# Siebel

## Business Process Framework: Workflow Guide

January 2025

Siebel
Business Process Framework: Workflow Guide

January 2025

Part Number: F87444-02



Authors: Siebel CRM Information Development Team

# Contents

# Preface

This preface introduces information sources that can help you use the application and this guide.

## Using Oracle Applications

To find guides for Oracle Applications, go to the Oracle Help Center at *https://docs.oracle.com/*.

## Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the *Oracle Accessibility Program website*.

## Contacting Oracle

### Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit *My Oracle Support* or visit *Accessible Oracle Support* if you are hearing impaired.

### Comments and Suggestions

Please give us feedback about Oracle Applications Help and guides! You can send an email to: *oracle_fusion_applications_help_ww_grp@oracle.com*.

# 1 Whats New in This Release

## Whats New in This Release

This chapter tracks the changes in the documentation. It includes the following topics:



### What's New in Siebel Business Process Framework: Workflow Guide, Siebel CRM 25.1 Update

The following table lists the changes in this revision of the documentation to support this release of the software.

| Topic | Description |
|---|---|
| *Viewing All Workflow Process/Task Steps in the Web Tools Workflow Process/Task Editor* | New topic. This documentation explains how to make all the steps of a Workflow Process viewable in the editor, no matter how many there are. |

### What's New in Siebel Business Process Framework: Workflow Guide, Siebel CRM 23.10 Update

The following table lists the changes in this revision of the documentation to support this release of the software.

| Topic | Description |
|---|---|
| *Removing a Workflow Process Instance from Workflow Instance Admin* | New topic. Functionality update to purging workflow instances. |

### What's New in Siebel Business Process Framework: Workflow Guide, Siebel CRM 22.7 Update

The following table lists the changes in this revision of the documentation to support this release of the software.

| Topic | Description |
|---|---|
| *Consequence of Pausing the Process Simulator in DR* | New topic. Workflow Processes paused in one Workspace cannot be resumed in another Workspace. |

## What's New in Siebel Business Process Framework: Workflow Guide, Siebel CRM 22.4 Update

The following table lists the changes in this revision of the documentation to support this release of the software.

| Topic | Description |
|---|---|
| *Validating the Workflow Process* | Modified topic. The Validate Tool is now supported in Web Tools. The Applet Menu in Web Tools has a new Validate option, which you can use to validate objects of a single object type. |

## What's New in Siebel Business Process Framework: Workflow Guide, Siebel CRM 21.12 Update

The following table lists the changes in this revision of the documentation to support this release of the software.

| Topic | Description |
|---|---|
| *Setting the Monitoring Level Parameter* | Modified topic. Step 2 in this procedure has changed. |
| *Collecting Data About Workflow Process Properties* | Modified topic. Step 6a in this procedure has changed. |
| *Deploying a Workflow Process to Siebel Mobile Web Client* | Modified topic. Step 2 in this procedure has changed. |
| *Verifying the Functionality of the Workflow Process* | Modified topic. Step 7b in this procedure has changed. |
| *About Using the Process Simulator*<br>*Process Simulator*<br>*Preparing to Use the Process Simulator*<br>*Using the Process Simulator* | Modified topics. Since Siebel CRM 21.9 Update, the Process Simulator is supported in Web Tools. |
| *Notifying Mobile Users Who Are Not Synchronized* | Modified topic. The example procedure shown in this topic is new. |

# What's New in Siebel Business Process Framework: Workflow Guide, Siebel CRM 21.9 Update

The following table lists the changes in this revision of the documentation to support this release of the software.

| Topic | Description |
|---|---|
| *Restricting Client Access to Workflows* | New topic. Describes the Restrict Client Access for WFs system preference and the purpose of setting it. |
| *Process Simulator*<br>*Using the Process Simulator* | Modified topics. If inspecting a business service for the first time, then you must manually add it and any associated method(s) to execute the business service in the Process Simulator. |
| *Monitoring a Workflow Process* | Modified topic. Workflow Process Monitoring is not available when inspecting a Developer Workspace. |
| *Identifying a Workflow Process That a Runtime Event Starts* | Modified topic. Describes how to identify a Workflow Process that a runtime event starts. |
| *Workflow Admin Service Business Service* | Modified topic. The ability to import a Workflow Process from a non Workspace-enabled environment to a Workspace-enabled environment is currently not supported. |

# What's New in Siebel Business Process Framework: Workflow Guide, Siebel CRM 21.6 Update

The following table lists the changes in this revision of the documentation to support this release of the software.

| Topic | Description |
|---|---|
| *Concatenating a Process Property*<br>*Using the Watch Window in the Process Simulator*<br>*Using the Process Simulator*<br>*Validating and Testing the Workflow Process*<br>*Watch Window* | Modified topics. In Web Tools, the Watch Window is called Workflow Process Data. |

# What's New in Siebel Business Process Framework: Workflow Guide, Siebel CRM 21.4 Update

The following table lists the changes in this revision of the documentation to support this release of the software.

| Topic | Description |
| --- | --- |
| *Overview of Siebel Workflow Process* | Modified chapter. Describes Siebel Workflow, Workflow Process, and Workflow Policy. |
| *Siebel Workflow Process Runtime Architecture* | Modified topic. Describes the components in the runtime architecture that Siebel Workflow Process uses. |
| *About Developing a Workflow Process* | Modified topic. Describes a typical approach to developing a Workflow Process: |
| *Using the Process Designer*<br>*Modifying a Workflow Process*<br>*Diagramming a Workflow Process*<br>*Adding or Removing a Connector Point*<br>*Editing a Workflow Process*<br>*Adding Workflow Process Steps*<br>*Using the Multi Value Property Window* | Modified topics. Since Siebel CRM 20.12 Update, the Workflow Editor is supported in Web Tools. You can click Edit Workflow Process (the pencil icon) or drill down on the Process Name in the Workflow Processes list to open and modify existing Workflow Processes. |
| Various topics | Various topics have been removed from the guide. |

# 2 Overview of Siebel Workflow

## Overview of Siebel Workflow

This chapter provides an overview of Oracle's Siebel Workflow, Workflow Process, and Workflow Policy. It includes the following topics:

- *About Siebel Workflow*
- *About Workflow Processes*
- *About Workflow Policies*
- *About Using this Book*

## About Siebel Workflow

*Siebel Workflow* is a set of tools that allow you to define, manage, and enforce your business processes and establish business automation in Siebel CRM. Siebel Workflow coordinates business processes across many Siebel CRM components.

*Workflow Processes* are the flowchart-like objects with branching logic that operate at an application object manager level, while *Workflow Policies* use triggers and have predefined actions that operate at the database layer.

### Business Requirements that Siebel Workflow can Meet

Siebel Workflow can provide the following types of functionality:

- Escalating an event, such as a Service Request exceeding its Service Level Agreement window.
- Routing and assigning work.
- Enforcing authorization and transition rules.

Using Siebel Workflow Processes help to ensure:

- Efficiency
- Quality of Service
- Contractual obligations are met
- Maximization of profitability

#### Examples of Meeting Business Requirements

A service department can use a Workflow Process that meets the following business requirements:

- **Implement standards to process calls.** If Siebel CRM assigns a Severity 1 call, then send an email to the new owner.
- **Meet contracted service agreements.** If a customer purchases a support agreement, then make sure a callback occurs within two hours and make sure the problem is resolved within four hours.

A sales department can use a Workflow Process to meet the following business requirements:

- **Implement discount authority.** If a sales representative quotes a discount that exceeds the maximum discount allowed, then get the approval of the district sales manager or VP of Sales.
- **Manage the pipeline.** To promote sufficient levels of prospects at each stage of the sales cycle, each sales representative manages their own pipeline. If an area of the pipeline requires attention, then alert the representative or manager.
- **Make sure forecasting is accurate.** If a user forecasts an opportunity but never closes it, then flag the opportunity. If a forecast includes a wide discrepancy from the actual revenue, then flag the opportunity.

# Technologies that Siebel CRM Uses to Automate Business Processes

The following table describes some of the technologies that Siebel CRM uses to automate a business process. Siebel Workflow coordinates Workflow Processes and other repository objects, such as Workflow Policies, to implement a complete Workflow Process. Each of these technologies helps to automate a business process. To coordinate the services that these technologies provide, a Workflow Process calls each technology directly or interacts with them through the Siebel event model.

| Automation Technology | Description |
|---|---|
| Siebel Workflow Process Designer | Uses a familiar flowcharting interface that allows you to define business processes for your company. Includes one or more process steps, such as a start step, sub process step, decision step, or task step. |
| Workflow Policy | Allows you to define the Workflow Policy conditions and actions that can start a Workflow Process. If Workflow Policy conditions are met, then the policy action runs the relevant Workflow Process. A Workflow Policy creates an event that references a database operation. A Workflow Policy can perform a simple action, such as sending an email message, or creating an activity or assignment. |
| Siebel Workflow Process Task | Allows you to define a user interface that is similar to a wizard, with multiple steps and interactive operations that can include branching and decision logic, which guides the user through a task. It allows the user to navigate backward or forward, and to pause or resume a task. For more information, see *Siebel Business Process Framework: Task UI Guide* . |
| Assignment Manager | Allows you to define rules that can assign a record to a user according to skill, workload, and availability. Allows you to reassign a Workflow Process. For more information, see *Siebel Assignment Manager Administration Guide* . |
| SmartScript | Guides the user through data entry work. Supports call scripting and basic support for transaction level commits. For more information, see *Siebel SmartScript Administration Guide* . |
| Activity Template | Allows you to define a series of steps that the user performs. Effective for handling asynchronous and offline work. For more information on Activity Template, see *Siebel Applications Administration Guide* . |
| State Model | Manages how and when the user can change the status of a record according to a current value and the user position. Can also enforce directional progression of status. For example, to force an opportunity to move forward but not backward through a pipeline. For more information on the State Model, see *Siebel Applications Administration Guide* . |

| Automation Technology | Description |
|---|---|
| | |
| Personalization Engine | Handles runtime events, such as application events, applet events, and business component events. A Workflow Process uses the Personalization engine to handle these events. A Workflow Process that a runtime event starts or resumes registers itself with the Personalization engine when Siebel CRM activates the Workflow Process. If a runtime event occurs in a user session, then the Personalization engine calls the Workflow Process in the local object manager. |

# About Using Workflows in Workspaces

Note the following about using Siebel workflows in Workspaces:

- You cannot perform any operation on workflows that are in the MAIN/Integration branch - you have to create Developer Workspaces under the MAIN/Integration branch (first) to perform the operations on workflows.
- Workflows cannot be deleted.
- You have to deliver a Workspace to activate a workflow.
- Once a Workspace is delivered, workflows under that Workspace are activated and published into the Runtime table.

For more information about Workspaces and using Workspaces, see *Using Siebel Tools* .

# About Invoking Workflows

You can use one of the following methods to invoke a workflow:

- Calculating fields through the InvokeSiebelMethod
- Applet User Property such as Named Method
- RTEs
- Running a script
- Using other workflows
- Applying workflow policies
- Applying web services

# Restricting Client Access to Workflows

You can control client access to Workflow Processes by setting the Restrict Client Access for WFs system preference to TRUE. The purpose of setting this system preference to TRUE is to prevent Workflow Processes from being called over SOAP/UI (for example, from the browser console using the Workflow Process Manager). Typically, if the Workflow Process Manager is permitted as a Business Service, then every Workflow Process can be called, which could pose a security risk.

If the Restrict Client Access for WFs system preference is set to TRUE, then all Workflow Processes are blocked from being called over SOAP/UI by default. You must then unblock the particular Workflow Processes that you want to access

by adding them as Application User Props (`ClientWorkflowNameX`) for the applications where you want to allow the Workflow Processes.

# About Workflow Processes

A Workflow Process is Oracle's Siebel Workflow representation of a business process or the graphic representation of a sequence of automated steps that support a business process. It includes one or more steps that indicate when a business process starts and ends, and specifies inputs and outputs for individual steps as well as for the entire Workflow Process.

Workflow Processes can be simple (such as entering a product order) or complex (such as managing a call center). A Workflow Process operates at the application object manager level and can perform operations on data, can include conditional logic, and can call a business service, task, or sub process. For more information about Workflow Processes, see the following topics:

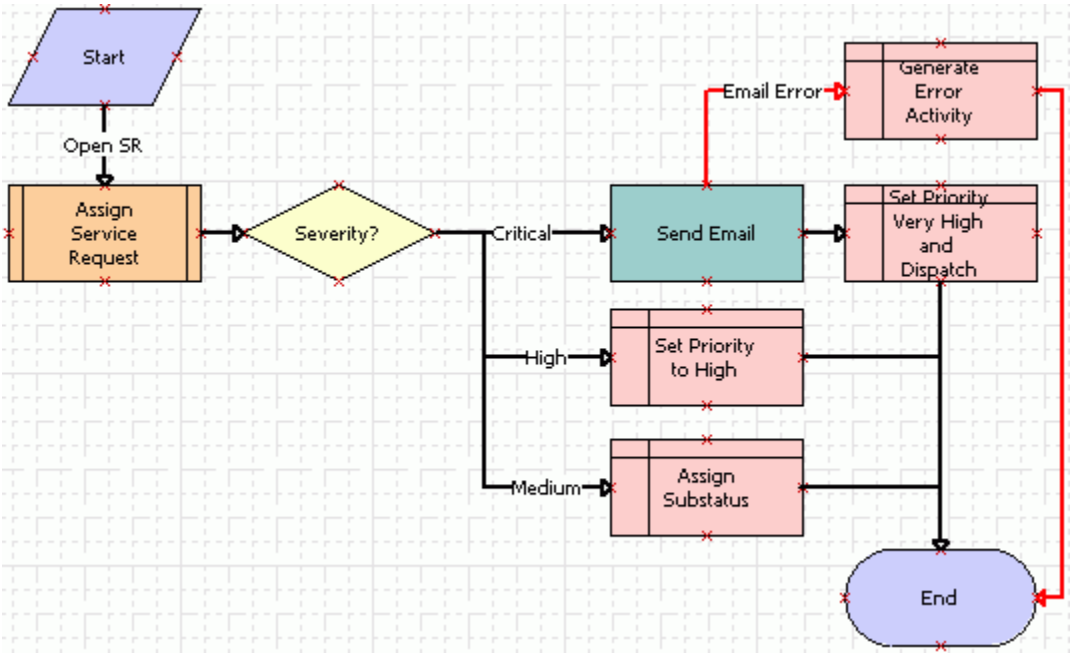
- *Scenario that Resolves Service Requests*
- *Creating a Workflow Process*
- *Adding Workflow Process Steps*
- *Manipulating Data in a Workflow Process*
- *Testing a Workflow Process*
- *Administering a Workflow Process*
- *Configuring a Workflow Process*
- *Example Workflow Processes*
- *Example Workflow Processes That Call a Business Service*

## Scenario that Resolves Service Requests

The scenario in this topic describes how a Workflow Process can automate a business process. In this scenario, a service manager for a high volume service agency thinks the organization cannot resolve customer issues in a timely manner. To improve how the organization tracks and manages service requests, the service manager decides to use a Workflow Process that automates the service request management process.

The goal is to meet a Service Level Agreement commitment that makes sure the organization resolves new service requests (SRs) within a specific amount of time. The service manager requires Siebel CRM to assign SRs according to the representative who is available and who possesses the skills that are required to resolve the SR. If the SR requires immediate attention, then Siebel CRM must notify the SR owner.

The developer uses the Workflow Process Designer to define the business process that Siebel CRM uses when a user creates a new service request. The following figure shows an example of the Workflow Process. It includes the steps that Siebel CRM uses if a new service request is received by the the organization.




The following table describes each step, as shown in this figure, that the new service request scenario uses. The following summarizes the flow in the New Service Request Workflow Process:

- If a user creates an SR, then Siebel CRM starts the Workflow Process.
- It calls the Siebel Assignment Manager to assign the SR to the service representative who is available and who possesses the skills required to resolve the SR.
- According to the SR severity, the Workflow Process can use the Siebel Communications Server to send an email message that notifies the representative.
- Automating this process helps the company achieve faster turnaround time to resolve SRs and to meet service commitments.

| Step Name | Step Type | Description |
|---|---|---|
| Start | Start | Every Workflow Process includes a start step that it uses to start the Workflow Process. |
| Open SR | Branch Connector | The connector that emanates from the start step defines the condition that starts this Workflow Process. |
| Assign Service Request | Sub Process | Siebel CRM assigns the service request to the appropriate service representative, according to assignment rules that the sub process runs. |
| Severity? | Decision Step | The decision step uses the service request severity to determine the next step to run: Critical, High, or Medium. |
| Send Email | Business Service | If the service request priority is critical, then the business service step sends an email to the assigned service representative. |
| Set Priority to High | Siebel Operation | The Set Priority to High step sets the service request priority to High. |

| Step Name | Step Type | Description |
|---|---|---|
| | | |
| Assign Substatus | Siebel Operation | The Assign Substatus step sets the substatus to Assigned. |
| Email Error | Error Exception Connector | If Siebel CRM cannot deliver the email or if the Send Email business service step returns some other error, then the Workflow Process runs an error exception. |
| Generate Error Activity | Siebel Operation | If an error occurs, then the Generate Error Activity step creates an activity that manages this error. |
| Set Priority Very High then Dispatch | Siebel Operation | The Set Priority Very High then Dispatch step sets the service request priority to Very High and the substatus to Dispatch. |
| End | End | The Workflow Process ends. |

# About Workflow Policies

A Workflow Policy is a systematic expression of a rule which includes one or more Workflow Policy conditions and one or more Workflow Policy actions. If the Workflow Policy conditions for a Workflow Policy are true, then Siebel CRM runs the Workflow Policy action. A Workflow Policy creates an event that references a database operation. A Workflow Policy can perform an action, such as, sending an email message or creating an activity or assignment.

A Workflow Policy is contained by one Workflow Policy group and is related to one Workflow Policy object. For more information about Workflow Policies, see the following topics:

- *Starting a Workflow Process from a Workflow Policy*
- *Using Predefined Workflow Policies*
- *Defining Custom Workflow Policies*
- *Administering Workflow Policies*
- *Monitoring, Testing, and Migrating Workflow Policies*

# About Using this Book

This book uses the following terms:

- A *user* is a person who uses the client of a Siebel Business Application to access Siebel CRM data.
- The *client* (hardware or software) connects to or accesses a Siebel Business Application, such as, Siebel Call Center.
- The *server* is the Siebel Server, unless noted otherwise.
- An *administrator* is anyone who uses an administrative screen in the client to configure Siebel CRM. The Administration - Server Configuration screen is an example of an administrative screen.

- A *predefined object* is an object that comes already defined with Siebel CRM.
- The term *focus* indicates the currently active object in the client. To indicate the object that is in focus, Siebel CRM typically sets the border of this object to a solid (blue) line.
- The terms *drag* and *drag and drop* are used in this guide to describe how to move items from one place on the UI to another using a mouse or similar device. You move an item by first selecting the item and (with the mouse button depressed) then moving the item elsewhere on the UI (where you release the mouse button).

Depending on the software configuration you purchase, your Siebel Business Application might not include all the features that this book describes.

Computer font indicates a value you enter or text that Siebel CRM displays. For example:

```
This is computer font
```

Italic text indicates a variable value. For example, the `n` and the `method_name` in the following example are variables:

```
Named Method <n>: <method_name>

<-- For example: -->
Named Method 2: WriteRecord
```

This guide assumes that you understand and possess the following the basic skills:

- Know how to use the Object Explorer and Object List Editor.
- Know how to define object properties, applets, and applet controls.
- Know how to use the menu bar.
- Know how to check out and check in projects.

For more information, see *Using Siebel Tools* and *Configuring Siebel Business Applications* .

# How this Book Describes Objects

This book describes how an object, such as a user property, does something. For example:

- The Copy Contact user property copies contacts.

  In reality, the Copy Contact user property only includes information that some other Siebel CRM component uses to copy contacts.

To describe how Siebel CRM uses the value that a property contains, this book typically only describes the property name. For example, assume Siebel CRM displays the value that the Display Name property contains. This is a property of a tree node object. This book only states the following:

- Siebel CRM displays the Display Name property of the tree node.

  In reality, Siebel CRM displays the value that the Display Name property contains.

# How this Book Describes Relationships Between Objects

An object definition includes properties and a property includes a value. For example, the Business Object property of the Account Address view contains a value of Account. To describe this relationship, this book states the following:

- The Account Address view references the Account business object.

Sometimes the relationship between objects occurs through multiple objects. For brevity, this book does not always describe the entire chain of relationships that exists between objects through the entire Siebel object hierarchy. For example, because the Account business object references the Account business component, and the Account Address view references the Account business object, this book states the following:

- The Account Address view references the Account business component.

# Getting Help from Oracle

If you require help from Oracle for using object types, you can create a service request (SR) on My Oracle Support. Alternatively, you can phone Global Customer Support directly to create a service request or get a status update on your current SR. Support phone numbers are listed on My Oracle Support. You can also contact your Oracle sales representative for Oracle Advanced Customer Services to request assistance from Oracle's Application Expert Services.

# 3 Siebel Workflow Process Architecture

## Siebel Workflow Process Architecture

This chapter describes the architecture that Siebel Workflow Processes use. It includes the following topics:

- *Siebel Workflow Process Runtime Architecture*
- *Object Hierarchy That Workflow Processes Use*
- *About Developing a Workflow Process*
- *About Using the Process Simulator*
- *About Activating a Workflow Process*

The following topics include more information of an architectural nature:

- *Overview of Workflow Policy Objects*
- *Sequence That a Workflow Policy Uses*

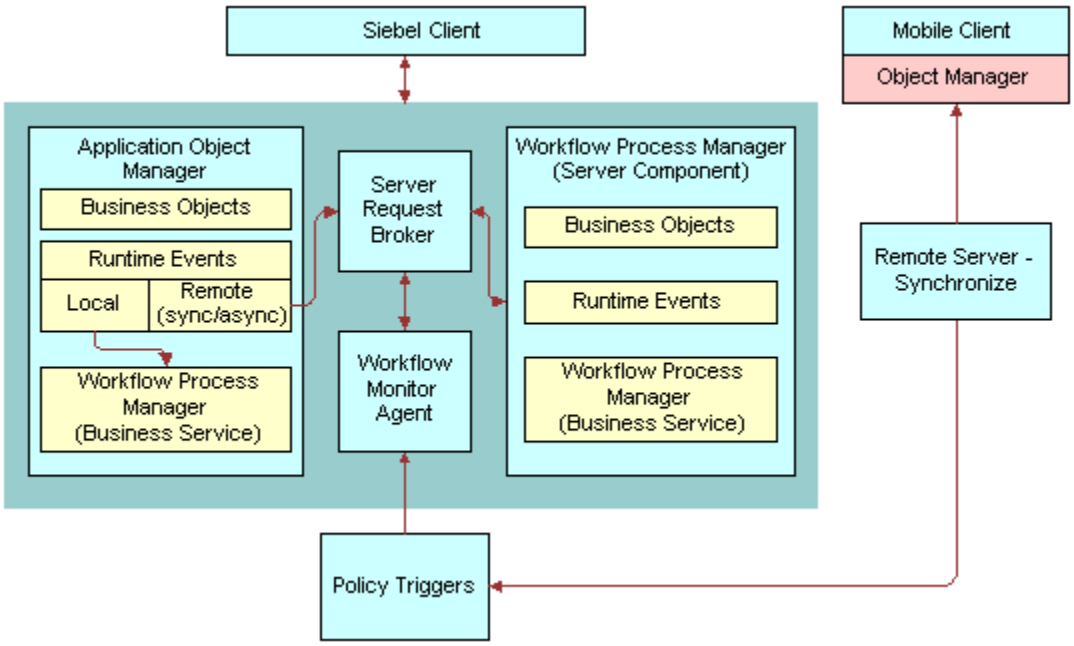
For more information, see *Using Siebel Tools* .

## Siebel Workflow Process Runtime Architecture

A Workflow Process can run as a business service or as a server component in the runtime environment. You can invoke a Workflow Process either synchronously or asynchronously.

- In a synchronous scenario, everything is carried out *in process* (for example, in the Application Object Manager that the user is currently interacting with) and the Workflow Process will run to completion before returning control to the calling process.
- In an asynchronous scenario, the Server Request Broker and Server Request Manager are invoked to run operations in the background (*out-of-process*). For more information, see *Server Request Broker Server Component*, *Workflow Process Manager Server Component*, and *Server Requests Business Service*.

The following figure shows the components in the runtime architecture that Siebel Workflow Process uses.




As shown in this figure, the components in Siebel Workflow Process runtime architecture are as follows:

- **Application Object Manager.** Siebel business objects, runtime events (local and remote, synchronous and asynchronous) and the Workflow Process Manager (Business Service) reside on the Application Object Manager.
- **Server Request Broker Server Component.** The Server Request Broker brokers the request to the appropriate server component. For more information, see *Server Request Broker Server Component*.
- **Workflow Process Manager (Server Component).** The Workflow Process Manager server component runs the Workflow Process as a business service. For more informaiton, see *Workflow Process Manager Server Component* and *Components of the Workflow Management Server Component Group*.

  **Workflow Process Manager (Business Service).** For more information, see the following:

  - *How the Business Service Determines Where the Workflow Process Runs*
  - *How the Workflow Process Runs in the Workflow Process Manager Server Component*

# Server Request Broker Server Component

The Workflow Engine sends a request to the Server Request Broker asynchronously, and then the Server Request Broker brokers the request to the appropriate server component. It does the following work:

- Sends asynchronous messages from an interactive server component to the Workflow Engine
- Communicates, asynchronously, between the Workflow Engine and batch components
- Schedules repeated server tasks that Siebel CRM runs periodically in the Workflow Engine

The Server Request Broker also does load balancing. If it receives a request, then it routes this request to a server component that resides in the current server. If this server component is not available in the current server, then the Server Request Broker sends it to other servers where the Workflow Process Manager is active. It does this on a round robin basis.

A Workflow Process also uses the Server Request Broker to resume a Workflow Process that is waiting. The Server Request Broker periodically queries a database table to identify server tasks that it must resume.

For more information, see *Server Requests Business Service* and *Siebel System Administration Guide* .

# Workflow Process Manager Server Component

The Workflow Process Manager (WfProcMgr) is a server component that uses the Siebel Object Manager. It runs a Workflow Process as a business service. It hosts the business object layer and the data object layer that allows Siebel CRM to run multiple object managers and multiple server tasks for each object manager. The term Workflow Process Manager refers to the component that runs Workflow Processes asynchronously in the background (out-of-process).

The Workflow Process Manager can be in one of the following states:

- **Online.** The Workflow Process Manager is active and can receive and process requests.
- **Not online.** For example, Inactive, Shutdown, or Offline. Siebel CRM cannot process requests to the Workflow Process Manager or Workflow Process Batch Manager (WfProcBatchMgr) server components. If Siebel CRM saves a request to the Siebel database, and if it sends this request in DirectDb, then it can resend the request later when the Workflow Process Manager comes back online. If Siebel CRM does not save this request, then the request is lost.

For more information, see *Starting a Workflow Process from the Workflow Process Manager*.

# How the Business Service Determines Where the Workflow Process Runs

The Workflow Engine includes the Workflow Process Manager business service and the Workflow Process Manager (Server Request) business service. The following table shows where the Workflow Process runs for each business service. The developer must decide whether to invoke the Workflow Process Manager (synchronous scenario) or the Workflow Process Manager (Server Request) (asynchronous scenario).

| Business Service | Location Where the Workflow Process Run |
|---|---|
| Workflow Process Manager | Synchronously, in the object manager of the Siebel application. |
| Workflow Process Manager (Server Request) | Asynchronously, in the Workflow Process Manager server component. |

# How the Workflow Process Runs in the Workflow Process Manager Server Component

A Workflow Process Manager server component that Siebel CRM configures and optimizes to run the Workflow Process Manager business service can run a Workflow Process in the background (asynchronous scenario). The Workflow

Process Manager server component works as the object manager that runs the Workflow Process, including any application logic that the Workflow Process uses.

The Workflow Process Manager accepts the Workflow Process name in the following ways:

- Through the Process Name server component parameter. For example, if Siebel CRM starts a server task from the Server Manager or from a repeating server component request.
- Through the Encoded Args server component parameter. For example, if the Workflow Monitor Agent business service or the Server Requests business service sends the request.

# Components of the Workflow Management Server Component Group

The following table describes the components of the Workflow Management server component group.

| Server Component | Alias | Description |
|---|---|---|
| Workflow Process Manager | WfProcMgr | The Workflow Process Manager server component does the following:<br>• Executes a single time when invoked by some other process.<br>• Acts as the Application Object Manager that runs Workflow Processes.<br>• Is a specialized server component configured and tuned to run Workflow Processes.<br>• Is similar to a server component that provides a multithreaded environment. |
| Workflow Process Batch Manager | WfProcBatchMgr | The Workflow Process Batch Manager server component does the following.<br>• Executes the same workflow over and over again against different records.<br>• Acts as the Application Object Manager that runs Workflow Processes.<br>• Is a specialized server component configured and tuned to run Workflow Processes.<br>• Is similar to a server component that provides a multithreaded environment. |
| Workflow Monitor Agent | WorkMon | Runs and monitors Workflow Policies, and then runs actions when the conditions of a Workflow Policy are met. |
| Workflow Action Agent | WorkActn | Logs requests in the action request table (S_ESCL_ACTN_REQ) for a policy group and calls actions that the Workflow Policy uses. |
| Workflow Recovery Manager | WfRecvMgr | Polls the Workflow Engine to identify Workflow Process instances that are running on the Siebel Server. Recovers failed instances and resumes instances that are waiting beyond a due date. For more information, see *Recovering a Workflow Process*. |
| Generate Triggers | GenTrig | The Generate Triggers server component does the following:<br>• Allows you to create database triggers that Workflow Policies use to identify records that match Workflow Policy conditions<br>• Must be rerun if you create, update, or delete a policy |

| Server Component | Alias | Description |
|---|---|---|
| | | • Can be run from the Server Manager command line or through a command line interface.<br>For more information, see *Overview of Creating Database Triggers*. |

# Object Hierarchy That Workflow Processes Use

You can modify predefined Siebel objects and define new objects that meet a business requiremet for your organization. Just as you modify the data model, business logic, and define the user interface for Siebel CRM, you can aslo define a Workflow Process that Siebel uses to automate a business process for your organization. Workflow Processes are defined in the Siebel Repository just like Applets, BusComps, and other objects.

# About Developing a Workflow Process

The following describes a typical approach to developing a Workflow Process:

1. **Define.** Create a definition for the Workflow Process, including the Process Properties, Steps and Connectors using the Workflow Process Designer.
2. **Simulate and Debug.** Simulate the Workflow Process, which uses a Siebel Application (such as, Call Center) to step through the steps and ensure it is working as required. For more information, see *Process of Testing a Workflow Process*.
3. **Deliver.** Commit the Workflow Process to the parent Workspace in which you designed it. This compiles it into the Runtime Repository tables in that Integration Workspace (or MAIN).
4. **Migrate and Test.** Like any other Workspace-enabled repository object, you can migrate the Workflow Process's Runtime Repository definition to downstream environments for further testing.
5. **Migrate to Production.** Once the Workflow Process has been fully tested and the Integration Workspace used for testing is delivered to the Workspace tied to Production, it will be migrated to the Production environment like any other Runtime Repository object.

# About Using the Process Simulator

You can use the Process Simulator to test a Workflowy Process. Testing your Workflow Process before you deliver it to its parent Integration Branch makes sure that it works correctly and that the results meet your business requirements.

You use the following to simulate a Workflow Process:

1. **Repository data.** The Process Simulator accesses object definitions from the repository that are part of the Workflow Process or that the Workflow Process references.
2. **Runtime data.** During simulation, data is accessed or modified depending on how you define the Workflow Process. It can access data in the runtime database, such as customer data that resides in various fields of an opportunity.

If you run a Workflow Process from the Process Simulator, then it runs in the Application Object Manager. You can start a Workflow Process in the Application Object Manager or in a server session of the Workflow Process Manager.

For more information, see *Process Simulator*.

# About Activating a Workflow Process

Workflow Processes are active in the Siebel CRM application if they are active in the Siebel Repository.

For example, there are over 1,800 Workflow Processes in the out-of-the-box Siebel CRM repository but only a few are required for the application to function. Some of these are *system workflows*, for example, those that support the Siebel Migration Application or Application Deployment Manager modules. The remainder are needed only for customers using particular functionality. For example, a Consumer Goods customer only needs the workflows related to Consumer Goods functionality and does not need workflows related to the Financial Services functionality.

In a new Siebel CRM deployment, only the system workflows are active, and the remainder are not. Customers who wish to activate the out-of-the-box industry specific functionality must activate the relevant workflows as documented in the Siebel CRM Bookshelf Guide for that industry.

The process for activating a particular industry-specific workflow is consistent with any other Siebel CRM repository change; that is, you create a Workspace, activate the workflow, and deliver it. For more information about managing Workspaces, see *Using Siebel Tools* .

Once a Workflow Process is active, you can still make changes to the runtime settings related to monitoring and logging. For more information, see *Monitoring, Testing, and Migrating Workflow Policies*.

# 4 Developing Workflow Processes

## Developing Workflow Processes

This chapter describes how to develop a Workflow Process. It includes the following topics:

- *Roadmap for Developing Workflow Processes*
- *Process of Analyzing Business Requirements*
- *Process of Planning Workflow Processes*
- *Job Roles Used to Develop a Workflow Process*

## Roadmap for Developing Workflow Processes

To develop a Workflow Process, perform the following processes:

1. *Process of Analyzing Business Requirements*
2. *Process of Planning Workflow Processes*
3. *Creating a Workflow Process Object Definition*
4. *Diagramming a Workflow Process*
5. *Process of Testing a Workflow Process*

## Process of Analyzing Business Requirements

This process is a step in *Roadmap for Developing Workflow Processes*.

The first step in developing a Workflow Process includes analyzing your business requirements to determine the rules and business processes that the Workflow Process must automate. To define the processes to automate, a business analyst uses a Siebel CRM application and then determines the most appropriate automation solution. The developer who defines the Workflow Process often participates as a technical consultant during this analysis.

To analyze business requirements, do the following tasks:

1. *Gathering Information for Planning a Workflow Process*
2. *Identifying Business Process Actions*
3. *Identifying an Automation Solution*

### Gathering Information for Planning a Workflow Process

This task is a step in *Process of Analyzing Business Requirements*.

Gathering information for planning a Workflow Process involves the following:

- Determining how your organization currently handles business processes. For more information, see *Analyzing Existing Performance of a Business Process*.

- Determining how the business process must perform in the future. For more information, see *Identifying Areas for Improvement*.

## Analyzing Existing Performance of a Business Process

Current business processes provide the basis of what you define when you use Siebel Workflow. If you currently use an automated system, then you must gather information about the business processes that this system handles. It is also important to understand the limitations or problems of the current system that the Workflow Process must solve.

When analyzing the existing performance of a business process, research the following areas of the current business process:

- Existing process information. The following sources might include existing process information:
  - Current business processes that are automated
  - Management guidelines
  - Written guidelines for business process rules or approval paths
  - Written or unwritten internal procedures
- Measures for improvement or new process requirements

For example, assume you must document a new work item, such as a service request, from the moment that someone opens the service request to the moment someone closes it. You can include information about the decision steps in the business process, such as, if someone must escalate the service request and describe the approval path that Siebel CRM must run if the service request is high priority compared to low priority.

## Identifying Areas for Improvement

After you gather the required information about existing business processes, review it to determine if improvement opportunities exist or if a new business process is required. When identifying areas of improvement, consider each of the following areas for improvement:

- New management guidelines or business requirements that must be considered
- Current problems that must be solved
- Areas you must make more visible
- Customer satisfaction issues
- Workflow Processes you must automate

# Identifying Business Process Actions

This task is a step in *Process of Analyzing Business Requirements*.

A business process includes actions that your organization must perform to meet business requirements. Siebel CRM provides a number of predefined actions. The following are some example predefined actions:

- **Notifications.** Send an email, page, or fax.
- **Siebel Operations.** Insert or update information in the Siebel database.
- **Integration Messages.** Request to send or receive data from an external system.
- **Assignment.** Use Assignment Manager to assign an object to someone.

- **Navigation.** Navigate a user to a specific view through a user interact step or a call.
- **Server Request.** Use the Siebel Server Request Broker to run a server process.

Siebel CRM calls a business service method to start each of these actions. This situation is true except for a Siebel operation. You can use these actions in a variety of settings and technical configurations. For example, you might identify a specialized action that calls a Workflow Process, such as calculate credit risk. You can define a custom business service to use a specialized action. A Workflow Process can call a predefined business service or a custom business service. For more information, see *Example Workflow Processes That Call a Business Service* and *Integration Platform Technologies: Siebel Enterprise Application Integration* .

### To identify the business process actions

- Map the requirements you identified in *Gathering Information for Planning a Workflow Process* to potential predefined Siebel actions.

## Identifying an Automation Solution

This task is a step in *Process of Analyzing Business Requirements*.

After you determine business process requirements and the actions that must be performed to meet those requirements, you can identify an automation solution.

### To identify an automation solution

1. Identify the solution that most closely meets the business process requirements. The following table describes the advantages and limitations of the following automation solutions: Workflow Process, Workflow Policy, and Siebel Script.

| Solution | Advantages | Limitations |
|---|---|---|
| Workflow Process | This solution includes the following advantages:<br>o Visual representation of business logic is simple to understand and maintain<br>o Remote synchronous and asynchronous capability provides compatibility across Siebel CRM for scalability and long-running transactions | This solution includes the following limitations:<br>o The semantics for control are not as rich as with scripting<br>o Limited control of flow for iteration through record sets<br>o Limited direct access to object methods |
| Workflow Policy | This solution includes the following advantages:<br>o Replies to a database event regardless of whether or not an Object Manager server component starts the policy<br>o Can realize higher transaction throughput for a simple transaction | This solution includes the following limitations:<br>o Policy changes might require database downtime<br>o More difficult to define than other alternatives<br>o Provides only a limited range of actions |
| Siebel Script | This solution includes the following advantages:<br>o Familiar to many developers<br>o Provides a set of semantics | This solution includes the following limitations:<br>o More difficult to maintain and upgrade<br>o Slower performance |

| Solution | Advantages | Limitations |
|---|---|---|
| | ○ Is flexible | |

2. Determine if a Workflow Process or a Workflow Policy can meet the requirement(s). The following table describes when to use a Workflow Process or Workflow Policy solution for certain key requirements. For more information about Workflow Policy, see *Defining Custom Workflow Policies*.

| Requirement | Possible Solution |
|---|---|
| Capture business layer logic. | Use a Workflow Process. Workflow Process Manager and runtime events capture business layer logic. |
| Use features that a Workflow Process supports but that Workflow Policy does not support. | Use a Workflow Process. A Workflow Process can provide pause, stop, and error handling capabilities. |
| Implement complex comparison logic or flow management. | Use a Workflow Process. A Workflow Process is recommended to develop and deploy, and to perform complex comparison logic and flow management. For example, through IF, THEN, ELSE, or CASE. |
| Call a business service. | Use a Workflow Process. A Workflow Process can call a business service. |
| Use a repeating component request. | Use a Workflow Process. You can set up a Workflow Process from a repeating server component request but not from a Workflow Policy. |
| Repetitive, manual processing. | Use a Workflow Process. A Workflow Process is recommended for repetition, timeliness, and for cross functional routing through a business process. |
| Process an event in a timely fashion. | Use a Workflow Process or a Workflow Policy. A Workflow Process is recommended for repetition, timeliness, and for cross functional routing through a business process. |
| Perform escalations and notifications. | Use a Workflow Process or a Workflow Policy. A Workflow Process is recommended for repetition, timeliness, and for cross functional routing through a business process. |
| Capture data layer logic. | Use a Workflow Policy. Workflow Policy Manager captures data layer logic. Data coming into Siebel CRM in the data layer is not captured in the business layer. This requirement typically indicates a potential candidate for a Workflow Policy. Example data coming into Siebel CRM in the data layer includes data coming through EIM or MQ channels. |
| Do bulk data uploads (where Siebel Enterprise Integration Manager is required). | Use a Workflow Policy. Workflow Policy Manager is recommended if bulk data upload occurs through Siebel EIM. |

| Requirement | Possible Solution |
| --- | --- |
| | |

# Process of Planning Workflow Processes

This process is a step in *Roadmap for Developing Workflow Processes*.

To plan a Workflow Process, do the following tasks:

1. *Determining How to Start a Workflow Process*
2. *Determining the Workflow Process Decision Logic*
3. *Determining Workflow Process Actions*
4. *Determining Error Handling*
5. *Examining Seed Workflow Processes*

If your work in *Process of Analyzing Business Requirements* determined that the Workflow Process is the most appropriate solution, then you can continue planning the Workflow Process. When planning a Workflow Process you determine how to build the Workflow Process, including making design decisions, such as which workflow mode to use, the events to define, the rules to define, actions that the Workflow Process runs, and so on.

# Determining How to Start a Workflow Process

This task is a step in *Process of Planning Workflow Processes*.

During the planning phase of a development effort you can determine if a runtime event, user event, Workflow Policy, or a script starts the Workflow Process. For more information, see *Starting a Workflow Process*.

Before deciding how to start a Workflow Process, consider the advantages and limitations of each configuration that can start a Workflow Process (described in the following subtopics) and then chose the configuration that most closely matches the business requirement.

- *Using a Workflow Policy to Start the Workflow Process*
- *Using an Event to Start the Workflow Process*
- *Using a Script to Start the Workflow Process*
- *Configurations That Start a Workflow Process*

## Using a Workflow Policy to Start the Workflow Process

A Workflow Policy can start a Workflow Process after a database change. If the Workflow Policy conditions are met, then an action occurs. In some situations, the action calls the Workflow Process Manager server component to run a Workflow Process. Processing that a Workflow Policy starts does not occur in real time. The following are typical uses of a Workflow Process:

- EIM batch processing
- Siebel EAI inserts and updates
- Manual changes from the user interface
- Assignment Manager assignments

- Siebel Remote synchronization

## Using an Event to Start the Workflow Process

You can use the following events to start a Workflow Process:

- **Runtime event.** A runtime event occurs if a change occurs in the client or in the business component. Processing that a runtime event starts occurs in real time.
- **User event.** A user event is a unique event that is internal to Siebel Workflow. It starts or resumes a long-running Workflow Process. The User Event business service creates a user event.

