairs:

**Note :**

Action Sequence	Netcool Activity Create 1
-----------------	------------------------------

13. Navigate to the **Arguments** applet.
14. Check that a record has been created with the following field-value pairs:

**Note :**

Argument Value	ProcessName Export Activity (http)
----------------	---------------------------------------

## Generating triggers

These configuration lists create triggers in the Siebel database to support the workflow policies just created and starts the Workflow Monitor Agent reference.

To generate triggers:

1. Log in to Siebel Communications and connect to the Server database.
2. Create a **Generate Triggers** component request.
3. Navigate to **Site Map > Administration - Server Management > Jobs**.
4. In the Jobs list, select the **New** button.
5. In the Component/Job field, select **Generate Triggers** from the drop-down list.
6. Navigate to the **Job Parameters** applet.
7. Create the following five parameters for the component request, selecting the **New** applet button for each of the following field-value pairs:

**Note :**

Name Value	EXEC true
Name Value	Table Owner dbo
Name Value	Table Owner Password SIEBEL
Name Value	Privileged User SIEBEL
Name Value	Privileged User Password SIEBEL

8. Submit the job by selecting the **Start** button in the Jobs list.
9. Refresh the view until the job is completed, with a Status value of Success.

## Creating a new workflow monitor agent component

These are the lists involved in creating a new workflow monitor agent component.

To create a new workflow monitor agent component:

1. Navigate to **Site Map > Administration – Server Configuration > Enterprises > Component Definitions**.
2. In the Component Definitions list, copy the Workflow Monitor Agent component and enter (or confirm) the following field-name values in the new record:

Component	Netcool WF Monitor Agent
Alias	Netcool WorkMon
Component Type	Workflow Monitor Agent
Component Group	Workflow Management
Run Mode	Background

3. Select the **Menu** applet and select **Save Record**.
4. Refresh the **Component Definitions** list and verify that the new component has a state of **Creating**.
5. Define parameters for the new component definition by confirming or editing values in the **Component Parameters** list as follows:

Action Interval	1
Auto Restart	true
Default References	1
Group Name	Netcool Group
Sleep Time	30
Use Action Agent	false

**Note :** Some of these parameters may not be visible in the parameter list. For information about setting these values, see [“Frequently asked questions” on page 33](#).

6. Enable the new component definition as follows:
  - a. In the Component Definitions list, click the Enable applet button.
  - b. Verify that the state changes from **Creating** to **Active**. Click the **Synchronize** applet button.
7. Log out of the Siebel client.
8. Stop the Siebel Server service
9. Stop the Siebel Gateway Name Server service.
10. Restart the Siebel Gateway Name Server service.
11. Restart the Siebel Server service.
12. Log back into Siebel Communications and verify that the Netcool WF Monitor Agent is running as follows:
  - a. Navigate to **Site Map > Administration – Server Management > Components**
  - b. Query for the Netcool WF Monitor Agent component in the **Components** list and check that its state is **Running**.
  - c. Navigate to **Site Map > Administration – Server Management > References**
  - d. Query for the Netcool WF Monitor Agent component in the **References** list and check that its reference state is **Running**.
  - e. The **Status** field should show **Sleeping for x seconds...** where *x* changes cyclically from 30, to 20, to 10 when the view is refreshed.

## Configuring a bidirectional gateway

This section explains how to configure a bidirectional gateway.

The following properties must be set up in the `G_SIEBEL.conf` gateway configuration file, in addition to those added for the unidirectional gateway:

1. Set `GW_HOST` to the host name of the machine on which the gateway reader module is running. This is the same machine that is set as the URL Template Value (see [“Setting up workflows” on page 19](#)).
2. Set `GW_PORT` to the port on which the gateway reader module listens for data from Siebel. This will be the same machine that is set as the URL Template Value (see [“Setting up workflows” on page 19](#)).
3. Set `HTTP_TIMEOUT` to the timeout value being used by the Gateway reader module.

## Configuring action SQL

These are the steps involved in configuring the action SQL files.