You can define an event, from the administrative interface, for one of the following object types:

- Application
- Object
- Business Component

## Using a Script to Start the Workflow Process

A script can start a Workflow Process programmatically as a business service. The Workflow Process Manager server component includes APIs that you can use with a script to start a Workflow Process from an external system. The Object Manager starts a script. You can add a script to one the following object types:

- Application
- Applet
- Business Component
- Business Service

## Configurations That Start a Workflow Process

The following table describes the configurations (Workflow Policy, Event, and Script) that start a Workflow Process, including the limitations for each configuration.

| Configuration | Description | Limitations |
|---|---|---|
| Workflow Policy | Use a Workflow Policy if Siebel CRM must detect and react to data changes that occur outside of the Object Manager. For example, changes that occur in Siebel Remote or Siebel EIM. | A Workflow Policy includes the following limitations:<br>• Making changes requires database downtime<br>• Relatively complex to define |
| Event | Use an event in the following situations:<br>• You must configure a basic entry point for a Workflow Process or a simple custom action. | An event includes the following limitations:<br>• You cannot write script on the object event to reply directly to an event.<br>• Can be more difficult to send the event context to business logic.<br>• Only detects data changes that the Object Manager component makes. |
| Script | Use a script in the following situation:<br>• You want to perform some complex processing before deciding to invoke the Workflow Process. | Invoking a Workflow Process from a script includes the following limitation:<br>• Because a script in written in the Application Object Manager, it cannot respond directly to database layer |

| Configuration | Description | Limitations |
|---|---|---|
| | | events, such as Siebel Remote or EIM changing a record at the database level. |

# Determining the Workflow Process Decision Logic

This task is a step in *Process of Planning Workflow Processes*.

You can configure the decision logic that guides the flow of control in a Workflow Process. For more information, see *Configuring Decision Conditions for a Workflow Process*. Use the following procedure to determine the decision logic that the Workflow Process uses.

## To determine the Workflow Process decision logic

1. Examine the business analysis work you have completed so far. This will help you decide if the business process requires decision conditions and what those conditions are.
2. Map the requirements to the Workflow Process decision logic. The following table describes the configurations that use decision logic in a Workflow Process, and the limitations of each configuration.

| Configuration | Description | Limitations |
|---|---|---|
| Decision step in Workflow Process | A Workflow Process step that determines the flow between alternative branches in the Workflow Process.<br><br>Use a decision step in Workflow Process if you require a simple decision that uses one or more alternative branches in a Workflow Process.<br><br>Each connector that emanates from a decision step can contain one or more decision conditions. If the conditions evaluate to TRUE for the connector, then flow proceeds down the branch that the connector represents. | A conditional expression does not support the following operators:<br><br>o AND<br>o OR<br>o Order of precedence, as determined by parentheses in an equation |
| Scripted Business Service | A script in a business service action step that evaluates a potentially complex set of inputs and returns a simplified output that a decision step can evaluate.<br><br>Use a scripted business service if a decision step cannot meet the decision logic that your business requires. | Reduces readability and increases complexity because decision logic resides a script. |
| Wait Step | Allows you to pause a Workflow Process for an amount of time or until an event occurs. | The Object Manager must call the releasing event. |

| Configuration | Description | Limitations |
|---|---|---|
| | Use a wait step if you must support an escalation that depends on time or a long-running Workflow Process that can last for days or weeks. For example, waiting for a customer reply. | |
| Other Specialized Decision Frameworks | A Workflow Process can directly or indirectly use other decision frameworks. For example, personalization rules, assignment rules, or EAI Dispatch Service.<br><br>Use other specialized decision frameworks if a specialized decision framework is required. For example, if Siebel CRM must assign work to a person according to the expertise that this person possesses. | Limitations vary depending on the decision framework. |

## Using Decision Conditions with a Decision Step

You can use a decision condition to determine the flow that Siebel CRM must take on a branch in a Workflow Process. A decision step can exist with multiple connectors where each connector represents a logical branch. Siebel CRM can evaluate a decision condition for each connector that provides branching. A decision condition can make a comparison between two of the following items:

- Process properties
- Business component fields
- Literal values

A comparison can include the following terms:

- Two values that are equivalent.
- One value exists among a series of other values. For example, child record values, One Must Match, or All Must Match.
- Greater than (>) or less than (<).
- Between or Not Between.
- Null or Not Null.

For an example that uses the Compose Condition Criteria dialog box, see *Defining a Decision Condition for a Decision Step*. For a description of properties that the Compose Condition Criteria dialog box displays, see *Creating a Decision Condition on a Branch Connector*.

# Determining Workflow Process Actions

This task is a step in *Process of Planning Workflow Processes*.

This tasks involves identifying the type of actions that the Workflow Process must perform to meet the business requirements. For more information, see the following subtopics:

- *Data Manipulation in a Workflow Process*
- *Uses and Limitations of Workflow Process Actions*
- *Using the Business Service Step*
- *Using the Siebel Operation Step*

## Data Manipulation in a Workflow Process

A Workflow Process operates on business objects and business components. It references a single business object. In the context of these data layer objects, Siebel CRM creates or updates data while the Workflow Process runs. A Workflow Process can modify the following data:

- Business component data
- Process property data
- Siebel Common Object data

When you create a Workflow Process, a set of predefined process properties are also created. You can think of a process property as a local variable that is active while a Workflow Process instance runs. Siebel CRM can use the process property as input and output to various steps in a Workflow Process. The Process Instance Id is one example of a predefined process property. For more information, see *Using Process Properties*.

## Uses and Limitations of Workflow Process Actions

The following table describes the actions that the Workflow Process can perform, and the limitations of those actions.

| Action | Description | Limitations |
|---|---|---|
| Business Service step | Use a business service step if you must run a potentially complex and reusable set of logic.<br><br>A Workflow Process step that calls a business service method.<br><br>The business service can be predefined or scripted. | Creating and deleting business services can impact system performance. Caching can reduce overhead.<br><br>Incorporating too much logic in a business service can limit re-usability for the business service and can reduce the transparency of the Workflow Process. |
| Siebel operation step | Use a Siebel operation step if you must run simple record operations in the Workflow Process.<br><br>A Workflow Process step that does inserts, updates, and queries on business component data. | A search specification can update multiple records but it cannot get and iterate through a set of records so that subsequent actions for the Workflow Process can run for each record. |

## Using the Business Service Step

A Business Service step can run a predefined or a custom Business Service method. The following items are examples of predefined Business Services:

- Assignment Manager requests
- Notification through the Communications Server
- Server requests
- Integration requests from Siebel EAI

You can write a custom Business Service in Siebel VB or Siebel eScript. If you add a Business Service step, then you must define the Business Service, the Business Service method, input arguments, and output arguments for that step. Siebel CRM passes input arguments in a process property, business component data, or as a literal value.

The following list includes some Business Services that a Workflow Process typically uses:

- Outbound Communications Manager
- Synchronous Assignment Manager Requests
- Server Requests
- Report Business Service
- Audit Trail Engine
- EAI Business Services, such as EAI Siebel Adapter, EAI XML Converter, and related Business Services
- FINS Data Transfer Utilities and Data Validation Manager

For more information, see *Adding a Business Service Step*, and *Predefined Business Services*.

If you require specialized functionality, then you can define a custom Business Service that performs a specific action. You can define a Business Service in the Siebel Repository or in the Administration - Business Service screens of a Siebel CRM application (such as, Siebel Call Center or Siebel Financial Services).

### Using the Siebel Operation Step

The Siebel operation step allows you to configure Siebel CRM to do database operations. For example:

- Query
- Insert
- Delete
- Update
- NextRecord
- Upsert
- PreviousRecord
- QueryBiDirectional

A Siebel operation step references a single business component. After you define a Siebel operation step, you can use the Search Specification property to locate the records that Siebel CRM must use or modify. Examples of a Siebel operation step include creating an activity if the user creates a new service request, or updating a comment field if a service request is open too long. For more information, see *Adding a Siebel Operation Step*.

## Determining Error Handling

This task is a step in *Process of Planning Workflow Processes*.

The planning phase is an appropriate time to plan for how to recover from a failed Workflow Process. Error handling can range from using a simple stop step to defining a separate Workflow Process that handles the error. For more information, see *Handling Errors*.

### To determine error handling

1. Determine what, if any, error handling is required to meet the business requirements.

2. Identify the error handling action that can meet the business requirements.

## Examining Seed Workflow Processes

This task is a step in *Process of Planning Workflow Processes*.

A *seed Workflow Process* is a Workflow Process that comes predefined with Siebel CRM. In many cases, these can be used rather than creating a Workflow Process from scratch. Even if the seed Workflow Process is not perfect for your business requirement, it may be close enough to your requirement to serve as a good base from which to start customizing the seed Workflow Process as required.

Each of the out-of-the-box seed Workflow Processes is explained in the Bookshelf Guide related to that particular area - for example, seed Workflow Processes handling incoming and outgoing emails are described in *Siebel Email Administration Guide* while those Workflow Processes assocciated with Consumer Goods processes are desribed in *Siebel Consumer Goods Guide* .

# Job Roles Used to Develop a Workflow Process

This topic describes job roles that you can use to develop a Workflow Process. Job roles, job titles, and division of labor might vary significantly for your organization. The following job roles are associated with developing a Workflow Process:

- The business analyst considers business requirements for your organization and identifies the business processes to automate.
- The developer creates or modifies a Workflow Process and other objects, business services, and programs in the Siebel CRM repository. Custom objects, business objects and programs can also be defined.
- The workflow administrator uses Siebel Workflow to monitor a Workflow Process in the Siebel client. To activate a Workflow Policy, the workflow administrator creates database triggers in a script and defines them in the Siebel database. The workflow administrator then starts the Siebel Server processes that run the Workflow Process and Workflow Policy. This person is typically a system administrator, database administrator, or someone from the Information Services department.
- Once a Workflow Process or Workflow Policy is defined to execute under certain conditions, the Workflow Process or Workflow Policy will happen automatically based on end-user actions, without any need for the user to know it exists.

# 5 Using the Siebel Workflow Process Development Environment

## Using the Siebel Workflow Process Development Environment

This chapter describes how to use the Siebel Workflow Process development environment. It includes the following topics:

- *Using the Process Designer*
- *Validating a Workflow Process*
- *Deploying a Workflow Process*

## Using the Process Designer

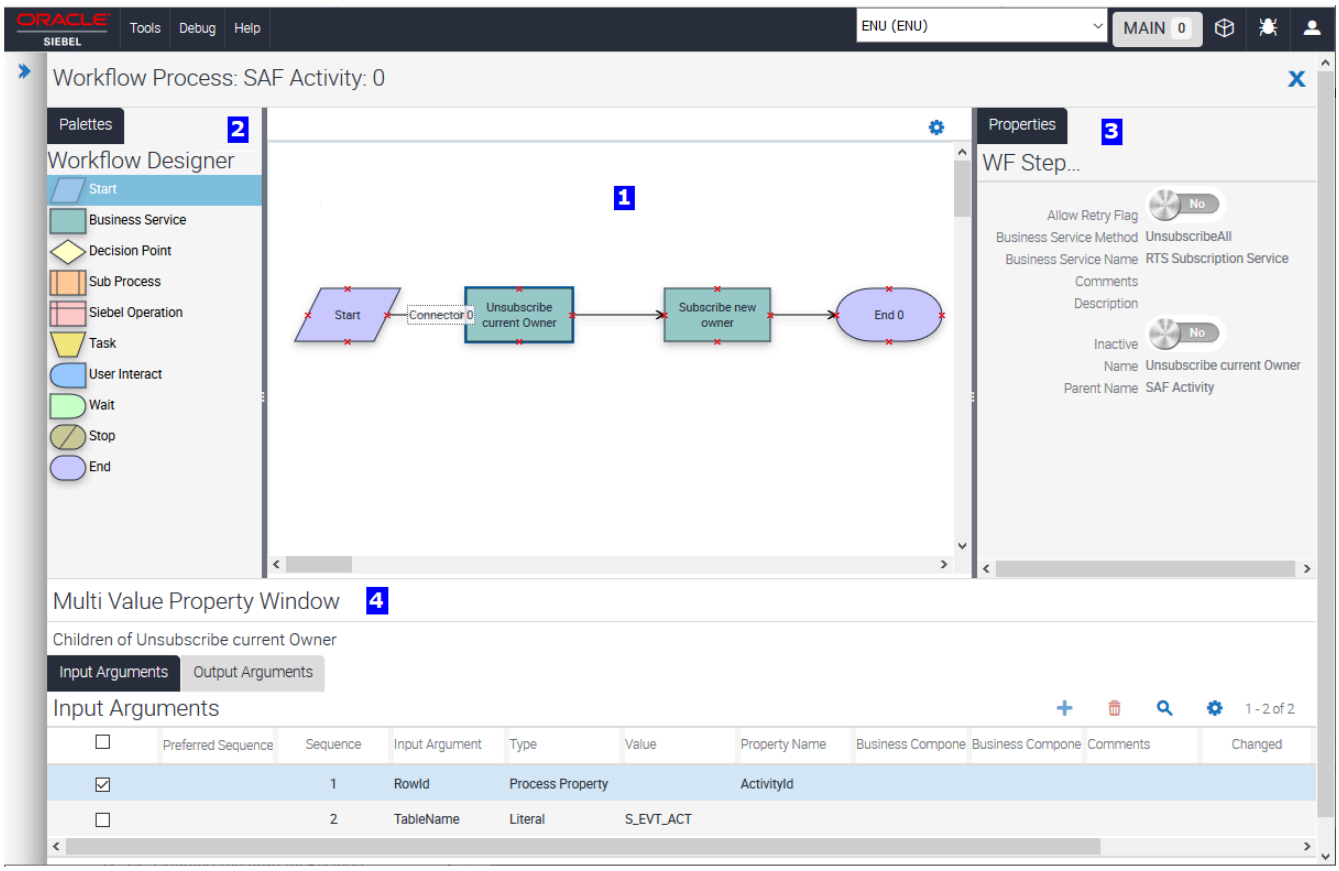
This topic describes the Process Designer and how you use it to define a Workflow Process. As shown in the following figure, the Process Designer contains the following elements:

1. **Canvas.** A work area where you create and define the Workflow Process. For more information, see *Creating a Workflow Process*.

   Click the Workflow Process Canvas Menu (the cogwheel icon) to configure the following options for the canvas: Enable Dynamic Connectors, Connection Points, Show Grid, Snap to Grid, Zoom (in and out).
2. **Palettes pane.** A pane that contains icons that represent the various step types you can add to a Workflow Process. The step types are Start, Business Service, Decision, Sub Process, Siebel Operation, Task, User Interact, Wait, Stop, and End step. To add a step to a Workflow Process, select it from the palette, then drag and drop the step on the canvas. For more information, see *Adding Workflow Process Steps*.

   To increase or decrease the size of the Palettes pane, drag the pane border to resize the pane. To hide the Palettes pane, drag the pane border to reduce the size of the pane so that it is essentially hidden.
3. **Properties pane.** A pane where you to define properties for an individual Workflow Process step or for the overall Workflow Process. For more information, see *Defining a Property for a Workflow Process Step*. The Properties pane is context-sensitive:
   - If you choose a step or connector, then the properties for the selected step or connector are shown in the Properties pane.
   - If no step or connector is selected, then the properties for the Workflow Process are shown in the Properties pane.

   To increase or decrease the size of the Properties pane, drag the pane border to resize the pane. To hide the Properties pane, drag the pane border to reduce the size of the pane so that it is essentially hidden.
4. **Multi Value Property Window (MVPW) pane.** A pane where you to define multiple Process Properties for a Workflow Process and Input and Output Arguments for a Workflow Process step. For more information, see *Using the Multi Value Property Window*.




# Viewing All Workflow Process/Task Steps in the Web Tools Workflow Process/Task Editor

When a Workflow Process or a Task has many steps (more than three thousand) the Workflow Process/Task editor will not show these by default. If your Workflow Process or Task has more than three thousand steps (including connectors) you must add an optional System Preference and set it to an appropriate value to see them all.

Steps to follow to view more steps in the Workflow Process/Task Editor:

1. Login to Web Tools.
2. Open the System Preferences by **Clicking Tools** > **System Preferences** in the Application Toolbar.
3. Query for `WFTaskMaxRowsAllowedInEditor`.

> **Note:** If it does not exist, the default of 3000 will be used to display Steps and Connectors in the Workflow Process/Task Editor. This System Preference is optional, so you do not have to define it to get the default of 3000 steps visible in the editor.

If you want to increase this number add a new row with the values in the following table substituting your number for the default of 3000.

| Name | Value | Description |
|---|---|---|
| System Preference Name | `WFTaskMaxRowsAllowedInEditor` | This is the System Preference that will drive how many steps are visible in the Workflow Process/Task Editor. This System Preference is optional. When it is not defined the default value is 3000 for steps and connectors. |
| System Preference Value | 3000 | The value you want to allow the editor to display. This includes Workflow/Task Steps and connectors. The default is 3000. If you define this to be less than 3000 this number will be ignored. |

**Note:** In your application log file you may see the following if your log level is set to three (INFO):

- **"Max number of steps and branches allowed for the Workflow/Task editors in Web Tools is 3000"** – This shows the current setting. The default is 3000. If you have increased this value that value will be displayed here.
- **"The System Preference – `WFTaskMaxRowsAllowedInEditor` has been set below the default of 3000. This value, 1200, will be ignored so that all steps are visible in the Workflow/Task editors in Web Tools."**– Someone set the value to less than 3000 so the application will ignore this and still display 3000.

# Modifying a Workflow Process

You can edit and modify Workflow Processes when you are in an open Developer Workspace.

## To modify a Workflow Process

1. In the Process Designer, open the Workflow Process you want to modify.

   For more information, see *Editing a Workflow Process*.
2. Modify the Workflow Process by adding and removing steps and connectors or changing step and connector properties.

   For more information, see *Using the Process Designer* and *Adding Workflow Process Steps*.

# Data that a Workflow Process Can Access During Configuration

The data objects that are available for use as you design your Workflow Process originate in the Siebel Repository. Any change in the repository data is immediately available for you to use in a Workflow Process. You can use configuration data (such as a business component field or other repository information) while you create your Workflow Process. Consider the following examples:

- Assume that a List of Values (LOV), such as Account Status, contains the values Gold, Silver, and Bronze. You can use a newly added LOV in a decision condition of your Workflow Process, while you define it.
- If you add a new field to a business component, then this new field is immediately available in the Process Designer.

Runtime data is not available for you to use while you define a Workflow Process, such as an account name, ZIP code, or other transactional data. To use runtime data when you define a Workflow Process, you can make the data available through a process property. If necessary, you can use an unbounded drop-down list to hard code runtime data into your Workflow Process. For more information, see *Using Process Properties*.

# Validating a Workflow Process

Before you deliver a Workflow Process, you must test the Workflow Process to ensure that it does what it is supposed to do without error. Testing a Workflow Process involves validating and then simulating the Workflow Process using the Process Simulator. For more information, see *Testing a Workflow Process*.

# Deploying a Workflow Process

As soon as you deliver a Workflow Process (see *Delivering a Workflow Process*) to the Integration Workspace (or MAIN), it is compiled into the Runtime Repository and is active in that Workspace. If a migration is performed from that Workspace to any downstream Runtime Repository environment, then the workflow will also be active in that environment.

For example, a developer creates a Workflow Process with a runtime event that launches the workflow upon the creation of a new account, and delivers it to the parent Integration Workspace. If that Workspace context is active, then creating a new account will launch the workflow.

If migration to a downstream environment from that Workspace occurs, then creation of a new account in that target Runtime Repository environment will also trigger the workflow.

For more information about Workspaces, see *Using Siebel Tools* .

# 6 Creating a Workflow Process

## Creating a Workflow Process

This chapter describes how to create a Workflow Process. It includes the following topics:

- *Creating a Workflow Process Object Definition*
- *Defining the Primary Business Component*
- *Diagramming a Workflow Process*
- *Adding or Removing a Connector Point*

## Creating a Workflow Process Object Definition

This task is a step in *Roadmap for Developing Workflow Processes*.

If you cannot locate an existing Workflow Process that meets your business requirements, then you should create a new one. For more information, see *Creating the Workflow Process*.

For examples that guide you through the complete steps involved in creating different types of Workflow Processes, from start to finish, see *Example Workflow Processes*.

See also *Object Hierarchy That Workflow Processes Use*.

## Defining the Primary Business Component

If you want your Workflow Process to respond to actions that occur on a particular business component, then that business component must be the primary business component for a business object. For example, to respond to an action on an account, such as the creation of a new account, the following must be true:

1. The Workflow Process definition must specify the Business Object (in this case, Account) on the Workflow Process definition.
2. The Account Business Component must be the primary BusComp in some Business Object, which it is - that is, the Account Business Object.

### Updating Fields in Nonprimary Business Component

You can configure Siebel CRM to update fields that reside in a nonprimary business component.

To update fields that reside in a nonprimary business component, make sure that a link exists between the primary business component and the business component that contains the field that Siebel CRM must update.

# Diagramming a Workflow Process

This task is a step in *Roadmap for Developing Workflow Processes*.

Diagramming Workflow Process steps is an important part of defining a functioning Workflow Process. The Process Designer allows you to create a visual representation of the entire flow, including decision steps and conditional logic. You can define the details for each step while you define each step in the Workflow Designer, or you can define the entire flow, and then define details for each step. For more information, see *Using the Process Designer*.

## To diagram a Workflow Process

1. Select the Start step from the Palettes pane, then drag and drop it on the canvas.

   A Workflow Process must contain only one Start step. For more information, see *Adding a Start Step*.
2. Select one or more other steps from the Palettes pane, then drag and drop them on the canvas.

   A Workflow Process can contain one or more steps that perform an action, such as a business service step, decision step, sub process step, task step, stop step, wait step, or Siebel operation step. For more information, see *About Step Types*.
3. Select an End step from the Palettes pane, then drag and drop it on the canvas.

   A Workflow Process must contain at least one End step. For more information, see *Adding an End Step*.
4. Define the flow or path of the Workflow Process by adding Connectors as follows:
   - (Siebel Tools) Select Connector from the palette, then drag and drop it on the canvas. Select the Connector on the canvas and attach it (or one end of it) to a step, then move the other end of the Connector to connect to the next step in the flow.
   - (Web Tools) Use the mouse to draw a connector line from the connector node on one step to the connector node on another step.
5. Repeat the previous step until every step in the Workflow Process is connected correctly.

   A connector that emanates from some step types can provide conditional logic for the Workflow Process. For more information, see *Adding a Branch Connector*.

# Adding or Removing a Connector Point

> **Note:** This task applies only to Siebel Tools.

The following procedure shows how to add or remove a connector point in Siebel Tools.

## To add or remove a connector point

1. Select the connector or error exception connector.
2. To add a connector, right-click, click the Edit menu and then select the Add Point menu item.
3. To delete a selected connector, right-click, click the Edit menu and then select the Remove Point menu item.

# 7 Adding Workflow Process Steps

## Adding Workflow Process Steps

This chapter describes how to edit a Workflow Process, including how to add steps and connectors to a Workflow Process and define step properties. It includes the following topics:

- *About Step Types*
- *Naming a Workflow Process Step or a Process Property*
- *Editing a Workflow Process*
- *Adding a Start Step*
- *Adding a Business Service Step*
- *Adding a Decision Step*
- *Adding a Sub Process Step*
- *Adding a Siebel Operation Step*
- *Adding a Task Step*
- *Adding a User Interact Step*
- *Adding a Wait Step*
- *Adding a Stop Step*
- *Adding an End Step*
- *Differences Between the End Step and the Stop Step*
- *Adding a Workflow Process Connector*
- *Defining a Property for a Workflow Process Step*

For information about defining a decision condition and adding a branch connector, see *Configuring a Decision Condition*.

## About Step Types

You use the Palettes pane in the Process Designer to define a Workflow Process. For more information on the Process Designer, see *Using the Process Designer*.

The Palettes pane contains the following step types, which you can add to your Workflow Process as required:

- Start - for more information, see *Adding a Start Step*.
- Business Service - for more information, see *Adding a Business Service Step*.
- Decision - for more information, see *Adding a Decision Step*.
- Sub Process - for more information, see *Adding a Sub Process Step*.
- Siebel Operation - for more information, see *Adding a Siebel Operation Step*.

- Task - for more information, see *Adding a Task Step*.
- User Interact - for more information, see *Adding a User Interact Step*.
- Wait - for more information, see *Adding a Wait Step*.
- Stop - for more information, see *Adding a Stop Step*.
- End - for more information, see *Adding an End Step*.

After you define the steps in a Workflow Process, you must add and define (if required) the connectors in the Workflow Process and (optionally) fine-tune the properties for each Workflow Process step. For more information, see the following:

- To add and define Workflow Process connectors, see *Adding a Workflow Process Connector*.
- To define the properties for a Workflow Process step, see *Defining a Property for a Workflow Process Step*.
- To define Workflow Process error exceptions, see *Example of Error Exception Handling* and *Defining an Error Exception Connector to Handle an Update Conflict*.

# Naming a Workflow Process Step or a Process Property

The name that you use for a Workflow Process step or process property must be unique. You cannot use some symbols, such as the period (.), in the name of a process property.

If you add a new step, then the Workflow Designer populates the Name property of the step with a name and number combination that you can change. The step or connector type determines this name. For example:

- Business Service 0 for a business service step
- Siebel Operation 0 for a Siebel operation step

The number differentiates instances of the same type of step or connector that the Workflow Process contains. For example, Business Service 0 and Business Service 1. You should change this default name to something more meaningful to reflect the purpose of the step.

# Editing a Workflow Process

You use the Process Designer to define and modify a Workflow Process. For more information on the Process Designer, see *Using the Process Designer*.

## To edit a Workflow Process

1. In the Object Explorer, click Workflow Process.
2. In the Workflow Processes list, locate and select the Workflow Process you want to modify, then do one of the following as required:
   - (Siebel Tools) Right-click the Workflow Process and select the Edit Workflow Process option.
   - (Web Tools) Select the Workflow Process and then click Edit Workflow Process (the pencil icon). You can also drill down on the Process Name in the Workflow Process list to open the Workflow Process that you want.

> **Note:** If the Workflow Process you want to open is in an editable Workspace, then Edit Task Flow (the pencil icon) will be available, otherwise Preview (the eye icon) will be available to open the Workflow Process in read-only mode.

The Workflow Process opens in the Process Designer along with the following panes: Canvas, Palettes, Properties, and Multi Value Property Window (MVPW).

3. Select the type of step that you want to add from the Palettes pane, then drag and drop it on the canvas.
4. In the Properties pane:
   - In the Name property field, enter a name for the step. When you navigate out of the property pane, the step name will be updated on the canvas. For more information, see *Naming a Workflow Process Step or a Process Property*.
   - (Optional) In the Description property field, enter a value that describes the purpose of the step.
   - Define other properties for the Workflow Process step, as required.

   For more information, see the topic about the Workflow Process Step object type in *Siebel Object Types Reference* .

# Adding a Start Step

A *start step* is a type of Workflow Process step that indicates the starting point (or entry point) of the Workflow Process. A Workflow Process can contain only one start step.

If you want to start a Workflow Process conditionally, however, then define an event or a decision condition on the connector emanating from the start step (not on the start step itself). For example, you can configure the connector that emanates from a start step to do the following:

- **Define the decision conditions that must be met to start the Workflow Process.** For example, to handle an open service request, you can define a condition of `Status=Open` on the connector. For more information, see *Configuring a Decision Condition*.
- **Define a runtime event that starts the Workflow Process.** For example, to create an activity if the revenue for an opportunity is greater than $10,000, you can define a WriteRecord runtime event, and then define a workflow that inserts the activity if the user saves an opportunity that meets the decision condition. For more information, see *Starting a Workflow Process from a Runtime Event*. For an example that uses a runtime event, see *Defining a Workflow Process That Creates an Activity for a Sales Representative*.

For more information, see *Starting a Workflow Process*.

## To add a start step

1. Select the Start step from the Palettes pane, then drag and drop it on the canvas.

2. In the Properties pane:
   - In the Name property field, enter a name for the Start step (this is typically Start). When you navigate out of the property pane, the step name will be updated on the canvas. For more information, see *Naming a Workflow Process Step or a Process Property*.
   - (Optional) In the Description property field, enter a value that describes the purpose of the step.
   - Define other properties for the Start step, as necessary.

   For more information, see the topic about the Workflow Process Step object type in *Siebel Object Types Reference* .
3. (Optional) To define a runtime event that starts the Workflow Process:
   a. Add another step (for example, a Business Service step) to the Workflow Process.
   b. Add a Connector between the Start step and Business Service step, and then define the runtime event on this connector.

      For more information, see *Starting a Workflow Process from a Runtime Event*.

# Adding a Business Service Step

A *business service step* is a type of Workflow Process step that allows you to run a predefined or custom service during the execution of a workflow. The following are examples of predefined business services:

- **Notification.** Siebel CRM uses the Outbound Communication Server business service to send a notification to an employee or a contact.
- **Assignment.** Assignment Manager calls the Synchronous Assignment Manager Request business service to assign an object in a Workflow Process.
- **Server task.** You can use the Asynchronous Server Requests or the Synchronous Server Requests business service to run a server component task.

For a list of some of the more commonly used predefined business services, see *Predefined Business Services*.

To define your own custom business service, you can define a service in the Siebel Repository or the Administration - Business Service view in the Siebel client. You can use Siebel VB or Siebel eScript to define your own custom business service that you call from a Workflow Process.

## To add a business service step

1. Select the Business Service step from the Palettes pane, then drag and drop it on the canvas.
2. In the Properties pane:
   - In the Name property field, enter a name for the business service step. When you navigate out of the property pane, the step name will be updated on the canvas. For more information, see *Naming a Workflow Process Step or a Process Property*.
   - (Optional) In the Description property field, enter a value that describes the purpose of the business service step.
   - In the Business Service Name property field, choose the name of the business service that this Workflow Process should call. The drop-down list contains the business services that are defined in the Siebel

Repository or the Siebel client. For more information about defining a custom business service, see *Integration Platform Technologies: Siebel Enterprise Application Integration* .

> **CAUTION:** When a Workflow Process invokes a custom Business Service, only the server scripts associated with that Business Service are executed. Do not include browser script in the Business Service if the logic in that script should be evaluated during execution of the Workflow Process.

- In the Business Service Method property field, choose the business service method that the Workflow Process should invoke. The choices available for this property depend on the business service you defined.
- Define any input arguments for the business service step as required.

For more information, see the topic about the Workflow Process Step object type in *Siebel Object Types Reference* .

3. In the Multi Value Property Window (MVPW) pane, define input arguments and output arguments for the selected Business Service step.

   For more information, see *Defining a Step Argument in the MVPW*.

# Using the Pass By Reference Feature with a Business Service

You can use the PassByRef user property on a predefined business service. For more information, see *Using the Pass By Reference Feature with a Sub Process*.

## To use the Pass By Reference feature with a business service

1. In the Object Explorer, click Business Service.
2. In the Business Services list, locate the predefined business service you want to modify.

   You cannot use the Pass By Reference feature with a custom business service; you can use it only with a predefined business service.
3. In the Object Explorer, expand the Business Service object, and then click Business Service User Prop.
4. In the Business Service User Prop list, add a user property using values from the following table.

| User Property | Value |
|---|---|
| PassByRef | True |

5. Make sure the Pass By Reference check box for the business service step contains a check mark.

   If you do not do this, then the Workflow Engine returns an error.

# How a Business Service Calls a Shell Script

A Siebel process that runs on Windows or UNIX runs as the user process that starts Siebel Services. If a Workflow Process calls a business service that calls a (Windows or UNIX) shell script, then this shell script runs as the Siebel Service Owner account.

# Adding a Decision Point

A *decision step* is a type of Workflow Process step that evaluates one or more decision conditions to determine the next step that the Workflow Process runs. For example, assume a Workflow Policy starts the Workflow Process, and that multiple violations of a Workflow Policy condition occur during the action interval of the Workflow Monitor Agent. When this Workflow Process runs, the decision step determines the branch to pursue according to the current value of the business component field.

## To add a decision step

1. Select the Decision step from the Palettes pane, then drag and drop it on the canvas.
2. In the Properties pane:
   - In the Name property field, enter a name for the decision step. When you navigate out of the property pane, the step name will be updated on the canvas. For more information, see *Naming a Workflow Process Step or a Process Property*.
   - (Optional) In the Description property field, enter a value that describes the purpose of the decision step.
   - Define other properties for the decision step, as necessary.

   For more information, see the topic about the Workflow Process Step object type in *Siebel Object Types Reference* .
3. (Optuonal) Define a decision condition on a connector emanating from the decision step.

   For more information, see *Creating a Decision Condition on a Branch Connector*.

# Adding a Sub Process Step

A *sub process step* is a type of Workflow Process step that allows you to start a separate Workflow Process from an existing Workflow Process. A Workflow Process can contain one or more sub process steps. You can also define a sub process step that supports Pass By Reference. For more information, see the following:

- *Passing a Process Property In and Out of a Workflow Process Step*.
- *Using the Pass By Reference Feature with a Sub Process*.
- See the topic about the Workflow Process object type in *Siebel Object Types Reference* .

## To add a sub process step

1. Make sure the Workflow Process that the sub process step calls exists.

   An object definition for the Workflow Process that the sub process step calls must exist before you can add a sub process step. If it does not exist, then you must define it. For more information, see *Creating a Workflow Process Object Definition*.
2. Select the Sub Process step from the Palettes pane, then drag and drop it on the canvas.

3. In the Properties pane:
   - In the Name property field, enter a name for the sub process step. When you navigate out of the property pane, the step name will be updated on the canvas. For more information, see *Naming a Workflow Process Step or a Process Property*.
   - (Optional) In the Description property field, enter a value that describes the purpose of the sub process step.
   - In the Subprocess Name property field, select the Workflow Process that the sub process step calls.
   - Define other properties for the sub process step, as necessary.

   For more information, see the topic about the Workflow Process Step object type in *Siebel Object Types Reference* .
4. In the Multi Value Property Window (MVPW) pane, define input arguments and output arguments for the selected Sub Process step.

   For more information, see *Defining a Step Argument in the MVPW* and *Adding an Input Argument on a Sub Process Step*. See also *Recipient Argument Fields*.

To edit the Workflow Process that the sub process references, double-click the sub process step.

# Adding an Input Argument on a Sub Process Step

You can add an input argument to a sub process step. This input argument allows you to populate a process property in the sub process step. For example, Siebel CRM passes the Object Id from the main Workflow Process to the sub process through an input argument. If the sub process references a different business object, then you must send the relevant Row Id of the target object as the Object Id process property of the sub process.

If the sub process creates a child object, then the Object Id that Siebel CRM passes to a sub process must not contain the Object Id of the parent process. If a sub process creates a child object, then it is recommended that the Object Id that Siebel CRM passes to a sub process be null.

For more information about how to add an input argument to a sub process step, see *Defining a Step Argument in the MVPW*.

# Using the Pass By Reference Feature with a Sub Process

If a sub process modifies a large amount of data, then it must also copy a large amount of data. This configuration can result in a negative impact on performance and scalability. You can use the Pass By Reference feature so that Siebel CRM passes a pointer to the data instead of copying the data.

The following figure includes an example Workflow Process that passes a large property set from the Create Siebel Message step to the 11i Process Order step. This figure contains the following steps and connectors:

1. A Start step
2. A Create Search Spec step
3. A Create Siebel Message step
4. An 11i Process Order step
5. An End step
6. The following Connectors: A connecter between step 1 and 2, step 2 and 3, step 3 and 4, step 4 and 5.


If you configure a sub process step to support Pass By Reference, then it is not necessary to map the output argument in the sub process step for the hierarchical properties that Siebel CRM passes. The sub process step overwrites the passed input hierarchical argument. Optionally, you can define the sub process step to modify the passed input hierarchical argument.

### To use the pass by reference feature with a sub process

1. In the Workflow Processes list, choose a Workflow Process that another Workflow Process references as a sub process.
2. In the Properties pane, set the Pass By Ref Hierarchy Argument process property to TRUE.

   In the Worfklow Processes list, the repository definition now includes a check mark for the Pass by Ref Hierarchy property of the Workflow Process that is the sub process.

# Adding a Siebel Operation Step

This topic describes how to add the Siebel operation step. It includes the following topics:

- *Configuring the Operation Property*
- *Using a Siebel Operation Step with a Search Specification*
- *Using a Siebel Operation Step to Update a Field That References a Multi-Value Group*
- *Using a Siebel Operation Step with a Calculated Field*
- *Using a Siebel Operation Step to Traverse a Record Set*
- *How Siebel Operations and Workflow Policy Programs Use Different Object Layers*

The *Siebel operation step* is a type of Workflow Process step that does an operation on a business component, such as insert, update, or query. You define a Siebel operation step for the business component that a business object references. The Workflow Process references this business object. If you configure Siebel CRM to update a business component that this business object does not reference, then you can use the Workflow Process Designer to start a sub process or to associate the business component with the business object. For more information, see *Defining the Primary Business Component*.

## To add a Siebel operation step

1. Select the Siebel Operation step from the Palettes pane, then drag and drop it on the canvas.
2. In the Properties pane:
   - In the Name property field, enter a name for the Siebel operation step. When you navigate out of the property pane, the step name will be updated on the canvas. For more information, see *Naming a Workflow Process Step or a Process Property*.
   - (Optional) In the Description property field, enter a value that describes the purpose of the Siebel operation step.
   - In the Operation property field, choose an operation. For more information, see *Configuring the Operation Property*.

- In the Business Component property field, choose the business component that contains the data that the Siebel operation modifies.
- Define other properties for the Siebel operation step, as necessary.

For more information, see the topic about the Workflow Process Step object type in *Siebel Object Types Reference* .

3. (Optional) In the Multi Value Property Window (MVPW) pane:
   - Define input arguments and output arguments for the selected Siebel Operation step. For example, if you chose Insert or Upsert in step 3, then make sure you add the required arguments in the MVPW pane as follows:
     - In the Field Name field, define the name of the field that Siebel CRM must update.
     - In the Type field, choose an input argument type, and then define other fields, depending on the type.

     For more information, see *Arguments You Can Define for a Process Property*.
   - (Optional) In the MVPW pane, define a search specification as follows:
     i. In the Type field, choose a search specification type, as described in the following table.

| Type | Description |
|---|---|
| Literal | A static string where the runtime value is the same as the value that you define in the Workflow Process Designer. |
| Expression | A Siebel expression that references a runtime variable. Siebel CRM can evaluate the expression only at runtime. |

     ii. In the Search Specification field, enter a search specification.
     iii. If the search specification Type is Expression, then choose a business component.

     For more information, see *Using a Siebel Operation Step with a Search Specification*.

# Using a Siebel Operation Step to Update a Process Property

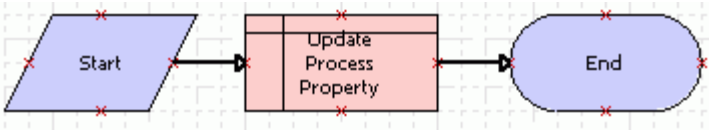
You can define a test Workflow Process that uses a Siebel operation to update a process property. For example, you can use the sum of two business component integer fields that provide input to a decision that resides downstream in the Workflow Process. Siebel CRM uses the sum of the two required fields to update a process property in the Siebel operation step. You can then use this property on a subsequent branch connector. The example in this topic references the Action business object.

## To use a Siebel operation step to update a process property

1. Define a new Workflow Process with the values shown in the following table.

| Property | Value |
|---|---|
| Process Name | Updating a Process Property |
| Business Object | Action |
| Workflow Mode | Service Flow |

   For more information, see *Creating the Workflow Process*.
2. Add the following steps and connectors until your Workflow Process resembles the flow illustrated in the following figure:
   a. A Start step
   b. An Update Process Property Step
   c. An End step
   d. The following Connectors: A connector between step a and b and between step b and c.




   For more information, see *Adding Workflow Process Steps* and *Diagramming a Workflow Process*.
3. Click the canvas, making sure no Workflow Process step or connector is chosen.
4. In the Multi Value Property Window (MVPW) pane, add a new process property with the values shown in the following table.

| Field | Value |
|---|---|
| Name | Custom Process Property |
| Display Name | Custom Process Property |
| In/Out | In/Out |
| Business Object | Action |

   For more information, see *Using Process Properties*.
5. Click the Update Process Property step, and then click the Search Spec Input Arguments tab in the MVPW pane.

6. Add a new record with the values shown in the following table.

| Field | Value |
|---|---|
| Expression Business Component | List of Values |
| Filter Business Component | Query |
| Search Specification | `[Name] = 'Personal' AND [Type] = 'TODO_TYPE'` |
| Type | Literal |

7. Select the Update Process Property step in the Workflow Process Designer, and then click the Output Arguments tab in the MVPW pane.
8. Add a new record with the values shown in the following table.

| Field | Value |
|---|---|
| Property Name | Custom Process Property |
| Type | Expression |
| Value | `[Class Code]` |
| Business Component Name | List Of Values |

9. Validate and then simulate the Workflow Process.

   For more information, see *Process of Testing a Workflow Process*.
10. Implement this configuration in your production Workflow Process.

# Using a Siebel Operation Step with the Object Id Process Property

When Siebel CRM finishes an Insert or Upsert operation, it stores the Row Id of the record that it created. It stores this Row Id in the Siebel Operation Object Id process property. It passes the Object Id of the Workflow Process to the Siebel operation step. It is not necessary to define a search specification unless Siebel CRM must update child records. For example, if a Workflow Process:

- References the Service Request business object, then it must update the service request but it is not necessary to define a search specification.

- Must update activity records for the service request, then you can enter a search specification that queries for the activity that Siebel CRM must update. If you do not do this, then the update step updates every activity that Siebel CRM associates with the service request.

If Siebel CRM runs a Siebel operation, then the Object Id cannot be null unless it inserts this Object Id into the primary Object Id. If the Workflow Process does not contain an Object Id, then the Siebel operation step returns an error.

The Siebel Operation Object Id process property returns the following values if a query operation modifies child records:

- The Row Id if one record matches.
- Null or no value if no records match.
- An asterisk (*) if multiple records match. It provides this result to distinguish from a value that returns a unique record or no record. Typically, a unique record matches or no records match. In most situations, multiple records do not match and Siebel CRM returns an asterisk.

The following option returns the Row Id of a matching row:

only

The Insert, Update, and Upsert operations update the Siebel Operation Object Id process property of the Row Id for the record.

# Using a Siebel Operation Step with the Upsert Operation

The Upsert Operation can do an update or insert operation depending on whether records exist in the Siebel database. Hence the name, Upsert. For example, assume that a search specification on a Siebel operation step queries the Siebel database and returns one of the following results:

- **One or more records.** Upsert updates these records according to the Workflow Process configuration.
- **No records.** Upsert inserts a record according to the Workflow Process configuration.