When a modification is made to a Siebel case, the changes are reflected in the ObjectServer. To determine what actions take place in the ObjectServer, you need to edit the action SQL files. These files are ObjectServer SQL-based, therefore you have all of the language facilities available to you. You must create an action SQL file for each of the event types.

The following SQL statement updates the Severity field:

```
update alerts.status set Severity = $(siebelfield:Severity);
```

In this example, the \$ represents an expansion statement and information enclosed within the parentheses must be resolved. The siebelfield statement specifies the Siebel field. The Severity statement specifies that severity conversions must be used and applied. The colon (:) is a statement separator.

## Cache storage

The gateway makes extensive use of certain data fields that are kept in cache storage.

Cache storage ensures the unique identification of IBM Tivoli Netcool/OMNIbus alerts at all times.

The cache fields that are always available are:

```
Serial  
ServerSerial  
ServerName
```

## Close action SQL file

The close action SQL file determines what action should be taken in the ObjectServer when a Siebel case is closed.

The following example performs an ObjectServer delete:

```
# Example Siebel Action SQL for the IBM Tivoli Netcool/OMNIbus Gateway Server  
#  
# Siebel Problem Close (Deletion) Action SQL.  
#  
delete from alerts.status where Serial = $(Serial);
```

## Update action SQL file

The update action SQL file determines what action should be taken in the ObjectServer when a Siebel case is updated.

The following example updates the ObjectServer and sets a data field. The Agent field is resolved to Siebel field number 2:

```
# Example Siebel Action SQL for the IBM Tivoli Netcool/OMNIbus Gateway Server  
#  
# Siebel Problem Update Action SQL.  
#  
update alerts.status set Agent = '$(2)',  
update alerts.status set Severity = $(11:Severity) where Serial = $(Serial);
```

## Journal action SQL file

The journal action SQL file determines what action should be taken in the ObjectServer when a journal is amended.

The following is an example of a journal action SQL file.

```
# Example Siebel Action SQL for the IBM Tivoli Netcool/OMNIbus Gateway Server
#
# Siebel Problem Journal Action SQL.
#
insert into alerts.journal values ('$(5):0:$(1)',$(Serial),0,
$(1), '$(3)',',',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ');
```

## Gateway configuration mappings

This section describes the gateway configuration file that maps ObjectServer fields to Siebel server destination field names.

### Mapping syntax

Mappings in the Gateway for Siebel Communications are described below.

Mappings in the Gateway for Siebel Communications configuration file must use the following syntax:

```
CREATE MAPPING mappingname
(
  'SiebelFieldXMLTag' = '@fieldname' .....
) ;
```

Where:

- *mappingname* is the name of the map configuration to be created
- *@fieldname* is the name of a field in the ObjectServer alerts.status table



**Attention :** Make sure that the port number you specify for the GW\_PORT attribute in the G\_SIEBEL.conf configuration file is different from the port specified for the gateway using nco\_xigen. If the port numbers specified are the same, the gateway will lose its connection to Siebel immediately after a journal entry is made.

### Example configuration file

This is an example of a Siebel configuration file.

The following shows a complete G\_SIEBEL.conf Siebel configuration file.

```
Example Siebel configuration for the Netcool/OMNIbus Gateway Server 3.4
# Start up the reader - connect to the Object Server NCOMS
#
START READER SERVER_READER CONNECT TO NCOMS;
#
# Create a mapping from the Object Server's fields into Siebel's fields
#
CREATE MAPPING SIEBEL_OPEN_MAP
(
  'Abstract'= '@ServerName + @Serial ',
  'Account'= 'Netcool + Account',
  'CreatedByName'= 'SADMIN',
  'Severity'= '@Severity' CONVERT TO STRING,
  'Status'= 'Open'
);
CREATE MAPPING SIEBEL_UPDATE_MAP
(
  'Abstract' = '@ServerName + @Serial + : + @Summary',
  'SRNumber' = PROBLEM_NUMBER,
  'Severity' = '@Severity' CONVERT TO STRING
);
CREATE MAPPING SIEBEL_CLOSE_MAP
```



```

(
'SRNumber'= PROBLEM_NUMBER,
'Status'= 'Closed'
);
CREATE MAPPING SIEBEL_JOURNAL_MAP
(
'SRNumber' = PROBLEM_NUMBER,
'Description'= JOURNAL_TEXT
);
#
# Start up the writer
#
START WRITER SIEBEL_WRITE
(
TYPE = SIEBEL,
DATE_FORMAT = '%m/%d/%Y %H:%M:%S',
#
# These maps have been defined above
#
OPEN_MAP = SIEBEL_OPEN_MAP,
UPDATE_MAP = SIEBEL_UPDATE_MAP,
CLOSE_MAP = SIEBEL_CLOSE_MAP,
JOURNAL_MAP = SIEBEL_JOURNAL_MAP,
#
# Information for sending events to Siebel. The SIEBEL_VERSION
# is the version of siebel being connected to. Currently supported
# are 6 and 7.
# The SWE properties refer to the location of the Siebel Web Engine.
# The USERNAME and PASSWORD are the Siebel Username and Password.
#
SIEBEL_VERSION= 7,
SWE_HOST= 'tiho',
SWE_PORT= '80',
USERNAME = 'MMUSE',
PASSWORD = 'MMUSE',
EAI_PAGE= 'http://tiho/eai_enu/start.swe',
KEY_FIELD= 'Abstract',
THREADPOOL_SIZE = 10,
# Workflows set up within Siebel to import the data. These
# reference the workflow outlines in the EAI.cfg and
# EAPPS.cfg files.
#
SR_UPSERT_WF= 'wf1',
SR_GET_ID_WF= 'wf2',
ACTIVITY_UPSERT_WF= 'wf3',
#
# Integration Object definitions and corresponding XML
# Tags for updating Siebel from Omnibus34.
#
SR_INSERT_INT_OBJ= 'Netcool SR Insert',
SR_INSERT_XML_TAG= 'ListOfNetcoolSrInsert',
SR_MODIFY_INT_OBJ= 'Netcool SR Modify',
SR_MODIFY_XML_TAG= 'ListOfNetcoolSrModify',
SR_DELETE_INT_OBJ= 'Netcool SR Delete',
SR_DELETE_XML_TAG= 'ListOfNetcoolSrDelete',
SR_QUERY_INT_OBJ= 'Netcool SR Get Id',
SR_QUERY_XML_TAG= 'ListOfNetcoolSrGetId',
ACTIVITY_INSERT_INT_OBJ= 'Netcool Activity Insert',
ACTIVITY_INSERT_XML_TAG= 'ListOfNetcoolActivityInsert',
#
# Gateway location where Siebel can send any changes to its
# data. This allows bi-directional functionality.
#
GW_HOST = 'HOSTNAME',
GW_PORT= '7000',
HTTP_TIMEOUT= '60.0',
FEEDBACK_SERVER = 'NCOMS',
FEEDBACK_FIELD = 'Location',
EVENT_ID= 'SRNumber',
COUNTERPART = SERVER_READER,
CONVERSIONS_TABLE= 'conversions.siebel',
#
# These Integration Object definitions are used by the
# bi-directional side of the gateway. Siebel uses them to
# format the data for export. The gateway uses them to
# determine which operation has been performed by Siebel, and
# so they must be different.
#
MODIFIED_SR_INT_OBJ= 'Netcool SR Modify',
DELETED_SR_INT_OBJ= 'Netcool SR Delete',
INSERTED_JRNL_INT_OBJ= 'Netcool Activity Insert',
#

```

```

# The sql files are used by the bi-directional side of the gateway.
# They determine what actions the Gateway will perform with the
# incoming Siebel event.
#
OPEN_ACTION_SQL = '/opt/Omnibus/gates/siebel/open_action.siebel.sql',
UPDATE_ACTION_SQL =
'/opt/Omnibus/gates/siebel/update_action.siebel.sql',
CLOSE_ACTION_SQL =
'/opt/Omnibus/gates/siebel/close_action.siebel.sql',
JOURNAL_ACTION_SQL =
'/opt/Omnibus/gates/siebel/journal_action.siebel.sql',
#
# CLOSE_PROBLEMS will determine whether the corresponding case is closed when
# a netcool event is deleted.
#
CLOSE_PROBLEMS = TRUE,
#
# DEFAULT_CODE is the message that is returned when a case fails to be
# created.
# The default message is 'default'.
#
DEFAULT_CODE = 'Insert failed'
);
#
# Add a route from the reader to the writer so the info gets passed
#
ADD ROUTE FROM SERVER_READER TO SIEBEL_WRITE;
#
# End of file
#

```

## Writer attributes

You use writer attributes to specify how the gateway provides a communication flow between Netcool/OMNIBus and Siebel Communications. You can override the default values by setting attributes in the configuration file.