For example, assume the following occurs:

1. Siebel CRM navigates the user to a view where they can can click a check box to create a new contact.

   The Yes default for the check box causes Siebel CRM to create a new contact.
2. The user enters the contact information.
3. The user navigates away from the view, and then returns to the view.

   When the user returns to the view, Siebel CRM might have created another contact. Siebel CRM must determine if the contact already exists. If the contact exists, then it must update the contact. If the contact does not exist, then it must create a new contact.

For another example, assume a Workflow Process runs in the background at midnight, processing orders that an external system sends to Siebel CRM:

- If the order does not exist, then Upsert creates a new record.
- If the order already exists but must be updated, then Upsert updates the record.

# Configuring the Operation Property

If you configure this step to update or insert a field that contains a dependency, then you must make sure the field is valid. For example, if a service request process and a Workflow Process update the area and sub area fields, then you must make sure the values chosen for the subarea field are valid for this area.

If your configuration uses a DB2 for z/OS database, and if you must use a Siebel operation step to query this database, then it is recommended that you set the Operation property to QueryBiDirectional and not to Query. Using QueryBiDirectional avoids leaving an open thread on DB2 for z/OS.

# Using a Siebel Operation Step with a Search Specification

You can use a search specification to filter the business component records that a Siebel operation modifies. For example, if a Workflow Process for the Account object must only update opportunities that possess a lead quality of Poor, then you can define a search specification that only returns these opportunities.

> **CAUTION:** Define a search specification for a Siebel operation as efficiently as possible so that the search returns the smallest possible set of rows. A search specification that returns a large set of rows can severely degrade performance.

## Adding a Literal Search Specification

If the search specification is of type Literal, then Siebel CRM interprets it literally. For example, `[Status] LIKE '*Open*'`. A search specification of type Expression allows you to create a search specification dynamically. For example, consider the following search specification:

```
"[Contact Id] = ' " + [&New ID] + " ' "
```

If the New ID process property is 1-ABC at runtime, then Siebel CRM interprets this search specification to the following value:

```
[Contact Id] = '1-ABC'
```

## Referencing a Business Component Field in a Search Specification

Each side of an expression can reference a business component field:

- The Filter Business Component defines the business component on one side - which is Account in the example shown in the following table.
- The Expression Business Component defines the business component on the other side - which is Contact in the example shown in the followign table.

| Field | Value |
|---|---|
| Filter Business Component | Account |
| Expression Business Component | Contact |

| Field | Value |
|---|---|
| Expression | `"[Id] = [Account Id]"` |

Siebel CRM uses the following format to evaluate this expression:

```
"[Account.Id] = [Contact.Account Id]"
```

## Using Compound Expressions and Substitutions in a Search Specification

You can use compound expressions and substitutions in a search expression. For example, consider the following generic format:

```
"([Field1] > '" + [&Process Property Name] + "') OR ([Field2] IS NULL) OR ([Field3]
IS NULL)"
```

Siebel CRM translates this format to the following value:

```
([Field1] > 'value') OR ([Field2] IS NULL) OR ([Field3] IS NULL)
```

An expression can include a literal representation of a business component field. For example:

```
"([Open] > '" + [&dFromDate] + "') OR ([Open] IS NULL)"
"([Open] > '" + [&dFromDate] + "') AND ([Status] IS NULL)"
```

where `dFromDate` is a custom process property

For more information about using a process property as a substitution variable, see *Referencing a Process Property*.

# Using a Siebel Operation Step to Update a Field That References a Multi-Value Group

A Siebel operation step can indirectly update a field that references a multi-value group. For example, assume Siebel CRM must update the Account Team business component field. This field references a multi-value group, so you cannot use a Siebel operation step to update it. You can define a business component named Account Team, and then associate it with the Account business object. You can then choose Account Team as the business component that the Siebel operation step updates. For more information about multi-value groups, see *Configuring Siebel Business Applications* .

# Using a Siebel Operation Step with a Calculated Field

A Siebel operation step cannot update a calculated field because, typically, the calculated field requires values from the fields of another business component. Instead, you can use an expression to perform calculations.

# Using a Siebel Operation Step to Traverse a Record Set

To traverse a record set, you can use the NextRecord, PrevRecord, and QueryBiDirectional operations on the Siebel operation step in conjunction with the Update, Query, and Insert operations. You can use the following operations to traverse the records of a child business component of the business object that the Workflow Process references:

- **NextRecord.** Changes the active row of the business component to the next record.
- **PrevRecord.** Changes the active row to the previous record.

You must configure Siebel CRM to perform a query on the business component before it uses another Next Record or Previous Record operation. If the Workflow Process traverses active records or if it uses the PreviousRecord operation, then use the QueryBiDirectional operation. Otherwise, use the Query operation.

You cannot use NextRecord, PrevRecord, and QueryBiDirectional to traverse records of the primary business component. These operations traverse records only for a child business component.

## Stopping a Traverse a Record Set Operation

You set the NoMoreRecords output argument in the Siebel operation to TRUE. Siebel CRM uses this argument when the Siebel operation attempts to read the next record but finds that no more records exist in the record set to traverse. You set NoMoreRecords to TRUE:

- In the forward direction for Next Record
- In the backward direction for Previous Record

You can assign NoMoreRecords to a process property. You can use it in conjunction with a decision step to exit the loop after navigating through every record in the record set. To support this output argument, the Process Designer displays an Output Argument column in the Output Arguments tab of the Siebel operation step.

## Counting the Number of Records Siebel CRM Gets or Updates

You can use the NumAffRows (Number of Affected Rows) output argument with the following operations:

- **Query.** NumAffRows returns the number of rows that Siebel CRM gets as a result of the query.
- **Update or Upsert.** NumAffRows returns the number of rows that Siebel CRM updated.

# How Siebel Operations and Workflow Policy Programs Use Different Object Layers

The Siebel operation step uses a different object layer than a Workflow Policy program to update data. For example:

- Assume a Workflow Policy calls a Workflow Policy program that updates a service request. This configuration proceeds through the data layer where the state model does not apply.
- Assume a Workflow Policy calls a Workflow Process that includes a Siebel operation step that updates a service request. This configuration proceeds through the object layer, where the state model does apply.

When determining to use a Siebel operation or Workflow Policy program, you must consider how these operations interact with the various object layers.

# Adding a Task Step

A *task step* is a type of Workflow Process step that starts a task from a Workflow Process. If a Workflow Process calls a task, then Siebel CRM does the following:

- Adds this task to the user Inbox.
- Sets the Workflow Process state to waiting until the user finishes this task.
- When the task finishes, Siebel CRM resumes the Workflow Process at the next step that is immediately downstream from the task step.

## To add a task step

1. Make sure the task that the Workflow Process calls is already defined.

   For information about defining a task, see *Siebel Business Process Framework: Task UI Guide* .
2. In the Workflow Process, make sure that the Workflow Mode property is set to Long Running Flow.
3. Select the Task step from the Palettes pane, then drag and drop it on the canvas.
4. In the Properties pane:
   - In the Name property field, enter a name for the task step. When you navigate out of the property pane, the step name will be updated on the canvas. For more information, see *Naming a Workflow Process Step or a Process Property*.
   - (Optional) In the Description property field, enter a value that describes the purpose of the task step.
   - In the Task Name property field, select the task that this Workflow Process must run.

     After you define a task step in the Workflow Process Designer and set the Task Name property, you can double-click the task step to open the Task Editor.
   - Set the Inactive property to FALSE.
   - Define other properties for the task step, as necessary.
5. In the Multi Value Property Window (MVPW) pane, add new input, output, and recipient arguments as required.

   For more information, see *Using Process Properties*, and *Siebel Business Process Framework: Task UI Guide* .

# Adding a User Interact Step

The *user interact step* is a type of Workflow Process step that allows you to control the flow of Siebel views that Siebel CRM displays in the client. It can guide the user through a flow of Siebel views according to user interactions or according to a defined set of actions. Siebel CRM can modify this flow as the business environment changes.

The user interact step can use a process property as an input argument. It can use this argument to dynamically set view names in an interactive Workflow Process.

## To add a user interact step

1. Select the User Interact step from the Palettes pane, then drag and drop it on the canvas.
2. In the Properties pane:

- In the Name property field, enter a name for the user interact step. When you navigate out of the property pane, the step name will be updated on the canvas. For more information, see *Naming a Workflow Process Step or a Process Property*.
- (Optional) In the Description property field, enter a value that describes the purpose of the user interact step.
- In the User Interact View property field, select the view that this Workflow Process must show in the client.

  At runtime, Siebel CRM displays this view in the client.
- Define other properties for the user interact step, as necessary.

For more information, see the topic about the Workflow Process Step object type in *Siebel Object Types Reference* .

3. If the user interact step requires conditional logic, then define a decision condition.

   You can define a decision condition on one or more connectors that emanate from the user interact step. For more information, see *Configuring a Decision Condition*.
4. If the Workflow Process is interactive, then make sure the branch that emanates from the user interact step is associated with a runtime event.

   For more information, see *Defining a Runtime Event for a User Interact Step*.

# User Interact Step Behavior

A user interact step behaves as follows:

- Sends a request to the Siebel Web Engine to create the view, and then display this view in the client. The Siebel Web Engine can create only one view at a time. You cannot combine a user interact step with another action, such as displaying a message box or simultaneously creating another view. You cannot simultaneously use a user interact step and start a task.
- Waits in the user session memory for a runtime event to resume processing. If a runtime event is not defined, then the Workflow Process resumes.
- If the user manually navigates out of the view that the user interact step displays, then the Workflow Process remains in the user session memory. Siebel CRM deletes this Workflow Process instance from memory when it terminates the user session or if it starts another Workflow Process in the same user session.
- If not already in the run state, then the user interact step queries the primary business component of the business object that the Workflow Process references. This query searches for a record where the Row Id matches the value that the Object Id property contains.
- If the user interact step includes conditional logic, then Siebel CRM resumes the Workflow Process after the user interact step finishes. It resumes the Workflow Process only if the record ID of the runtime event that is registered in the decision condition matches the value that the Object Id process property contains.

# User Interact Step Restrictions

The following restrictions apply if you define a user interact step:

- A Workflow Process that runs in the Workflow Process Manager server component must not contain a user interact step. If the workflow runs in background mode or in batch mode, then it cannot include a user interact step.

- Siebel CRM supports the user interact step only if a script or a runtime event starts the Workflow Process, and only if it runs this Workflow Process locally in the Application Object Manager.
- You cannot use the user interact step to display or start the Search Center.
- The user interact step or any activity that starts in the background must not interfere with the work that the user is performing.

## Defining a Runtime Event for a User Interact Step

A branch that emanates from a user interact step in an interactive Workflow Process must be associated with a runtime event. An error occurs during validation if each of the following statements are true:

- The Workflow Mode property of the Workflow Process is set to Interactive Flow.
- The Workflow Process contains a user interact step and a runtime event is not defined on the outgoing branch.

To avoid this error, if your configuration does not require the runtime event in the user interact step, then you can change the Workflow Mode property of the Workflow Process to 7.0 Flow. For more information, see *Setting the Workflow Mode Property*.

## Configuring a User Interact Step to Set a View Name

You can associate the name of a view with a process property so that Siebel CRM can set this view name dynamically at runtime.

### To configure a user interact step to set a view name

- Enter the following string in the User Interact View property of a user interact step:
  `[&ProcessPropertyName]`

  The Workflow Engine recognizes this string and assigns the view name at runtime.

# Adding a Wait Step

The *wait step* is a type of Workflow Process step that pauses a Workflow Process for a specific amount of time or until an event occurs. You can pause a Workflow Process instance for seconds, minutes, hours, or days.

If a Workflow Process includes a wait step, then it is persistent, by default.

You can use the wait step for testing and development. Unlike the Siebel operation step, you can use the wait step without affecting business component data. For an example, see *Defining a Runtime Event in a One-to-Many Relationship*.

### To add a wait step

1. In the Workflow Process, set the Workflow Mode property to Interactive Flow. For more information, see *Setting the Workflow Mode Property*.

2. Select the Wait step from the Palettes pane, then drag and drop it on the canvas.
3. In the Multi Value Property Window (MVPW) pane, define input arguments and output arguments for the wait step.

   For more information, see *Using Process Properties*.

   If you define a duration argument that is greater than 60 seconds, then you can use minutes or a larger unit of measure so that Siebel CRM refreshes business component data. If you use minutes or higher, then Siebel CRM resumes the Workflow Process from the Workflow Process Manager. If you use any duration other than seconds, then Siebel CRM considers the Workflow Process as persistent.

## Resuming a 7.0 Workflow Process

If you set the Workflow Mode property of a Workflow Process to 7.0 Flow, then you can set the SleepMode input argument to one of the following values to determine how the wait step resumes the Workflow Process:

- **Local.** Siebel CRM resumes the Workflow Process in the session that started this Workflow Process.
- **Remote.** The Workflow Process Manager server component on the Siebel Server resumes the Workflow Process. Siebel CRM runs a component request for the Workflow Process Manager according to the time that the Duration input argument of the wait step specifies.

If you set the Workflow Mode property of a Workflow Process to something other than 7.0 Flow, then Siebel CRM ignores the SleepMode input argument. For more information, see *Setting the Workflow Mode Property*.

## Setting the Processing Mode Property of a Wait Step

If you configure Siebel CRM to use SetFieldValue to start a Workflow Process, then it is recommended that you set the Processing Mode property to Local Synchronous. SetFieldValue can occur without committing data. Setting the Processing Mode to Remote Synchronous or Remote Asynchronous when using SetFieldValue might result in a Workflow Process that runs *out of process,* which means that Siebel CRM runs in a completely different thread and probably at a different time. In this situation, the Workflow Process cannot access data that Siebel CRM has not committed. It cannot access all the current runtime data. For more information, see *Server Requests Business Service*.

# Adding a Stop Step

The *stop step* is a type of Workflow Process step that Siebel CRM uses to display an error message in the client, and then terminate a Workflow Process.

If Siebel CRM uses a stop step in a sub process, then this stop step stops the sub process and it also stops the parent Workflow Process. It is not necessary to define a stop step in the parent workflow to stop the sub process.

## To add a stop step

1. Select the Stop step from the Palettes pane, then drag and drop it on the canvas.
2. In the Properties pane:
   - In the Name property field, enter a name for the stop step (this is typically Stop). When you navigate out of the property pane, the step name will be updated on the canvas. For more information, see *Naming a Workflow Process Step or a Process Property*.

- (Optional) In the Description property field, enter a value that describes the purpose of the stop step.
- In the Error Code property field, choose an error code.

  If the predefined error messages that the Error Code property of the stop step displays do not meet your requirements, then you can define a custom error message on the stop step. For more information, see *Defining a Custom Error Message on a Stop Step*.
- In the Error Message property field, type in the error message.
- Define other properties for the stop step, as necessary.

For more information, see the topic about the Workflow Process Step object type in *Siebel Object Types Reference* .

3. In the Multi Value Property Window (MVPW) pane, define input arguments for the stop step.

   The input arguments for a stop step contain the substitution variables that Siebel CRM displays in the error message. A percent sign (%) identifies a substitution variable. To define the substitution value, you enter it in the Name field. For more information, see *Arguments You Can Define for a Process Property*.

# How Siebel CRM Handles the Stop Step

The following table describes how Siebel CRM handles the stop step.

| How Called | Where the Workflow Process Runs | Work That the Workflow Process Manager Performs |
|---|---|---|
| A Workflow Policy calls a Workflow Process that contains a stop step. | Not applicable | Exits and writes an error message to the log file. |
| A script or a runtime event calls a Workflow Process that contains a stop step. | Workflow Process Manager Object Manager | Writes an error message to the log file. |
| A script or a runtime event calls a Workflow Process that contains a stop step. | Application Object Manager | Displays an error message in the client. |

# Calling a Workflow Process that Includes a Stop Step

It is recommended that you use the stop step only in a Workflow Process that a script calls. For example, assume a Workflow Process includes a stop step that displays a custom error message. When Siebel CRM runs this Workflow Process, the custom error message includes stack information. Siebel CRM cannot suppress this stack information.

Assume a Workflow Process that a script starts contains an end step that defines the error message in a process property. Assume this Workflow Process encounters an error, and that the subsequent step is an end step that does not display messages. Siebel CRM returns control to the script. This script examines the value that the process property contains. It then uses the following function to display the message:

`RaiseErrorText`

The error dialog displays the error text but it does not include Workflow Process or stack trace information.

# Defining a Custom Error Message on a Stop Step

This topic describes how to define a custom error message on a stop step.

## To define a custom error message on a stop step

1. Add a stop step to your Workflow Process as shown in *Adding a Stop Step*.
2. Select the stop step on the canvas and in the Properties pane, set the Error Code property to the following value:

   **`WF_ERR_CUSTOM_1`**

   The Error Message property is set to `%1`, by default.
3. In the Multi Value Property Window (MVPW) pane:
   - Add a new input argument for the stop step. For more information, see *Arguments You Can Define for a Process Property*.
   - Set the Name field to the substitution variable that appears in the Error Message property in the Properties pane. In this situation, this value is `%1`.
   - Type in the custom text message in the Value field.

     At runtime, Siebel CRM replaces the `%1` substitution variable with the value that you enter in the Value field.

## Defining Multiple Custom Error Messages for a Stop Step

You can use multiple customizable codes in the Error Code property of a stop step. These codes use the following format:

**`WF_ERR_CUSTOM_x`**

Each **`WF_ERR_CUSTOM_x`** code is unique. You can use each of these error codes only one time, and for a specific purpose. If you must display multiple custom error messages, then use **`WF_ERR_CUSTOM_2`**, **`WF_ERR_CUSTOM_3`**, and so on, instead of using `%1`, `%2` for the same **`WF_ERR_CUSTOM_x`**.

# Adding an End Step

An *end step* is a type of Workflow Process step that specifies when a Workflow Process ends. It also provides one last chance to save output arguments to a process property. A Workflow Process must contain at least one end step.

## To add an end step

1. Select the End step from the Palettes pane, then drag and drop it on the canvas.
2. In the Properties pane:

- In the Name property field, enter a name for the end step (this is typically End). When you navigate out of the property pane, the step name will be updated on the canvas. For more information, see *Naming a Workflow Process Step or a Process Property*.
- (Optional) In the Description property field, enter a value that describes the purpose of the end step.
- Define other properties for the end step, as necessary.

For more information, see the topic about the Workflow Process Step object type in *Siebel Object Types Reference* .

3. In the Multi Value Property Window (MVPW) pane, define output arguments for the end step.
For more information, see *Using Process Properties*.

# Differences Between the End Step and the Stop Step

The stop step sets the state for the Workflow Process to `In Error`. The end step sets the state to `Completed`. It is important to consider this situation if you use the Workflow Monitor Agent to start a Workflow Process.

If the Ignore Errors parameter of the Workflow Monitor Agent is set to False, and if a Workflow Process includes:

- **A Stop step**, then Workflow Monitor Agent exits with an error.
- **An End step**, then Workflow Monitor Agent does not exit with an error.

# Adding a Workflow Process Connector

You use Workflow Process connectors to define the flow or path of the Workflow Process. To define the properties of connector in a Workflow Process, you use the Properties pane in the Process Designer.

## To add a Workflow Process connector

1. Go to the Workflow Processes list and locate and select the Workflow Process you want to modify.
2. Define the flow or path of the Workflow Process by adding connectors as follows:
   - (Siebel Tools) Select Connector from the palette, then drag and drop it on the canvas. Select the Connector on the canvas and attach it (or one end of it) to a step, then move the other end of the Connector to connect to the next step in the flow.
   - (Web Tools) Use the mouse to draw a connector line from the connector node on one step to the connector node on another step.
3. Select the connector you added in the previous step and in the Properties pane:
   a. Enter or modify the connector name as required.
   The connector name must be unique in the Workflow Process. If it is not unique, then you cannot save the Workflow Process.
   b. Set the Type property.
   In most situations, this value is Condition (if defining a decision condition on the connector) or Default (if using this connector as an exit route). You can set other values as required. For more information about connector properties, see the topic about the Connector object type in *Siebel Object Types Reference* .

If you define multiple branches on a step, then see *Adding Multiple Branches to a Single Workflow Process Step* for caution information.

c. Enter comments.

4. Repeat steps 2 and 3 until every step in the Workflow Process is connected and defined correctly.

A connector that emanates from some step types can provide conditional logic for the Workflow Process. For more information, see *Adding a Branch Connector*.

For an example showing how to define a runtime event on a connector, see *Starting a Workflow Process from a Runtime Event*.

# Defining a Property for a Workflow Process Step

The properties of a Workflow Process step can include input and output arguments, search specifications, operations, and so on. To define some of these properties, you can use the WF Steps list or the Properties pane and the Multi Value Property Window (MVPW) pane in the Workflow Process Designer. You can define each step property for only one Workflow Process step at a time.

## To define a property for a Workflow Process step

1. With the Workflow Process Designer open, click the step whose properties you must define.
2. Use the Properties pane to define properties for the selected step.

   For more information, see *Defining Properties for Workflow Process Steps*.
3. Use the MVPW pane to define input arguments, output arguments, and search specifications.

   For more information, see *Using the Multi Value Property Window* and *Defining Properties and Arguments for Workflow Process Steps* .

# 8 Manipulating Data in a Workflow Process

## Manipulating Data in a Workflow Process

This chapter describes how to configure a Workflow Process to manipulate and process data. It includes the following topics:

- *Using Process Properties*
- *Passing Data to and from a Workflow Process*
- *Configuring Decision Conditions for a Workflow Process*
- *Accessing Data from a Runtime Event in a Workflow Process*
- *Using the Timestamp*

For information on manipulating data validation rules, see *Siebel Order Management Infrastructure Guide* .

## Using Process Properties

This topic describes how to use process properties. It includes the following topics:

- *Overview of the Process Property*
- *Arguments You Can Define for a Process Property*
- *Using the Multi Value Property Window*
- *Using Predefined Process Properties*
- *Defining a Custom Process Property*
- *Using Process Properties to Manipulate Data*

### Overview of the Process Property

A *process property* is a container that stores a value that the Workflow Process gets from the Siebel database or that it derives before or during processing. The value that a process property contains can be used in the following ways:

- **To pass information between objects.** For example, between steps in a Workflow Process, between a Workflow Process and a sub process, or between a Workflow Process and a business service. You define a process property as an input argument or output argument on a Workflow Process step.
- **To make decisions.** You can base a decision branch on the value that a process property contains.
- **In expressions.** You can use the value that a process property contains in an expression.

When a Workflow Process finishes, the final value of each process property is available as a separate output argument that Siebel CRM can pass to another object. For more information about:

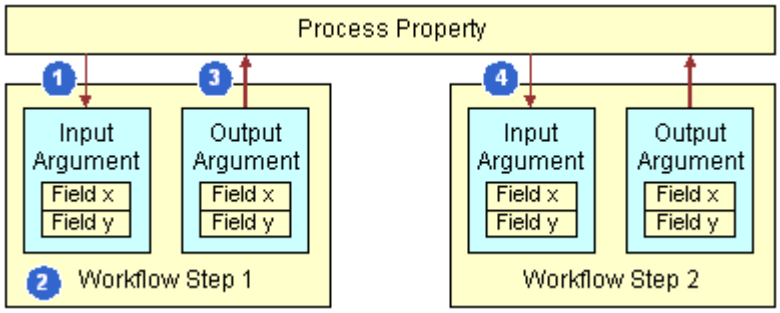
- Arguments that you can use with a process property, see *Process Property Fields and Arguments*.

- Property sets, see *Integration Platform Technologies: Siebel Enterprise Application Integration* .

## Process Property and the Property Set

A property set is a hierarchical structure that contains name and value pairs, known as properties, at each level in the hierarchy. The process property stores data that is associated with the property set. A Workflow Process uses a property set to send data to and from a Workflow Process step.

The following figure describes how Siebel CRM uses a process property in a Workflow Process.




As illustrated in this figure, Siebel CRM typically does the following when it uses a process property in a Workflow Process:

1. The process property sends data to Workflow Step 1 as an input argument.
2. Workflow Step 1 modifies the data according to the configuration for Step 1.
3. An output argument on Workflow Step 1 sends data from the step to the process property.
4. An input argument on Workflow Step 2 brings the data that Step 1 uses into Workflow Step 2, where the Workflow Process can use it according to the internal configuration that is defined for workflow Step 2.

You can define the process property and step arguments in the Multi Value Property Window (MVPW) pane. For more information, see *Using the Multi Value Property Window*.

## How Siebel CRM Sets the Starting Value of a Process Property

The following table describes how Siebel CRM sets the starting value of a process property when a Workflow Process starts.

| Process Property (Type) | Start Value |
|---|---|
| A process property of type string, number, or date and that includes an In/Out type of In or In/Out. | The value that the input property with the same name contains, if one exists. |
| A hierarchy process property that includes an In/Out type of In or In/Out. | The value that child input property sets contain. |
| A process property that contains a Default String. | The value that the Value property of the input property set contains. |

## How Siebel CRM Sets the Ending Value of a Process Property

The following table describes how Siebel CRM sets the ending value of a process property when a Workflow Process ends.

| Process Property (Type) | End Value |
|---|---|
| A process property that is of type string, number, or date, and that includes an In/Out type of In or In/Out, | Stored as a property in the output property set. |
| A hierarchy process property that includes an In/Out type of In or In/Out, | Stored as a child property set. |
| A process property with the name `[Value]` is defined.<br><br>You can also use the process property `[Value]` to send binary data in a sub process step. Siebel CRM can set the starting value of the input argument `[Value]` of the sub process with data that resides in the main process, and then the sub process step can get this data. For more information, see *Passing Data to and from a Workflow Process*. | Stored in the Value property of the output property set. |

# Arguments That You Can Define for a Process Property

An argument is a part of the process property hierarchy that allows you to define values for fields. An argument field is a variable on an argument that allows you to define a value that determines the parameters of a process property.

The following table describes the different type of arguments that you can define for a process property.

| Argument | Description |
|---|---|
| Input Argument | Brings data into a Workflow Process step. |
| Output Argument | Sends data out of a Workflow Process step. |
| Search Specification | Defines a search specification for a Siebel operation step. |
| Recipient | Defines a recipient for a task step or a sub process step. |

A Workflow Process step can include several types of arguments. The type of step determines the arguments that you can define. The Type field for an argument affects other fields in the argument that you must define. For more information, see *Defining a Step Argument in the MVPW*.

## Input Arguments of the Business Service Step, Sub Process Step, and Wait Step

An input argument allows you to define the value for a field that Siebel CRM must send to the business service method. Many methods require an input argument. For more information, see *Input Argument Fields*.

### Output Arguments of the Business Service Step, Sub Process Step, and Siebel Operation Step

An output argument can contain the result of work that the business service method performs. Siebel CRM can store an output argument in a process property. Siebel CRM cannot use a calculated field as a value for an input argument or an output argument. If you must use a calculated value, then use an expression. For more information, see *Output Argument Fields*.

### Process Properties of the Business Service Step

Siebel CRM can use the process property of a business service step as an input argument or output argument for a business service method. The Data Type field of the process property must contain one of the following values:

- Hierarchy
- Integration object

If Siebel CRM must call a Workflow Process through a business service, then you can map the data that the input and output property sets contain to or from a process property. This configuration is useful if Siebel CRM must run a Workflow Process from a script.

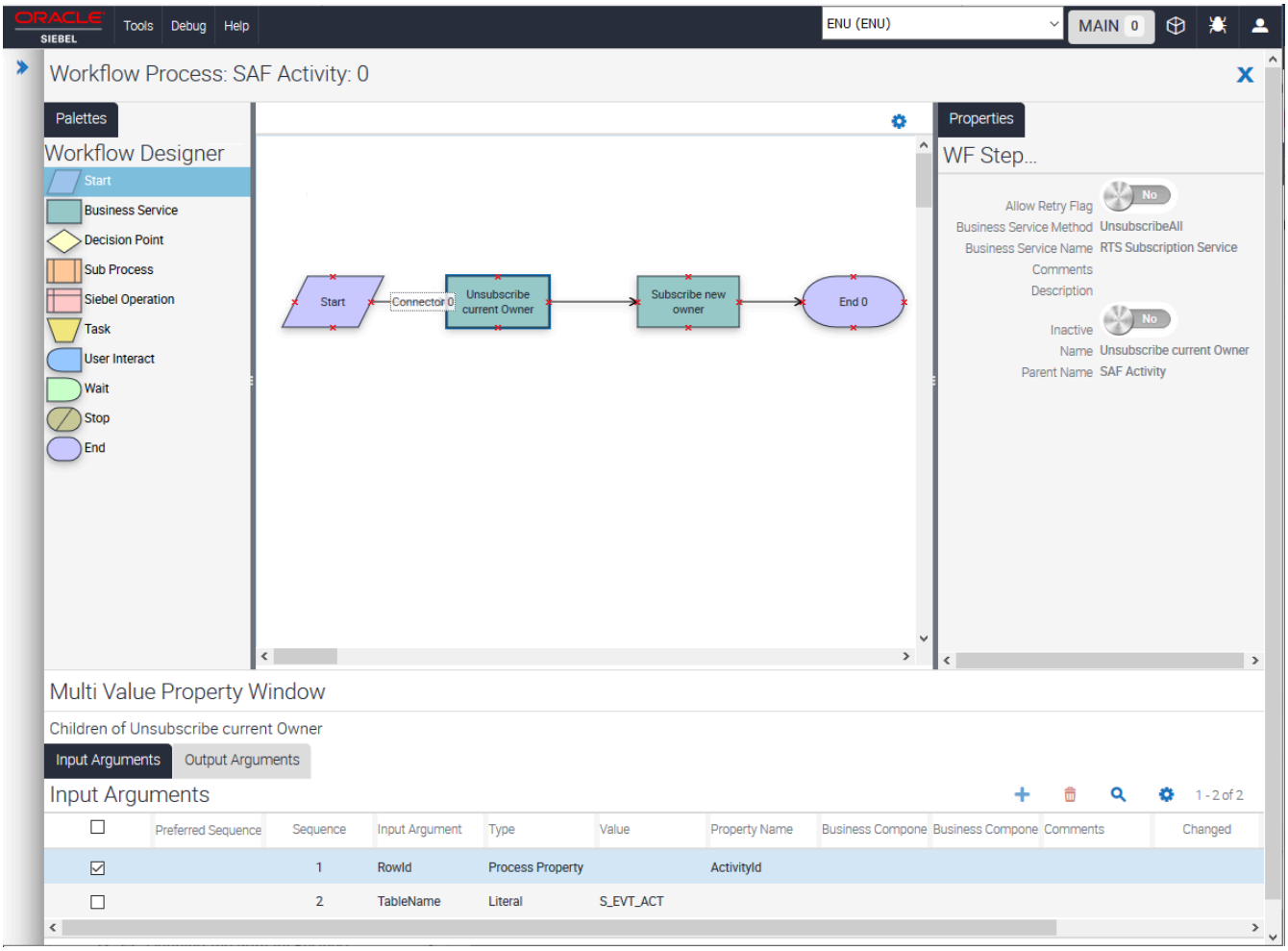
## Using the Multi Value Property Window

The tabs described in the following table appear in the Multi Value Property Window (MVPW) pane. You use the MVPW pane to define the Process Properties, Input Arguments and Output Arguments for a Workflow Process.

For more information about how process properties and step arguments (input/output) work with one another, see *Process Property and the Property Set*.

| Tab | Description | How to Access the Tab |
|---|---|---|
| Process Properties | Stores values that Siebel CRM applies to the entire Workflow Process. | Click anywhere on the Process Designer canvas, making sure not to select a step or connector, and the Process Properties tab appears in the MVPW pane. |
| Input Arguments | Communicates information about input arguments between a process property and a Workflow Process step. | Click a step in the Workflow Process and then go to the Input Arguments tab in the MVPW pane to define input arguments for the step as required . |
| Output Arguments | Communicates information about output arguments between a process property and a Workflow Process step. | Click a step in the Workflow Process and then go to the Output Arguments tab in the MVPW pane to define output arguments for the step as required . |

The example MVPW pane in the following figure shows the input arguments (`RowId` and `TableName`) for the Unsubscribe Current Owner step.




## Viewing a Process Property in the MVPW

When no step or connector is selected and you click the canvas in the Process Designer, then the Process Designer displays the child objects for that Workflow Process. The Process Properties tab in the Multi Value Property Window (MVPW) pane displays the definitions of the process properties.

### To view a process property in the MVPW pane

1. Click anywhere in the Process Designer canvas, making sure not to select a step or connector.
2. In the MVPW pane, go to the Process Properties tab.
3. Examine the records that the MVPW pane displays.

## Viewing a Step Argument in the MVPW

When you click a step in the Process Designer canvas, then the Process Designer displays the input arguments and output arguments for the selected step in the Multi Value Property Window (MVPW) pane. Depending on the type of step you select, the MVPW pane displays one or more tabs.

### To view a step argument in the MVPW pane

1. Click a step in the Process Designer canvas.
2. In the MVPW pane, go to the Input Arguments or Output Arguments tab.

3. Examine the records that the MVPW pane displays.

## Defining a Step Argument in the MVPW

You can define an input argument or an output argument for a step in the Multi Value Property Window (MVPW) pane.

### To define a step argument in the MVPW pane

1. In the Workflow Process Designer, click the step where you want to define an argument.
2. In the MVPW pane, click the Input Arguments tab (or Output Arguments tab as required).
3. In the MVPW pane, click the Input Arguments Menu (the cogwheel icon) and then select New Record.
4. Define the new input argument by completing the Type field and other fields as necessary.

   The value you choose in the Type field determines how you define the other fields. The following table describes how the Type field affects how you complete the other fields.

   For example, to define an input argument on a Siebel operation step, then you must define the Input Argument field.

| Type Field Options | Work You Perform |
|---|---|
| Business Component | If you select Business Component in the Type field, then you must:<br>o Choose a business component in the Business Component Name field.<br>o Choose a business component field the Business Component Field field. |
| Expression | If you select Expression in the Type field, then enter an expression in the Value field. Siebel CRM evaluates the expression you enter at runtime. You can also click the button to open the Expression Builder. |
| Literal | If you select Literal in the Type field, then enter a string in the Value field. The value you enter defines the literal value for the argument. |
| Process Property | If you select Process Property in the Type field, then choose a Property Name in the Property Name field. Process Property is only available for an input argument. |
| Output Argument | If you select Output Argument in the Type field, then Choose a property in the Output Argument field. You can choose a process property that is of type In/Out or Out. Output Argument is only available for an output argument. |

# Using Predefined Process Properties

The following table describes some of the key Siebel predefined process properties.

| Predefined Process Property | Description |
|---|---|
| Object Id | The Siebel Row Id of the record that Siebel CRM uses when it starts a Workflow Process. That is, the Object Id process property is the Siebel Row Id of the primary business component record that is passed to the Workflow Process from the caller. For more information, see the following:<br>• *Runtime Event Behavior* |

| Predefined Process Property | Description |
|---|---|
| | • *Changing the Active Row in a Workflow Process Step*<br>• *Using the In/Out Process Property*<br>• *Running a Workflow Process to Avoid a Response Data Insert* |
| Error Code | An error number that Siebel CRM uses if a step returns an error. The Workflow Engine automatically populates this process property if an error occurs. |
| Error Message | The text error message if a step returns an error. The Workflow Engine automatically populates this process property if an error occurs. |
| Siebel Operation Object Id | The Siebel Row Id of the record that Siebel CRM updates, creates, or queries during a Siebel operation step. The Workflow Engine automatically populates this process property when a Siebel operation step is run. |
| Process Instance Id | A unique number that Siebel CRM assigns to the Workflow Process instance that is currently running. The Workflow Engine automatically populates this process property when a Workflow Process runs and if persistence is activated. |

## Runtime Event Behavior

If a runtime event starts a Workflow Process, then Siebel CRM sends the business component instance that starts the runtime event to Siebel Workflow. The row that the Object Id property defines receives this runtime event. Siebel Workflow receives and processes this event only if the business component and the Workflow Process references the same business object. If Siebel Workflow can receive this event, then Siebel CRM sets the Object Id process property to the active row of the primary record in the business object.

If the business component that starts the runtime event is not the primary business component, then Siebel CRM does not update the active row of the business component in the Object Id process property, and it must get this row through some extra processing.

The long-running, interactive, and service Workflow Processes use the following behavior:

- The Object Id must match the Row ID of the active row for the primary business component. If the Object Id process property is different from the active row, then Siebel CRM runs the primary business component again to make the active row the same as the Object Id.
- To change the active row, you can configure Siebel CRM to assign a new Row Id to the Object Id property. If Siebel Workflow detects that a new Row Id is assigned to the Object Id property, then it runs the business component again and makes the new Row ID the active row.
- If you set the Object Id to an empty string, then Siebel Workflow does not enforce the must match rule. However, Siebel Workflow cannot use objects that require an Object Id, such as the runtime event or Siebel operation step, until it sets the Object Id to a new Row ID.

## Changing the Active Row in a Workflow Process Step

You can add a step that changes the active row.

### To change the active row in a Workflow Process step

1. Add a Siebel operation or business service step that performs an operation that causes Siebel CRM to change the active row.

   For more information, see *Adding Workflow Process Steps*.
2. Update the value of the Object Id process property to the new value of the active Row Id that resides in the output argument of the Workflow Process step you added.

   After a Workflow Process step finishes and Siebel Workflow evaluates the output arguments, it makes sure the Object Id matches the active row. The Object Id property in this step must include the change to the active row.

## Using the In/Out Process Property

If necessary, you can run a Workflow Process to avoid receiving response data. For example, to prevent Siebel CRM from inserting records in the S_SRM_DATA table that can cause a heavy backlog. If the Type field of a process property is Out or In/Out, then Siebel CRM sends a reply to the object that called the Workflow Process. For example, it sends a reply to the Server Request Processor. If the caller:

- Receives a reply that is not null, then it writes this reply in the S_SRM_DATA table.
- Receives no reply, then it does not write a reply in the S_SRM_DATA table.

Siebel CRM does the following:

- If it inserts a request in the S_SRM_REQUEST table, then it also inserts one or more rows in the S_SRM_DATA table for the component request parameters and input arguments that the request uses.
- When it sends this request, the MSG_TYPE_CD column of the S_SRM_DATA table contains a value of REQ_DATA, which indicates that Siebel CRM must request data or input for the request.
- If the request finishes, then it inserts a set of rows in the S_SRM_DATA table and sets the value of the MSG_TYPE_CD column to REQ_RESPONSE, which indicates that the request returned a reply to the caller. In this situation, the request is in the S_SRM_REQUEST table, which indicates that the Server Request Processor component is the caller.

## Running a Workflow Process to Avoid a Response Data Insert

You can run a Workflow Process to avoid response data inserts.

### To run a Workflow Process to avoid a response data insert

1. Do one of the following:

   - Leave the In/Out property of the process property set to In/Out until immediately upstream from the end of the Workflow Process.
   - Immediately upstream from the end of the Workflow Process, call another step that nulls out values that Siebel CRM stores in the process properties.

     If you use this configuration, then Siebel CRM does not send a reply to the Server Request Processor and it does not insert a record in the S_SRM_DATA table.
2. Review the process properties, and then set the In/Out property to NONE.

# Defining a Custom Process Property

You can define a process property in the Multi Value Property Window (MVPW) pane or in the WF Process Props list. You can define process properties while you diagram a Workflow Process or you can diagram the entire workflow and then define properties for individual steps. It is recommended that you define process properties before you define an input argument or output argument on an individual step because some of these arguments might reference a process property. For more information, see *Using Process Properties*.

## To define a custom process property

1. Click the canvas in the Process Designer, making sure no Workflow Process step or connector is chosen.
2. In the MVPW pane, go to the Process Properties tab.
3. Add a new (custom process property) record by completing the following fields:
    - Enter a Name for the process property. For more information, see *Naming a Workflow Process Step or a Process Property*.
    - Define the Data Type field using values from the following table.

| Data Type | Type of Data the Process Property Holds |
|---|---|
| String | Series of characters |
| Number | Numeric value |
| Date | Date value |
| Hierarchy | Hierarchical data, as in a property set |
| Binary | Binary value<br><br>Siebel CRM encodes binary data in the original character set where it was entered. This data type code is appropriate for data that describes itself, such as XML code. |
| Integration Object | Integration object<br><br>The Siebel Runtime Repository uses the integration object definition for this object. |
| Alias | XPath notation for pointing to a child in a hierarchical process property |

| Data Type | Type of Data the Process Property Holds |
|---|---|
| | |

The String data type code is appropriate for strings, text data, and non XML data even though Siebel CRM can store XML data as string data. Siebel CRM assumes that string data in a Workflow Process is UTF 16 String encoded. Literal or Expression data types are UTF 16 String.

You cannot modify a data type after you define it.

- Define the Default String, Default Date, or Default Number property, as necessary.

  Siebel CRM uses the value that you define in one of these properties when it runs a Workflow Process. If the corresponding data type for the value is chosen, then you only define one of these default values. For example, if you define the Date data type, then you can define a Default Date.

4. Repeat step 2 and 3 to define more process properties, as required.

# Using Process Properties to Manipulate Data

This topic describes how to use a process property to manipulate data. It includes the following topics:

- *Passing a Process Property In and Out of a Workflow Process Step*
- *Passing a Process Property by Reference*
- *Passing Data Through a Process Property to an Error-Workflow Process*
- *Concatenating a Process Property*
- *Referencing a Process Property*
- *Externalizing Workflow Properties*

## Passing a Process Property In and Out of a Workflow Process Step

It is recommended that the Type field in the top-level of a hierarchical argument not contain data. If Siebel Workflow passes a hierarchical process property to a Workflow Process step, then it overwrites the Type field in the top-level process property for the duration of the call. It does this so that the process property matches the name of the argument that the configuration for the input argument defines. This configuration applies only for a long-running, interactive, or service Workflow Process.

For example, assume MyTree is a process property and that the data type for this process property is Hierarchy. MyBusSvc is a business service that contains a hierarchical input argument named SomeTree. The following table describes the configuration for this process property.

| Input Argument | Type | Property Name |
|---|---|---|
| SomeTree | Process Property | MyTree |
| Business Object | N/A | The name of the associated business object. |
| Created By | N/A | The name of the person who creates the process. |

| Input Argument | Type | Property Name |
|---|---|---|
| Created | N/A | The date that the process is created. |

If you use the values that this table describes to define the input argument, then the call to MyBusSvc receives a child in the input process property for the argument where the Process Property Type field is set to SomeTree instead of MyTree. The rest of the data in the child process property remains the same as the contents of the MyTree process property.

Siebel Workflow expects an output argument of a Workflow Process step that passes out a hierarchy as a child of the output property set. To locate the child, Siebel Workflow examines the Type field of the child.