The following table describes the attributes used to create a Siebel writer.

**Note :** For details of the data type definitions for the alerts tables forwarded to the ObjectServer, see the *IBM Tivoli Netcool/OMNIBus Administration Guide, SC23-6381*.

Attribute name	Type	Description
ACTIVITY_INSERT_INT_OBJ	<i>string</i>	Use this attribute to specify the integration object defining Siebel field names populated by journal inserts from Tivoli Netcool/OMNIBus.
ACTIVITY_INSERT_XML_TAG	<i>string</i>	Use this attribute to specify the Siebel XML Tag for the Netcool Activity Insert integration object.
ACTIVITY_UPSERT_WF	<i>string</i>	Use this attribute to specify the alias for the workflow declared in the Siebel <code>eai.cfg</code> file. This workflow implements the EAI Netcool Insert Activity (http) business service.
CLOSE_ACTION_SQL	<i>string</i>	Use this attribute to specify the name of the file containing the close action SQL to be performed at the ObjectServer. No action is taken if no attribute is supplied. Used in bidirectional implementations of this gateway.
CLOSE_MAP	<i>string</i>	Use this attribute to specify the name of the close map.

Table 3. Writer attributes (continued)

Attribute name	Type	Description
CLOSE_PROBLEMS	<i>string</i>	Use this attribute to specify whether a service request is closed in Siebel if its corresponding NetcoolEvent is deleted:  TRUE: Enables service request closure in Siebel when events are deleted.  FALSE: Disables service request closure.  The default is TRUE.
CONVERSIONS_TABLE	<i>string</i>	Use this attribute to specify the conversion table within the ObjectServer that the gateway uses.  <b>Note:</b> The table specified must be created and the correct conversion values inserted before this attribute can function correctly.
COUNTER_PART	<i>string</i>	Use this attribute to specify the name of the counterpart reader in a bidirectional gateway configuration. This defines a link between the reader and the writer of one half of the bidirectional gateway and prevents the gateway from sending an alert straight back to the ObjectServer from which it was received.  <b>Note :</b> The reader must be running for the attribute value to be applied.
DATE_FORMAT	<i>string</i>	Use this attribute to specify the format of date required.  The default is %m/%d/%y %H/%M/%S.
DEFAULT_CODE	<i>string</i>	Use this attribute to specify the message that is returned when a alert fails to be created.  The default is default.
DELETED_SR_INT_OBJ	<i>string</i>	Use this attribute to specify the integration object definitions used by Siebel to format Service Request closure data for export to the gateway.
EAI_PAGE	<i>string</i>	Use this attribute to specify the web page for interfacing with Siebel Web Engine for communication with Siebel server.  <b>Tip :</b> You must specify a value for this attribute.
EVENT_ID	<i>string</i>	Use this attribute to specify the Siebel field name containing Siebel's unique identifier for service requests.
FEEDBACK_FIELD	<i>string</i>	Use this attribute to specify the name of field in FEEDBACK_SERVER where the feedback mechanism should write the service request number.

Table 3. Writer attributes (continued)

Attribute name	Type	Description
FEEDBACK_SERVER	<i>string</i>	Use this attribute to specify the name of ObjectServer where the feedback mechanism should write the service request number. <b>Tip :</b> You must specify a value for this attribute.
FILTER_CONDITION	<i>string</i>	Use this attribute to specify which records are retrieved when the cache is being rebuilt on startup. The gateway ignores this attribute if the REBUILD_CACHE attribute is set to FALSE.
GW_HOST	<i>string</i>	Use this attribute to specify the hostname where the gateway is running and where Siebel should send HTTP messages. <b>Tip :</b> You must specify a value for this attribute.
GW_PORT	<i>integer</i>	Use this attribute to specify the port on which gateway listens for HTTP connections from Siebel. <b>Tip :</b> You must specify a value for this attribute.
HTTP_TIMEOUT	<i>integer</i>	Use this attribute to specify the timeout value used in the HTTP communication between the gateway and Siebel.
INSERTED_JRNL_INT_OBJ	<i>string</i>	Use this attribute to specify the integration object definitions used by Siebel to format Service Request Activity (journal) data for export to the gateway.
JOURNAL_ACTION_SQL	<i>string</i>	Use this attribute to specify the name of the file containing the journal action SQL to be performed at the ObjectServer. No action is taken if no attribute is supplied. Used in bidirectional implementations of this gateway.
JOURNAL_MAP	<i>string</i>	Use this attribute to specify the name of the journal map.
KEY_FIELD	<i>string</i>	Use this attribute to specify the key field used in the Netcool SR Insert integration object populated with a unique value by Tivoli Netcool/OMNIBus.
MODIFIED_SR_INT_OBJ	<i>string</i>	Use this attribute to specify the integration object definitions used by Siebel to format service request update data for export to the gateway.
OPEN_ACTION_SQL	<i>string</i>	Use this attribute to specify the name of the file containing the open action SQL to be performed at the ObjectServer. No action is taken if no attribute is supplied. Used in bidirectional implementations of this gateway.

Table 3. Writer attributes (continued)

Attribute name	Type	Description
OPEN_MAP	<i>string</i>	Use this attribute to specify the name of the open map.
PASSWORD	<i>string</i>	Use this attribute to specify the password of the user specified in USER attribute. <b>Tip :</b> You must specify a value for this attribute.
REBUILD_CACHE	<i>string</i>	Use this attribute to specify whether the gateway queries the ObjectServer and rebuilds the cache on startup. The default is FALSE.
SIEBEL_VERSION	<i>integer</i>	Use this attribute to specify the version of Siebel being to which the gateway connects. <b>Tip :</b> You must set this attribute to 7.
SR_DELETE_INT_OBJ	<i>string</i>	Use this attribute to specify the integration object defining Siebel field names modified by deletes from Tivoli Netcool/OMNIBus.
SR_DELETE_XML_TAG	<i>string</i>	Use this attribute to specify the Siebel XML tag for the Netcool SR Delete integration object.
SR_GET_ID_WF	<i>string</i>	Use this attribute to specify the alias for the Workflow declared in the Siebel eai.cfg file. This workflow implements the EAI Netcool Get Service Request Id (http) business service.
SR_INSERT_INT_OBJ	<i>string</i>	Use this attribute to specify the integration Object defining Siebel field names populated by inserts from Netcool/OMNIBus.
SR_INSERT_XML_TAG	<i>string</i>	Use this attribute to specify the Siebel XML Tag for the Netcool SR Insert Integration Object.
SR_MODIFY_INT_OBJ	<i>string</i>	Use this attribute to specify the integration Object defining Siebel field names modified by updates from Tivoli Netcool/OMNIBus.
SR_MODIFY_XML_TAG	<i>string</i>	Use this attribute to specify the Siebel XML Tag for the Netcool SR Modify Integration Object.
SR_QUERY_INT_OBJ	<i>string</i>	Use this attribute to specify the integration object facilitating Siebel's response to Tivoli Netcool/OMNIBus inserts by providing new service request SR numbers.
SR_QUERY_XML_TAG	<i>string</i>	Use this attribute to specify the Siebel XML tag for the Netcool SR Get Id integration object.