The string in the Type field must match the name of the Output Argument field, as defined in the output argument applet of the step. If Siebel Workflow finds such a child, then it does the following:

- Copies the data into the destination process property.
- Discards the Type field on the incoming hierarchy.
- Replaces the value that the Type field contains with the name of the process property.

## Passing a Process Property by Reference

A sub process that modifies a large amount of data must copy this data through input arguments and output arguments. This situation can negatively affect performance and scalability. Pass By Reference is a feature that allows you to pass a pointer to the property set rather than passing all the data that this property set contains. For more information, see *Using the Pass By Reference Feature with a Business Service*.

## Passing Data Through a Process Property to an Error-Workflow Process

Siebel CRM can pass more than just a system defined process property to an error-Workflow Process. It can pass the following information:

- **Workflow Process instance.** To pass a Workflow Process instance to a process property that resides in an error-Workflow Process, you must redefine these process properties in the error-Workflow Process. You must use the same name and data type in each of these Workflow Processes.
- **Property set of a Workflow Process instance.** To pass the property set of a Workflow Process instance to an error-Workflow Process, you must define a hierarchy process property in the Workflow Process and in the error-Workflow Process. You must use this hierarchy property to pass the property set.
- **The name of a Workflow Process instance.** To pass the name of a Workflow Process instance, you must define a process property in this Workflow Process and in the error-Workflow Process. You must configure Siebel CRM to pass the name of the Workflow Process instance through this process property.

### To pass data through a process property to an error-Workflow Process

1. Define an error exception connector.

   For more information, see *Example of Error Exception Handling*.
2. Click the stop step, and then define an input argument in the MVPW pane using values from the following table.

| Field | Value |
|---|---|
| Name | `%1` |

| Field | Value |
|---|---|
| | |
| Type | Literal |
| Value | Error Message (or a valid string) |

3. In the Workflow Processes list, create a new Workflow Process that references the error-Workflow Process.

## Concatenating a Process Property

The example in this topic describes how Siebel CRM can use four process properties in a Workflow Process to concatenate three string values. The three process properties contain the following values in the Default String property:

- Welcome
- to
- Siebel

### To concatenate a process property

1. In the Workflow Processes list, create a new Workflow Process with the values shown the following table.

| Property | Value |
|---|---|
| Process Name | Concatenate |
| Workflow Mode | Interactive Flow |

   For an example, see *Creating the Workflow Process*.
2. Open the Workflow Process you defined in step 1, and then create a flow that contains a Start step, followed by a Concatenate step, followed by an End step.

   For more information, see *Diagramming a Workflow Process* and *Adding Workflow Process Steps*.
3. In the Multi Value Property Window (MVPW) pane, define four process properties with the values shown in the following table.

| Name | In/Out | Default String | Data Type |
|---|---|---|---|
| ProcessProperty1 | In/Out | Welcome | String |
| ProcessProperty2 | In/Out | to | String |
| ProcessProperty3 | In/Out | Siebel | String |

| Name | In/Out | Default String | Data Type |
|---|---|---|---|
| | | | |
| ProcessProperty4 | In/Out | (no value) | String |
| Business Object | N/A | Business object of the parent process. The business object determines the business components that are available for conditions. This is a read-only field. | Business object of the parent process. The business object determines the business components that are available for conditions. This is a read-only field. |
| Created By | N/A | This value is automatically entered based on the log on name of the user. | This read-only value is automatically entered based on the log on name of the user. |
| Created | N/A | This value is automatically entered. | This read-only value is automatically entered. |

For more information, see *Using the Multi Value Property Window*.

4. Click the Wait step and go to the Output Arguments tab in the MVPW pane.

   You can use the wait step to test and develop a Workflow Process. For more information, see *Adding a Wait Step*.

5. Define an Output Argument for the wait step with the values shown in the following table.

| Type | Output Arguments | Value |
|---|---|---|
| Expression | ProcessProperty4 | `[&ProcessProperty1]+' '+[&ProcessProperty2]+' '+[&ProcessProperty3]` |

   The ampersand (`&`) identifies the text that immediately follows the ampersand as the name of a process property. This process property can also be the name of a business component field. ProcessProperty1, ProcessProperty2, and ProcessProperty3 can be of different types, such as string or date.

   You must set the Data Type property for the process property in this example to String. If you use a binary type with an expression, then an error might occur. You cannot use an expression to set a process property whose Data Type property is set to binary. An example error in this situation is a process property that includes truncated data.

6. Validate and then simulate the Workflow Process.

   For more information, see *Process of Testing a Workflow Process*.

7. After control returns from the Application Object Manager:
   - (Siebel Tools) Right-click the Process Simulator window, and then click Watch Window.

- (Web Tools) The Workflow Process Data window is automatically open in the Process Simulator and populated with the Process Properties for the Workflow Process.

  **Note:** In Web Tools, the Watch Window is called Workflow Process Data.

8. In the Watch window, expand PS:Property Set.
9. Make a note of the values that the Process Designer displays for the process properties you defined in step 3.
10. Click Simulate Next.

    In the Watch window, ProcessProperty4 now contains a concatenation of values from ProcessProperty1, ProcessProperty2, and ProcessProperty3.

## Referencing a Process Property

To reference a process property in an expression, use the following format:

```
[&PropName]
```

For example:

```
[&Object Id] like '99-28B1T'
```

In this example, the `&` indicates a process property and the `Object Id` is the name of the process property.

The following example shows how Siebel CRM can use a process property as a substitution variable. This example uses an input argument to create a message body:

```
The Activity #" + [&Object Id] + ", owned by " + [&First Name]+" "+[&Last Name] + "
is three days past the Due Date.
```

In this example, Object Id, First Name, and Last Name are defined as process properties.

For more examples of using a process property as a substitution variable, see *Using Compound Expressions and Substitutions in a Search Specification* and *Using Substitution Variables in an Expression*.

## Externalizing Workflow Properties

It is recommended that you define properties for a Workflow Process that are externalized and that are not hard coded. Hard coding a property in a Workflow Process requires you to change the object definitions when you deploy the Workflow Process between environments. For example, if a Workflow Process sends an email to a customer list, and if you hard code this list in a process property, then the Workflow Process does not run correctly in the production environment. You must make sure the Workflow Process reads this kind of data dynamically.

# Passing Data to and from a Workflow Process

This topic describes how to configure Siebel CRM to pass data to a Workflow Process and to get data from a Workflow Process. It includes the following topics:

- *Passing Values to a Workflow Process*
- *Passing Values from a Workflow Process*
- *Passing a Constant from a Workflow Policy Action to a Workflow Process*
- *Examples Scripts That Pass Data to and from a Workflow Process*

If Siebel CRM uses the Workflow Process Manager business service to start a Workflow Process, then it can pass data to and get data from this Workflow Process. To pass a property set, a sub process step uses the same conventions that this business service uses. For more information, see *Using the In/Out Process Property*.

# Passing Values to a Workflow Process

The input property set must include the ProcessName property, which specifies the name of the Workflow Process to run. You can include other values in the property set, such as strings, numbers, and property sets. The Workflow Process Manager business service passes these values.

If the input property set includes a property in the top-level property set, and if the name in this property set matches the Name property of the Workflow Process, then Siebel CRM initializes simple data type process properties that are marked In or In/Out, such as the following:

- String
- Number
- DateTime

To initialize the value for each process property in the Workflow Process, Siebel CRM uses the value in the corresponding process property in the input property set.

If the input property set includes a child, and if the Type field of this child contains a string that matches the name of the hierarchical process property in the Workflow Process, then Siebel CRM does the following:

- Initializes the hierarchical data type process properties that are marked In or In/Out.
- Copies the matching child and everything below the child in the input property set to the process property.

# Passing Values from a Workflow Process

A Workflow Process that starts programmatically might not return an output. For example, an interactive Workflow Process can start programmatically but it can pause, and the output from the call that Siebel CRM uses to start it might reflect the state that exists at an intermediate point. For this reason, only a Workflow Process that is certain to complete in one call can provide output in the output arguments of a call to the Workflow Process Manager business service. A service Workflow Process is the only type of Workflow Process that runs to completion in this way.

# Passing a Constant from a Workflow Policy Action to a Workflow Process

The Run Workflow Process program can pass a constant from a Workflow Policy action to a Workflow Process.

## To pass a constant from a Workflow Policy action into a Workflow Process

1. Navigate to the Workflow Policy Program list, and then query the Name property for Run Workflow Process.
2. Navigate to the Workflow Policy Program Arguments list.
3. Add a new argument.

4. Choose the File menu, and then the Save menu item.

   For example, add a new argument with the Name property set to IOName. Make sure the Visible property contains a check mark.
5. In the Workflow Policy Program list, click the Run Workflow Process object.
6. Set the default value.

   The default value varies according to IOName.
7. Open the Process Designer for the Workflow Process to which Siebel CRM must pass the constant.
8. In the MVPW pane, create a new record with the Name argument set to the same name defined in the argument in step 4.

   For example, Name must be the same as IOName. For more information, see *Using Process Properties*.
9. Call the Workflow Policy.

   For more information, see *Starting a Workflow Process from a Workflow Policy*.

# Examples Scripts That Pass Data to and from a Workflow Process

The following examples in this topic pass data to and from a Workflow Process:

- *Using a Script to Start a Workflow Process and Create an Input Property Set*
- *Using a Script to Create Property Sets, Set Values in PS, Add Child PS, and Call BS with Input PS*
- *Using a Script to Access Parameters for a Workflow Process*

## Using a Script to Start a Workflow Process and Create an Input Property Set

The following example script calls the Workflow Process Manager business service. This business service creates an input property set named psInputs for the business service. This script defines strings as properties. It puts these strings in the input property set:

```
var msgName = "Siebel Agent Authorization Retrieval";
var reqSubType = "CICS Services Request";
var reqType = "AgentAuthorizationReq";
var CICSServiceName = "Consumer Auto Agent Authorization Retrieval";
var processName = "Consumer Auto VBC VBC Template";
var reqFileName = "C:\\<path to file>\\XMLMessages\\AgentAuthorizationVBCReq-final.xml"
var resFileName = "C:\\<path to file>\\XMLMessages\\AgentAuthorizationVBCResponse-
final.xml"
```

## Using a Script to Create Property Sets, Set Values in PS, Add Child PS, and Call BS with InputPS

The following example script shows how to create property sets, set values in `PropertySets`, add child PS to parent `psInputs`, and call the business service with the `Input PropertySets`:

```
//Create Property Sets
var psRequest = app.NewPropertySet();
var psAgentNumTag = app.NewPropertySet();
var sAgentID;

//Set values in PropertySets
sAgentID = app.LoginName();
```

```
psRequest.SetType("XMLHierarchy");
psAgentNumTag.SetType("DataAgentNumber");
psAgentNumTag.SetValue(sAgentID);
psRequest.AddChild(psAgentNumTag);

//Add child PS to parent psInputs
psInputs.AddChild(psRequest); //Pass in Property Set
psInputs.SetProperty("RequestURLTemplate", requestURLTemplate); //Pass in string
psInputs.SetProperty("RequestSubType", reqSubType);
psInputs.SetProperty("ReqType", reqType);
psInputs.SetProperty("MessageName", msgName);
psInputs.SetProperty("CICSServiceName", CICSServiceName);
psInputs.SetProperty("ProcessName", processName);
psInputs.SetProperty("Request File Name", reqFileName);
psInputs.SetProperty("Response File Name", resFileName);

//Invoke the business service with the Input PropertySets.
var svc = TheApplication().GetService("Workflow Process Manager");
svc.InvokeMethod("RunProcess", psInputs, psOutputs); //Call the Workflow
var sErr = psOutputs.GetProperty("status"); //Check the Workflow status
```

## Using a Script to Access Parameters for a Workflow Process

The example in this topic accesses workflow parameters for a running Workflow Process.

### To use a script to access parameters for a Workflow Process

1. Define a business service that includes the required methods and parameters.
2. Access the business service from the Workflow Process.
3. In the business service step in the Workflow Process, pass the Workflow Process properties to the business service method arguments.

   For more information, see *Passing a Process Property In and Out of a Workflow Process Step*.
4. Use the following script to use the values of the business service argument and assign them to profile attributes:

   ```
   TheApplication().SetProfileAttr("WFRunning", Inputs.GetProperty("Workflow Running");
   ```

   A *profile attribute* is a string that is visible to any object that exists in a user session. A separate session exists for each user. A user cannot access the strings that another user session contains.
5. Use the profile attributes for more processing.

   Siebel CRM places the required information in the profile attributes. You can use the typical procedures to configure Siebel CRM to access these profile attributes. For more information, see *Siebel Personalization Administration Guide* .

# Configuring Decision Conditions for a Workflow Process

This topic describes how to use the Compose Condition Criteria dialog box to configure a decision condition for a Workflow Process. It includes the following topics:

- *Configuring a Decision Condition*
- *Configuring Fields in the Compose Condition Criteria Dialog Box*
- *Configuring Expressions in the Expression Builder*

# Configuring a Decision Condition

To define conditional logic in a Workflow Process, you define conditions and values that determine the path that the Workflow Process pursues. For example, you can define a decision condition that uses the value of a priority field:

- If the priority is equal to high, then the Workflow Process pursues a branch that includes a step that sends an email to a vice president.
- If the priority is equal to medium, then the Workflow Process sends an email to an engineer.

A connector that emanates from the following step types can provide conditional logic for the Workflow Process: start step, decision step, wait step, or user interact step. For more information, see *Adding a Branch Connector*.

## Viewing an Example of a Decision Condition

This book includes several examples that use a decision condition. Here are two examples:

- If the Revenue field in the Opportunity business component is greater than $10,000, then insert an activity for the parent opportunity. To examine the example that uses this condition, see *Defining a Decision Condition for a Decision Step*.
- Use a CASE statement to identify one of several branches to pursue in the Workflow Process, depending on the value of the Priority field in the Service Request business component. Insert an activity with a value for the commit time according to the result of the CASE statement.

## Adding a Branch Connector

A branch connector (also known as a connector) is a type of connector in a Workflow Process that can include a decision condition. You define conditional logic on a connector that emanates out of the step, and not on the step itself. A branch connector typically emanates from a start step, decision step, wait step, or user interact step.

### To add a branch connector

1. In the Process Designer, add a branch connector as follows:
    - (Siebel Tools) Select Connector from the palette, then drag and drop it on the canvas. Select the connector on the canvas and attach it (or one end of it) to a step, then move the other end of the connector to connect to the next step in the flow.
    - (Web Tools) Use the mouse to draw a connector line from the connector node on one step to the connector node on another step.

    For more information, see *Diagramming a Workflow Process*.
2. Select the connector you added in the previous step and in the Properties pane:
    a. Enter or modify the connector name.

    The connector name must be unique in the Workflow Process. If it is not unique, then you cannot save the Workflow Process.

    b. Set the Type property.

    In most situations, this value is Condition (if defining a decision condition on the connector) or Default (if using this connector as an exit route). You can set other values as required. For more information, see the topic about the Connector object type in *Siebel Object Types Reference* .

    If you define multiple branches on a step, then see *Adding Multiple Branches to a Single Workflow Process Step* for caution information.

c. Enter comments.

3. If required, define a decision condition for the connector.

   If you want the connector to incorporate conditional logic, then double-click the connector to open the Compose Condition Criteria dialog box. For more information, see *Creating a Decision Condition on a Branch Connector*.

## Creating a Decision Condition on a Branch Connector

You can use the Compose Condition Criteria dialog box to define a decision condition on a branch connector. A branch connector can be a connector that emanates from a start step, decision step, wait step, or user interact step. The Compose Condition Criteria dialog box constrains the values that it lists according to the Business Object property of the Workflow Process. For an example that uses conditional logic on a connector, see *Defining a Decision Condition for a Decision Step*.

### To create a decision condition on a branch connector

1. In the Process Designer, double-click the connector where you must define a decision condition.
2. In the Compose Condition Criteria dialog box, create a condition.

   For more information, see *Configuring Decision Conditions for a Workflow Process*.
3. Click OK.

## Adding Multiple Branches to a Single Workflow Process Step

A start step, decision step, wait step, or user interact step can each reference multiple branch connectors.

> **CAUTION:** If you define multiple branch connectors, then make sure you define at least one connector with the Type property set to Default. This configuration makes sure an exit route exists if a work item does not meet any of the decision conditions. If you do not do this, then you might create an infinite loop.

### To add multiple branches to a single Workflow Process step

- Complete the procedure described in *Adding a Branch Connector* for each branch that you must define for the step. Each branch can contain a separate decision condition.

## Using Parallel Processing in a Workflow Process

Siebel Workflow does not support parallel processing. Make sure you define decision conditions so that the Workflow Process can only proceed along one connector. If you do not do this, then Oracle cannot predict the runtime behavior of the Workflow Engine.

# Configuring Fields in the Compose Condition Criteria Dialog Box

The following table describes the fields you set in the Compose Condition Criteria dialog box to define a decision condition. For more information, see *Configuring Expressions in the Expression Builder*,

| Field | Description | Possible Value |
|---|---|---|
| Compare To | Indicates where the comparison value is coming from. | Required. The following choices are available:<br>• Business Component<br>• Process Property<br>• Expression<br>• Applet |
| Operation | Identifies the comparison operation. | The following choices are available:<br>• **This Must Match.** The current value must match exactly, including case.<br>• **One Must Match.** One or more values must match exactly, including case.<br>• **All Must Match.** All of the values must match exactly, including case.<br>• **None Can Match.** None of the values can match exactly, including case.<br>• **This Must Match (ignore case).** The current value must match without regard to case.<br>• **One Must Match (ignore case).** One or more values must match without regard to case.<br>• **All Must Match (ignore case).** All of the values must match without regard to case.<br>• **None Can Match (ignore case).** None of the values can match without regard to case.<br>• **Greater Than.** Value must be greater than the comparison value.<br>• **Less Than.** Value must be less than the comparison value.<br>• **Between.** Value must be between a range of values.<br>• **Not Between.** Value cannot be between a range of values.<br>• **Is Null.** Value must be null.<br>• **Is Not Null.** Value cannot be null. |
| Object | The name of the associated business object. | You choose this value from a drop-down list of business objects. |
| Field | A required value when Applet is the Compare To value. | The name of the field in the named applet. |
| Values | Not applicable. | The Values property is dynamic. It uses the Compare To property. Siebel CRM uses the Values property to store data to use when it evaluates the decision condition. |

# Configuring Expressions in the Expression Builder

You use the Expression Builder in the Compose Condition Criteria dialog box to define a decision condition. This topic describes parts of the Expression Builder, including Compare To options, comparison operations, and patterns.

- *Using Compare To Options*
- *Using Simple Comparison Operations*
- *Using Iterative Comparison Operations*
- *Using Substitution Variables in an Expression*

## Using Compare To Options

The following table describes Compare To options and their required and optional values.

| Compare To | Operation | Object | Field | Value |
|---|---|---|---|---|
| Applet. Compare the runtime value of an applet control or list column to a literal. | Comparison operation. All operations are allowed. | The name of the applet. | The column in the applet. | One or more literals for comparison. |
| Business Component. Compare the runtime value of a business component field to a literal. | Comparison operation. All operations are allowed. | The name of the business component. | The field in the business component. | One or more literals for comparison. |
| Expression. Evaluate expressions and determine if they return true. | Specifies how to evaluate multiple expressions. Only applicable if the Values property includes multiple expressions.<br>The following choices are available:<br>• All Must Match<br>• None Can Match<br>• One Must Match<br>• This Must Match<br>For more information, see *Using Simple Comparison Operations* and *Using Substitution Variables in an Expression*. | Optional. If the expression references a business component field, then the object is a business component name. You can reference the fields of only one business component in the expression. | N/A | One or more expressions. |
| Process Property. Compare the runtime value of a process property to a literal. | Comparison operation. All operations are allowed. | The name of the process property. | N/A | One or more literals for comparison. |

## Using Simple Comparison Operations

A *simple comparison operation* involves a comparison that Siebel CRM can express in a simple expression without using an iterative operation. The following table describes values for the following simple comparison operations: Between, Not Between, Greater Than, Less Than, Is Null, and Is Not Null.

| Comparison | Runtime Value That Results in a Successful Comparison | Description |
|---|---|---|
| Between | Between two predefined literal values. | This comparison requires two values in the Values property. The Process Designer enforces this requirement. |
| Not Between | Not between two predefined literal values. | This comparison requires two values in the Values property. The Process Designer enforces this requirement. |
| Greater Than | Greater than a predefined literal value. | Logically, only one value is required in the Values property. |
| Less Than | Less than a predefined literal value. | Logically, only one value is required in the Values property. |
| Is Null | Null or empty. | No value is required in the Values property. |
| Is Not Null | Not null or empty. | No value is required in the Values property. |

## Using Iterative Comparison Operations

An *iterative comparison operation* is a type of operation that uses multiple iterations to get a Boolean result. It does these iterations on the literal values or expressions that the Values property contains, or on child business component records.

The following configurations can result in different iterative behavior:

- **You define multiple values in the Values property.**
- **Multiple records of the child business component exist in the current workset.** A *child business component* is a business component that is not the primary business component in the current business object that the Workflow Process references. If you set the Compare To option to Business Component or to Expression, and if you define the Object field as the name of a child business component, then Siebel CRM uses this child business component in the comparison. If Siebel CRM returned multiple records the last time it processed records in the child business component, then multiple records can exist in the current workset of the child business component. For more information, see *Defining the Primary Business Component*.

### All Must Match Operation

The following options are available for the All Must Match operation:

- **Multiple value behavior.** If you set the Compare To option to a value other than Expression, and if at least one value matches, then Siebel CRM defines a literal and this comparison succeeds. If the Compare To option is Expression, and if each expression evaluates to true, then this comparison succeeds.

- **Multiple child business component record behavior.** If the comparison succeeds for every record, then Siebel CRM uses child business component records for comparison and the overall comparison succeeds.

  Consider an example where Siebel CRM uses All Must Match and multiple child business components. Account is the business object. The following table describes the decision condition.

| Compare To | Operation | Object | Field | Values |
|---|---|---|---|---|
| Business Component | All Must Match | Contact | First Name | Jane, Julie |

  In this example, if a record in the Contact business component in the current workset contains a first name of Jane or Julie, then the comparison succeeds. The workset is the child contact record for the account record that the Workflow Process is processing.

## This Must Match Operation

For the comparison to succeed, field values of the active row of the current instance of the child business component must match the decision condition values that you define for the connector. The This Must Match operation includes the following options:

- **Multiple value behavior.** If the values involved are literal values, and if at least one value matches, then the comparison succeeds. If the values involved are expressions, and if at least one expression evaluates to true, then the comparison succeeds.
- **Multiple child business component record behavior.** If the comparison succeeds for the current record, then Siebel CRM uses only the current record of the child business component for comparison and the overall comparison succeeds.

If you use Remote Asynchronous, then the Workflow Process resumes from the Workflow Process Manager. The Workflow Process Manager does not preserve the instance of the child business component and the active row for the child business component is lost. For this reason, the branch connector stops working. If you must use Remote Asynchronous, then use an operation other than This Must Match. Otherwise, use Local Synchronous.

## None Can Match Operation

The None Can Match operation includes the following options:

- **Multiple value behavior.** If the values involved are literal values, and if none of the values match, then the comparison succeeds. If the values involved are expressions, and if none of the expressions evaluate to true, then the comparison succeeds.
- **Multiple child business component record behavior.** If the comparison fails for every record, then Siebel CRM uses the records of the child business component for the comparison and the overall comparison succeeds.

## One Must Match Operation

The One Must Match operation includes the following options:

- **Multiple value behavior.** If the values involved are literal values, and if at least one value matches, then the comparison succeeds. If the values involved are expressions, and if at least one expression evaluates to true, which is the same as This Must Match, then the comparison succeeds.
- **Multiple child business component record behavior.** If the comparison succeeds for at least one of the records, then Siebel CRM uses records of the child business component for comparison and the overall comparison succeeds.

### Other Iterative Comparison Operators

Other iterative comparison operators include the following:

- All Must Match (Ignore Case)
  Same as All Must Match except string comparisons are case insensitive.
- This Must Match (Ignore Case)
  Same as This Must Match except string comparisons are case insensitive.
- None Can Match (Ignore Case)
  Same as None Can Match except string comparisons are case insensitive.
- One Must Match (Ignore Case)
  Same as One Must Match except string comparisons are case insensitive.

## Using Substitution Variables in an Expression

You can use a process property or a field in a business component as a substitution variable in an expression. For more information see *Referencing a Process Property* and *Referencing a Business Component Field You*.

For more information about the operators, expressions, and decision conditions that you can use in a Workflow Process, see *Siebel Developer's Reference* .

### Referencing a Business Component Field You

You must use the following format to reference a business component field:

```
[field_name]
```

For example, the Contact business component uses the following format:

```
[First Name] like 'Jane'
```

You specify the business component name in the Object field of the Expression Builder.

### Example of Using an Expression to Compare Values

Tthe following table describes how to use an expression to compare the value of the Object Id process property to the value of the Account Id field in the Contact business component.

| Compare To | Operation | Object | Values |
|---|---|---|---|
| Expression | This Must Match | Contact | `[Account Id] like [&Object Id]` |

# Accessing Data from a Runtime Event in a Workflow Process

This topic explains how to configure Siebel CRM to access runtime event data from a Workflow Process.

# To access data from a runtime event in a Workflow Process

1. If necessary, display the WF Step Branch object type in the Object Explorer.
2. In the Workflow Process Designer, define the properties for a WF Step Branch using values described in the following table.

| Property | Value |
|---|---|
| Name | Enter a name for the branch. |
| Type | Condition |
| Event Object Type | Applet |
| Event | InvokeMethod |
| Event Object | Choose the name of the event object that starts the Workflow Process. |
| Sub Event | NewRecord |
| Comments | Optional |
| Event Cancel Flag | The default is blank. |

3. Set up process properties in the Process Designer.

   For more information, see *Using Process Properties*.
4. Click the first step in the Workflow Process, and then add a new output argument for each process property that Siebel CRM must populate that references the runtime event.

   To add a new output argument, you enter values in the Output Arguments tab in the Multi Value Property Window (MVPW) pane. Use the example values described in the following table.

| Field | Value |
|---|---|
| Property Name | Enter a name for the property. |
| Type | Expression |
| Value | `GetProfileAttr("RestrucOut")` |

| Field | Value |
|---|---|
| | When you define the Profile Attribute field, note the name you give to each of the property values for the Profile Attributes. |
| Output Argument | Leave default value. |
| Business Component Name | Leave default value. |
| Business Component Field | Leave default value. |
| Comments | Optional |

5. Deliver the Workspace.
6. Query the Event field for the runtime event you defined in step 4.
7. Drill down on the Action Set Name field.
8. Add a new Action for each process property that the runtime event updates.
   For example, to set the ACU Transaction ID and name it Set ACU Trans ID, you can create a new action. You must set the following for each new action:
   - Set the Action Type property to Attribute Set.
   - Set the Profile Attribute to match the value you used in the GetProfileAttr call in the Process Designer, such as TransType.
   - Set the Set Operator value to Set.
   - Use the Expression Builder to assign the Value field an appropriate value, such as the literal string FA-0001.
   - Set the Sequence for the action to be less than the default sequence for the Workflow_XXXXXXX Action.
   - Make sure the Sequence setting of the Workflow_XXXXXXX Action is the highest number so that this action occurs after every other action.

   If you modify the Workflow Process, then Siebel CRM resets the sequence for the workflow action to 1 and you must repeat this step.

# Using the Timestamp

You can use the time stamp to get the current system time and to do time arithmetic that uses the current time. The arithmetic that Siebel CRM uses with time information for a Workflow Process is different from the arithmetic that it uses for a Workflow Policy program. It must use the seconds operand of the Timestamp function in a scale of minutes, which is a fraction of the whole day. For example, to display an arithmetic operation as 30 minutes, Siebel CRM displays the argument according to the following format:

```
Timestamp()+(30/(24*60))
```

where `(24*60)` are the number of minutes in the day

For more information, see *Using Date Calculations in the Conditions List*.

# 9 Testing a Workflow Process

## Testing a Workflow Process

This chapter describes how to test a Workflow Process. It includes the following topics:

- *About the Testing Tools*
- *Process of Testing a Workflow Process*
- *Validating the Workflow Process*
- *Using the Process Simulator*
- *Troubleshooting Validation and Simulation Problems*

For more information about developing planning strategies for testing the Siebel application, see *Testing Siebel Business Applications* .

## About the Testing Tools

This topic describes the tools you can use to test a Workflow Process. It includes the following topics:

- *Validate Tool*
- *Process Simulator*
- *Event Logs*

### Validate Tool

The *Validate Tool* is an error correction tool that enforces the semantic consistency of a Workflow Process. You can use the Validate Tool before you simulate or deliver a Workflow Process to ensure that there are no configuration errors. For example, you can use the Validate Tool to make sure an error Workflow Process does not itself contain an error Workflow Process. When you validate a Workflow Process, warnings will be returned describing any errors, if there are any. The Validate dialog box lets you correct the errors before you run the Process Simulator or deliver the Workspace. You can keep the Validate dialog box open to view the errors (since the dialog is not modal) while you use the Process Designer to correct the problems, or you can proceed to deliver the Workflow Process without fixing the validation errors.

For more information about using the Validate Tool, see the following topics:

- *Validating the Workflow Process*
- *Troubleshooting Problems That Occur During Validation*

## Process Simulator

The *Process Simulator* is a simulation tool that allows you to step through a Workflow Process while you view the results of each step. Simulating a Workflow Process before you deliver it to its parent Workspace helps to verify that the Workflow Process produces the expected results.

For more information about how to use the Process Simulator to test that a Workflow Process functions correctly, see the following topics:

- *Using the Process Simulator*
- *Troubleshooting Problems That Occur During Simulation*

For more information about using the Business Service Simulator in situations where you cannot use the Process Simulator, see *Using the Business Service Simulator to Diagnose a Workflow Process*.

> **Note:** The Process Simulator shows a list of services that have previously been inspected in the simulator. If inspecting a business service for the first time, then it will not appear in this list so you must manually add the business service and any associated method(s) to execute the business service in the simulator.

## Event Logs

You can set monitoring levels for event logs so that you can view detailed information about a Workflow Process. This information is useful if you cannot perform real-time debugging or if the Process Simulator is not available. Event logs can result in large log files. For more information, see *Monitoring Levels for Tracing and the Event Log*. For information about using the Log File Analyzer, see *Siebel System Monitoring and Diagnostics Guide* .

# Process of Testing a Workflow Process

This process is a step in *Roadmap for Developing Workflow Processes*.

The tasks involved in the process of testing a Workflow Process are as follows:

1. *Validating the Workflow Process*
2. *Preparing to Use the Process Simulator*
3. *Using the Process Simulator*

# Validating the Workflow Process

The following procedure describes how to validate a Workflow Process. This task is a step in *Process of Testing a Workflow Process*.

# To validate a Workflow Process

1. In the Object Explorer, locate and select the Workflow Process that you want to validate.
2. Click the Applet Menu (the cogwheel icon) and then select the Validate option.
3. In the Validate dialog box that opens, click Options.
4. In the Validation Options dialog box, review the validation rules that are available and specify which rules to enforce or ignore during validation:
   a. Select/deselect the check box next to each rule that you want to enforce/ignore.
   b. (Optional) Click Advanced Options to open the Advanced Options dialog where you can select objects to exclude from (object type, Workspace or batch) validation.
   c. (Optional) Once the list of rules to enforce or ignore is set, click Save to save the information to a text file. This helps if you need to review or share the information later with your team.
   d. (Optional) Select the check box next to the following options as required:
      - **Do Not Report Warnings.** Select this option to report only errors and ignore warnings.
      - **Abort Validation After <x_number> Errors.** Select this option and then enter a number (for example 5) so that object validation will stop after <x_number> of errors are identified.
   e. Click OK.

      Note that if you select Ignore All or deselect all the rules in the Validation Options dialog, then you have chosen to ignore the rules for this Workflow Process and will receive the following message:

      ```
      There are no rules currently enforced. You may update the list
      of rules to be enforced from the 'Options' button.
      ```

5. In the Validation dialog box, click Start to start the validation process.

   Siebel starts validating the Workflow Process and returns the results in the Validation dialog when finished:

   - For example:

     `Total tests failed: 0` or

     `Total tests failed: 3`
   - Validation Errors and Warnings appear in the Validate list. Each row in the Validate list identifies a rule violation for the Workflow Process. You can correct or ignore a validation error, as follows:
     - If the Rule column in the Validate dialog box contains the number 98, then you must correct the problem. The validation engine uses the number 98 to identify an error that you must correct and cannot ignore.
     - If the Rule column in the Validate dialog box does not contain 98, then you should evaluate the possible problem, but you are not required to make a change.

     Not every error is fatal. Some errors are only warnings. For more information, see *Troubleshooting Problems That Occur During Validation*.
   - Select an error in the Validate list and then click Go To to drill down on the object that causes the error.
   - (Optional) Click Save As to save the validation results to a `validation.log` file on your local machine.

     Depending on your browser naming conventions, the validation results will be saved to `validation(1).log` if *validation.log* already exists or `validation(2).log` if *validation(1).log* already exists, and so on.

   For more information about validating objects including batch validation, see *Using Siebel Tools* .

# Using the Process Simulator

This task is a step in *Process of Testing a Workflow Process*.

The following procedure describes how to run the Process Simulator on a Workflow Process. For more information about:

- Using the simulator, see *About Simulating an Interactive Workflow Process*
- Usage guidelines, see *Guidelines for Using the Process Simulator*.
- Watch Window, see *Using the Watch Window in the Process Simulator*
- Examples that use the Process Simulator, see *Preparing Example Data for the Simulation*.

**Note:** The Process Simulator shows a list of services that have previously been inspected in the simulator. If inspecting a business service for the first time, then it will not appear in this list so you must manually add the business service and any associated method(s) to execute the business service in the simulator.

## To use the Process simulator

1. Close every Siebel client session.

   To avoid encountering a multiple client session error, you must close every client session before you start the Process Simulator. If a Siebel client session is running, then you cannot start the Process Simulator.
2. Make sure the Microsoft Taskbar does not include any Siebel client icons.
3. In the Process Designer, locate and select the Workflow Process that you want to simulate.
4. Do one of the following, as required:

   - (Siebel Tools) Right-click the Workflow Process and then select the Simulate Workflow Process option.
   - (Web Tools) Select the Workflow Process and then click Simulate Workflow Process (the Simulate icon) on the applet banner.

   In the read-only version of the Workflow Process that opens in the Simulator, the start step is highlighted. The Start Simulation button on the Simulator toolbar is also activated.
5. Click Start Simulation on the Simulator toolbar.

   The simulator starts a new instance of the Siebel client according to the debug settings you entered in *Preparing to Use the Process Simulator*. You use this Siebel client instance as the runtime environment for the simulation. You are not required to do any other work in this Siebel client instance unless the simulated Workflow Process is an interactive Workflow Process. For more information, see *About Simulating an Interactive Workflow Process*.

   After the Siebel client starts and the Simulation In Progress dialog box closes, control passes back to the Process Simulator, the start step runs and control pauses at the next step in the Workflow Process. If the first step runs as expected, then the next step in the Workflow Process is highlighted in the Process Simulator view.

   **Note:** The Simulation In Progress dialog box appears only in Siebel Tools - it does not appear in Web Tools.

6. Click Simulate Next on the Simulator toolbar to run the highlighted step.

   As you step through the Workflow Process, you can examine the results of each step (Process Properties and Business Component values) in the Watch Window until the Workflow Process finishes. For more information, see *Using the Watch Window in the Process Simulator*.

   > **Note:** In Web Tools, the Watch Window is called Workflow Process Data.

7. Continue clicking Simulate Next until the last step finishes.

   After the last step in the Workflow Process runs, the simulation automatically ends. You can also click Stop Simulation on the Simulator toolbar to stop the simulation before the last step runs.
8. If the Workflow Process does not run as expected, then correct the problem and restart the simulation.

   For more information, see *Troubleshooting Problems That Occur During Simulation*.

   > **Note:** Any changes made to a Workflow Process definition during execution requires a restart.

9. (Optional) Examine the test results in the Siebel client.

   If the Workflow Process modifies record data, you can examine test results in the Siebel client. For an example, see *Simulating the Workflow Process*.
10. Close the Simulator window.

# About Simulating an Interactive Workflow Process

If you use the Process Simulator to simulate an interactive Workflow Process, then you must perform some actions in the Siebel client while the simulator runs. For example, when *Using the Process Simulator* to simulate an interactive Workflow Process:

1. If the Process Simulator highlights a user interact step, then click Simulate Next.

   Siebel CRM opens the view that the user interact step references in the Siebel client.
2. Switch to the Siebel client and make sure the runtime event runs in the client according to the definition for the user interact step.

   Control passes back to the user after the user interact step runs successfully in the Siebel client.

   The Process Simulator then moves to the next step in the Workflow Process.

# Guidelines for Using the Process Simulator

If you use the Process Simulator, then it is recommended that you adhere to the following guidelines:

- If the Workflow Process does not contain a user interact step, then do not navigate or click UI elements in the Siebel client while the Process Simulator runs.
- Do not navigate outside of the Process Simulator while the simulation runs.
- It might be necessary to use `ALT+TAB` to return to Siebel Tools or Web Tools to proceed with the simulation after the Siebel client starts.
- After the simulation starts, it is possible that the client window will not return focus to the client. In such a case, you manually return to the simulation - for example, by using `ALT+TAB` in a Windows environment.

- You can use the Process Designer to change values in step properties, and then return to the Process Simulator to test the Workflow Process. After changing values in step properties, you must stop the Process Simulator, change these values, and then restart the simulation.
- You can change process properties at any time and continue with the simulation from that point. This allows you to test various paths through the Workflow Process without having to restart the simulator.
- You can hide the Object Explorer and the Properties pane as required. You can resize the main window so that it covers only a part of the display area.
- If you define a wait step in seconds, then the Process Simulator simulates a wait period according to the number of seconds that you define for this wait step. If you define the unit of time in minutes or greater, then the Simulator continues to the next step.
- It is not necessary for a Workflow Process to be active to run it in the Process Simulator. The simulator ignores activation date, expiration date, and status.

# Using the Watch Window in the Process Simulator

The Process Simulator includes a Watch Window (called Workflow Process Data in Web Tools) that dynamically displays values for the business component and process properties of the Workflow Process. For more information about using the Watch Window in conjunction with the Workflow Utilities business service, see *Workflow Utilities Business Service*:

## To use the Watch Window (in Siebel Tools)

1. Select the Workflow Process you want to simulate.
2. Click Start Simulation on the Simulator toolbar, and then wait for the Siebel client to start the simulation of the Workflow Process.
3. Right-click the Process Simulator window, and then click Watch Window.

   The first column in the Watch Window displays the name of the property set type. The second column displays the current value for the property.

   The Watch Window is not available until after you complete step 1. You must start the simulation and wait for control to return to Siebel Tools before you open the Watch Window.

## To use the Watch Window (in Web Tools)

**Note:** In Web Tools, the Watch Window is called Workflow Process Data.

1. Select the Workflow Process you want to simulate.
2. Click Simulate Workflow Process (the Simulate icon) on the applet banner to open the Process Simulator window.
3. (Optional) If this is the first time running the Simulator or if you need to change to a different application for simulating, then click the Workflow Process Menu (the cogwheel icon), click Application, and then select the application you want to test with.
4. Click Start Simulation on the Simulator toolbar to start the simulation of the selected Workflow Process.

   The Simulator starts the specified application and populates the Workflow Process Data (Watch) window with the initial Process Properties for the Workflow Process.

### Watch Window Property Set

The following table describes the information that is available in the Watch Window. As each step finishes during a simulation, the Watch Window displays the Workflow Process Properties and the current value(s) of affected Business Component records.

| Property Set | Description |
|---|---|
| Simulator Status | Displays real-time status information for the simulation. For example, Step Completed or Simulation Ended Successfully. |
| Process Properties | Displays the current value for each process property that is defined for the Workflow Process. For example, the value `7-4HWSV` displays for the Object Id process property. |
| BusComp | Displays business component user data for fields in the business component for the record that the simulator is currently processing. For example, the value `40000` displays for the Revenue field. |

**Note:** In Web Tools, the Watch Window is called Workflow Process Data.

## Using the Process Simulator with Different Workflow Modes

You can use the Process Simulator to test most Workflow Processes. For example, you can use the Process Simulator to test a Workflow Process that runs in the Siebel client, including a service Workflow Process, a 7.0 flow Workflow Process, an interactive Workflow Process, or a Workflow Process that a runtime event starts. You cannot use the Process Simulator to test a long-running Workflow Process or a Workflow Process that involves an asynchronous server component - see *Using the Process Simulator with a Server Component*.

## Using the Process Simulator with a Server Component

You cannot use the Process Simulator to test a long-running Workflow Process or a Workflow Process that invokes an asynchronous server component, such as, one of the following: Workflow Process Manager, Server Request Broker, Assignment Manager, or Communications Server.

Server components that execute in-line with the current Workflow Process can be invoked. For example, let's say you want to use the Outbound Communications Manager Business Service to send an email.

- If you set the Process Mode method argument to "Local", it will send the email in the same context as the Workflow Process and will work fine.
- However, if you set the Process Mode argument to "Remote", then it sends a request to the Server Request Manager to call the Outbound Communications Manager as a background job (in another process). This cannot be simulated.

Therefore, the technique for testing via Simulation is to set the Process Mode argument to "Local" and after you are sure that it works, you can change it to "Remote" if that is what you actually want.

The same applies to any similar processing scenario, where there is almost always a synchronous and an asynchronous request. The exception is Server Request Broker, where typically the server does something asynchronously.

# Using the Process Simulator to Start a Workflow Process

You can use the Process Simulator to start a Workflow Process, and then immediately test and debug it. You can debug Workflow Process steps while you define them in the Repository.

When you use the Process Simulator to start a Workflow Process, it is run in the context of an Application Object Manager (AOM) such as Siebel Call Center.

For more information, see *Starting a Workflow Process*.

# Preparing to Use the Process Simulator

The following procedure describes how to prepare to use the Process Simulator in Siebel Tools. You do not have to do anything particular in Web Tools to prepare to use the Process Simulator other than to configure the Debugger Service - see *Using Siebel Tools* for more information. This task is a step in *Process of Testing a Workflow Process*.

## To prepare to use the Process Simulator (Siebel Tools)

1. Make sure that you deliver and publish all sub processes that the Workflow Process you want to simulate calls.
2. Locate and select the Workflow Process you want to simulate.
3. Right-click the Workflow Process and then select the Simulate Workflow Process option.
4. Click View and then select the Options menu item.
5. Click the Debug tab.
6. Define the fields in the Debug tab using values from the following table, where `$` represents settings that are specific to your setup.

| Field | Value |
|---|---|
| Executable | **`$SiebelClient\BIN\siebel.exe`**<br><br>Make sure you enter the full path for the siebel.exe file. |
| CFG file | **`$SiebelClient\BIN\ENU\uagent.cfg`**<br><br>Make sure you enter the full path for the uagent.cfg configuration file. |
| Browser | **`$\Program Files\Internet Explorer\iexplore.exe`** |
| Working directory | **`$SiebelClient\BIN`** |
| Arguments | **`/h`** |

| Field | Value |
|---|---|
| | |
| User name | `$username` |
| Password | `$password` |
| Data source | `$datasource` |

7. Click OK.

# Preparing the Process Simulator for Use with a Script

**Note:** The ability to use the Process Simulator in conjunction with Siebel Server Script Debugger is currently not supported in Web Tools.

You can use the Process Simulator with a Workflow Process that references a business service or a business component that contains a script. If this script includes a breakpoint, and if the Arguments field in the debug configuration contains `/h`, then the simulator might not perform as expected. A *breakpoint* is a marker on a line of Basic code that instructs Basic to pause at that line so that the Siebel Debugger can examine the program state. The `/h` argument instructs the Siebel Debugger to open the Watch window.

## To prepare the simulator for use with a script

1. Remove every breakpoint from the script that resides on a business service or a business component that the Workflow Process references.
2. Remove the `/h` argument from the debug configuration:
   - Click the View menu, then select the Options menu item.
   - Click the Debug tab.
   - Remove the `/h` argument from the Argument field.

   For more information, see *Using Siebel Tools* .

# Consequence of Pausing the Process Simulator in DR

If you have started to test a Workflow Process (WP1_testA in WorkspaceX) using the Workflow Process Simulator in a Design Time (DR) environment and you pause the Workflow Process in the Process Simulator, then note the following:

- You will not be able to test or resume WP1_testA in a different workspace.
- If you change the Workspace context by opening a different Workspace (for example, WorkspaceY), then you will receive an error message similar to the following:

  ```
  Cannot execute the Workflow Process definition 'XX-XXXXXX' because it either
  ```

```
does not exist or is not Active, it has expired, or it was recently updated and
republished using a different Row ID (SBL-BPR-00158).
```

To resolve this, simply return to or open WorkspaceX where WP1_testA was paused.

- If you install a new monthly update while WP1_testA is paused, then you will receive an error message similar to the following:

```
Cannot execute the Workflow Process definition 'XX-XXXXXX' because it either
does not exist or is not Active, it has expired, or it was recently updated and
republished using a different Row ID (SBL-BPR-00158).
```

In this case, the paused WP1_testA instance cannot be resumed.

This only applies to DR environments – this does not occur in Runtime Repository (RR) environments such as Test, QA, or Production. This behavior is by design.

# Troubleshooting Validation and Simulation Problems

This topic describes guidelines for resolving validation and simulation problems. It includes the following subtopics:

- *Troubleshooting Problems That Occur During Validation*
- *Troubleshooting Problems That Occur During Simulation*

## Troubleshooting Problems That Occur During Validation

The following table shows how to troubleshoot some common validation problems by describing a symptom and then providing the corresponding solution to resolve the problem.

| Symptom | Diagnostic Steps or Cause | Solution |
|---|---|---|
| Connector is not attached correctly. | N/A | Make sure you connect the connectors for the Workflow Process correctly. |
| Conditional branch is not defined for a Decision step. | N/A | Make sure you define at least one conditional branch that emanates from each decision step in the Workflow Process. |
| Business service and business service method are not defined for a business service step. | N/A | Make sure each business service step in the Workflow Process is not missing a business service or a business service method. |
| Business component is missing from a Siebel Operation step. | N/A | Make sure each Siebel operation references a business component. |
| Name of the sub process is not defined for a sub process step. | N/A | Make sure each sub process step references a sub process. |

# Troubleshooting Problems That Occur During Simulation

The following table shows how to troubleshoot some common simulation problems by describing a symptom and then providing the corresponding solution to resolve the problem.

| Symptom | Diagnostic Steps or Cause | Solution |
|---|---|---|
| When attempting to start a simulation, the Process Simulator does not run. | If a runtime event is defined on the connector that emanates from a start step, then the Process Simulator waits for the event and never reaches the next step. | Do one of the following:<br>• Remove the runtime event from the simulation.<br>• Add a Default branch. It is recommended that you define two branches:<br>o One branch with Type set to Condition that waits for the runtime event<br>o One branch with Type set to Default |
| When attempting to rerun the Process Simulator, an error dialog displays one of the following messages:<br>• `Still waiting on Workflow Process Simulator`<br>• `The runtime client is not responding` | Navigating in the Siebel client during or after running the Process Simulator can cause a failure during subsequent attempts at running the simulator. | Do not navigate in the Siebel client while the Process Simulator runs or after it ends. For more information, see *Guidelines for Using the Process Simulator*. |
| After completing the user interact step, the Process Simulator fails to proceed to the next step in the Workflow Process. | A branch from a user interact step might be missing an associated runtime event. | Make sure the branches from the user interact step reference the runtime events that are associated with actions that the user performs on the user interact step. |
| After the Siebel client starts, control does not return to Siebel or a dialog box indicates that the client is still starting, even after a long wait. | N/A | Restart processes that are currently running. |
| An error message that is similar to the following message appears while the Process Simulator runs:<br>`Error loading siebel operation step definition <step_name>. The defined step definition is not valid.` | Errors in the Workflow Process configuration can occur during the simulation. These errors can result from errors in how objects are defined or how the Workflow Process uses objects. | Close the Process Simulator, correct the configuration errors, and then run the simulation again. |

# 10 Administering a Workflow Process

## Administering a Workflow Process

This chapter describes how to administer a Workflow Process. It includes the following topics:

- *Managing a Workflow Process*
- *Monitoring a Workflow Process*
- *Diagnosing a Failed Workflow Process*
- *Recovering a Workflow Process*
- *Upgrading a Workflow Process*

**Note:** If a Workflow Process is active in the Design Time Repository (DR), it will automatically be migrated to downstream Runtime Repository (RR) environments, and be available to any calling process, including runtime events.

## Managing a Workflow Process

You typically use the Workflow Instance Admin view and the Workflow Instance Monitor view to manage Workflow Processes.

- **Workflow Monitoring Configuration view.** This view allows you to configure the Monitoring Level (also Replication Level and Analytics Level) for a given Workflow Process. For more information, see *Monitoring a Workflow Process*.
- **Workflow Instance Admin view.** This view shows all live (in progress) Workflow Processes that are in a run state, wait state, or error state. This view also shows the related instances and process properties for a selected Workflow Process.
- **Workflow Instance Monitor view.** This view shows what happens after a Workflow Process runs and as such collects logging information about all Workflow Process instances, step instances and aggregate data. In this view. The Workflow Process applet lists all active Workflow Processes and the Process Instances, Step Instances, and Aggregate Data tabs show more detailed information for a selected Workflow Process.

Some typical tasks you perform when managing Workflow Processes include the following:

- *Viewing Runtime Instances of a Workflow Process*
- *Administering Workflow Process Instances*
- *Stopping a Workflow Process Instance*
- *Stopping a Workflow Process Instance with a Script*
- *Removing a Workflow Process Instance from Workflow Instance Monitor*
- *Removing a Workflow Process Instance from Workflow Instance Admin*
- *Removing a Workflow Process from the Runtime Environment*
- *Delivering a Workflow Process*

- *Deploying a Workflow Process as a Web Service*
- *Monitoring Workflow Process Instances*
- *Manually Recovering a Workflow Process Instance*

# Viewing Runtime Instances of a Workflow Process

The Workflow Instance Admin view shows all live (in progress) Workflow Processes that are in a run state, wait state, or error state. This view also shows the related instances and process properties for a selected Workflow Process.

## To view runtime instances of a Workflow Process

1. Navigate to the Administration - Business Process screen, then the Workflow Instance Admin view. This view contains the information described in the following table.

| Section Name | Description |
|---|---|
| All Workflow Process Instances | Lists all Workflow Process instances that are in a run state, wait state, or an error state. |
| Related Instances | Shows the related instances for the seleted record in the All Workflow Process Instances list. |
| Process Properties | Displays the process properties for selected Workflow Process Instance in the Related Instances list. |

2. In the All Workflow Process Instances list, query the Process Name field for the Workflow Process you want. The related instances and process properties for the selected Workflow Process appear in the Related Instances and Process Properties sections respectively.

# Administering Workflow Process Instances

The Workflow Instance Admin view displays Workflow Processes that have persistence set and that are in a run state, wait state, or an error state. If a Workflow Process contains a wait step, or if the Auto Persist property of a Workflow Process contains a check mark, then Siebel CRM sets persistence for a 7.0 Workflow Process.

## To administer Workflow Process instances

1. Navigate to the Administration - Business Process screen, then the Workflow Instance Admin view.
2. In the All Workflow Process Instances list, query the Process Name field for the Workflow Process you want.
3. In the Related Instances section, administer (stop, start, and resume) the Workflow Process instances as required. For more information, see the following topics:
   - *Stopping a Workflow Process Instance*
   - *Manually Recovering a Workflow Process Instance*
4. Modify the process properties for a selected Workflow Process instance as required:
   - In the Related Instances section, select the Workflow Process instance that you want.

- In the Process Properties section, change the Process Properties values as required.

  You should change the value of these process properties before you resume a Workflow Process instance that is in a wait or an error state.

# Stopping a Workflow Process Instance

If persistence is defined for a Workflow Process, then you can stop an instance of that Workflow Process. Siebel CRM removes the instance after it stops. Stopping an instance is different from purging an instance. For more information, see *Removing a Workflow Process Instance from Workflow Instance Monitor*.

**CAUTION:** Siebel CRM cannot resume a stopped instance of a Workflow Process.

## To stop a Workflow Process instance

1. Navigate to the Administration - Business Process screen, and then the Workflow Instance Admin view.
2. In the All Workflow Process Instances list, query the Process Name field for the Workflow Process you want to stop.
3. In the Related Instances list, select the instance you want to stop.
4. Click the Related Instances Menu (the cogwheel icon) and select Stop Instance.

# Stopping a Workflow Process Instance with a Script

You can use the `_StopInstance` method in a script to stop a Workflow Process instance. For example, assume you must stop an instance of an interactive Workflow Process that the user paused or that is suspended on a wait step. In this situation, Siebel CRM cannot determine the Process Instance Id from the script. You must know the Process Instance Id in order to use the `_StopInstance` method.

## To stop a Workflow Process instance with a script

- Call the `_StopInstance` method on the Workflow Process Manager business service.

  The following example hard codes the Process Instance Id:

```
var bs_WF = TheApplication().GetService("Workflow Process Manager");
var ps_inputs = TheApplication().NewPropertySet();
var ps_outputs = TheApplication().NewPropertySet();
ps_inputs.SetProperty("ProcessInstanceId", "1-IIT");
bs_WF.InvokeMethod("_StopInstance" , ps_inputs, ps_outputs);
```

# Removing a Workflow Process Instance from Workflow Instance Monitor

If a Workflow Process instance contains a status of Stopped, Completed, Completed Abnormal or In Error, then you can remove it from the Workflow Instance Monitor view. If you must remove a Workflow Process instance that is paused, then you must first stop the instance.

**Note:** Workflow Instances that are **In Error** are deleted regardless of the date you specify. All other status codes will use your specified date.

## To remove a Workflow Process Instance from Workflow Instance Monitor

1. Navigate to the Administration - Business Process screen, Workflow Instance Monitor view, then the Process Instances view.
2. In the Process Instances list, select the Workflow Process Instance that you want and click Purge.
3. In the Workflow Instance Monitor Purge dialog box that appears, specify a date.
4. Click Purge.

   The selected Workflow Process Instance is removed and will no longer appear in the Workflow Instance Monitor view.

# Removing a Workflow Process Instance from Workflow Instance Admin

If a Workflow Process instance contains a status of Stopped, Completed, Completed Abnormal or In Error, then you can remove it from the Workflow Instance Admin view. If you must remove a Workflow Process instance that is paused, then you must first stop the instance.

**Note:** Workflow Instances that are **In Error** are deleted regardless of the date you specify. All other status codes will use your specified date.

## To remove a Workflow Process Instance from Workflow Instance Admin

1. Navigate to the Administration - Business Process screen, Workflow Instance Admin view, then the Process Instances view.
2. Click Purge in the All Workflow Process Instances Applet.
3. In the Workflow Instance Purge dialog box that appears, specify a date.
4. Click Purge.

   The Workflow Instances with the said status and before the specified date are removed from the Workflow Process Instances and will no longer appear in the Workflow Instance Admin view.

# Removing a Workflow Process from the Runtime Environment

Use the following procedure to remove a Workflow Process from the runtime environment.

## To remove a Workflow Process from the runtime environment

1. Deactivate the Workflow Process in the development environment.
    a. In the Object Explorer for the development environment, click Workflow Processes, and select the Workflow Process that you want to deactivate in the Workflow Processes list.
    b. In the Status field, set the value to Inactive (Siebel Tools) or Y (Web Tools).
2. Run migration from the development environment to the Runtime Repository environment.

    The Workflow Process in the runtime environment is now removed.

# Delivering a Workflow Process

Delivering a Workflow Process to its parent integration branch makes it active in any Application Object Manager (AOM) with that Workspace Context. For more information about Workspaces, see *Using Siebel Tools* .

The steps involved in delivering a Workflow Process are:

1. Modify the Workflow Process in the Developer Workspace.
2. Test the updated Workflow Process.
3. Deliver the Workspace to its parent Integration Workspace.

On delivering the Workflow Process, the following happens:

- The runtime definition of the Workflow Process is created as an entry in the S_RR_WORKFLOW table.
- If a runtime event is used to start the Workflow Process, then runtime events are created at that time.

  Note that during FullPublish, the S_RR_WORKFLOW table and related runtime event tables are reset to reflect the version of the Workflow Process that remains in the MAIN Workspace.

**Note:** If there are different runtime definitions of the same Workflow Process, modified in different Workspaces, then the appropriate definition will be picked up based on the Workspace Context defined in the AOM definition or selected by the user from the Workspace Dashboard at runtime.

# Deploying a Workflow Process as a Web Service

You can deploy a Workflow Process as a Web service.

**Note:** This task applies only to Siebel Tools.

## To deploy a Workflow Process as a Web service

1. In the Workflow Process Designer, examine the definitions in the Multivalue Property Window to determine if the Data Type field for a process property is set to Integration Object. If so, make sure a value is defined in the Integration Object field for each of these process properties.

   If the Workflow Process uses hierarchical data in a business service, then you must set Data Type for the process property to Integration Object, and then define an integration object in the Integration Object field. If no integration object is defined, then an error that is similar to the following occurs:

   ```
   The selected Workflow Process contains hierarchy type process properties without
   having the integration object name defined.
   ```

   For more information, see *Using Process Properties*.
2. Locate and select the Workflow Process that you want to deploy as a Web service.
3. Right-click the Workflow Process and then select the Deploy as a Web Service option.
4. In the Expose Workflow Process as Web Service dialog box that appears:
   - Define the operation name for the new Web service. You can define the underlying method name of the business service without spaces, such as CreateAccount. Use a business service method that is defined in the Workflow Process.
   - Define the URL that identifies the address for the Web service. Replace webserver with a valid host name and lang with a valid language code, such as enu.
5. (Optional) To create a Web Services Description Language (WSDL) file, select the Generate WSDL check box, and then specify a location to save the WSDL file.
6. Click Finish.

   The Deploy as a Web Service Wizard creates the Web service.
7. To view the Web service, do the following:
   a. Navigate to the Administration - Web Services screen in the Siebel client, then the Inbound Web Services view.
   b. In the Inbound Web Services list, query the Name field for the Workflow Process name you specified in step 4.

   To display Web services that Siebel CRM creates from Workflow Processes and business services, query the Namespace field for the following value:
   ```
   http://siebel.com/CustomUI
   ```

For more information, see *Integration Platform Technologies: Siebel Enterprise Application Integration* .

# Monitoring a Workflow Process

This topic describes how to monitor and troubleshoot a Workflow Process in a production environment.

> **Note:** Workflow Process monitoring is not available when inspecting a Developer Workspace. To debug a Workflow Process in a Developer Workspace, use the Process Simulator. For more information, see *Using the Process Simulator*.

The following tools and views are available to monitor and troubleshoot a Workflow Process:

- Workflow Monitoring Configuration view, Workflow Instance Monitor view, and Workflow Instance Admin view The Workflow Monitoring Configuration view allows you to configure the Monitoring Level (see *Setting Monitoring Levels for a Workflow Process*), Replication Level (see *Deploying a Workflow Process to Siebel Mobile Web Client*) and Analytics Level (see *Collecting Data About Workflow Process Properties*) for a given Workflow Process.
- Event logging
- WF Process Metric object
- SARM (Siebel Application Response Measurement)
- Siebel FDR (Flight Data Recorder)

Use these tools and views to monitor and troubleshoot the following for a Workflow Process:

- **Progress and status information.** Use the Workflow Instance Monitor view and the Workflow Instance Admin view to monitor the progress and status of each Workflow Process. For more information, see *Setting Monitoring Levels for a Workflow Process*.
- **Operation details.** Use event logging to trace operations that each Workflow Process step performs. For more information, see *Monitoring Levels for Tracing and the Event Log* and *Increasing Tracing Levels for Server Components of the Workflow Management Server*.
- **Workflow Process metric information.** Use the WF Process Metrics object to collect data that is associated with the properties of the Workflow Process. For more information, see *Collecting Data About Workflow Process Properties*.
- **Performance measurement data.** Use SARM to capture timing data that measures the performance of a Workflow Process. For more information, see *Collecting Timing Data*.
- **Failure analysis records.** Use Siebel FDR to capture the events that might cause a system failure. For more information, see *Collecting Data About System Behavior and Server Component Behavior*.

# Setting Monitoring Levels for a Workflow Process

The monitoring level you set determines the frequency that Siebel CRM uses to log data and the data that the views display for a Workflow Process instance. For more information, see the following topics:

- *Setting the Monitoring Level Parameter*
- *Guidelines for Setting the Monitoring Level*
- *Monitoring Level and Compatibility with the 7.0 Workflow Process*
- *Monitoring Workflow Process Instances*

## Setting the Monitoring Level Parameter

The following procedure describes how to set the monitoring level parameter for a Workflow Process.

When Siebel CRM creates a Workflow Process instance, it reads the monitoring level from the Workflow Process. This level remains constant throughout the lifetime of the instance unless Siebel CRM pauses it. If Siebel CRM pauses an instance, then it rereads the monitoring level from the Workflow Process when it resumes the instance.

### To set the monitoring level parameter

1. In the Siebel client, navigate to the Administration - Business Process screen, then the Workflow Monitoring Configuration view.

2. In the Monitoring Configuration list, query the Name field for the Workflow Process that you want.
3. Set the required value in the Monitoring Level field.

   A monitoring level of 0 (None) indicates that monitoring is turned off for the Workflow Process. A monitoring level of 1, 2, 3, or 4 indicates that monitoring is turned on for the Workflow Process.

   The following table describes the different Monitoring Levels you can set and the frequency with which a Workflow Process instance, a Workflow Process step instance, and the Workflow Process properties are recorded..

| Monitoring Level | Record Process Instance | Record Step Instances | Record Process Properties |
|---|---|---|---|
| 0 None | No | None | None |
| 1 Status | Yes | None | None |
| 2 Progress | Yes | All Steps | None |
| 3 Detail | Yes | All Steps | All Steps |
| 4 Debug | Yes | All Steps | All Steps |

## Guidelines for Setting the Monitoring Level

The monitoring level determines how frequently Siebel CRM saves data to the disk. The runtime environment for Siebel Workflow sets this frequency according to the type of Workflow Process and the monitoring level you choose. Monitoring can cause a performance overhead. It is recommended that you set the monitoring level to one of the following values:

- **0 (None) or 1 (Status)** for a Workflow Process that Siebel CRM runs in a production environment.
- **2 (Progress) or higher** only to debug a Workflow Process. If you set the monitoring level to 4 (Debug), then Siebel CRM saves the Workflow Process state to the database after every Workflow Process step.

## Monitoring Level and Compatibility with the 7.0 Workflow Process

Siebel CRM sets monitoring levels for a 7.0 Workflow Process according to the following value of the persistence frequency:

- **NONE.** Siebel CRM sets the monitoring level to NONE.
- **ON_PAUSE or EVERY_STEP.** Siebel CRM turns persistence on and sets the monitoring level according to the following value of the persistence level:
- **ALL_STEPS.** Siebel CRM sets the monitoring level to PROGRESS.
- **CURRENT_STATE.** Siebel CRM sets the monitoring level to STATUS.

Beginning with the Siebel CRM 8.1 release, persistence and monitoring are separate features that serve different purposes. Persistence affects quality of service and is controlled during development. Monitoring is an administrative tool and is controlled during deployment. Monitoring typically does not affect the functionality of a Workflow Process.

## Monitoring Workflow Process Instances

The Workflow Instance Monitor view allows you to monitor the progress and status of Workflow Process instances, step instances, and aggregate data.

If monitoring is turned on for a Workflow Process, then the Workflow Process instance remains in the Workflow Instance Monitor view after it finishes and Siebel CRM no longer displays it in the Workflow Instance Admin view. Siebel CRM might not display any records in the Step Instances or Aggregate Data views, depending on the monitoring level that is set for the Workflow Process you choose.

### To monitor instances of a Workflow Process

1. Navigate to the Administration - Business Process screen, then the Workflow Instance Monitor view.

   The Workflow Process list that appears displays the definitions of Workflow Processes (according to Name, Version, Business Object, Status, and so on) that have monitoring turned on. A monitoring level of 1, 2, 3, or 4 indicates that monitoring is turned on for a Workflow Process. For more information, see *Setting the Monitoring Level Parameter*.
2. In the Workflow Process list, query the Name field for the Workflow Process that you want.
3. Monitor and review the progress and status information for the selected Workflow Process as required. The following table describes the view tabs and corresponding information available for a Workflow Process.

| View Tab | Description |
|---|---|
| Process Instances | Displays the related log instances for the selected Workflow Process. |
| Step Instances | Displays the steps and process properties for the selected Workflow Process. |
| Aggregate Data | Displays aggregate data as a chart view for the selected Workflow Process. |

   For more information, see *Removing a Workflow Process Instance from Workflow Instance Monitor*.

# Monitoring Levels for Tracing and the Event Log

You can use tracing levels to troubleshoot a Workflow Process. Setting a trace level over the default level affects performance. It is recommended that you reset the trace level back to the default value after you finish troubleshooting.

The following table describes the *Events* that are monitored for tracing and logging purposes. The corresponding trace *Level* for each event is also specified.

You must specify event log levels as follows:

- On the Workflow Process Manager component for asynchronous Workflow Processes.
- On the Application Object Manager (for example, Call Center) for synchronous Workflow Processes.

| Event | Level | Description |
|---|---|---|
| Workflow Definition Loading (DfnLoad) | 3 | Traces object definitions for Workflow Processes and Workflow Process steps that Siebel CRM loads into memory. |
| Workflow Engine Invoked (EngInv) | 4 | Traces methods that Siebel CRM calls and arguments that it passes to the Workflow Engine. |
| Workflow Process Execution (PrcExec) | 4 | Traces work that Siebel CRM does to create and complete Workflow Process instances. Traces one of the following process properties: Get or Set. |
| Workflow Step Execution (StpExec) | 4 | Traces work that Siebel CRM does to create and complete Workflow Process steps, evaluate decision conditions, call a business service, and insert or update a business component. |
| Workflow Performance (WfPerf) | 4 | Measures overall runtime for a Workflow Process. |
| Workflow Performance (WfPerf) | 5 | Measures runtime of the Workflow Process and of individual Workflow Process steps. |
| Workflow Recovery (WfRecv) | 3 | Traces recovery status and progress for a Workflow Process instance. Applicable only to the Workflow Recovery Manager server component. |
| Workflow Recovery (WfRecv) | 4 | Traces recovery details for a Workflow Process instance. Applicable only to the Workflow Recovery Manager server component. |

# Increasing Tracing Levels for Server Components of the Workflow Management Server

You can configure Siebel CRM to create a more detailed trace file that you can use to troubleshoot the Workflow Process Manager, Workflow Process Batch Manager, or Workflow Recovery Manager. You can increase tracing levels before the Siebel Server process runs.

## To increase tracing levels for server components of the workflow management server

1. In the Siebel client, navigate to the Administration - Server Configuration screen, Servers, Components, and then the Events view.
2. In the Components list, select the server component that Siebel CRM must trace.

   These server components include the Workflow Process Manager, the Workflow Process Batch Manager, or the Workflow Recovery Manager.
3. Go to the Events tab to view the configurable event types for the server component.
4. Change the value in the Log Level field to 3, 4, or 5.

   By default, Siebel CRM sets the Log Level value to 1. For more information about log levels, see *Setting Monitoring Levels for a Workflow Process* and *Monitoring Levels for Tracing and the Event Log*.

5. (Optional) If required, repeat step 4 to increase the tracing level for the Object Manager SQL log event to continue troubleshooting.

Siebel CRM creates more tracing information as you increase the value of the Component Event Configuration Log Level. For more information about the different log level values that you can set, see *Siebel System Monitoring and Diagnostics Guide* .

# Collecting Data About the Workflow Process Properties

A *Workflow Process metric* is a feature that allows you to collect data that is associated with the property of the Workflow Process. To collect metrics, you define the subset of metrics that Siebel CRM captures, and then activate metric collection after you deploy the Siebel CRM application.

## To collect data about Workflow Process properties

1. In the Object Explorer, go to the WF Process Metrics object type.
2. Locate and select the Workflow Process you want to monitor.
3. In the Object Explorer, expand the Workflow Process object, and then click WF Process Metric.
4. Create a new record:.
   - In the Metric Name property, select a metric from the list of predefined metric names.
   - For the Property Name property, select a process property from the list of process properties that are defined for the Workflow Process.
   - Make sure the Inactive property does not contain a check mark.

     The Inactive property allows you to turn off a metric and turn it on later.
5. Deliver the Workflow Process.
6. In the Siebel client, navigate to the Administration - Business Process screen, then the Workflow Monitoring Configuration view and do the following:
   a. In the Monitoring Configuration list, query the Name field for the Workflow Process that you delivered in step 5.
   b. In the Analytics Level field, select either Property or All.

   This step allows Siebel CRM to collect metrics data at runtime for the specified Workflow Process.

# Collecting Timing Data

The Workflow Process Manager server component uses Siebel Application Response Measurement (SARM) to collect timing data. You can also use this data to monitor Siebel CRM performance. SARM saves this information to binary files. For more information about configuring SARM and the SARM analyzer, see *Siebel Performance Tuning Guide* .

The following table describes the SARM levels (SARM Level 1 and 2) that apply to a Workflow Process.

| SARM Level | Description |
|---|---|
| 1 | The Workflow Process Manager business service records the time that Siebel CRM requires to run a business service method. |

| SARM Level | Description |
|---|---|
| | RunProcess and ResumeInstance are examples of business service methods that run in the Workflow Process Manager. |
| 2 | The Workflow Process Manager business service records the time required to do the following:<br>• Run a business service method<br>• Run a Workflow Process step<br>• Write monitoring data to the disk<br>SARM level 2 can help you measure the logging overhead that occurs if you increase the monitoring level of a Workflow Process. |

The following table describes the area and subareas that apply to each SARM level.

| Level | Area | Subarea | Description |
|---|---|---|---|
| 1 | WORKFLOW | CORDR_RESUME | Resume a paused Workflow Process. |
| 1 | WORKFLOW | CORDR_EXECUTE | Run a Workflow Process. |
| 1 | WORKFLOW | ENGNE_INVOKE | Call a method of the Workflow Process Manager business service. |
| 2 | WORKFLOW | STEPS_EXSTEP | Run a Workflow Process step. |
| 2 | WORKFLOW | MONTR_WRTE | Write monitoring data of a Workflow Process to disk. |

# Collecting Data About System Behavior and Server Component Behavior

Log files that the Siebel Flight Data Recorder (FDR) creates include information about system behavior and server component behavior that occurs at runtime. Siebel CRM logs the settings and events that occur that might contribute to a failure. The `.fdr` extension identifies these log files. You can use these `.fdr` files to troubleshoot and analyze the settings and events that occur prior to the failure. Siebel CRM stores `.fdr` log files in the following directory:

```
ORACLE_HOME\binary
```

where:

- `ORACLE_HOME` is the root directory where you installed the Siebel Server.

The FDR instrumentation points that Siebel CRM uses in the Workflow Process Manager business service and the Workflow Recovery Manager business service collects detailed data. For more information, see *Siebel System Monitoring and Diagnostics Guide* .

# Deploying a Workflow Process to Siebel Mobile Web Client

The Replication field in the Workflow Monitoring Configuration view allows you to configure Siebel CRM to route a Workflow Process to the Siebel Mobile Web Client. Routing only those Workflow Processes that the Siebel Mobile Web Client requires allows you to reduce the amount of data that the local database stores.

## To deploy a Workflow Process to Siebel Mobile Web Client

1. Navigate to the Administration - Business Process screen, then the Workflow Monitoring Configuration view.
2. In the Monitoring Configuration list, select the record that you want.
3. In the Replication field, select All.

   The available replication values are described in the following table.

| Value | Description |
|---|---|
| All | Siebel CRM routes the Workflow Process to Siebel Mobile Web Clients and regional nodes.<br><br>Changing the value in the Replication field from None to All adds the Workflow Process and related records to the Siebel Mobile Web Client or regional node when it synchronizes with the Siebel Server. |
| Regional | Siebel CRM routes the Workflow Process to the regional nodes only. |
| None | Siebel CRM does not route the Workflow Process to the Siebel Mobile Web Client or the regional nodes. |

## Testing Routing Behavior with Full Copy Nodes

If Siebel CRM extracts a regional node with the routing group set to FULL COPY, then it routes a Workflow Process with Replication set to None to the Siebel Mobile Web Client.

### To test routing behavior with full copy nodes

1. Modify two existing or define two new Workflow Processes, one with Replication set to All and the other with Replication set to None.
2. Extract a regional node with the routing group set to FULL COPY.
3. Go to the Workflow Monitoring Configuration view on the Replication Node and verify that the Workflow Process is there.

## Using a Runtime Event with a Mobile Client

Some variation exists in the way a runtime event performs in a Siebel Mobile Web Client when compared with other Siebel client types. If you configure Siebel CRM to use a runtime event with a mobile client, then it is recommended that you set the processing mode to Local Synchronous or Remote Asynchronous. If Siebel CRM uses Remote Asynchronous

for a mobile client, then it starts the Workflow Process after it synchronizes with the Siebel Server. For more information, see *Remote Synchronous Processing*.

## Notifying Mobile Users Who Are Not Synchronized

You can define a Workflow Process that sends a notification email to mobile users who are not synchronized. For example, the following procedure shows how to send an email to Siebel Remote Users who have not synchronized recently.

### To notify mobile users who are not synchronized

1. Create a new Business Object named Sync Status.
2. Add the following Business Object Components:
   - Sync Doc Status
   - Sync Client Status
3. On the Sync Status business object, specify Sync Doc Status as the Primary Business Component
4. On the Business Component definition for Sync Doc Status, do the following:
   a. Add two new fields named Emp Id and Emp Email Address. Do not populate any attributes, other than the name at this point.
   b. Create a new join named S_CONTACT with the join specification shown in the following table.

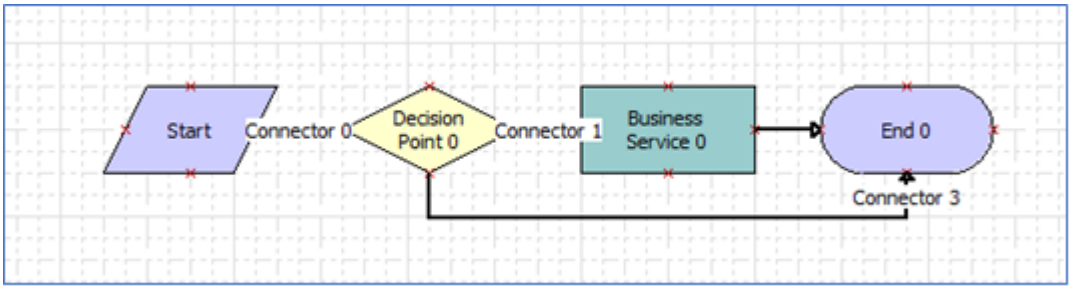
| Name | Destination Column | Source Field |
|---|---|---|
| Id | ROW_ID | Emp Id |

   c. Update the Emp Id and Emp Email Address fields as shown in the following table.

| Name | Join | Column |
|---|---|---|
| Emp Id | S_NODE | EMP_ID |
| Emp Email Address | S_CONTACT | EMAIL_ADDR |

5. Create a Workflow Process named Remote Client Sync Email with the Business Object as Sync Status.
6. Add the following steps and connectors until the (Remote Client Sync Email) Workflow Process resembles the flow illustrated in the following figure:
   a. A Start step.
   b. A Decision Point 0 step.
   c. A Business Service 0 step.
   d. An End 0 step.
   e. The following connectors:
      - Connector 0 between step a and step b.
      - Connector 1 between step b and step c.

- Connector between step c and step d.
- Connector 3 between step b and step d.




7. Modify the (Remote Client Sync Email) Workflow Process as follows:

   a. Start step - No changes.
   b. Connector 0 - No changes.
   c. Decision Point 0 step - Rename to "Delinquent Sync?".
   d. Connector 1 - Rename to "Yes" and add the conditions specified in the following table.

| Compare To | Operation | Business Component Name | Business Component Fields | Value |
|---|---|---|---|---|
| Business Component | One Must Match (Ignore Case) | Sync Doc Status | LOCAL_FLG | `Y` |
| Business Component | One Must Match (Ignore Case) | Sync Doc Status | TYPE | `SESSION` |
| Expression | One Must Match (Ignore Case) | Sync Client Status | N/A | `([Effective End Date] > Today() or [Effective End Date] is NULL)` |
| Expression | One Must Match (Ignore Case) | Sync Client Status | N/A | `(Today() - [LAST_UPD] >= 1 and [LAST_UPD] <= Today())` |

> **Note:** In the Expression line, change the `>=1` to `>=N`, where N is the number of days you consider the user to be delinquent in synchronizing.

e. Connector 3:
   - Rename to "No"
   - Change the Type attribute to "Default"

f. Business Service 0 step:
   - Rename to "Send Reminder Email"
   - Select the "Outbound Communications Manager" Business Service and the "Send Message" Method.
   - Configure the Input Arguments as shown in the following table. Note that additional arguments may be added if required.

| Input Argument | Type | Value | Business Component Name | Business Component Field |
|---|---|---|---|---|
| Communication Profile | Literal | <specify a Communications Profile> | N/A | N/A |
| To Address List | Business Component | N/A | Sync dock Status | Emp Email Addr |
| HTML Body | Literal | <reminder message of your choice> | N/A | N/A |

g. End 0 step - No changes required, but you can rename to simply "End".

8. Create a Server Job for component "Workflow Process Batch Manager" for the "Remote Client Sync Email" workflow to run daily (or at the frequency required).

# Diagnosing a Failed Workflow Process

This section includes following subtopics:

- *Diagnosing a Failed Workflow Process in a Production Environment*
- *Troubleshooting Problems That Occur When a Workflow Process Runs*

## Diagnosing a Failed Workflow Process in a Production Environment

You can use the following tools to diagnose Workflow Processes that fail in a production environment:

- Tracing and Instance Monitoring. You can turn on tracing for the server component that runs the Workflow Process. Example server components include the Workflow Process Manager, Workflow Process Batch Manager, or the Application Object Manager. For more information, see the following topics:
  - *Setting the Monitoring Level Parameter*
  - *Monitoring Levels for Tracing and the Event Log*

- *Increasing Tracing Levels for Server Components of the Workflow Management Server*

- Business Service Simulator - for more information, see *Using the Business Service Simulator to Diagnose a Workflow Process*.

## Using the Business Service Simulator to Diagnose a Workflow Process

You can use the Business Service Simulator to diagnose a Workflow Process that has failed.

### To use the Business Service Simulator to diagnose a Workflow Process

1. Run the Workflow Process from the Business Service Simulator using the Workflow Process Manager business service.

   This step runs the Workflow Process in the Application Object Manager.
2. In the Siebel client, navigate to the Administration - Business Service screen, then the Simulator view.
3. In the Simulator list, create a new record with the values shown in the following table.

| Field | Value |
|---|---|
| Service Name | Workflow Process Manager |
| Method Name | RunProcess |
| Iterations | 1 |

4. In the Input Arguments list, create a new record, and then do the following:
   a. Set the Test Case # field to 1.
   b. Select and open the Property Name field.
5. In the multivalue applet that appears, click New, and then set the field values as shown in the following table.

| Field | Value |
|---|---|
| Property Name | ProcessName |
| Value | <Type in the name of the Workflow Process> |

6. Click Save.
7. Repeat step 5 and 6 as necessary for other parameters, including RowId, that Siebel CRM passes to the Workflow Process.
8. In the multivalue applet, click OK.
9. In the Simulator list of the Simulator view, click Run.

**Tip:** To increase the data that is available to you for troubleshooting, you can set the monitoring level to 4-Debug, start the Workflow Process with the Business Service Simulator, and then review the information about the Workflow Process in the Workflow Instance Monitor view.

## Avoiding Excessive Records in the S_WF_PROP_VAL Table

The S_WF_PROP_VAL table stores the values of the process properties for a Workflow Process. When a Workflow Process runs, Siebel CRM creates records in the S_WF_PROP_VAL table along with a new S_WF_STEP_INST record.

The S_WF_PROP_VAL table might become very large over time because a Workflow Process typically contains five or more process properties. As a result, Siebel CRM might add five records to the S_WF_PROP_VAL table for each Workflow Process instance. If you enable Persistence on a large number of Workflow Processes, then the S_WF_PROP_VAL table might become very large. To avoid this situation, you can disable persistence on your custom Workflow Processes.

### To avoid excessive records in the S_WF_PROP_VAL table

- To disable persistence, set the Auto Persist property to NO on your custom Workflow Processes.

  For more information, see *Configuring Events*.

# Troubleshooting Problems That Occur When a Workflow Process Runs

The following table describes some common problems that might occur when a Workflow Process runs and how to resolve those problems.

| Symptom | Diagnostic Steps and Cause | Solution |
|---|---|---|
| The DisplayApplet Runtime Event starts a Workflow Process the first time the applet is accessed, but not on subsequent times the Applet is accessed. | The DisplayApplet Runtime Event is a UI event. If the default Web UI framework uses a cache, then Siebel only starts the event the first time it accesses a non-cached view. | To make the Runtime Event start each time the Applet is accessed, set the EnableViewCache parameter in the .cfg file to FALSE. |
| After Siebel CRM starts a Workflow Process from a runtime event, it does not get the Row Id of the record on which the event occurs. | Runtime Events pass the Row Id of the Primary Business Component in the active business Object to the Workflow Process "Object Id" Process Property. For more information, see *Defining the Primary Business Component*. | To get the Row Id of the active business component, you can use a search specification. |
| Siebel CRM displays the following error message:<br><br>`Cannot resume Process x-xxxxx for Object-id x-xxxxx. Please verify that the process exists and has a waiting status.` | Deleting existing instances of the Workflow Process does not help. This error typically occurs in the following situation:<br><br>1. A Workflow Process instance starts and pauses, waiting for a runtime event.<br>2. The runtime event starts. Siebel CRM resumes the instance and runs it to completion.<br>3. The runtime event starts a second time. The Workflow Engine attempts to resume the instance but fails because the instance is no longer in a Waiting state.<br>These steps must occur in the sequence described and in the same user session in order for Siebel CRM to report the error message. As a result, the | Ignore the error message and proceed.<br><br>The Purge feature only works on instances that Siebel CRM has stopped and finished. To delete a persistent or incomplete instance, you first must manually stop the instance. |

| Symptom | Diagnostic Steps and Cause | Solution |
|---|---|---|
| | error message disappears when the Siebel CRM application restarts. | |
| Unable to access a different business object from a Workflow Process. | Siebel Workflow Processes are constrained to a single business object. | Use a sub process step to access a different business object. |
| Siebel CRM displays the following error message:<br><br>`Unable to initiate the process definition process_name.` | N/A | Make sure that each of the following is true:<br><br>• The Workflow Process exists.<br>• The repository definition of the Workflow Process is not marked as Inactive. |
| Siebel CRM displays the following error message:<br><br>`The selected record has been modified by another user since it was retrieved.` | A Workflow Process attempted to update a record that another user or task updated since Siebel CRM initially retrieved the Workflow Process. | Define an error exception connector to handle the update conflict. For more information, see *Defining an Error Exception Connector to Handle an Update Conflict*. |

# Recovering a Workflow Process

This section describes how to use the Workflow Recovery Manager to recover a Workflow Process. It includes the following topics:

- *About the Workflow Recovery Manager*
- *Architecture of the Workflow Recovery Manager*
- *Using the Workflow Recovery Manager*
- *Recovering Workflow Process Instances*
- *Guidelines for Recovering a Workflow Process*

## About the Workflow Recovery Manager

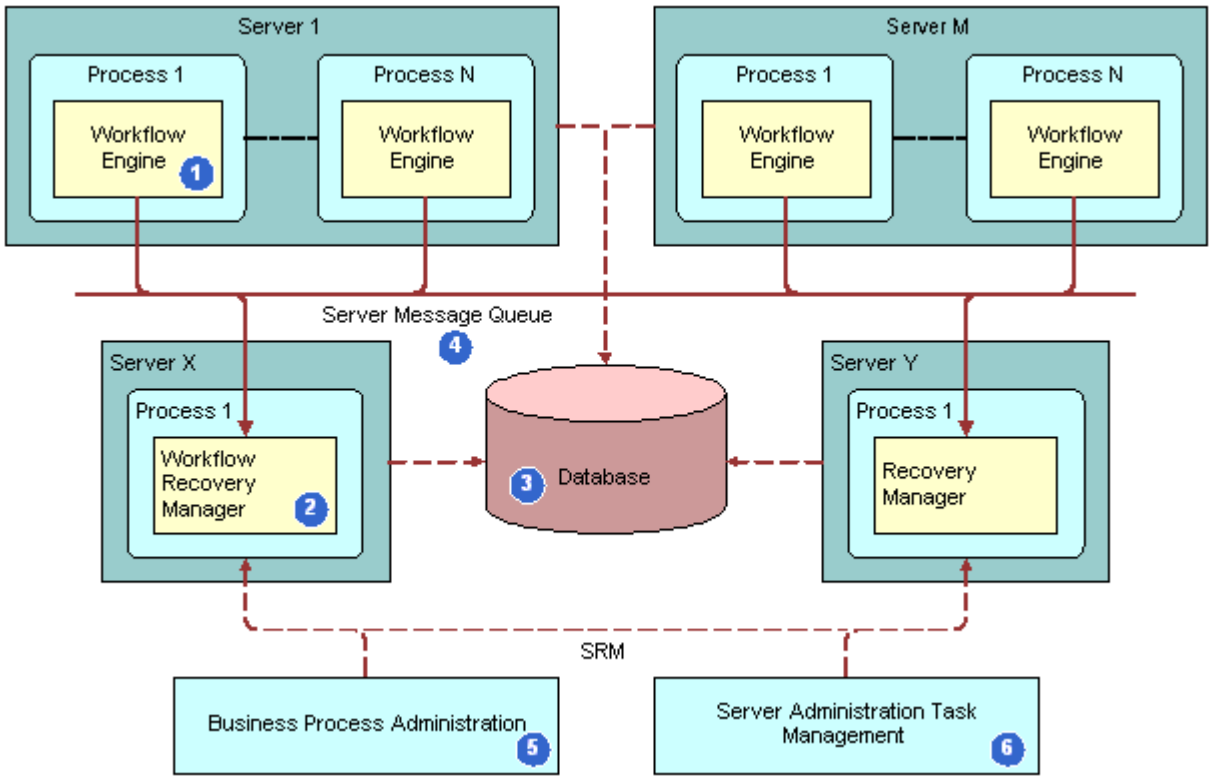
The Workflow Recovery Manager provides the following key functionality:

- Recovers interrupted instances of a long-running Workflow Process if the Siebel Server fails. When Siebel CRM recovers an instance, the Workflow Engine attempts to continue running this instance. If forward progress is not possible, then Siebel CRM marks the instance as IN_ERROR and the user must manually resume the Workflow Process.
- Siebel CRM resumes a Workflow Process instance at the last checkpoint. The Workflow Engine determines the checkpoint according to hints that you define during development. The Workflow Engine can persist relevant data, and then use it to resume running the Workflow Process after a failure.

- Siebel CRM load balances the work it performs to recover Workflow Process instances. One server thread can determine the recovered instances. This server thread delegates resuming the recovered instance to other server threads in a load balanced manner.
- Siebel CRM can recover a Workflow Process without interfering with the Workflow Engine. One server thread can recover Workflow Process instances while other threads in the Workflow Engine still handle new requests to run Workflow Process.

# Architecture of the Workflow Recovery Manager

The following figure illustrates the architecture that Siebel CRM uses to recover a Workflow Process.




As shown in this figure, the Workflow Recovery Manager architecture includes the following components:

1. **Workflow Engine.** The Workflow Process Manager server component runs a long-running Workflow Process. As this Workflow Process runs, the Workflow Engine saves the execution state to the Siebel database at the appropriate time.
2. **Workflow Recovery Manager.** A batch server component identifies the Workflow Process instance that a server failure interrupts. If Workflow Recovery Manager detects an instance that is interrupted, then it forwards this instance to the Workflow Engine. The Workflow Recovery Manager does not rerun the Workflow Process instance.
3. **Database.** A database stores the run state of the Workflow Process. Siebel CRM uses the persistent record to restore the run state when the Workflow Process resumes running from a failure.
4. **Server Message Queue.** A queue through which the Workflow Engine and the Workflow Recovery Managers sends messages.
5. **Business Process Administration View.** A view that allows you to manually send a request to the Workflow Recovery Manager to recover an interrupted Workflow Process.

6. **Server Administration Task Management.** Allows you to define a recurrence server task for the Workflow Recovery Manager. The recurrent server task requests the Server Manager to periodically scan for interrupted Workflow Processes.

# Using the Workflow Recovery Manager

You use the Workflow Recovery Manager to perform the following tasks:

1. *Enabling the Workflow Recovery Manager*
2. *Preparing to Test the Workflow Recovery Manager*

## Enabling the Workflow Recovery Manager

This task is a step in *Recovering a Workflow Process*.

The status of the Workflow Recovery Manager is Online after it starts. Siebel CRM does not display any server tasks unless it recovers a Workflow Process instance after a server failure. If you run the following command, and if Siebel CRM returns no server tasks, then this situation indicates that there are no failed instances, which is expected behavior:

```
List Tasks For Comp Wfrecvmgr
```

A Workflow Recovery Manager that is in an Active status indicates that it is up and running.

### To enable the Workflow Recovery Manager

1. In the Server Manager command-line interface, enter the following command:

   ```
   enable compgrp workflow
   ```

   This step makes sure that Siebel CRM activates the component group for Siebel Workflow on the Siebel Server. For more information about Server Manager usage, see *Siebel System Administration Guide* .
2. In the Siebel client, navigate to the Administration - Server Management screen, and then the Components view.
3. Query for Workflow Recovery Manager.
4. If Workflow Recovery Manager is not Online, then click Start Up.

   This step starts the Workflow Recovery Manager. It is not necessary to set other parameters.

## Preparing to Test the Workflow Recovery Manager

This task is a step in *Recovering a Workflow Process*. The following procedure describes how to prepare to test the Workflow Recovery Manager.

### To prepare to test the workflow recovery manager

1. Connect to the Server Manager from the Siebel Server.

   You must use the `srvrmgr>` prompt. For more information about Server Manager usage, see *Siebel System Administration Guide* .
2. To determine if the Workflow component group is active, enter the following command:

   ```
   srvrmgr>list compgrp
   ```

3. If the Workflow component group is not active, then do the following:
   a. Enter the following command:

```
srvrmgr> enable compgrp workflow
```

b. Stop, and then restart the Siebel Server.

4. In the Siebel client, navigate to the Administration - Workflow Process screen, then the Workflow Monitoring Configuration view.
5. Make sure a Workflow Process is running, and then fails.

   In order for the Workflow Recovery Manager to attempt recovery, a Workflow Process instance must run, and then fail. To make sure such a workflow exists, you can examine the Active Workflow Processes list.

# Recovering Workflow Process Instances

If the Workflow Process Manager server component fails, then the Siebel Server resumes the interrupted Workflow Process instances when it restarts. The Workflow Recovery Manager recovers these instances according to the state information that the Workflow Engine saves for each instance.

## Automatically Recovering a Workflow Process Instance

If the Workflow Process Manager server component fails due to an event that occurs outside of the Workflow Process Manager server component, such as a server failure, then Siebel Workflow resumes the interrupted Workflow Process instances when the Siebel Server restarts.

You can manually recover instances that Siebel Workflow cannot automatically recover. For example, if the Siebel Server fails while a Siebel operation updates a record, then the Workflow Process cannot determine if the Siebel operation finished. It might be necessary for you to manually make sure the Workflow Process finished this update before resuming the Workflow Process.

Siebel CRM can automatically recover a Workflow Process that runs in the server component. It cannot automatically recover a Workflow Process that runs on a local database.

## Manually Recovering a Workflow Process Instance

You can correct and resume a Workflow Process instance that encounters errors. For example, if the Communications Server is not available, and if a Workflow Process uses email to send a notice, then Siebel CRM sets the status for this Workflow Process to In Error. You can activate the Communications Server, and then resume the Workflow Process.

### To manually recover a Workflow Process instance

1. In the Siebel client, navigate to the Administration - Business Process screen, then the Workflow Instance Admin view.

   You can recover instances that Siebel CRM marks for manual recovery from the Workflow Instance Admin view. For more information, see *Administering Workflow Process Instances*.
2. In the Related Instances list, click the Related Instances Menu (the cogwheel icon) and select one of the options defined in the following table.

| Menu Item | Description |
|---|---|
| Resume Instance - Next Step | To determine the next step to run, the recovery skips the current Workflow Process step, and then evaluates the decision conditions that emanate from the current step. |

| Menu Item | Description |
|---|---|
| Resume Instance - Current Step | The recovery retries the current Workflow Process step. If the current step is a sub process step, then it starts a new instance of the sub process. |

If the Resume Instance options are not available in the Related Instances Menu, then see *How Call Depth Affects Resuming a Workflow Process*.

### How Call Depth Affects Resuming a Workflow Process

Siebel CRM can resume a Workflow Process instance only if the Call Depth setting of this instance is the highest among the related instances. If the instance is part of a set of related instances, and if one or more of these instances includes a higher Call Depth level, then Siebel CRM disables the Resume Instance - Next Step and Resume Instance - Current Step options in the Related Instances Menu. For example, assume multiple related instances exist, and that these instances include Call Depth settings of 0,1,2,3, and 4. Assume you choose the record with level 3. In this situation, Siebel CRM disables these Resume controls because level 3 is not the highest Call Depth level that it set among the related instances.

## Guidelines for Recovering a Workflow Process

It is recommended that you adhere to the following guidelines to recover a Workflow Process:

- To reduce the number of checkpoints and minimize runtime overhead, you should modify the Allow Retry Flag property on Siebel operation steps and business service steps. If the Workflow Recovery Manager cannot determine from which step it must recover a Workflow Process instance, then it marks this instance for manual recovery. To recover the instance, the Workflow Recovery Manager examines the state information for the instance that the Workflow Engine saves at recovery checkpoints. To optimize performance, the Workflow Engine determines the recovery checkpoints according to the step type and the Allow Retry Flag property on the step.
- Separate a complex business service into a number of simpler business services. It is easier to recover from a smaller business service.
- To avoid transient or permanent system problems, use multiple Workflow Recovery Managers. For example, a situation can exist in an environment where the subnet in which the Workflow Recovery Manager resides is frequently down. It is typical for only one Workflow Recovery Manager to exist as long as the Siebel Server can restart it if this server fails.

# Upgrading a Workflow Process

For important upgrade information about Workflow Processes, such as Workflow Premerge, Repository Merge, Workflow Postmerge, and Logging, see *Siebel Database Upgrade Guide* .

## Upgrading the Runtime and Repository Schema

If Oracle changes any Siebel Workflow tables in the repository or the runtime schema or if Oracle adds new columns to the repository or runtime schema, then these new columns will be added to the Siebel database as part of the upgrade.

Oracle does not change Workflow Processes that exist in the runtime deployment table during an upgrade. To reactivate Workflow Processes after a merge finishes, you must create a Workspace, set `Inactive='N'` for the all the workflows that need to be deployed and then deliver the Workspace. For more information, see *Siebel Database Upgrade Guide* .

# 11 Configuring a Workflow Process

## Configuring a Workflow Process

This chapter describes options for configuring a Workflow Process. It includes the following topics:

- *Setting the Workflow Mode Property*
- *Starting a Workflow Process*
- *Handling Errors*
- *Configuring Events*
- *Configuring Batch Processing*
- *Configuring a Workflow Process for a Multilingual Environment*

## Setting the Workflow Mode Property

This topic includes the following subtopics:

- *Types of Workflow Process Modes*
- *Options for the Workflow Process Mode Property*
- *Options for an Interactive Workflow Process*
- *Options for a Long-Running Workflow Process*
- *Enabling Workflow Process Persistence*

### Types of Workflow Process Modes

To determine the runtime behavior of a Workflow Process, you set the Mode property of a Workflow Process to use one of the following values:

- **Service Flow.** A Workflow Process that runs a set of operations.

  For more information, see *About the Service Workflow Process*.
- **Interactive Flow.** A Workflow Process that helps the user to navigate Siebel views.

  For more information, see *About the Interactive Workflow Process*.
- **Long-Running Flow.** A Workflow Process that can last for hours, days, or months.

  For more information, see *About the Long-Running Workflow Process*.
- **7.0 Flow.** A Workflow Process that is compatible with Siebel CRM version 7.7 or earlier.

  For more information, see *About the 7.0 Workflow Process*.

## About the Service Workflow Process

A *service Workflow Process* is a type of Workflow Process that runs a set of operations to complete a work unit from beginning to end. A service Workflow Process includes no special behavior. Siebel CRM can call it as a sub process from another service Workflow Process, an interactive Workflow Process, or a long-running Workflow Process, but not from a 7.0 Workflow Process. A service Workflow Process includes the following limitations:

- It cannot wait for a runtime event or pause for time.
- It cannot include a user interact step or a wait step. The Process Designer does not allow you to add a user interact step or a wait step to a service Workflow Process.

For an example service Workflow Process, see *Defining a Workflow Process That Creates an Activity for a Sales Representative*.

## About the Interactive Workflow Process

An *interactive Workflow Process* is a type of Workflow Process that assists and controls navigation for a user across Siebel views and screens. It includes one or more user interact steps and it typically includes a runtime event.

For an example interactive Workflow Process, see *Options for an Interactive Workflow Process*.

### Interactive Workflow Process in a UI Context

An interactive Workflow Process can run only in a user session context. It cannot run in the Workflow Process Manager server component. It cannot run in a server component that does not include a user session context, such as the Enterprise Application Integration (EAI) Object Manager server component. To run in a user session context means the Workflow Process must run in an Application Object Manager that contains a user session context, such as one of the following Siebel CRM applications:

- Siebel Call Center
- Siebel Service
- Siebel eSales
- Siebel Service Portal

These applications support view navigation, view display, and so on, which is necessary when using an interactive Workflow Process because they display views through the user interact step and they require interaction with a user who clicks buttons and links.

## About the Long-Running Workflow Process

A *long-running Workflow Process* is a type of Workflow Process that is persistent. It can last for hours, days, or months. The Send Order to External Workflow Process is an example of a long-running Workflow Process. It sends an order to a system that is external to Siebel CRM, and then waits for a reply. You can use a long running Workflow Process to define a single Workflow Process that handles an entire business process transaction that coordinates between multiple sub processes. The Quote to Cash business process is an example where a long-running Workflow Process is useful.

To create a collaborative long-running Workflow Process, you can assign a sub process to a user. The Workflow User Event Service business service creates a user event that can span from one user or session to another user or session. For more information, see *Workflow User Event Service Business Service*.

You cannot use a user interact step in a long-running Workflow Process, but you can add an interactive Workflow Process as a sub process step in a long-running Workflow Process.

If you create a long-running Workflow Process, then you must use a user event and not a runtime event to start or resume the Workflow Process.

For more information, see *Options for a Long-Running Workflow Process* and *Defining a Sub Process that Assigns a Long-Running Workflow Process*.

## About the 7.0 Workflow Process

A *7.0 Workflow Process* is a type of Workflow Process that provides backward compatibility for an existing Workflow Process that was defined in a Siebel release prior to Siebel CRM version 7.7. If a Workflow Process was defined earlier than version 7.7, and if you upgrade to version 7.7, then the existing Workflow Process becomes a 7.0 Flow Workflow Process by default.

If you must upgrade a Workflow Process that was defined prior to Siebel CRM version 7.7 to version 8.0, then you must first define the Workflow Process as a 7.0 Workflow Process. You must upgrade a pre 7.7 Workflow Process to version 7.7 before you can upgrade it to version 8.0.

### 7.0 Flow Usage

If you create a new Workflow Process, then it is strongly recommended that you do not use the 7.0 Flow workflow mode. If you do not define a workflow mode for a Workflow Process in a release that occurs prior to Siebel CRM version 8.1, then Siebel CRM assumes the mode is 7.0 Flow. If you create a new Workflow Process, then make sure you do not set the Workflow Mode property to 7.0 Flow so that Siebel CRM does not assume the mode is 7.0 Flow. Starting with Siebel version 8.1, if you create a new Workflow Process, then the Workflow Mode property is set to Service Flow by default.

# Options for the Workflow Process Mode Property

This topic describes how the Workflow Process mode property determines the types of steps a Workflow Process can contain. For more information, see the following:

- *Workflow Process Mode Property Usage with Wait Step and User Interact Step*
- *Workflow Process Mode Property Usage with Sub process Step*

## Workflow Process Mode Property Usage with Wait Step and User Interact Step

Typically, a Workflow Process that includes a wait step is a long-running Workflow Process only if the wait step waits for an amount of time. The Workflow Process Manager will resume the Workflow Process after the wait finishes.

If a wait step waits for a runtime event in a user session, however, then the Workflow Process typically includes a user interact step - making the Workflow Process an interactive Workflow Process. If a Workflow Process includes a user interact step, then you must set the Mode property of the Workflow Process to Interactive Flow.

## Workflow Process Mode Property Usage with Sub Process Step

A Workflow Process can call multiple sub processes, but the Mode property of the calling Workflow Process must be consistent with the Mode property for the sub process.

| Mode Property of the Calling Workflow Process | Mode Property of the Sub Process |
|---|---|
| Service Flow | Service Flow |
| Interactive Flow | You can use any of the following modes: |

| Mode Property of the Calling Workflow Process | Mode Property of the Sub Process |
|---|---|
| | • Service Flow<br>• Interactive Flow |
| Long Running Flow | You can use any of the following modes:<br>• Service Flow<br>• Interactive Flow<br>• Long Running Flow |
| 7.0 Flow | 7.0 Flow |

# Options for an Interactive Workflow Process

This topic includes the following subtopics:



For an example of an interactive Workflow Processes, see *Defining a Workflow Process That Attaches an Activity Plan to an Opportunity*.

## About the Synthetic Event

A *synthetic event* is a runtime event that controls navigation for a Workflow Process. Examples of synthetic events include Suspend, Resume, Next, and Back. To control navigation, the Workflow Engine interprets these synthetic events to navigate the user backward or forward through a Workflow Process, or to pause or resume a Workflow Process. For example, assume the Siebel client includes an Account Note view that includes an Account Entry Applet, and that this applet includes Back, Next, and Save synthetic events. These buttons allow the user to move forward or backward from the Account Note view, or to pause an interactive Workflow Process, and then return to it later to resume the Workflow Process.

You can define buttons, and then associate methods in the interactive Workflow Process to these buttons. For example, to configure the Next synthetic event, you associate a method with an outgoing connector on a user interact step. You set the method name of the synthetic event in the MethodInvoked property of the button control.

## Synthetic Event Methods

The following table describes the different synthetic event methods.

| Synthetic Event Method | Description |
|---|---|
| FrameEventMethodWFNext | Uses an applet to move the user forward in an interactive Workflow Process. To use a synthetic event that resumes a Workflow Process from a step that is different from the current step on which the Workflow Process is waiting, use a prefix of BF in the method name. For example, FrameEventMethodWFBF*event_name*. Use this optional BF prefix to define backward and forward behavior for the synthetic event. |
| EventMethodWFNext | Operates the same as FrameEventMethodWFNext except when navigation occurs through the business component. |
| FrameEventMethodWFBack<br><br>EventMethodWFBack | Moves the user backwards in the interactive Workflow Processs. |
| SaveWorkflow | Pauses and saves the interactive Workflow Process and makes it appear in the user Inbox. |
| ResumeLastIntFlow | Resumes the interactive Workflow Process that Siebel CRM most recently ran.<br><br>ResumeLastIntFlow is not specific to a particular Workflow Process. Siebel CRM can call it from elsewhere in a Siebel CRM application. The button that corresponds to this event can reside in an applet. It is recommended that you place this button in the task bar where Siebel CRM locates the Site Map icon. |

## Comparison of a Synthetic Event to a User Event

A synthetic event differs from a user event as follows:

- A user event is internal to Siebel. Siebel CRM uses it only to resume a Workflow Process from the Workflow Process Manager server component. It uses a user event only in a long running Workflow Process.
- A synthetic event is a runtime event that Siebel CRM uses to resume a Workflow Process from the Application Object Manager where it creates the synthetic events. Siebel CRM can use a synthetic event only with an interactive workflow Workflow Process or with a 7.0 Workflow Process.

## Forward and Backward Navigation Between Views

To define navigation in an interactive Workflow Process, you can use a synthetic event. If the user clicks Next or Back, then Siebel CRM navigates the user to the next or to the previous view without losing the context of the Workflow Process instance.

Siebel CRM uses a synthetic event to allow the user to navigate backward through views. It uses a runtime event to allow forward navigation but not backward navigation.

To navigate back and forth, the following requirements must be met:

- The resumed Workflow Process must be an interactive Workflow Process or a 7.0 Workflow Process.
- The calling event must be a navigation event of a Workflow Process. For example, the calling event must include a name such as InvokeMethod and a sub event with a name such as FrameEventMethodWFBF *event_name* or EventMethodWFBF *event_name*, where *event_name* is the name of the event, such as Next.

The following table describes backward and forward navigation that is available for Workflow Process modes

| Workflow Process Mode Supported | Workflow Process Mode Not Supported |
|---|---|
| Siebel CRM supports the following modes with backward and forward navigation:<br>• Interactive Flow<br>• 7.0 Flow | Siebel CRM does not support the following modes with backward and forward navigation:<br>• Service Flow<br>• Long Running Flow |

## Guidelines for Configuring Backward Navigation

If you configure Siebel CRM to use backward navigation, then it is recommended that you adhere the following guidelines:

- Make sure that any Workflow Process steps that Siebel CRM repeats as a result of using backward navigation do not result in Siebel CRM performing duplicative actions on data.
- Note that backward navigation does not undo changes that the Workflow Process makes. It only modifies the current step counter to reference a previous step.

## Configuring a Synthetic Event Button for Next and Back Events

The following procedure describes how to configure a synthetic event button for next and back events.

### To configure a synthetic event button for next and back events

1. Define a Next button or a Back button on an applet where the event is called.

   Siebel CRM must display the button in an applet that references the primary business component. For example, if the Workflow Process references the Service Request business object and if the view displays activities that are associated with a service request, then Siebel CRM must display the button in the applet that references the Service Request business component. In this example, if Siebel CRM displays the button in an applet that references the Activities business component, then the synthetic event button does not work. For more information, see *Defining the Primary Business Component* and *Using Siebel Tools* .
2. Define the MethodInvoked property of the button control as the name of the associated event.

   For example, FrameEventMethodWFBack for backward navigation.
3. In the Process Designer, associate the applet type runtime event.

   For example, assign FrameEventMethodWFBack to the outgoing connectors of the user interact step in the Workflow Process that receives the event. Assign the event using values from the following table.

| Property | Value |
|---|---|
| Event Type | Applet |
| Event Obj | AppletName |
| Event | InvokeMethod |
| Sub Event | [Method Name]. For example, FrameEventMethodWFBack. |

| Property | Value |
|---|---|
| | |

## Configuring a Synthetic Event Button for the Save Workflow Process Event

The following procedure describes how to configure a synthetic event button for the save Workflow Process event.

### To configure a synthetic event button for the save Workflow Process event

1. Define a Save button on an applet where Siebel CRM calls the event.
   For more information, see *Using Siebel Tools* .
2. Set the MethodInvoked property of the button control to reference the Save Workflow Process event.
3. Use the following script to call the event handler that the Workflow Process uses to handle the button click event. This script also sends the contextual information for the event to the Workflow Process event handler, which is the name of the view where the event occurs. It is not necessary for you to define the event in the Workflow Process:

```
function WebApplet_PreCanInvokeMethod(MethodName, (&CanInvoke)
{
var returnValue = ContinueOperation;
// Recognize SaveWorkflow event, which is used to save Interactive flow
if (MethodName == "SaveWorkflow")
{
CanInvoke = "TRUE";
returnValue = CancelOperation
}
return (returnValue);
}

function WebApplet_PreInvokeMethod (MethodName)
{
var returnValue = ContinueOperation;
try
{
switch (MethodName)
{
// Handle SaveWorkflow event.
// Call Workflow Process Manager to save the interactive
// flow(s) that is waiting in the current view.
case "SaveWorkflow":

var Inputs = TheApplication().NewPropertySet();
var Outputs = TheApplication().NewPropertySet();
// Event name ("SaveWorkflow"), view name, and the rowId
// of the active row of the underlying buscomp are
// three required parameters for handling the event
Inputs.SetProperty("Event Name", MethodName);
var viewName = TheApplication().ActiveViewName();
Inputs.SetProperty("Sub Event", viewName);
var bc = this.BusComp ();
var bcId = bc.GetFieldValue ("Id");
Inputs.SetProperty("RowId", bcId);
var workflowSvc = TheApplication().GetService("Workflow Process Manager");
workflowSvc.InvokeMethod("_HandleSpecialEvent", Inputs, Outputs);
returnValue = CancelOperation
break;

// Add other cases for different custom methods here.
}
```

```
} // end try
finally
{
 bc = null;
 workflowSvc = null;
 Inputs = null;
 Outputs = null;
} // end finally
return(returnValue);
} // end function
```

## Configuring a Synthetic Event Button for the Resume Last Flow Event

The following procedure describes how to configure a synthetic event button for the resume last flow event.

### To configure a synthetic event button for the resume last flow event

1. Define a Resume button on the applet where Siebel CRM must call the event.

   For more information, see *Using Siebel Tools* .
2. Set the MethodInvoked property of the button control to reference the ResumeLastIntFlow event.
3. Use the following script to call the event that the Workflow Process uses to handle the button click. This script passes the contextual information for the event to the Workflow Process event handler. This handler is the name of the view where the event occurs. It is not necessary for you to define this event in the Workflow Process:

```
function WebApplet_PreCanInvokeMethod (MethodName, &CanInvoke)
{
 if (MethodName == "ResumeLastIntFlow")
 {
 CanInvoke = "TRUE";
 return (CancelOperation);
 }
 return (ContinueOperation);
}

function WebApplet_PreInvokeMethod (MethodName)
{
 // Call Workflow Process Manager to resume the last-run interactive flow
 if (MethodName == "ResumeLastIntFlow")
 {
 var Inputs = TheApplication().NewPropertySet();
 var Outputs = TheApplication().NewPropertySet();
 var workflowSvc = TheApplication().GetService("Workflow Process Manager");
 workflowSvc.InvokeMethod("_ResumeLastInteractFlow", Inputs, Outputs);

 return (CancelOperation);

 }

 return (ContinueOperation);
}
```

## Pausing and Resuming an Interactive Workflow Process

Siebel CRM can resume an interactive Workflow Process that the user pauses. The user can navigate out of this Workflow Process, and then navigate back to it and continue where the user paused it.

For example, assume the user is an insurance agent and this user pauses an interactive Workflow Process because the user cannot finish a transaction. Some required information is missing, such as the social security number for a spouse.

When the user eventually gets the social security number, the user resumes the Workflow Process from the Inbox and enters the social security number. The Workflow Engine removes the Workflow Process from the Inbox after it finishes.

If the user pauses an interactive Workflow Process, then Siebel CRM places it in the user Inbox. It does this to allow the Workflow Processs owner to track and resume the Workflow Process.

The following table describes what Siebel CRM does if the user pauses a Workflow Process. If an interactive Workflow Process that is paused finishes, and then terminates, then Siebel CRM removes it from the user Inbox. To realize this behavior, the Auto Persist property for the paused Workflow Process must contain a check mark.

| Pause | Description | Result |
|---|---|---|
| Explicit pause. | The user clicks Suspend. | Siebel CRM saves the Workflow Process to the Siebel database and places it in the user Inbox. |
| Implicit pause | The user leaves a view that is part of the Workflow Process. | If the cache is full or if the user logs out of the Siebel CRM application, then Siebel CRM must remove the Workflow Process from the cache.<br><br>Siebel CRM saves the Workflow Process to the Siebel database and places it in the user Inbox. |

## Persisting a Workflow Process

If Siebel CRM pauses a Workflow Process and the Auto Persist property for the Workflow Process process is set to TRUE, then the Workflow Process can be persistent. If Siebel CRM does an implicit pause and if the user leaves a view that the Workflow Process displays, then Siebel CRM saves the Workflow Process instance to the cache but it does not create an Inbox item. If the user logs out and if the Auto Persist property is TRUE, then Siebel CRM creates an Inbox item in the cache. Siebel CRM displays the Inbox item only if the user logs out, and then logs back in to the Siebel application.

## How Siebel CRM Uses the Cache of an Interactive Workflow Process That Is Paused

If a user navigates out of an interactive Workflow Process, then it remains in memory so that Siebel CRM can resume it later in the same user session. The cache can contain no more than eight instances of an interactive Workflow Process that is paused. This limit applies to an individual user session. The limit is eight instances for interactive Workflow Process that Siebel CRM starts in the user session, not eight instances for each interactive Workflow Process. You cannot modify this eight instance limit.

The user session is one connection on the Application Object Manager server component, regardless if Siebel CRM logs this thread as a single user. If Siebel CRM reaches the eight instance limit, then it does not prevent a new interactive Workflow Process from running. If a user starts an Workflow Process and if Siebel CRM has already cached eight instances in memory, and if the AutoPersist property for the Workflow Process is set to TRUE, then Siebel CRM replaces the oldest instance with this ninth instance. Siebel CRM replaces the oldest instance only if the cache reaches the cache limit.

## Event Handling with an Interactive Workflow Process That Is Paused

The Workflow Process handles events in the following sequence:

1. To examine the cache, it uses the matching criteria that the event defines. It does this to determine if Workflow Process instances exist in the cache that can receive an event.
2. Examines the Siebel database to determine if persistent instances of the Workflow Process exist that can receive an event.
3. Resumes instances it finds in steps 1 and 2.

### How Siebel CRM Handles an Interactive Workflow Process That it Cached

If the user logs out, then Siebel CRM examines the interactive Workflow Processes that it paused in the cache. It creates an Inbox item for each Workflow Process that contains a check mark in the Auto Persist property.

# Options for a Long-Running Workflow Process

This topic includes the following subtopics:

- *Defining a Sub Process that Assigns a Long-Running Workflow Process*
- *Defining a Long-Running Workflow Process That Assigns a Task to a User*

## Defining a Sub Process that Assigns a Long-Running Workflow Process

You can define a Workflow Process that assigns an interactive sub process to the user of a *collaborative Workflow Process*. An example of a collaborative Workflow Process is one that includes a requirement for approvals. It includes a logical path that runs across multiple users who work in collaboration to finish the work that the Workflow Process requires.

To define a collaborative Workflow Process, you use the Recipients properties on a sub process step. The login name determines assignment. The Position does not determine assignment. This login name can be one of the following items:

- A literal value
- Reside in a process property
- Reside in a business component field
- The result of an expression

The Process Designer does not validate this login name.

### To define a sub process that assigns a long-running Workflow Process

1. Define a sub process step.

   For more information, see *Adding a Sub Process Step*.
2. In the Recipients tab of the Multi Value Properties pane, set the Recipient Name field to the login name of the user who Siebel CRM assigned to the sub process.

   For more information, see *Using Process Properties*, and *Recipient Argument Fields*.

## Defining a Long-Running Workflow Process That Assigns a Task to a User

You can use use a long-running Workflow Process to assign a task to a user and then create an item for the task in the user Inbox. To create a new instance of this task and to run the task, the user can clicks the inbox item. For example, the following occurs in the Expense Report Approval long-running Workflow Process:

- An employee sends an expense report.
- Siebel CRM creates a new task named Review Expense Report.
- Siebel CRM assigns the Review Expense Report task to the employee's manager.

This situation requires a one-to-one assignment. To make this assignment, Siebel CRM can look up the Id for the manager from the business component that stores data about the user. When Siebel CRM gets the manager Id, it then creates a new item in the manager Inbox. This item references the Review Expense Report task.

### To define a long-running Workflow Process that assigns a task to a user

1. Use the Process Designer to open the long-running Workflow Process.
2. At the point where Siebel CRM must call the task, determine the Id of the user to whom it must assign the new task instance.

   Siebel CRM can use a business service that accesses Siebel Assignment Manager to apply an assignment rule. The input to this business service varies depending on the context that the assignment requires. The output from this business service is the Id of the user to whom Siebel CRM assigns the task.

   This logic depends on your business requirements. If the Id:

   - Already exists in a business component, then you must use a Siebel operation.
   - Does not exist, then you must use a business service to get it.
3. In the Owner Id input argument, identify the task step that creates a new item in the user Inbox.
4. Add a task step to the Workflow Process.

   If you add a task step to a Workflow Process, then it creates a new instance of the task and assigns it to the user. For more information, see *Adding a Task Step*.

# Enabling Workflow Process Persistence

*Workflow Process persistence* is a feature that allows Siebel CRM to store the state and values that reside in the process properties of a Workflow Process instance. It supports transactions that are long lived in a single Workflow Process. Workflow Process persistence allows you to create an end-to-end Workflow Process that can include a wait step, a sub process step, or other interruptions. It can maintain the active state of the Workflow Process over a long time period while work occurs in different parts of your enterprise. It can save or restore data when the Workflow Process resumes. If persistence is enabled on a Workflow Process and a user pauses it, then Siebel CRM adds an item for it in the user Inbox. The user can take a work break and resume the Workflow Process process at a later time.

Workflow Process persistence can resume a Workflow Process if any of the following events occur:

- Pause
- Server failure
- Session time out
- The user logs out of a Web session

Siebel CRM saves the Workflow Process state and process properties in the S_WFA_INSTANCE table and the S_WFA_INST_PROP table. Siebel CRM removes a persistent Workflow Process from memory after it finishes running.

## How Persistence Works with Different Workflow Process Modes

Siebel CRM uses Workflow Process persistence with the following types of Workflow Processes:

- **Interactive Workflow Process or 7.0 flow Workflow Process.** You set the Auto Persist property of the Workflow Process in the Workflow Processes list.

- **Long-running Workflow Process.** The Siebel Server sets persistence for a long-running Workflow Process at runtime.
- **Service Workflow Process.** Siebel CRM cannot use Workflow Process persistence with a service Workflow Process.

### Process Monitoring with Workflow Process Persistence

Some releases that occur earlier than Siebel CRM version 8.0 use Workflow Process persistence to monitor a Workflow Process. To control Workflow Process persistence, Siebel CRM adjusted the frequency and level settings on individual Workflow Process steps. Starting with Siebel CRM version 8.0, Workflow Process monitoring is separate from Workflow Process persistence and it is not necessary to set persistence for a long-running Workflow Process. If Workflow Process persistence is enabled for a 7.0 flow Workflow Process, then Siebel CRM enables Workflow Process persistence for it during an upgrade or import.

### Enabling Persistence for a Workflow Process

You can enable Workflow Process persistence for an interactive Workflow Process or for a 7.0 flow Workflow Process. The Siebel Server sets persistence for a long-running Workflow Process at runtime.

#### To define persistence for a Workflow Process

1. Locate the Workflow Process want to modify.
2. Set the Auto Persist property to YES.

# Starting a Workflow Process

This topic describes different ways to start a Workflow Process. It includes the following topics:

- *Starting a Workflow Process from a Workflow Policy*
- *Starting a Workflow Process from a Runtime Event*
- *Starting a Workflow Process from a Business Service*
- *Starting a Workflow Process from Another Workflow Process*
- *Starting a Workflow Process from the Workflow Process Manager*
- *Starting a Workflow Process from the Application Object Manager*
- *Starting a Workflow Process from a Script*
- *Starting a Workflow Process from a Custom Toolbar*
- *Other Configurations That Start a Workflow Process*
- *Using the Business Integration Manager to Start a Workflow Process*

For more information, see *Determining How to Start a Workflow Process*.

For information about starting a Workflow Process from a server component, see *Overview: Siebel Enterprise Application Integration* and *Siebel Email Administration Guide* .

## Starting a Workflow Process from a Workflow Policy

To start a Workflow Process from a Workflow Policy, you define a Workflow Policy action that uses the Run Workflow Process Workflow Policy program. To define a custom Workflow Policy program, you can also copy Run Workflow

Process, and then add program arguments that correspond to Workflow Process properties. This configuration allows you to use the policy program to pass data to the process properties. For more information, see *Defining Custom Workflow Policies*.

### To start a Workflow Process from a Workflow Policy

1. Navigate to the Administration - Business Process screen, then the Actions view.
2. In the Actions list, click New.
3. In the Program field, choose the Run Workflow Process program.
4. In the Arguments list, click New.
5. In the Argument field, choose ProcessName.
6. In the Value property, enter the name of the Workflow Process you want to start.
7. Navigate to the Administration - Business Process screen, then the Policy Groups view.
8. In the Policy Groups list, click New to create a new group and then name it.
9. Navigate to the Administration - Business Process screen, then the Policies view.
10. In the list for the Policies List, click New.
11. In the Conditions list, click New to define a Workflow Policy condition for the policy that must be met to start the Workflow Process.
12. In the Actions list, click New to create a new action and then enter the name of the action you defined in step 2.
13. Run the Generate Triggers server component.

    For more information, see *Overview of Creating Database Triggers*.
14. If you use the Run Workflow Process program, then make sure the Workflow Process Manager server component is online.
15. Run Workflow Monitor Agent.

    For more information, see *Running a Workflow Policy with Workflow Monitor Agent*.
16. Meet the conditions of the policy.

    Meeting these conditions starts the Workflow Process.

## Starting a Workflow Process from a Runtime Event

This topic describes how to use a runtime event to start a Workflow Process. .

### How a Runtime Event Starts a Workflow Process

This topic describes the process that Siebel CRM uses to start a Workflow process from a runtime event. This example assumes that the PreWriteRecord event starts the Workflow Process:

The typical steps that Siebel CRM uses to start a Workflow Process from a runtime follow. This example assumes that the PreWriteRecord event starts the Workflow Process. For more information, see *Configuring Runtime Events*

1. Components in the Application Object Manager detect the PreWriteRecord runtime event.
2. The PreWriteRecord runtime event is defined on the connector that emanates out of the start step and the Workflow Process starts.
3. The Object Manager calls the Workflow Engine. Siebel CRM passes the business object that caused the runtime event to the Workflow Engine.
4. The Workflow Engine uses the current row on the business object for data operations.

5. When invoked by a runtime event, the Workflow Process runs in the local context of the event. If the event occurs in a User Interface, then the Workflow Process allows User Interact steps.

## Choosing Between a Runtime Event and a Workflow Policy

If Siebel CRM must detect a database event, then you must use a Workflow Policy and not a runtime event. For example:

- If there is a requirement for a database action, such as a change to a record via EIM, then Workflow Policies are required.
- If that requirement does not exist, then a runtime event can be used.
  - If the runtime event can happen without a User Interface (UI), such as a change to a Business Component record via EAI, then the runtime event must be specified on the Business Component.
  - If the runtime event can happen without a User Interface (UI), such as a change to a Business Component record via EAI, then the runtime event must be specified on the Business Component.

For more information, see *Starting a Workflow Process from a Workflow Policy*.

## Comparison of Using a Runtime Event to a Using a Workflow Policy

The following table describes two different requirements for using a runtime event compared to a Workflow Policy.

| Requirement | Recommended Configuration |
|---|---|
| If the user creates a new service request, then set the priority to ASAP and send a notification email. | Use a Workflow Policy. It is the appropriate way to detect a Siebel database event. |
| If the user views an entry in an applet, then create a text file that contains some information. | Use a runtime event. Define a Workflow Process that includes the InvokeMethod runtime event defined on the start step. This event detects if the user views the Entry Applet. Use a business service step to create the text file. |

## Defining a Button That Starts a Workflow Process

You can define a button on an applet that starts a Workflow Process.

### To define a button that starts a Workflow Process

1. Define a Workflow Process.
2. Define a minibutton on an applet as follows:
   a. Set the Method Invoked property to a meaningful value, such as SendEmail.
   b. Define an applet user property that calls the method. Use values from the following table.

| Property | Value |
|---|---|
| Name | Named Method n |
| Value | `'INVOKESVC', 'Workflow Process Manager', 'RunProcess', '"ProcessName"','"eAuction XML Bulk Upload 1"', '"RowId"', '[Id]'` |

For example, the following code calls the eAuction XML Bulk Upload 1 Workflow Process and sends a pointer to the current record:

```
'INVOKESVC', 'Workflow Process Manager', 'RunProcess', '"ProcessName"',
'"eAuction XML Bulk Upload 1"', '"RowId"', '[Id]'
```

For more information, see the topic about the Named Method *n* applet user property in *Siebel Developer's Reference* .

3. (Optional) Use a runtime event to start this Workflow Process:
   a. Define a runtime event on the connector that emanates from the start step in the Workflow Process. Use values from the following table.

| Property | Value |
|---|---|
| Event Object Type | Applet |
| Event Object | Choose the applet where the button is defined. |
| Subevent | Enter the same value that appears in the MethodInvoked property of the minibutton. |
| Event Cancel Flag | TRUE |

   If you use this configuration to call a Workflow Process, then it is not necessary to add the Named Method *n* applet user property.
   b. If you write a script for PreInvokeMethod to run on the applet or the business component, then do the following:
      - Set the event cancel flag to FALSE.
      - Write a script in the WebApplet_PreInvokeMethod event of the applet that returns CancelOperation.
      - Use the BusComp_PreInvokeMethod event of the business component that the applet references to return the CancelOperation.
4. (Optional) To resume this Workflow Process if the user pauses it, you must define the runtime event on the connector that emanates from a user interact step or from a wait step. This event is a synthetic event. For more information, see *About the Synthetic Event*.

   Define the runtime event using values from the following table.

| Property | Value |
|---|---|
| Event | PreInvokeMethod |
| Event Cancel Flag | True. |

| Property | Value |
|---|---|
| | If this flag is not checked, then an error results when the Workflow Process runs. |
| Event Object | Name of the business component that the applet references. This applet contains the button. |
| Event Object Type | BusComp |
| Sub Event | Name of the method you set.<br><br>The Sub Event name must be unique. |
| Type | (Required) Condition |

5. Deliver the Workflow Process.

   For more information, see *Delivering a Workflow Process*.
6. In the Siebel client, navigate to the Administration - Runtime Events screen, then the Events view.

## Identifying a Workflow Process That a Runtime Event Starts

Use the following procedure to identify a Workflow Process that a runtime event starts.

### To identify a Workflow Process that a runtime event starts

1. Navigate to the Administration - Runtime Events screen, then the Events view.
2. In the Events list, query for Action Set Name starting with `Workflow_*` and then drill down on the Action Set Name of one of the records returned.
3. The Action Sets view that opens has the following sections: Actions and More Info.

   The Actions section lists the actions associated with the selected action set – an example action follows.

   The More Info section shows more detailed information about the selected action.

   **Example action:**

   ```
   Workflow_9SIA-D65JX_Go to User Prompt View_Process.DECISION
   ```

   Where:

   - `9SIA-D65JX` is a ROW_ID representing a record in the S_RR_WORKFLOW table.
   - `Go to User Prompt View` is the name of a step in the Workflow Process.
   - `DECISION` indicates that the Workflow Process step type is a *decision step*.

   **Note:** There will be more than one record in the Actions list if you have either multiple versions of a Workflow Process or if there are further runtime events on other steps in the same Workflow Process.

4. Query the S_RR_WORKFLOW table for the record with a ROW_ID of 9SIA-D65JX to determine the name of the Workflow Process.

```
SELECT NAME FROM S_RR_WORKFLOW WHERE ROW_ID = '9SIA-D65JX'
```

# Starting a Workflow Process from a Business Service

The following procedure show how to configure Siebel CRM to start a Workflow Process from a business service.

## To start a Workflow Process from a business service

1. In the Business Services list, add a new business service with the values shown in the following table.

| Property | Value |
|---|---|
| Name | Enter a name that you can reference in scripting. |
| Class | CSSSrmService |
| Display Name | Enter a name that Siebel CRM displays in workflow views. |

2. In the Object Explorer, expand the Business Service object and then click Business Service User Prop.
3. In the Business Service User Props list, add two new objects using values from the following table.

| Property | Value |
|---|---|
| Component | Enter the short name of the server component. For example, WfProcMgr. |
| Mode | Enter the mode of the Siebel Server request. For example, Async. |

4. (Optional) Enter more user properties for the Server Request Broker.

   For more information, see *Server Request Broker Server Component*.
5. In the Object Explorer, click Business Service Method.
6. In the Business Service Methods list, add a new object using values from the following table.

| Property | Value |
|---|---|
| Name | Enter a name that you can reference in scripting. |
| Display Name | Enter a name that Siebel CRM displays see in workflow views. |

7. In the Object Explorer, expand the Business Service Method object and then click Business Service Method Arg.
8. In the Business Service Method Arguments list, add records that are specific to the component that Siebel CRM calls.

   For example, ProcessName for WfProcMgr. This name is the short name of the Siebel Server component parameter.

## Guidelines for Starting a Workflow Process from a Business Service

If you configure Siebel CRM to start a Workflow Process from a business service, then it is recommended that you adhere the following guidelines:

- If Siebel CRM uses a business service to start a Workflow Process, then it is not necessary for you to define parameters for the Server Request Broker (SRBroker). For example, if Siebel CRM uses the Workflow Process Manager (Server Request) business service.
- If Siebel CRM calls the Siebel Server requests directly, then you must define parameters for the Server Request Broker.
- Server Request Broker and Server Request Processor (SRProc) are required to run a business service that calls a server component.
- To use Workflow Process Manager (Server Request), the Server Request Processor and Server Request Broker must be running.

For more information, see *Siebel System Administration Guide* . For more information on calling a business service, see *Siebel Object Interfaces Reference* .

# Starting a Workflow Process from Another Workflow Process

You can define a Workflow Process so that it asynchronously starts another Workflow Process. You configure Siebel CRM to start a Workflow Process that calls a custom business service. This business service uses the Asynchronous Server Requests business service to start another Workflow Process.

**Note:** If you want to run a sub process synchronously, then you must add a Sub process step. This procedure is only required for asynchronous requests.

## To start a Workflow Process from another Workflow Process

1. In the Process Designer, add a business service step to your Workflow Process.
2. Select the new business service and in the Properties pane, set the values as shown in the following table.
3. 

| Property | Value |
|---|---|
| Business Service | Asynchronous Server Requests |
| Method | SubmitRequest |

4. In the Multi Value Properties pane (MVPW) pane, select the Input Arguments tab and then add three new input arguments with the values shown in the following table.

| Input Argument | Type | Value | Property Name |
|---|---|---|---|
| Component | Literal | WfProcMgr | Leave this field empty. |
| WfProcMgr.ProcessName | Literal | AG Simple Test | Leave this field empty. |
| WfProcMgr.RowId | Process Property | (Leave this field empty) | Siebel Operation Object Id |

For more information, see *Arguments You Can Define for a Process Property*.

5. To set server component parameters, such as the Workflow Process Name, define the Input Argument Name using the following format:

```
[ComponentAlias].[argument_name_for_the_business_service_method]
```

# Starting a Workflow Process from the Workflow Process Manager

A Workflow Process can run in the Workflow Process Manager server component. It can start on the Siebel Server in the following ways:

- From a Workflow Policy that runs on the Siebel Server. For more information, see *Starting a Workflow Process from a Workflow Policy*.
- From a script that specifies the Server Request parameter. For more information, see *Starting a Workflow Process from a Script*.
- From a runtime event with the Processing Mode property set to Remote Synchronous or Remote Asynchronous. For more information, see *Starting a Workflow Process from a Runtime Event*.

Siebel CRM runs a Workflow Process that starts from a script in synchronous mode.

Siebel CRM does not support a Workflow Process that runs on the Siebel Server asynchronously and that uses a business service to call a user interface element, including navigation functionality, such as the user interact step.

For more information, see *Workflow Process Manager Server Component*.

## Remote Synchronous Processing

If a user starts a Workflow Process that runs on the Siebel Server in the Workflow Process Manager server component, then this Workflow Process runs only if the user is connected to the Siebel Server. If the user is not connected, then Siebel CRM queues the request and runs the Workflow Process when the user synchronizes or when the Siebel Server is available. For more information, see *Server Requests Business Service*.

# Starting a Workflow Process from the Application Object Manager

Running a Workflow Process in the Application Object Manager can be useful if Siebel CRM must use a business process with a mobile user or if you must define a business process that includes user navigation.

## To start a Workflow Process from the Application Object Manager

- Start the Workflow Process in one of the following ways:
  - From the Process Simulator
  - From a script that you define to run locally in the Application Object Manager
  - From a runtime event with the Processing Mode defined as local synchronous

# Starting a Workflow Process from a Script

You can use Siebel VB or Siebel eScript to start a Workflow Process from a script. A script can start a Workflow Process from most locations in a Siebel CRM application or from a program that is external to Siebel CRM. If a Workflow Process starts from a script, then you can define it to run on the Siebel Server or in the Object Manager. A Workflow Process that starts from a script runs in synchronous mode. Note the following:

- To run a Workflow Process on the Siebel Server, you must configure Siebel CRM to call the Workflow Process Manager (Server Request) service.
- To run a Workflow Process in the Application Object Manager, you must configure Siebel CRM to call the Workflow Process Manager service.

## Starting a Workflow Process from a Script in the Object Manager

The example shown in the following script starts a Workflow Process named My Account Process that starts in the Object Manager.

```
// Example: Starting a Workflow Process through scripting
function Invoke_Process()
{
 try {
 var svc = TheApplication().GetService("Workflow Process Manager");
 var psInputs = TheApplication().NewPropertySet();
 var psOutputs = TheApplication().NewPropertySet();
 psInputs.SetProperty("Process Name", "My Account Process");
 psInputs.SetProperty("Object Id",
 TheApplication().ActiveBusObject().GetBusComp("Account").GetFieldValue("Id"));
 svc.InvokeMethod("RunProcess", psInputs, psOutputs);
} // end try
finally
{
 psOutputs = null;
 psInputs = null;
 svc = null;
}
```

### Starting a Workflow Process from a Script That Sends Field Values to Process Properties

The example in the following script starts a Workflow Process named My Opportunity Process that starts in the Object Manager. Siebel CRM passes field values to the process properties that are defined in the Workflow Process:

```
//Example: Passing Field Values to Process Properties
function Invoke_Process()
{
var svc = TheApplication().GetService("Workflow Process Manager");
var Input = TheApplication().NewPropertySet();
var Output = TheApplication().NewPropertySet();
var bo = TheApplication().ActiveBusObject();
var bc = bo.GetBusComp("Opportunity");
var rowId = bc.GetFieldValue("Id");
var accountId = bc.GetFieldValue("Account Id");
Input.SetProperty("ProcessName", "My Opportunity Process");
Input.SetProperty("Object Id", rowId);
// Pass the value of the Account Id field to the Account Id process property
Input.SetProperty("Account Id", accountId);
svc.InvokeMethod("RunProcess", Input, Output);
```

### Using the Server Requests Business Service to Start a Workflow Process from a Script

You can use the Server Requests business service to asynchronously start a Workflow Process from a script. For more information, see *Using the Server Requests Business Service to Start a Workflow Process from a Script*.

## Starting a Workflow Process from a Custom Toolbar

The following procedure guides you through one example of starting a Workflow Process from a custom toolbar. In this example, a new button is added to the Toolbar and the Workflow Process starts when the button is clicked.

### To start a Workflow Process from a custom toolbar

1. (Optional) Display the Command object type and the Toolbars object type in the Object Explorer.
2. Navigate to the Business Services list and create a new business service with the values shown in the following table.

| Property | Value |
|---|---|
| Name | TestTBItem |
| Server Enabled | TRUE |

3. Navigate to the Commands list and create a new command with the values shown in the following table.

| Property | Value |
|---|---|
| Name | TestTBItem |
| Business Service | TestTBItem |
| Method | TestTBItem |
| Target | Server |

4. In the Object Explorer, click Toolbar, and then query the Name property in the Toolbars list for the following value:

```
HIMain
```

5. In the Object Explorer, expand the Toolbar object and click Toolbar Item.
6. In the Toolbar Items list, create a new record with the values shown in the following table.

| Property | Value |
|---|---|
| Name | TestTBItem |
| Command | TestTBItem |
| DisplayName | TestTB Item |
| Type | Button |
| HTML Type | Button |
| Position | 20 |

7. In the Siebel client, define a Workflow Process and make sure it can run successfully from the Process Simulator.

   For more information, see *Process of Testing a Workflow Process*.
8. Add the following server script to the business service that you defined in step 2 (TestTBItem). You must use the name of the Workflow Process that you specified in step 7:

```
function Service_PreCanInvokeMethod (MethodName, &CanInvoke)
{
 var returnValue = ContinueOperation;
```

```
 if (MethodName == "TestTBItem")
 {
 CanInvoke = "TRUE";
 returnValue = CancelOperation;
 }
 return (returnValue);
}

function Service_PreInvokeMethod (MethodName, Inputs, Outputs)
{
 var returnValue = ContinueOperation;
 if (MethodName == "TestTBIItem")
 {
 try {
 var psInput = TheApplication().NewPropertySet();
 var psOutput = TheApplication().NewPropertySet();
 var svc = TheApplication().GetService("Workflow Process Manager");
 psInput.SetProperty("ProcessName", "<WF Process Name Created Above>");
 svc.InvokeMethod("Run Process", psInput, psOutput);
 returnValue = CancelOperation;
 } // end try
 finally {
 svc = null;
 psOutput = null;
 psInput = null;
 } // end finally
 } // end if
 return(returnValue);
}
```

9. Test your changes using a Siebel Client Application Object Manager.

   Make sure Siebel CRM displays the new button in the toolbar. Click it to make sure the correct Workflow Process starts.

# Other Configurations That Start a Workflow Process

For information about other ways to start a Workflow Process, see the following topics:

- To start from a synthetic event, see *About the Synthetic Event*.
- To start from a user event, see *Configuring a User Event*.
- To start from the Process Simulator, see *About the Testing Tools*.
- To start from a server component, see *Overview: Siebel Enterprise Application Integration* and *Siebel Email Administration Guide* .

Use these configurations to test a Workflow Process in a test environment before deploying it in a production environment. You can also use these configurations to start a Workflow Process in production environments.

## Using the Business Integration Manager to Start a Workflow Process

Do not use Business Integration Manager or Business Integration Batch Manager, since the Business Integration Manager server component is obsolete.

- If your configuration uses the Business Integration Manager, then you must replace it with the Workflow Process Manager.
- If your configuration uses the Business Integration Batch Manager, then you must replace it with the Workflow Process Batch Manager.

# Handling Errors

A Workflow Process error is a deviation from the normal or expected flow of a Workflow Process. You can use error handling to notify the user that an error occurred and to terminate the Workflow Process.

This topic describes how to handle Workflow Process errors and includes the following topics:

- *Using an Error Exception Connector to Handle Errors*
- *Example of Error Exception Handling*
- *Defining an Error Exception Connector to Handle an Update Conflict*
- *Using a Stop Step to Handle Errors*
- *Using an Error Workflow Process to Handle Errors*
- *Recovering a Workflow Process*

For information about using the process property when handling errors, see *Passing Data Through a Process Property to an Error-Workflow Process*.

## Using an Error Exception Connector to Handle Errors

An *error exception connector* is a type of connector that can handle the following types of errors:

- **Siebel CRM error.** A failure that occurs while Siebel CRM sends an email notification.
- **User error.** A user who attempts to submit an order that is not complete.

You can use an error exception connector to change the flow in a Workflow Process. If an error occurs, then Siebel CRM enters the error code and error message in the Error Code and Error Message process properties. An error exception connector allows you to use values in these properties to define a decision condition.

The Process Designer allows you to define error exception connectors between two steps. If you click the error excception connector on the canvas, then the Properties pane displays the (WF Step Branch) properties for the connector.

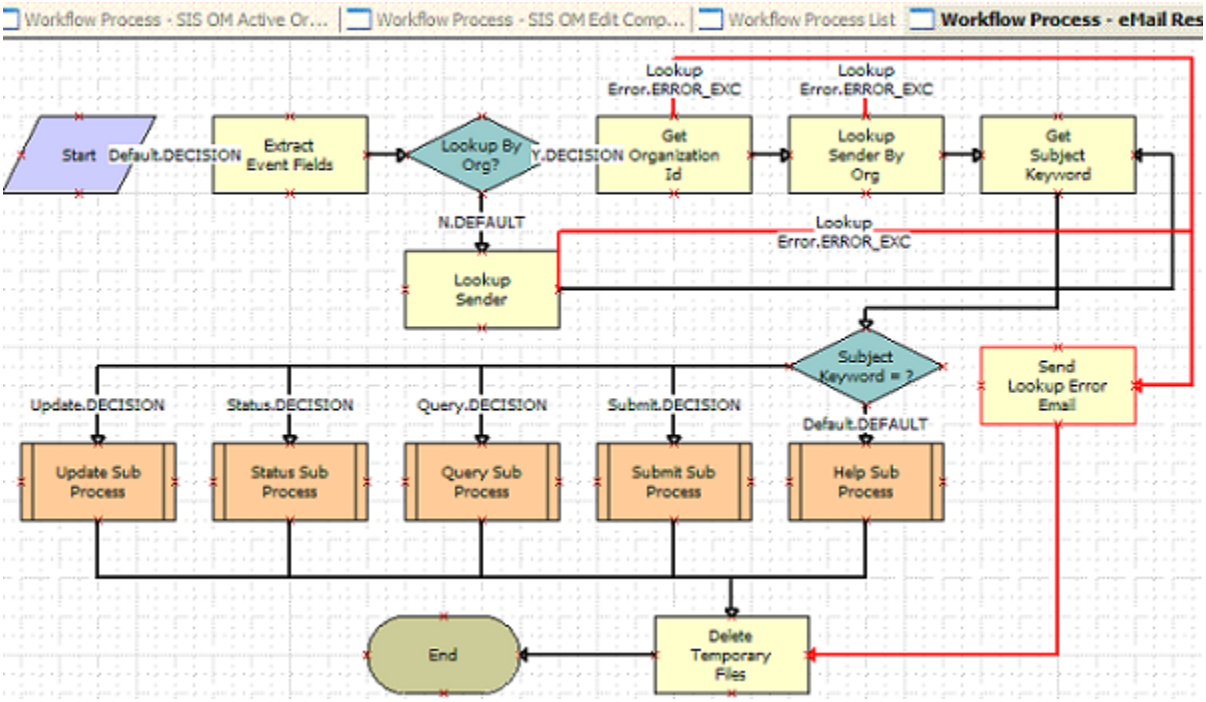
Siebel CRM evaluates exception logic for a step after the step finishes. If you must evaluate an exception before the step runs, then you must attach the error exception connector to an upstream step in the Workflow Process.

# Example of Error Exception Handling

The following image illustrates an example of handling an error exception using an Error Exception connector. In the example shown in this image:

- If the Get Organization ID step cannot get data, then the Workflow Process continues to the Lookup Sender by Org step.
- If this step (Lookup Sender by Org) fails, then the Workflow Process runs the Error Exception connector and uses the Send Lookup Error Email step to send an email.

The following procedure describes the typical steps involved in defining an Error Exception connector in the Process Designer.




## To define an Error Exception connector

1. In the Process Designer, locate and open the Workflow Process you want to modify.
2. (Optional) If an error should cause the Workflow Process to terminate, select the Stop step from the Palettes pane, then drag and drop it on the canvas.
3. Add a connector to the canvas between the step that could cause the error and the target step (either the Stop step created in step 2 or another step created to handle the error). For more information, see *Adding a Branch Connector*.
4. Select the connector you added in step 3 and set the Type to `Error Exception` or `User Defined Exception` in the Properties pane.
5. Double-click the Error Exception connector you defined in step 4 to open the Compose Condition Criteria dialog box.
6. Define the decision conditions that apply for the Error Exception connector.

   For more information, see *Creating a Decision Condition on a Branch Connector*.

# Defining an Error Exception Connector to Handle an Update Conflict

The example in this topic defines an error exception connector to handle an update conflict that occurs if Siebel CRM attempts to make multiple writes to the same record at the same time. If the Workflow Monitor Agent (WMA) starts a Workflow Process, and if this agent updates a record, then the WMA can fail if another Workflow Process attempts to update this record, if another user attempts to update this record, or if another WMA task attempts to update this record since the Workflow Process first gets this record. In this situation, Siebel CRM display an error message that is similar to the following:

```
The selected record has been modified by another user since it was retrieved.
```

To prevent the WMA task from failing, you can define an error exception connector that handles the update conflict that occurs while the Workflow Process runs.

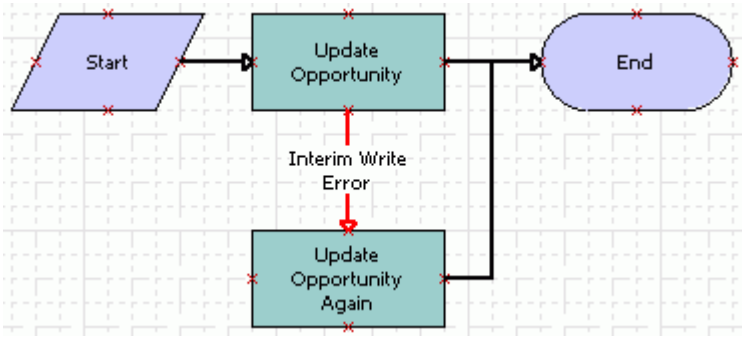
## To define an error exception connector to handle an update conflict

1. Create a new Workflow Process with the values shown in the following table. For an example, see *Creating the Workflow Process*.

| Property | Value |
|---|---|
| Process Name | Error Exception Example |
| Workflow Mode | Service Flow |
| Business Object | Opportunity |

2. Add the following steps and connectors until your Workflow Process resembles the flow shown in the following figure:

    a. A Start step
    b. An Update Opportunity step
    c. An Update Opportunity Again step
    d. An End step
    e. The following Connectors:
        - A connector between step (a) and (b), (b) and (d), and (c) and (d).
        - An Interim Write Error connector between step (b) and (c).




For more information, see *About Step Types* and *Diagramming a Workflow Process*.

3. Click the Update Opportunity business service step, and then use the Properties pane to define the properties shown in the following table.

| Property | Value |
|---|---|
| Business Service Name | ABC Update Opty |
| Business Service Method | Update Opty |

4. With the Update Opportunity business service step selected in the Process Designer, define an input argument in the Multi Value Properties pane (MVPW) pane with the values shown in the following table.

| Field | Value |
|---|---|
| Input Argument | Opportunity Id |
| Type | Process Property |
| Property Name | Object Id |
| Property Data Type | String |

| Field | Value |
|---|---|
| | |

For more information, see *Arguments You Can Define for a Process Property*.

5. Define a decision condition on the Interim Write Error exception connector with the values shown in the following table.

| Property | Value |
|---|---|
| Compare to | Process Property |
| Operation | One Must Match (Ignore Case) |
| Object | Error Code |
| Values | 0x8137 -- 0x6f74 |

Siebel CRM returns the 0x8137 -- 0x6f74 error code with the following error message:

```
The selected record has been modified by another user since it was retrieved.
```

For more information, see *Creating a Decision Condition on a Branch Connector*.

6. Define the Update Opportunity Again business service step, using the same values you used in step 3 and step 4. For the Name property, use Update Opportunity Again.
7. Validate and then simulate the Workflow Process.

For more information, see *Process of Testing a Workflow Process*.

8. Implement this configuration in your production Workflow Process.

The Update Opportunity Again step allows Siebel CRM to write to the opportunity again if the first attempt to update the opportunity fails due to the update conflict error. This example uses a business service step that updates opportunities. You can use the same configuration with other step types that update a record, such as a Siebel operation step or a sub process step, and with other types of records, such as accounts or contacts.

# Using a Stop Step to Handle Errors

A stop step can display an error message while a Workflow Process runs. It can include an exception handler that displays or logs a customizable error message that can include an expression. For example, to handle real-time processing for credit card authorization or for order validation. For more information, see *Adding a Stop Step*.

# Using an Error Workflow Process to Handle Errors

An *error Workflow Process* is a type of Workflow Process that runs if another Workflow Process reaches an error state. If this calling Workflow Process reaches an error state, then Siebel CRM stops the calling Workflow Process, passes system defined process properties to the error Workflow Process, and then runs the error Workflow Process.

A sub process can call an error Workflow Process. An error Workflow Process cannot contain a sub process.

## Using an error Workflow Process to handle errors

1. In the Workflow Processes list, locate the Workflow Process that must call the error Workflow Process.
2. In the Workflow Processes list, in the Error Process Name property, choose a Workflow Process.

   Siebel CRM uses the Workflow Process that you define in the Error Process Name property as the error Workflow Process.
3. If the error Workflow Process must change the state of the calling Workflow Process, then add an error exception connector to the calling Workflow Process.

   An error Workflow Process does not return values back to the calling Workflow Process.

## Benefits of Using an Error Workflow Process

A *universal exception handler* is an error Workflow Process that Siebel CRM can use to handle an error that occurs across multiple steps in a Workflow Process or across multiple Workflow Processes. You can use it to reduce clutter in a Workflow Process diagram. You can reuse a single error Workflow Process that handles the same error for multiple calling Workflow Process.

## How Siebel CRM Handles Errors

This topic describes how Siebel CRM handles errors for a Workflow Process and for a sub process.

### How Siebel CRM Handles Errors for a Workflow Process

If an error occurs in a Workflow Process and if the Error Process Name property for this Workflow Process:

- **Does not contain a value.** Then this Workflow Process remains in an In Error state and Siebel CRM returns an error code to the object that called this Workflow Process.
- **Contains a value.** Then Siebel CRM runs the error Workflow Process. The following table describes the possible outcomes when Siebel runs the error Workflow Process.

| Situation | Error State | Error Code | Result |
|---|---|---|---|
| The error Workflow Process handles the error successfully. | Completed | Siebel CRM does not return an error code to the object that called the error Workflow Process. | If the error Workflow Process encounters an end step, then error handling is successful and the error Workflow Process terminates immediately with a Completed state. |
| The error Workflow Process tries to handle the error but fails with a different error. | Error | Siebel CRM returns an error code to the object that called the error Workflow Process. You define this error code on | If the error Workflow Process encounters a stop step, then the error Workflow Process terminates immediately with In Error state. |

| Situation | Error State | Error Code | Result |
|---|---|---|---|
| | | the stop step of the error Workflow Process. | |
| The error Workflow Process cannot handle the error. | Error | Siebel CRM returns the original error code to the object that called the error Workflow Process. | If no start decision condition is met, then the error Workflow Process terminates immediately with In Error state. |

### How Siebel CRM Handles Errors for a sub process

If a sub process encounters an error, and if an error Workflow Process is defined in the Error Process Name property for this sub process, then the error Workflow Process finishes with one of the outcomes described in the following table.

| Situation | Error State | Error Code | Result |
|---|---|---|---|
| The error Workflow Process handles the error successfully. | Completed | N/A | If the error Workflow Process encounters an end step, then error handling is successful and the sub process terminates with a Completed state. Siebel CRM returns control to the calling Workflow Process. This calling Workflow Process continues to run from the next step. |
| The error Workflow Process attempts to handle the error but fails with a different error. | In Error | Siebel CRM returns an error code to the sub process. You define this error code on the stop step of the error Workflow Process. | If the error Workflow Process encounters a stop step, then error handling failed. Siebel CRM terminates the error Workflow Process and the calling Workflow Process with the In Error state. |

# Configuring Events

This topic describes how to configure runtime events and user events that Siebel CRM uses with a Workflow Process. It includes the following topics:

- *Configuring Runtime Events*
- *Configuring a User Event*

## Configuring Runtime Events

A Workflow Process uses runtime events to do the following:

- Allow real time monitoring of events
- Minimizes scripting and calling for a Workflow Policy

A Workflow Process uses the following types of events:

- Application event
- Business component event
- Applet event

A runtime event allows Siebel CRM to reply in real time to a user interaction. To start or resume a Workflow Process, you define a runtime event on a connector that emanates from a start step, wait step, or user interact step. To define a runtime event, you use the following properties of the WF Step Branch:

- Event Object Type
- Event Object
- Event
- Sub Event
- Event Cancel Flag

For more information, see *Starting a Workflow Process*, and *Siebel Personalization Administration Guide* .

## Using Runtime Events with a Long-Running Workflow Process

It is recommended that you do not configure Siebel CRM to use a runtime event to start a long-running Workflow Process because a runtime event is specifically related to a single user and a single session. It originates from Personalization functionality. Instead, it is recommended that you use an interactive Workflow Process or a service Workflow Process to handle a runtime event. After this Workflow Process finishes, Siebel CRM can create a user event that notifies a long-running Workflow Process.

## Using Runtime Events with a User Interact Step

If the Immediate Post Changes property of a business component field is TRUE, then Siebel CRM supports using the SetFieldValue event with a user interact step. If the field value changes, and if Immediate Post Changes is TRUE, then Siebel CRM does an immediate roundtrip to the Siebel Server, and then immediately recalculates the field or refreshes the view. It bypasses the PreSetFieldValue event in the browser script.

Siebel CRM does not support the following events with a user interact step:

- The DisplayRecord event.
- The DisplayApplet event.
- The Login event. You can use the WebSessionStart event instead.

For more information, see *Configuring Siebel Business Applications* .

## Using Runtime Events in a Business Object Context

Siebel CRM starts a Workflow Process that references a runtime event only if it detects the runtime event in the same business object context that the Workflow Process references. For example, assume the WriteRecord event starts a Workflow Process, and that the Business Object property for this Workflow Process is set to Service Request. To update the record, the user clicks the Service Requests List screen tab, updates the Status field, and then steps off the record. Siebel CRM writes the record in the context of the Service Request business object, and then starts the runtime event that is defined on theWorkflow Process.

If the user updates the Status field in a context other than the service request business object context, then Siebel CRM does not start the runtime event. For example, assume the user drills down on a Contact, clicks the Service Requests view tab, updates the Status field, and then steps off the record. In this situation, Siebel CRM writes the service request record in the context of the Contact business object and it does not start the runtime event.

## Defining a Runtime Event in a One-to-Many Relationship

The example in this topic defines a runtime event in a one-to-many relationship. If you define a runtime event to start a Workflow Process in reply to a change that a user makes to a child record in a one-to-many relationship, then you must configure Siebel CRM to start the Workflow Process according to the child ROW_ID. For example, assume you require

Siebel CRM to start a Workflow Process if the user updates a field in the activity of a service request. A service request can contain one or many activities, so Siebel CRM must start the Workflow Process according to the activity ROW_ID and not according to the service request. If you start the Workflow Process according to the service request ROW_ID, then the following occurs:

- Siebel CRM starts the Workflow Process if the user changes data in the form applet for the service request.
- Siebel CRM does not start the Workflow Process if the user changes data in the activities list applet.

## To define a runtime event in a one-to-many relationship

1. (Optional) For educational purposes, examine a one-to-many relationship:
    a. In the Siebel client, navigate to the Service Request screen, then the Service Request List view.
    b. Create a new service request.
    c. Click the SR# field.
    d. Use the Activities list to create two new activities.
    e. In the Activities list view, the Service Request form shows fields for the parent service request and the Activities list shows multiple activities for the parent.
2. In the Workflow Processes list, create a new Workflow Process with the values shown in the following table.

| Property | Value |
|---|---|
| Process Name | Update Service Request |
| Business Object | Service Request |
| Workflow Mode | Interactive Flow |

This Workflow Process references the Service Request business object. The runtime event that this Workflow Process uses occurs on the start step. It references the Action business component, which is a child of the Service Request business object. It includes a wait step for testing purposes. A wait step requires an Interactive Flow. For more information, see *Adding a Wait Step*. For an example, see *Creating the Workflow Process*.

3. Open the Workflow Process you created in step 2 and then add the following steps and connectors until your Workflow Process resembles the flow shown in the following figure:

    a. A Start step
    b. A Get Activity Id step
    c. An End step
    d. The following Connectors:

        - A Id Triggered connector between step (a) and (b).
        - A connector between step (b) and (c).


For more information, see *Adding Workflow Process Steps* and *Diagramming a Workflow Process*.

4. Click the Id Triggered connector and in the Properties pane, define values shown in the following table.

| Property | Value |
|---|---|
| Event Object Type | BusComp |
| Event | WriteRecord |
| Event Object | Action |

5. Click the canvas, making sure not to select any Workflow Process step or connector.
6. In the Multi Value Property Window (MVPW) pane, add a new process property with the values shown in the following table.

| Field | Value |
|---|---|
| Name | ActionBCRowId% |
| In/Out | In/Out |
| Business Object | Service Request |

For more information, see *Using Process Properties*.

To capture the activity ROW_ID, you define a process property and then use a wait step that reads a field from the child business component into the process property in the output argument for the wait step. It does not modify the underlying data.

7. Click the Get Activity Id step and go to the Output Arguments tab in the MVPW pane.

8. Create a New Record with the values shown in the following table.

| Field | Value |
|---|---|
| Property Name | ActionBCRowId% |
| Type | Expression |
| Value | Id |
| Business Component Name | Action |

   For testing purposes, this step reads the activity ROW_ID, which is the Id field, into the ActionBCRowId% process property. You do not need to define input arguments for the wait step.
9. Validate and then simulate the Workflow Process.

   For more information, see *Process of Testing a Workflow Process*.
10. Implement this configuration in your production Workspace.

    This example demonstrates how you can start a Workflow Process according to changes that Siebel CRM makes to a child in a one-to-many relationship. It includes steps that test the configuration. In a production environment, if it is not necessary to capture the child ROW_ID, then you can define the trigger for the runtime event on the connector that emanates from step 4.

## Using a Runtime Event with the Updated By Field

If a Workflow Process step includes a runtime event that uses a processing mode that runs locally to start or resume a Workflow Process, then the value in the Updated By field identifies the user who is currently logged into Siebel CRM.

If a Workflow Process runs asynchronously, then it will run as the user defined on the Workflow Process Manager component (typically, SADMIN).

## Runtime Events That You Cannot Use to Start a Workflow Process

To start a Workflow Process, Siebel CRM cannot use a runtime event that might not return a result. It can start a Workflow Process only in the record context of a business component. If business component record context does not exist, then Siebel CRM cannot start the Workflow Process and attempting to use the BusComp Query event to start it will fail.

## Using a Runtime Event More Than One Time

You cannot use the same runtime event more than one time in a given Workflow Process.

# Configuring a User Event

A *user event* is a unique event that is internal to Siebel Workflow. It starts or resumes a long-running Workflow Process from the Workflow Process Manager. It belongs to one of the following object types:

- Application
- Applet
- Business Component

Siebel CRM can use a user event to do the following:

- Start or resume a long-running Workflow Process that runs in the Workflow Process Manager (WFProcMgr) server component.
- Start a Workflow Process if the user event is attached to a start step.
- Resume a Workflow Process instance that is paused. You configure the user event on a Workflow Process step that can receive an input argument.
- Be used in a long-running Workflow Process that spans multiple users.

A long-running Workflow Process must use only a user event, and not a runtime event.

## Using the Workflow User Event Service Business Service to Create a User Event

To create a user event, Siebel CRM calls the Workflow User Event Service business service. Siebel CRM can use it only in a long-running Workflow Process.You cannot modify Siebel CRM to cause a long-running workflow to resume on a custom Workflow Process server component.

The Workflow User Event Service business service must communicate with the Workflow Process Manager to use a user event. Siebel CRM can call the Workflow User Event Service business service from a business service step, a script, a COM interface, or a Java interface. For example, assume a 7.0 Flow Workflow Process or a service Workflow Process starts a user event. A business service step calls the Workflow User Event Service business service to communicate with a long-running Workflow Process that runs in the background.

Most types of Workflow Process or business services can create a user event. It is recommended that you define only a long-running Workflow Process to receive a user event.

For more information, see *Workflow User Event Service Business Service*.

## Starting the Workflow User Event Service Business Service

This topic describes how to call the Workflow User Event Service business service to create a user event.

### To start the Workflow User Event Service business service

1. Add a business service step to a Workflow Process.
2. Click the business service step you added in step 1 and use the Properties pane to define values described in the following table.

| Property | Value |
|---|---|
| Business Service Name | Workflow User Event Service |

| Property | Value |
|---|---|
| | |
| Business Service Method | GenerateEvent |

3. In the Multi Value Property Window (MVPW) pane, create a new input argument for the step.

   For more information, see *Arguments You Can Define for a Process Property*.
4. In the Input Argument field, select Payload and define the other fields as appropriate.
5. Repeat step 3 and step 4, selecting Correlator Value and then User Event Name in the Input Argument field respectively.

## Configuring a Long-Running Workflow Process to Wait for a User Event

A long-running Workflow Process is the only type of Workflow Process that can wait for a user event. Other types of Workflow Processes can create a user event but not wait for a user event.

### To configure a long-running Workflow Process to wait for a user event

1. Open the Process Designer for the Workflow Process where you must define a user event.
2. In the Multi Value Property Window (MVPW) pane, set the Correlator Flag field for one of the process properties for the Workflow Process to TRUE.

   For more information, see *Using Process Properties*.
3. On the branch of the step that handles the event, such as a start step or a wait step, use the Properties pane to define values described in the following table.

| Property | Value |
|---|---|
| Type | Condition. If you set Type to Default rather than to Condition, then Siebel CRM does not call the user event. |
| User Event Name | Enter the name of the Workflow Process that you defined for the Value property in *Starting the Workflow User Event Service Business Service*. |
| User Event Timeout | Choose the name of the process property that holds the payload that Siebel CRM passes from the user event. |
| User Event Storage | Enter the timeout period for the user event.<br><br>The User Event Timeout works in a way that is similar to the timeout for a runtime event. If Siebel CRM does not receive a user event during the timeout period, then it resumes the Workflow Process after the timeout period expires. |

Siebel CRM does not queue a user event. If no recipient is waiting to accept the user event with the defined correlator, then Siebel CRM discards the event.

# Configuring Batch Processing

If Siebel CRM runs a Workflow Process for every record that is dated between X and Y of a business component, then you can configure the Workflow Process Batch Manager (WfProcBatchMgr) server component. It can run a Workflow Process one time for each record that the primary business component contains. For more information, see *Defining the Primary Business Component*.

For more information about running a Workflow Process in batch mode, see *Siebel System Administration Guide* .

## To configure batch processing

1. In the Siebel client, navigate to the Administration - Server Configuration screen, Enterprises view, then the Component Definitions view.
2. Click New in the Component Request form.
3. Click the selection button in the Component/Job field and in the dialog box that appears, select Workflow Process Batch Manager.
4. Click New in the Component Request Parameters form.
5. Click the selection button in the Name field and in the dialog box that appears, select the Workflow Process Name.

   The Workflow Process Batch Manager parameters are described in the following table.

| Parameter | Description |
|---|---|
| Workflow Process Name | (Required) Name of the Workflow Process that Siebel CRM runs. |
| Search Specification | Search specification that identifies the work items to process. Siebel CRM uses the Search Specification parameter only with the Workflow Process Batch Manager server component. If you define this parameter for any other server component, such as the Workflow Management server component, then Siebel CRM does not use it. For more information, see *Configuring Batch Processing with a Search Specification*. |

6. In the Value field, type in the name of the Workflow Process to run.
7. Click New to add another parameter.
8. Click the selection button in the Name field and then select Search Specification.
9. In the Value field, provide a search specification.

# Configuring Batch Processing with a Search Specification

To limit the number of records that Siebel CRM evaluates when it runs a Workflow Process in batch mode, you can define a search specification. Siebel CRM uses the Search Specification parameter on the Workflow Process Batch Manager to run the search specification on the primary business component. For each record it finds, the Workflow Process Batch Manager starts the Workflow Process and sets the Object Id process property to the current active row. For more information, see *Defining the Primary Business Component*.

If you do not define a search specification, then the Workflow Process Batch Manager runs the Workflow Process for each record of a particular type. For example, if 100 service requests exist, then it runs the Workflow Process 100 times, one time for each service request.

# Configuring Batch Processing to Run Repeatedly at a Specific Interval

To configure Siebel CRM to run a batch at a specific interval, you can use the Repeating Component Request feature. For example, you can configure a Workflow Process to run at 7 A.M. every Monday. For more information, see *Siebel System Administration Guide* .

# Configuring Batch Processing with Linked Fields

If Siebel CRM runs the Workflow Process Batch Manager, and if the Link Specification property is TRUE on a field of the primary business component, then Siebel CRM might return more records than expected. This situation can affect performance.

If the primary business component contains a link relationship with one or more nonprimary business components, as created through a Link on the current business object, then Siebel CRM passes the value that the field contains to a field in the nonprimary business component through this link. If Siebel CRM runs Workflow Process Batch Manager in batch mode, or if a custom business service queries business component records, then it is important to monitor performance and modify the configuration, if necessary. For more information, see *Defining the Primary Business Component*.

# Configuring Batch Processing with a Custom Business Service

If a custom business service that you define includes a loop that processes every business component record, and if it runs business service code on each of these records, then you must consider how this configuration uses the server component. The following table compares how the two server components work in this situation.

| Server Component | Description |
|---|---|
| Workflow Process Manager | The business service and the loop handle every record at one time. It is recommended that you use this configuration to run this business service and the Workflow Process. |
| Workflow Process Batch Manager | The business service for the Workflow Process already contains a loop. Using Workflow Process Batch Manager causes the business service loop to run for every record in the primary business component. This configuration can result in undesirable duplicate and repeated execution. |

# Configuring a Workflow Process for a Multilingual Environment

This topic describes how to define a Workflow Process to function correctly in a multilingual environment. It includes the following topics:

- *Using Literal Values in a Multilingual Environment*
- *Using Expressions in a Multilingual Environment*
- *Using Wait Steps with Global Time Calculations*
- *Using the Locale Code Parameter*

To configure a Workflow Process to work correctly in a language other than the base language, you must make sure the database that your configuration uses can handle Multilingual Lists Of Values (MLOV) for the type of language that the production environment uses. For example, if you must modify a workflow to use FRA, and if the base language is ENU, then make sure that List of Values entries exist for the FRA language type.

For more information, see *Siebel Global Deployment Guide* .

## Using Literal Values in a Multilingual Environment

A literal value is language dependent. A literal value in a dynamic drop-down list might contain content that a user creates at runtime. A Workflow Process cannot interpret these values in a predictable way. A Workflow Process that references a literal value is language dependent and cannot run in a multilingual environment. For more information, see *Siebel Global Deployment Guide* .

## Using Expressions in a Multilingual Environment

In a multilingual environment, it is important for workflows to be language agnostic. As an example, if we have a workflow decision step checking for the Account Status being **`Active`**, but that workflow is in a French environment, the value will be **`Actif`**, so comparing to the literal value would incorrectly return false.

For this reason, expressions involving comparison to MLOV-enabled values should leverage the Language Independent Code (LIC), which would be the same across all languages. This can be done using the built-in LookupValue expression as shown in the following examples.

### Decision Step Example

Assume a decision step compares Account Status to Active. The Account Status drop-down list bounds the Account Status field. You can set the Compare To property to Expression and set the Expression property to the following code:

```
[Account Status] = LookupValue ("ACCOUNT_STATUS", "Active")
```

### Business Service Step Example

To send an email to an expense approver, assume a business service step calls the Outbound Communications Manager. The Comm Recipient Group drop-down list bounds the Recipient Group argument. You can set the Type property to **`Expression`** and set the Value property to the following code:

```
LookupValue ("COMM_RECIP_SRC", "Comm Employee")
```

For more information about globalization, see *Siebel Global Deployment Guide* . For more information about defining Siebel Workflow to use MLOV capable fields, see *Configuring Siebel Business Applications* .

# Using Wait Steps with Global Time Calculations

Siebel CRM can use the following types of wait steps:

- **Absolute Wait.** A wait period that the duration controls. For example, an absolute wait that you set for 30 minutes waits 30 minutes from the time that the wait step starts the wait.
- **Service Calendar Wait.** Not absolute. For example, assume a service calendar wait begins at 6:00 P.M. but the service hours for the organization are 9:00 A.M. to 6:00 P.M.In this situation, the wait does not start until 9:00 A.M. the next morning. It runs from 9:00 A.M. to 9:30 A.M. instead of from 6:00 P.M.to 6:30 P.M.

## Using a Time Zone Setting with a Wait Step

If a Workflow Process runs as a server task, then you must shut down and then restart the Workflow Process Manager after you make any changes to the Time Zone user preference for the SADMIN user. Siebel CRM uses these changes only after it restarts the Workflow Process Manager, which is important if your deployment uses UTC because it might be necessary for you to set the Time Zone user preference.

Time zone settings do not affect an absolute wait, including server or user time zone preferences. You can use UTC with the server that contains the database that your Siebel CRM deployment uses. For more information, see *Siebel Global Deployment Guide* .

A service calendar wait step requires a time zone for delay computations. In this situation, Siebel CRM uses the time zone for the current user.

# Using the Locale Code Parameter

The Locale Code parameter for the Workflow Process Manager (WfProcMgr) server component includes formats for data, such as dates, times, numbers, and currency. Note the following:

- If a Workflow Process runs in the Workflow Process Manager, then the Workflow Process Manager formats data according to the format that you define in the Locale Code parameter.
- If a Workflow Process communicates with an application that is external to Siebel CRM or that writes data to a file, then it passes date, time, number, and currency data according to the format that you define in the Locale Code parameter.

# 12 Example Workflow Processes

## Example Workflow Processes

This chapter includes some example Workflow Processes. It includes the following topics:

- *Defining a Workflow Process That Creates an Activity for a Sales Representative*
- *Defining a Workflow Process That Attaches an Activity Plan to an Opportunity*
- *Defining a Workflow Process That Manages Research Activities for a Service Request*

## Defining a Workflow Process That Creates an Activity for a Sales Representative

In this topic you create a Workflow Process that creates an activity for a sales representative. The following tasks are involved in developing this example:

1. *Creating the Workflow Process*
2. *Adding Steps and Connectors to the Workflow Process*
3. *Defining Properties and Arguments for Workflow Process Steps*
4. *Defining the Runtime Event That Starts the Workflow Process*
5. *Defining a Decision Condition for a Decision Step*
6. *Preparing This Example for Testing*
7. *Simulating the Workflow Process*
8. *Verifying the Workflow Process*

In this example Workflow Process, if a user creates a new opportunity, then Siebel CRM evaluates the opportunity revenue. If the value of the opportunity is over $10,000, then it creates an activity that directs the sales representative to pursue the deal. The WriteRecordNew runtime event starts this Workflow Process.

### Creating the Workflow Process

This task is a step in *Defining a Workflow Process That Creates an Activity for a Sales Representative*.

You begin by creating the new Workflow Process.

#### To create the new Workflow Process

1. In the Object Explorer, click Workflow Process.
2. In the Workflow Processes list, create a new Workflow Process with the values shown in the following table.

| Property | Value |
|---|---|
| Process Name | Large Opportunity Creates Activity |

| Property | Value |
|---|---|
| Workflow Mode | Service Flow |
| Business Object | Opportunity |

# Adding Steps and Connectors to the Workflow Process

This task is a step in *Defining a Workflow Process That Creates an Activity for a Sales Representative*.

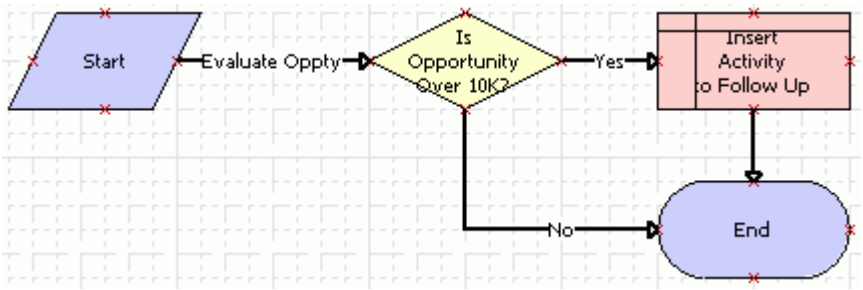
In this topic you add steps and connectors to the Workflow Process.

## To add steps and connectors to the Workflow Process

1. In the Process Designer, open the Workflow Process you defined in *Creating the Workflow Process*.
2. Add the following steps and connectors until your Workflow Process process resembles the flow illustrated in the following figure:

    a. A Start step
    b. An Is Opportunity Over 10K? decision step
    c. An Insert Activity to Follow Up opearation step
    d. An End step
    e. The following Connectors:

        - An Evaluate Oppty connector between step (a) and (b).
        - A Yes connector between between step (b) and (c).
        - A No connector between step (b) and (d)
        - A connector between step (c) and (d).




    For more information, see *Adding Workflow Process Steps* and *Diagramming a Workflow Process*.
3. Click the Is Opportunity Over 10 decision step and in the Properties pane, set the Name property to the value shown in the following table.

| Property | Value |
|---|---|
| Name | Is Opportunity Over 10k? |

| Property | Value |
|---|---|
| | |

The Properties pane is context-sensitive. If you click a step or connector in the Process Designer, then the properties for the step or connector appear in the Properties pane.

4. Click each remaining step in the Workflow Process in succession and use the Properties pane to change the Name property for each step according to the values shown in the following table.

| Step Type | Name Property |
|---|---|
| Siebel Operation | Insert Activity to Follow Up |
| End | End |

5. Click each connector in succession and use the Properties pane to change the Name property for the connector according to the values shown in the following table.

| Connector | Name Property |
|---|---|
| Between the start step and the decision step | Evaluate Oppty |
| Between the decision step and the Siebel operation | Yes |
| Between the decision step and the end step | No |

> **Tip:** The Workflow Process Designer sets the name of the connector that emanates from the start step to `Connector 0` by default.

# Defining Properties and Arguments for Workflow Process Steps

This task is a step in *Defining a Workflow Process That Creates an Activity for a Sales Representative*.

In this topic, you define properties and arguments for Workflow Process steps.

## To define properties and arguments for Workflow Process steps

1. Click the Insert Activity to Follow Up step and in the Properties pane, set the properties as shown in following table.

| Property | Value |
|---|---|
| Business Component | Action |
| Operation | Insert |

2. Select the Insert Activity to Follow Up step in the Process Designer and in the Multi Value Property Window (MVPW) pane, add an input argument with the values shown in the following table.

   You must first define the Field Name first, and then the Value.

| Field Name | Type | Value |
|---|---|---|
| Description | Literal | This is a large opportunity. Please call the customer ASAP. |

   For more information, see *Arguments You Can Define for a Process Property*.
3. Add a second property with the values shown in the following table.

| Field Name | Type | Value |
|---|---|---|
| Type | Expression | To Do |

# Defining the Runtime Event That Starts the Workflow Process

This task is a step in *Defining a Workflow Process That Creates an Activity for a Sales Representative*.

In this topic, you define the runtime event that starts the Workflow Process.

## To define the runtime event that starts the Workflow Process

1. In the Process Designer, click the Evaluate Oppty connector and in the Properties pane, define the values shown in the following table.

| Property | Value |
|---|---|
| Event Object Type | BusComp |

| Property | Value |
|---|---|
| | |
| Event Object | Opportunity |
| Event | WriteRecordNew |
| Type | Default |

When you define these properties, define them in the same order that this table lists them. Event Object and Event are context-sensitive according to the value you enter in Event Object Type. You must define Event Object Type first.

2. Choose the No connector and in the Properties pane, make sure the Type property is set to Default.
3. Choose the Yes connector and in the Properties pane, set the Type property to Condition.

# Defining a Decision Condition for a Decision Step

This task is a step in *Defining a Workflow Process That Creates an Activity for a Sales Representative*.

The following procedure shows how to define a decision condition for the decision step.

## To define a decision condition for a decision step

1. In the Process Designer, double-click the Yes connector to open the the Compose Condition Criteria dialog box.

   For more information, see *Adding a Branch Connector*.
2. In the Compose Condition Criteria dialog box, define a decision condition for the connector as follows:

   a. In the Compare To drop-down list, choose Business Component.

      The value you choose in the Compare To drop-down list instructs Siebel CRM to use an object, expression, or process property in the comparison. In this example, it compares the runtime value of a field from the Opportunity business component, as defined in the Object drop-down list. The items in the Object drop-down list are context-sensitive. The Workflow Process Editor changes them according to the value you choose in the Compare To drop-down list.

   b. In the Object drop-down list, choose Opportunity.

   c. In the Operation drop-down list, choose Greater Than.

      In this example, if the value you define in the Values window of the Compose Condition Criteria dialog box is greater than the runtime value of the Revenue field for the business component, then the Operation drop-own list defines the decision condition. You define the Revenue field in the Field window.

   d. In the Field window, choose Revenue.

   e. In the Values section, click New to add a new value and in the Add Value dialog box that appears, type 10000 and then click OK.

f. Click Add in the Compose Condition Criteria dialog box.
The Workflow Process Editor adds the decision condition that you define in the Compose Condition Criteria dialog box to the Conditions list (also in the dialog box). You can add another decision condition if required. For this example, you add only a single decision condition.
g. Click OK.

According to the decision condition defined in this example, at runtime, if the Revenue field of an opportunity is greater than 10000, then Siebel CRM runs the Yes branch. Otherwise, it runs the No branch.

After you define a decision condition, you can modify it as follows as required:

- **Update a condition.** Select the condition from the Conditions list in the Compose Condition Criteria dialog box, modify the condition, and then click Update.
- **Delete a condition.** Select the condition in the Conditions list in the Compose Condition Criteria dialog box, and then click Delete.
  When you delete a Condition:
  - It is deleted from the canvas and from the list applet for new Workflow Processes.
  - It is deleted from the canvas and becomes inactive in the list applet for existing Workflow Processes.

For more information, see *Creating a Decision Condition on a Branch Connector*.

# Preparing This Example for Testing

This task is a step in *Defining a Workflow Process That Creates an Activity for a Sales Representative*.

To prepare this example for testing, create an opportunity that matches the test criteria and note the Row Id of the opportunity. To run the test, use this Row Id in the properties of the Workflow Process.

## To prepare this example for testing

1. Validate the Workflow Process.
   For more information, see *Validate Tool* and *Validating the Workflow Process*.
2. Log in to the Siebel client and navigate to the Opportunities list.
3. Add an opportunity with the values shown in the following table.

| Name | Revenue |
|---|---|
| Large Opportunity to Test Workflow | $400,000 |

4. Select the record you just created in step 3, click Help and then click About Record.
5. Note the value of the Row Id in the Row # field.
6. Click anywhere on the canvas in the Process Designer, making sure not to select any Workflow Process step or connector.
   If you click an empty part of the canvas, then the Multi Value Property Window (MVPW) pane shows the process properties for the Workflow Process.

7. In the MVPW pane, go to the Object Id process property and set the Default String field for the property to the Row Id for the opportunity identified in step 5.

# Simulating the Workflow Process

This task is a step in *Defining a Workflow Process That Creates an Activity for a Sales Representative*.

In this topic you simulate the Workflow Process, and then examine simulation results.

## To simulate the Workflow Process

1. Validate and then simulate the Workflow Process using the Process Simulator to step through the entire Workflow Process – for more information, see *Process of Testing a Workflow Process*.
2. In the Siebel client, navigate to the Opportunities screen.
3. In the Opportunities list, drill down on the opportunity named Large Opportunity to Test Workflow.
4. Click the Activities view tab.

   A new Activity should exist and it must contain the description you defined in *Defining Properties and Arguments for Workflow Process Steps* . If it does, then the simulation finished successfully.
5. Remove the modifications you made for testing:
   - Close the Simulator.
   - In the WF Process Props list, choose the Object Id process property for your Workflow Process.
   - In the Properties pane, remove the Row Id from the Default String property for the Object Id Process Property.

# Verifying the Workflow Process

This task is a step in *Defining a Workflow Process That Creates an Activity for a Sales Representative*.

In this topic you verify the Workflow Process to make sure that it implements the required functionality.

## To verify the Workflow Process

1. Launch the Siebel Application Object Manager, such as Call Center or Siebel Communications.
2. Navigate to the Workspace Dashboard and open and inspect the Developer Workspace in which the Workflow Process was created or edited.
3. Verify that the Workflow Process implements the required functionality:
   - Create a new Opportunity record with a Revenue greater than $10,000.
   - Verify that an Activity has been associated with the Opportunity, including the description you defined earlier in this example.

     If this does not work as expected, then return to the Process Designer, make appropriate changes, and re-inspect the Developer Workspace in the Siebel Application Object Manager.

# Defining a Workflow Process That Attaches an Activity Plan to an Opportunity

In this topic you create a Workflow Process that attaches an activity plan to an opportunity. The following tasks are involved in developing this example:

1. *Creating the New Workflow Process*
2. *Defining Properties for Workflow Process Steps*
3. *Preparing Example Data for the Simulation*
4. *Validating and Testing the Workflow Process*
5. *Configuring the Button That Starts the Workflow Process*
6. *Verifying the Functionality of the Workflow Process*

In this example, you define a Workflow Process that creates an activity plan for an opportunity, and then navigates the user to a view that displays the new plan. The Workflow Process then waits for the user to enter more data and save changes before continuing. You use the Process Simulator and the Siebel client to test the Workflow Process.

## Creating the New Workflow Process

This task is a step in *Defining a Workflow Process That Attaches an Activity Plan to an Opportunity*.

To begin, you create the new Workflow Process, define step properties, and then validate the Workflow Process.

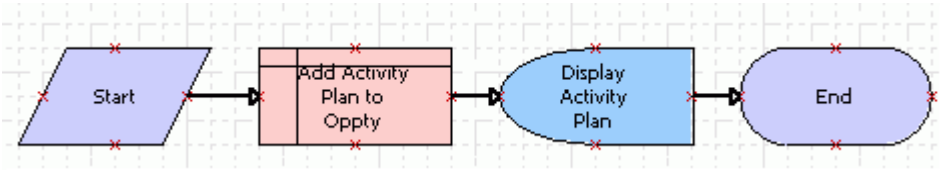
### To create the new Workflow Process

1. In the Process Designer, create a new Workflow Process with the values shown in the following table.

| Property | Value |
|---|---|
| Process Name | Create Plan |
| Business Object | Opportunity |
| Workflow Mode | Interactive Flow |

   For more information, see *Creating the Workflow Process*.
2. Add the following steps and connectors until your Workflow Process resembles the flow illustrated in the following figure:
   a. A Start step
   b. A Send Activity Plan to Oppty step
   c. A Display Activity Plan step
   d. An End step
   e. The following Connectors:

- A connector between step (a) and (b).
- A connector between between step (b) and (c).
- A connector between between step (c) and (d).




For more information, see *Adding Workflow Process Steps* and *Diagramming a Workflow Process*.

# Defining Properties for Workflow Process Steps

This task is a step in *Defining a Workflow Process That Attaches an Activity Plan to an Opportunity*.

In this topic you define properties for Workflow Process steps

## To define properties for Workflow Process steps

1. In the Process Designer, click the Add Activity Plan to Oppty step.
2. In the Properties pane, define the business component and the operation using the values shown in the following table.

| Property | Value |
|---|---|
| Business Component | Activity Plan |
| Operation | Insert |

If you configure an Insert operation on a business component, then you must supply a value for each business component fields that is required. To insert a new activity plan, you must provide the name of the template that Siebel CRM uses for the activity plan.

3. In the Process Designer, select the Add Activity Plan to Oppty step and in the Multi Value Property Window (MVPW) pane, add a new input argument with the values shown in the following table..

| Field Name | Type | Value |
|---|---|---|
| Template | Literal | A valid activity template name, such as Introductory Sales Call. |

For more information, see *Arguments You Can Define for a Process Property*.

4. In the Process Designer, select the Display Activity Plan step and in the Properties pane, define the property shown in the following table.

| Property | Value |
|---|---|
| User Interact View | Opportunity Activity Plan |

5. Click the connector that is located between the Display Activity Plan (user interact) step and the End step and in the Properties pane, define the properties shown in the following table.

| Property | Value |
|---|---|
| Event Object Type | BusComp |
| Event | WriteRecordUpdated |
| Event Object | Activity Plan |
| Type | Condition |

The WriteRecordUpdated runtime event signals the end of the Display Activity Plan (user interact) step. Siebel CRM must wait for the user to make and save a change to the Activity Plan before it resumes the Workflow Process. WriteRecordUpdated signals this change.

# Preparing Example Data for the Simulation

This task is a step in *Defining a Workflow Process That Attaches an Activity Plan to an Opportunity*.

Before you can test the Workflow Process, you create an opportunity that matches the test criteria. You create this opportunity, and then note the Row Id of the opportunity. To run the test, you then use this Row Id in the properties of the Workflow Process.

## To prepare example data for the simulation

1. Log in to the Siebel client application, such as Call Center or Communications.
2. Navigate to the Opportunities list.
3. Select an opportunity record, click Help and then click About Record.
4. Note the value of the Row Id that the Row # field contains, and then click OK.
5. In the Multi Value Property Window (MVPW) pane in the Process Designer, go to the Object Id process property and set the Default String field for this property to the Row Id of the opportunity you selected in step 3.

# Validating and Testing the Workflow Process

This task is a step in *Defining a Workflow Process That Attaches an Activity Plan to an Opportunity*.

In this topic, you validate and then test the Workflow Process.

## To validate and test the Workflow Process

1. Validate and then simulate the Workflow Process – for more information, see *Process of Testing a Workflow Process*.

   After the test reaches the user interact step in the simulation, you can use the Siebel client to change a value in the activity plan and then save your changes. For example, change the value in the Planned Start Date field or in the Description field. To reach the end step, you cannot click Next after the Opportunity Activity Plan view appears in the Siebel client. You must satisfy the condition that you defined on the branch to allow the Workflow Process to proceed to the end step.
2. Acknowledge the message that Siebel CRM displays at the end of the Workflow Process simulation.

   When Siebel CRM reaches the last step, it displays a message in the client that is similar to the following:

   ```
   Simulation terminated! Please check the Watch window for details.
   ```
3. Click Next to finish the end step.
4. In the Watch Window, view the Simulator Status field.

   > **Note:** In Web Tools, the Watch Window is called Workflow Process Data.

   If the Simulator Status field displays a message that is similar to the following, then the Workflow Process ended without error:

   ```
   Simulation ended successfully
   ```

# Configuring the Button That Starts the Workflow Process

This task is a step in *Defining a Workflow Process That Attaches an Activity Plan to an Opportunity*.

To start the Workflow Process in this example, you add a button to the Opportunity list. If the user clicks this button, then Siebel CRM starts the Workflow Process. For more information, see *Starting a Workflow Process*.

## To configure how the Workflow Process is started

1. In the Object Explorer, click Applet.
2. In the Applets list, query the Name property for Opportunity List Applet.
3. Edit the Applet Web Layout as follows:
   - In the Controls/Columns window, set the Mode to `3: Edit List`.

     Siebel sets the Mode to `1:Base` by default, so you must set it to `3:Edit List`. If the Controls/Columns window is not visible, then click View, Windows, and select the Controls Window menu item.

- Move a Button control from the palette to the layout, and then set the properties for the control using values from the following table.

| Property | Value |
|---|---|
| Caption-String Override | Create Plans |
| HTML Type | MiniButton |
| MethodInvoked | EventMethodCreateActivityPlan<br><br>**Note:** You must manually enter the value for the MethodInvoked property. |

A method that uses the following naming format does not require an applet or script on a business component to allow the event:

```
EventMethod a string value
```

- Preview the applet and make sure the new Create Plan button appears for the Opportunity list.
- Save your changes.

4. In the Object Explorer, click Workflow Process.
5. In the Workflow Processes list, query the Process Name property for the Create Plan Workflow Process that you created in *Creating the New Workflow Process* and then open the Workflow Process.
6. Click the connector that resides between the start step and the step named Add Activity Plan to Oppty, and then use the Properties pane to set properties for the connector using values from the following table.

| Property | Value |
|---|---|
| Type | Condition |
| Event Object Type | Applet |
| Event Object | Opportunity List Applet |
| Event | PreInvokeMethod |
| Subevent | EventMethodCreateActivityPlan |
| EventCancelFlag | TRUE |

| Property | Value |
|---|---|
| | |

This configuration defines a start condition for the Workflow Process so that Siebel CRM starts it if the EventMethodCreateActivityPlan event occurs on the Opportunity List Applet.

7. Validate and then simulate the Workflow Process.
   For more information, see *Process of Testing a Workflow Process*.

# Verifying the Functionality of the Workflow Process

This task is a step in *Defining a Workflow Process That Attaches an Activity Plan to an Opportunity*.

In this topic, you verify that the Workflow Process implements the required functionality.

## To verify the functionality of the Workflow Process

1. In the Siebel client, navigate to the Opportunities screen, then the All Opportunities view.
2. In the All Opportunities list, click Create Plan.
   This step starts the Workflow Process for an opportunity and navigates you to the Opportunity Activity Plan view.
3. Edit the Description field for the plan, and then save your changes.
4. Navigate to the Administration - Business Process screen, then the Workflow Instance Monitor view.
   For more information, see *Monitoring Workflow Process Instances*.
5. In the Workflow Process list, query the Name field for the Create Plan Workflow Process.
6. Click the Step Instances tab to view step instance details.
7. If the Workflow Process runs without error, then you can turn off monitoring:
   a. Navigate to the Administration - Business Process screen, then the Workflow Monitoring Configuration view.
   b. In the Monitoring Configuration list, query the name field for the Create Plan Workflow Process.
   c. Set the Monitoring Level field to 0-None.

# Defining a Workflow Process That Manages Research Activities for a Service Request

In this topic you create a Workflow Process that manages research activities for a service request. The following tasks are involved in developing this example:

1. *Creating the Workflow Process*
2. *Configuring the Workflow Process Steps and Connectors*
3. Validate and then simulate the Workflow Process – for more information, see *Process of Testing a Workflow Process*.
4. *Verifying the Workflow Process*

In this example, a Workflow Process detects if a user takes ownership of a service request, and then creates an activity of type Research. It then provides the service request owner with items that the owner can use to research the service request. You define a simple Workflow Process that a runtime event starts. You become familiar with the items that are involved in setting up this Workflow Process and you make sure that the Workflow Process and the runtime event work correctly.

# Creating the Workflow Process

This task is a step in *Defining a Workflow Process That Manages Research Activities for a Service Request*.

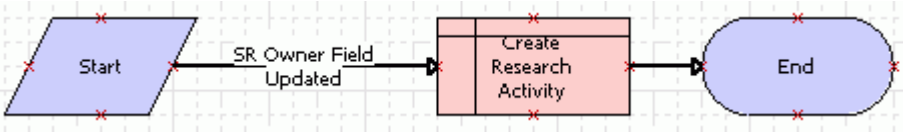
In this topic you create the Workflow Process.

## To create the Workflow Process

1. In the Process Designer, create a new Workflow Process with the values shown in the following table.

| Property | Value |
|---|---|
| Process Name | SR Assigned-Auto Create Activities |
| Business Object | Service Request |
| Group | (Leave this field empty.) |
| Workflow Mode | Service Flow |
| Auto Persist | No |
| Description | Automatically creates a research activity when SR ownership change is detected by using RTE |

   For more information example, see *Creating the Workflow Process*.
2. Add the following steps and connectors until your Workflow Process resembles the flow illustrated in the following figure:
   a. A Start step
   b. A Create Research Activity step
   c. An End step
   d. The following Connectors:
      - An SR Owner Field Updated connector between step (a) and (b).
      - A connector between step (b) and (c).



For more information, see *Adding Workflow Process Steps* and *Diagramming a Workflow Process*.

# Configuring the Workflow Process Steps and Connectors

This task is a step in *Defining a Workflow Process That Manages Research Activities for a Service Request*.

In this topic you configure the steps and connectors in the Workflow Process.

## To configure the steps and connectors in the Workflow Process

1. Click the SR Owner Field Updated connector and in the Properties pane, define properties for the connector as shown in the following table.

| Property | Value |
|---|---|
| Type | Condition |
| Event Object Type | BusComp |
| Event | SetFieldValue |
| Event Object | Service Request |
| Subevent | Owner |

2. Click the Create Research Activity step and in the Properties pane, define properties for the step as shown in the following table.

| Property | Value |
|---|---|
| Operation | Insert |
| Business Component | Action |

3. Select the Create Research Activity step and in the Multi Value Property Window (MVPW) pane, define two new input arguments for the step with the values shown in the following table.

| Field Name | Type | Value |
|---|---|---|
| Type | Literal | Research |

| Field Name | Type | Value |
|---|---|---|
| Description | Literal | Research the following: Siebel Bookshelf |

For more information, see *Arguments You Can Define for a Process Property*.

If you configure Siebel CRM to do an insert in a business component, then you must make sure that you define the fields that the business component requires. As an alternative, you can define a value in the Pre Default Value property for these fields. To access this property, you can expand the Business Component tree in the Object Explorer and then click Field.

# Verifying the Workflow Process

This task is a step in *Defining a Workflow Process That Manages Research Activities for a Service Request*.

In this topic you verify the Workflow Process to make sure that it implements the required functionality.

## To verify the Workflow Process

1. Launch the Siebel Application Object Manager, such as Call Center or Siebel Communications.
2. Navigate to the Workspace Dashboard and open and inspect the Developer Workspace in which the Workflow Process was created or edited.
3. Verify that the Workflow Process implements the required functionality:
    a. In the Siebel client, navigate to the Service Requests screen, and then the All Service Requests view.
    b. Create a new service request, and then step off the record.
    c. In the Service Request list, click the link in the SR # field.
    d. Examine the Activities tab. If the Workflow Process runs correctly, then Siebel CRM displays the following values in the Activities list:
        - Type: Research
        - Description: Will research the following: Siebel Bookshelf

# 13 Example Workflow Processes That Call a Business Service

## Example Workflow Processes That Call a Business Service

This chapter includes some example Workflow Processes that call a business service. It includes the following topics:

- *Examples That Use the Server Requests Business Service*
- *Examples That Use the Outbound Communications Manager Business Service*

## Examples That Use the Server Requests Business Service

In the following examples, the Workflow Process uses the Server Requests business service.

- *Using the Server Requests Business Service to Start a Workflow Process from a Script*
- *Using the Server Requests Business Service to Call EIM*

For more information, see *Server Requests Business Service*.

### Using the Server Requests Business Service to Start a Workflow Process from a Script

The following example code uses the Server Requests business service to start a Workflow Process from a script, and then passes field values to process properties:

```
//Example: Passing Field Values to Process Properties and start workflow from script
using the Server Request BS
function Invoke_Process()
{
try {
 var svc = TheApplication().GetService("Workflow Process Manager(Server Request)");
 var psInput = TheApplication().NewPropertySet();
 var psOutput = TheApplication().NewPropertySet();
 var bo = TheApplication().ActiveBusObject();
 var bc = bo.GetBusComp("Opportunity");
 var rowId = bc.GetFieldValue("Id");
 var accountId = bc.GetFieldValue("Account Id");
 psInput.SetProperty("ProcessName", "My Opportunity Process");
 psInput.SetProperty("Object Id", rowId);
 // Pass the value of the Account Id field to the Account Id process property
 psInput.SetProperty("Account Id", accountId);
 svc.InvokeMethod("RunProcess", psInput, psOutput);
}
finally
```

```
{
 psOutput = null;
 psInput = null;
 svc = null;
}
```

# Using the Server Requests Business Service to Call EIM

The example in this topic uses the Server Requests business service in a Workflow Process to call EIM. You can use a Workflow Process to start a server task. For example, you can use EIM to export base tables to `EIM_*` tables or to load `EIM_*` tables into base tables. Siebel CRM must pass the server component parameters that EIM requires in a child property set. You can use a wrapper business service for EIM, such as the Synchronous Assignment Manager Requests business service, or you can use the Server Requests business service.

Because the Server Requests business service can call a variety of server tasks, it does not contain definitions for parameters that are specific to a particular server component. Instead, it passes these parameters in a child property set that is not declared in the repository. To use Siebel to pass values in a child property set, you use the following dot notation:

```
[type].[property]
```

In this example, the Server Request Manager service does not care about the child type. It uses the first child it encounters and you can pass every parameter in the same child. For example

```
EIM.Config
```

## To use the Server Requests business service to call EIM

1. In the Workflow Processes list, create a new Workflow Process with the values shown in the following table.

| Property | Value |
|---|---|
| Process Name | EIM Export to IF (Tools) |
| Business Object | Account |
| Workflow Mode | Service Flow |

   For more information, see *Creating the Workflow Process*.
2. Open the Process Designer for the Workflow Process you defined in *Using the Server Requests Business Service to Call EIM*, and then create a new Workflow Process with the following steps and connectors until it resembles the flow in the following figure:
   a. A Start step
   b. An Export EIM step
   c. An End step.
   d. The following Connectors:
      - A connector between step (a) and step (b).

- A connector between step (b) and step (c).


For more information, see *Adding Workflow Process Steps* and *Diagramming a Workflow Process*.

3. Click the Export EIM step and in the Properties pane, define the values shown in the following table.

| Property | Value |
|---|---|
| Business Service Name | Synchronous Server Requests |
| Business Service Method | SubmitRequest |

4. Select the Export EIM step and in the Multi Value Property Window (MVPW) pane, add a new input argument with the values shown in the following table.

| Field | Value |
|---|---|
| Input Argument | Component |
| Type | Input Argument |
| Value | EIM |

For more information, see *Arguments You Can Define for a Process Property*.

5. Add another input argument to the Export EIM step with the values shown in the following table.

| Field | Value |
|---|---|
| Input Argument | EIM.Config |
| Type | Input Argument |
| Value | accnt.ifb |

6. To prepare the Workflow Process for testing, set up the accnt.ifb file in the following directory on the server where you will run the EIM process:

```
siebsrvr\Admin
```

7. Validate and then simulate the Workflow Process.

For more information, see *Process of Testing a Workflow Process*.

If the Workflow Process runs successfully, then it finishes without errors. Siebel CRM creates an EIM task log in the Server Tasks screen in the Siebel client each time you step through the Process Simulator.

8. Deliver the Workspace.

   After the Workflow Process is active, you can start it from a Workflow Policy, a script, or as a sub process from another Workflow Process.

# Examples That Use the Outbound Communications Manager Business Service

You can configure Siebel CRM to use the Outbound Communications Manager business service to send notifications through fax or email, such as notifications to contacts or employees. For more information about methods and arguments, see *Siebel CTI Administration Guide* .

For an example that uses the Outbound Communications Manager business service, see *Using the Outbound Communications Manager with a Substitution*

## Using the Outbound Communications Manager with a Substitution

The example in this topic uses a substitution variable when it calls the Outbound Communications Manager business service in a Workflow Process.

### To use the Outbound Communications Manager with a substitution

1. Create a new Workflow Process with the values shown in the following table.

| Property | Value |
|---|---|
| Process Name | Send SR Notification |
| Workflow Mode | Service Flow |
| Business Object | Service Request |

   For more information, see *Creating the Workflow Process*.

2. Add the following steps and connectors until your Workflow Process resembles the flow illustrated in the following image:
   a. A Start step
   b. A Send Notification step

c. An End step
d. The following Connectors:

- A connector between step (a) and step (b).
- A connector between step (b) and step (c).


For more information, see *Adding Workflow Process Steps* and *Diagramming a Workflow Process*.

3. Click the Send Notification business service step and in the Properties pane, define the values shown in the following table.

| Property | Value |
|---|---|
| Business Service Name | Outbound Communications Manager |
| Business Service Method | SendMessage |

4. Select the Send Notification step in the Process Designer and in the Multi Value Property Window (MVPW) pane, add a new input argument with the values shown in the following table.

5.

| Input Argument | Type | Value | Business Component Name |
|---|---|---|---|
| MsgBody | Expression | `"The Service Request Number" + [SR Number] + "was created with the abstract" + [SR Abstract]` | Service Request |

For more information, see *Arguments You Can Define for a Process Property*.

6. Add another input argument to the Send Notification step with the values shown in the following table.

| Input Argument | Type | Value |
|---|---|---|
| CommProfile | Literal | Enter the appropriate profile. The profile must correspond to a profile that the Siebel client displays in the Profiles view of the Administration - Communications screen, All Configurations view. When you enter the profile, you must use the exact same format that the Siebel client uses when it displays this profile. |

7. Validate and then simulate the Workflow Process.

For more information, see *Process of Testing a Workflow Process*.

# 14 Using Predefined Workflow Policies

## Using Predefined Workflow Policies

This chapter describes Workflow Policies, predefined Workflow Policies, and Workflow Policy programs that you can use. It includes the following topics:

- *About Workflow Policies*
- *Types of Predefined Workflow Policy Programs*
- *Using Predefined Workflow Policy Programs*
- *Using Predefined Workflow Policy Programs for Siebel Marketing*
- *Configuring a Predefined Workflow Policy*

## About Workflow Policies

This topic includes the following topics:

- *Overview of Workflow Policy Objects*
- *Structure of a Workflow Policy*
- *Sequence That a Workflow Policy Uses*
- *Hierarchy of Workflow Policy Objects*

You can use a Workflow Policy to start a Workflow Process. A Workflow Policy includes one or more Workflow Policy conditions. If the Workflow Policy conditions are met, then Siebel CRM runs the Workflow Policy action.

The name Workflow Policies replaces the name Workflow Manager, which Siebel CRM uses to refer to the Siebel Business Process automation tool in earlier Siebel CRM releases.

For more information, see *Administering Workflow Policies* and *Siebel Workflow Process and Workflow Policy Reference*

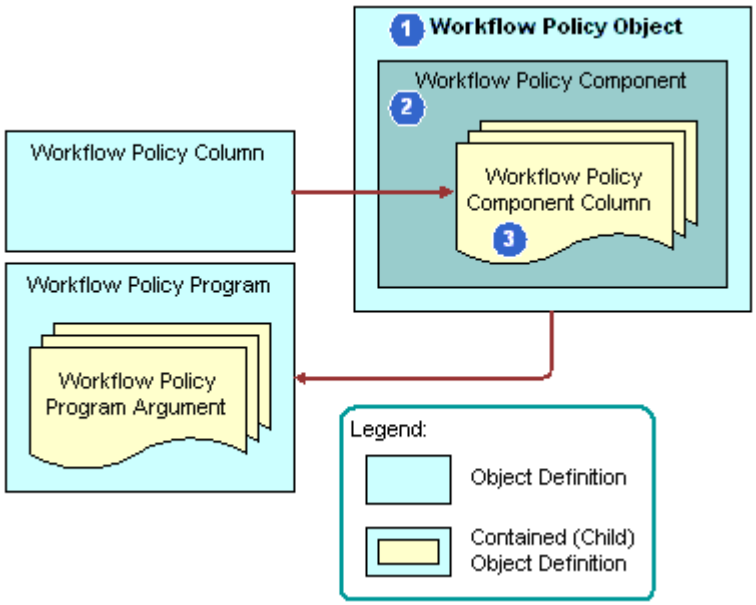
### Overview of Workflow Policy Objects

A Workflow Policy object provides the context in which Workflow Policies operate. The Workflow Policy object defines the set of tables and columns that a Workflow Policy monitors and how each table in the Workflow Policy object relates to the other tables. This collection of columns and the relations between the tables of the Workflow Policy object represent the entity in the Siebel Repository that you must monitor.

The settings in the Siebel Repository allow you to access predefined Workflow Policy objects for common business requirements, such as opportunities, service requests, or contacts. You can modify some predefined Workflow Policy objects through administrative screens in the Siebel client. To define a custom Workflow Policy object, you can make changes to the Siebel Repository.

# Relationships Between Objects of a Workflow Policy

The following figure illustrates the relationships that exist between Workflow Policy objects, Workflow Policy columns, and Workflow Policy programs.




As shown in this figure, the following relationships exist between Workflow Policy objects:

1. **Workflow Policy Object** A Workflow Policy object is a collection of Workflow Policy components. It is a parent to the Workflow Policy components and the Workflow Policy component columns.
2. **Workflow Policy Component** A Workflow Policy component is a logical mapping to a database table in the Siebel database. It defines the relations between the primary Workflow Policy component and other Workflow Policy components of a Workflow Policy object. Except for the primary Workflow Policy component, each Workflow Policy component defines a relation to another Workflow Policy component. To create this relation, Siebel CRM defines a source policy column and a target policy column. These source and target columns of a Workflow Policy component identify foreign key relations between the tables.

   A *primary* Workflow Policy component is a Workflow Policy component to which other Workflow Policy components are directly or indirectly related. Siebel CRM can define the Workflow Policy columns that are available for monitoring in the Workflow Policy from these Workflow Policy components. Each Workflow Policy object contains only one primary Workflow Policy component. The other Workflow Policy components of a Workflow Policy object are directly or indirectly related to the primary Workflow Policy component.
3. **Workflow Policy Component Column.** A Workflow Policy component column defines the columns in the Siebel database table that Siebel CRM can monitor. You can allow Siebel CRM to monitor these columns when you define Workflow Policy conditions for a Workflow Policy. For more information, see *How Siebel Indicates the Primary Workflow Policy Component*.

   To define a Workflow Policy object and its components, it is recommended that you familiarize yourself with the Siebel Data Model. For more information, see *Siebel Data Model Reference* on My Oracle Support.

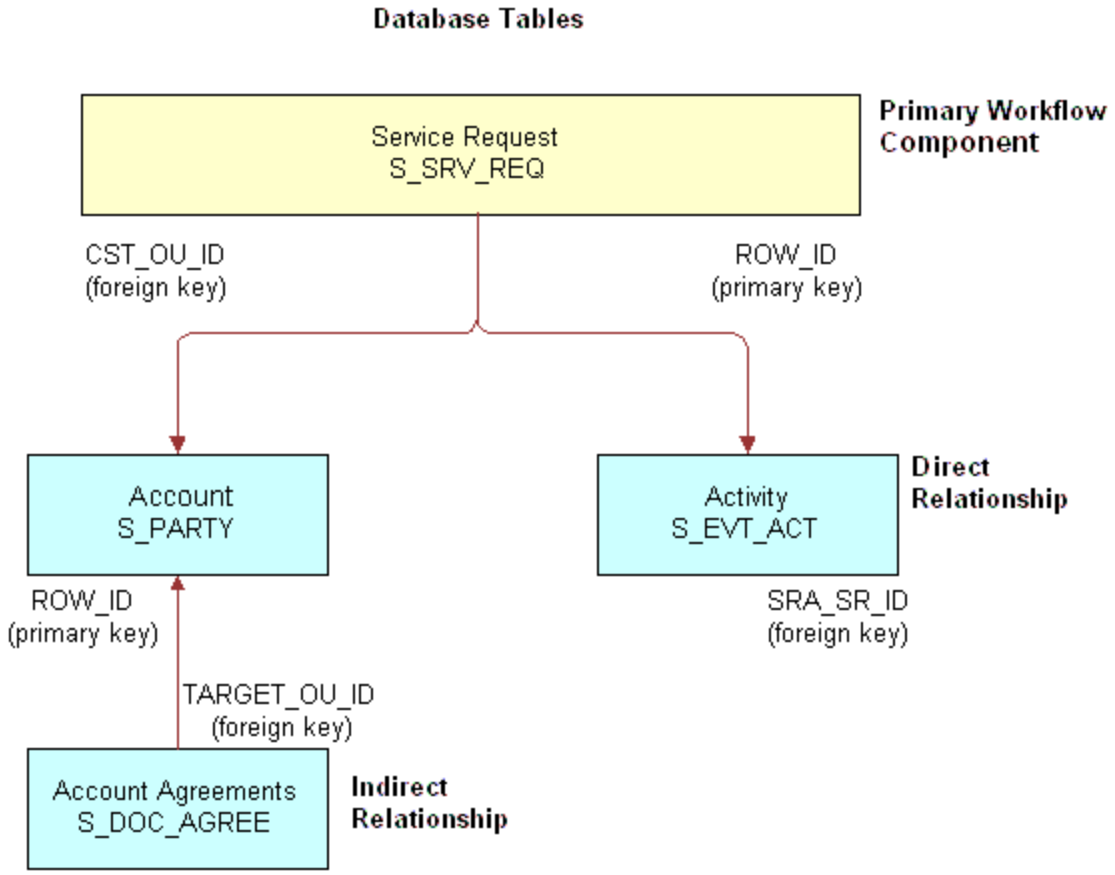
# Viewing the Hierarchy That Exists Between Workflow Policy Objects

Each Workflow Policy component can expose a number of Workflow Policy component columns. In the Object Explorer, a Workflow Policy component column is the child object of a Workflow Policy component, which is itself a child object of a Workflow Policy object. To view this hierarchy, see *Hierarchy of Workflow Policy Objects*.

# Example of an Entity Relationship Diagram for a Workflow Policy

The following figure displays the entity relationship diagram of four Workflow Policy components for a service request. It shows each of the components, their relations to one another, and the columns that are of interest as follows:

- There are four components: Service Request (S_SRV_REQ) labelled primary Workflow Policy component, Account (S_PARTY), Activity (S_EVT_ACT), and Account Agreements (S_DOC_AGREE).
- There is a direct relationship between Service Request and Account (via CST_OU_ID, foreign key).
- There is a direct relationship between between Service Request and Activity (via ROW_ID, primary key).
- There is an indirect relationship between Service Request and Account Agreements via the Account object (via the TARGET_OU_ID of S_DOC_AGREE referencing the ROW_ID of S_PARTY)



# Using a Workflow Policy to Monitor Tables

A Workflow Policy can monitor only Siebel database tables. You cannot use a Workflow Policy to monitor a database table that is external to Siebel CRM.

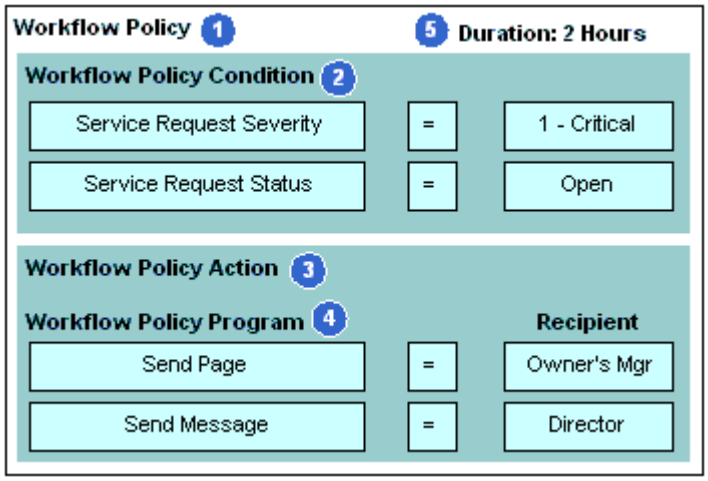
Workflow Policies can only be configured to monitor user data tables. Specific examples of tables that cannot be monitored include, but are not limited to, the following:

1. EIM tables.
2. Siebel Remote-related tables such as S_DOCK_TXN_LOG.
3. Repository tables of any kind.
4. Tables supporting server processing such as S_SRM_REQUEST.

# Structure of a Workflow Policy

A *rule* includes an IF/THEN structure that provides the basic underlying logic of a Workflow Policy.

A rule includes a Workflow Policy condition and a Workflow Policy action. If the Workflow Policy conditions are met, then Siebel CRM calls an action. The following figure shown an example of a Workflow Policy.




As shown in this figure, a typical Workflow Policy includes the following components:

1. **Workflow Policy.** A Workflow Policy includes conditions and actions. A Workflow Policy represents rules that the Siebel database monitors.
2. **Workflow Policy Condition.** This represents a situation that causes something to happen.
3. **Workflow Policy Action.** An action that Siebel CRM calls if a Workflow Policy condition is met.
4. **Workflow Policy Program.** A predefined program that provides arguments for the Workflow Policy action.
5. **Duration.** The amount of time that Workflow Policy conditions exist to allow the conditions of the policy to be met. For more information, see *Determining Settings for the Workflow Policy and Policy Conditions*.

A Workflow Process can support the functionality that is available with a Workflow Policy. It is recommended that you use a Workflow Policy to start a Workflow Process and to use a Workflow Process to define an action.

# Workflow Policy Condition

A *Workflow Policy condition* is an object that expresses an object or the relationship of an attribute to a value. For example, a Workflow Policy condition can target data, such as the severity of a service request. The Workflow Policy condition compares severity data to a value, such as 1-Critical. In this example, the combination of the following items define the condition:

- Data element
- Service request severity
- A comparison operation, such as the equal sign (=)
- The value, such as 1-Critical

The fact that the severity of a service request is 1-Critical is a problem only if the Workflow Policy condition remains valid for an extended amount of time, such as two hours. In this example, to make sure the condition remains valid, you can set a two hour duration on the Workflow Policy. This duration is part of the condition. Siebel CRM runs the policy actions only if the conditions are met in the time period that the duration measures.

A Workflow Policy action can occur if a duration is not set. For example, Siebel CRM can send an email to a sales manager each time a sales representative quotes a discount rate that exceeds 25 percent on revenue that is less than $100,