Table 3. Writer attributes (continued)

Attribute name	Type	Description
SR_UPSERT_WF	<i>string</i>	Use this attribute to specify the alias for the workflow declared in the Siebel <code>eai.cfg</code> file. This workflow implements the EAI Netcool Insert SR (http) business service.
SWE_HOST	<i>string</i>	Use this attribute to specify the hostname where Siebel Web Engine is running. You must specify a value for this attribute.
SWE_PORT	<i>integer</i>	Use this attribute to specify the port on the SWE_HOST host machine on which Siebel Web Engine is listening for HTTP requests from the gateway.
SEPARATOR_CHAR	<i>string</i>	Use this attribute to enable overriding of the internally used name-value pair separator character. The separator character specified must be a character that will not be used within any value assigned to a string-valued ObjectServer field.  The default is ^.  <b>Note :</b> The value of this property can be expressed as a character or as a hexadecimal. For example, the default value ^ can alternatively be written as 0x5e in hexadecimal.
THREADPOOL_SIZE	<i>integer</i>	Use this attribute to specify the number of threads that the gateway can use to forward events from the ObjectServer to Siebel. If this attribute is set to 1, the gateway forwards one event at a time. If this attribute is set to a value greater than 1, the gateway can forward multiple events simultaneously.  <b>Note :</b> To use multi-threading, you must configure Siebel correctly. You must set the Maximum Tasks parameter of the EAI Object Manager component to the same value as that set for this attribute.
TYPE	<i>string</i>	Use this attribute to specify the type of writer. You must set this attribute to SIEBEL.
UPDATE_ACTION_SQL	<i>string</i>	Use this attribute to specify the name of the file containing the update action SQL to be performed at the ObjectServer. No action is taken if no attribute is supplied. Used in bidirectional implementations of this gateway.
UPDATE_MAP	<i>string</i>	Use this attribute to specify the name of the update map.
USERNAME	<i>string</i>	Use this attribute to specify the username for logging in to Siebel.

## Frequently asked questions

---

Various questions arise as users work with the gateway. Answers to these questions are provided for your reference.

### Why can't I check out projects in Siebel Tools?

This is a known issue affecting new installations of Siebel 7.8.2. For a workaround, refer to Service Request SR #: 38-1532354709 on Siebel SupportWeb.

### Why don't activities entered in Siebel appear as journals in the ObjectServer?

The example Netcool Activity Create workflow policy defined in [“Creating the Netcool Activity Insert integration object”](#) on page 10 includes a condition of Activity Status = In Progress. If this condition is not satisfied when a new activity (the Siebel equivalent of a Netcool journal) is created for a Service Request in Siebel, it will not be forwarded to the gateway. Thus when creating a new Activity for a Service Request within Siebel, select the value **In Progress** from the drop-down list for the Status field.

**Note :** The workflow policy described in [“Creating the Netcool Activity Insert integration object”](#) on page 10 is an example that you can adapt to suit your requirements.

### How do I set Workflow Monitor Agent component parameters?

As noted in [“Creating a new workflow monitor agent component”](#) on page 23, some monitor agent parameters might not be visible within the **Component Parameters** list of the **Enterprise Component Definition View (Configuration)** of the **Server Configuration Administration** screen. Once a component has been enabled, synchronized and the Siebel server restarted, it becomes visible on the server's view, where more parameters are visible and can be adjusted. So, to perform configuration, use the following steps:

1. Configure all the required parameters which are visible.
2. From the Component Definitions list, enable and synchronize the Component.
3. Stop the Siebel Server and Siebel Gateway Name Server.
4. Restart these services and reconnect to Siebel Communications.
5. Navigate to **Site Map > Administration – Server Configuration > Servers**.
6. In the top *Siebel Server* applet, select the server on which you want to modify the parameter for Netcool WF Monitor Agent.
7. Under the **Component** sub-tab, query for Server Manager.
8. Under the component's **Parameters** sub-tab, query for Show Advanced Objects.
9. Change Value to True.
10. Log out of Siebel to allow the **Show Advanced Objects** to take effect in Server Manager.
11. Restart the Siebel server.

All the parameters should now be visible in **Administration - Server Configuration > Servers > Netcool WF Monitor Agent > Parameters**.
12. Change the parameters not previously visible.

**Note :** The parameters may be visible here only. They may still not be visible on **Administration - Server Configuration > Enterprises Component Definitions > Netcool WF Monitor Agent > Parameters**
13. Navigate to **Site Map > Administration – Server Configuration > Enterprises > Component Definitions** and select Netcool WF Monitor Agent and press the **Synchronise** button.
14. Log out of Siebel and stop the Siebel Server and Siebel Gateway Name Server services.
15. Restart these services.

## When no filter has been specified, what criteria does the gateway use to select ObjectServer alerts to send to the Siebel gateway?

If no filter has been specified, the gateway processes all alerts.

## How does the Siebel Gateway prioritize how it sends messages to Siebel?

There is an internal queue which is processed in a first-in, first-out (FIFO) manner. IDUC notifications from the ObjectServer are processed generically, and then put into a message queue for the specific gateway code. Within each notification, the order of processing varies depending upon configuration.

There are two valid configurations.

Configuration A:

1. Status inserts
2. Status updates
3. Status deletes
4. Journal inserts
5. Details inserts

or Configuration B:

1. Status deletes
2. Status updates
3. Status inserts
4. Journal inserts
5. Details inserts

The order that is used is selected in the configuration file when creating the ObjectServer reader using the IDUC\_ORDER clause. This selects configuration A. The default is configuration B. To verify which order is being used, check the debug messages in the log:

```
Reader Reader :Pushing data to routes IUD
```

```
Reader Reader :Pushing data to routes DUI
```

**Note :** All inserts, updates, and deletes from one IDUC notification are processed before any further notifications are processed.

## How do I change GW\_HOST and/or GW\_PORT?

The GW\_PORT and GW\_HOST attributes define where the gateway listens for Siebel workflows to send events as XML over HTTP to the gateway. If you have already configured the gateway and Siebel and now need to change either value, use the following steps:

1. Edit the gateway G\_SIEBEL . conf configuration file, changing the values for the GW\_HOST and GW\_PORT attributes as required.
2. Log in to Siebel Tools and connect to the Local database.
3. Check out the **EAI Business Services** project.
4. Navigate to the Object Explorer (OE).
5. Select **Workflow Process**.
6. Do the following for each process (Export Activity (http), Export SR Delete (http), and Export SR Modification (http)):
  - Query for the process name.
  - Click the **Revise** button.
  - Select the workflow process to change.



- Right-click and select **Edit Workflow Process**.
  - Select the **Send** step in the diagram.
  - Right-click and select **Show Import Argument**.
  - For HTTP Request URL Template, change the value to `http://new_host:new_port`.
  - Click **Deploy**.
7. Check in the EAI Business Services project.
  8. Log in to Siebel Communications and connect to the server source.
  9. Navigate to **Site Map > Administration > Administration – Business Process > Workflow Deployment > Repository Workflow Process**.
  10. Query for the updated workflow processes.
  11. Click the **Activate** button for each process in turn.
  12. Query for the workflow processes in the Active Workflow Process list.
  13. Check that they are now all active.
  14. Stop and restart the Siebel server.

See also the following:

- Siebel FAQ 1555: How do you start a Workflow Monitor Agent in Siebel Version 7.x?
- Technical Note 581: Siebel 7.7 Workflow Best Practices

Both of these are available from Siebel SupportWeb.

### **Why does the gateway lose its connection to Siebel immediately after a journal?**

This is due to setting the GW\_PORT attribute in the G\_SIEBEL.conf configuration file to the same value as the port number set for the gateway in nco\_xigen. Change the port number specified for the gateway GW\_PORT attribute using the instructions specified by the previous frequently asked question.



---

## Appendix A. Notices and Trademarks

This appendix contains the following sections:

- Notices
- Trademarks

### Notices

---

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation  
Licensing 2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106-0032, Japan

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who want to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation  
Software Interoperability Coordinator, Department 49XA

3605 Highway 52 N  
Rochester, MN 55901  
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

#### COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (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.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

## Trademarks

---

IBM, the IBM logo, ibm.com, AIX®, Tivoli, zSeries, and Netcool are trademarks of International Business Machines Corporation in the United States, other countries, or both.

Adobe, Acrobat, Portable Document Format (PDF), PostScript, and all Adobe-based trademarks are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

Intel, Intel Inside (logos), MMX, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java™ and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Linux® is a trademark of Linus Torvalds in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.







Part Number:

SC23-6394-03



(1P) P/N: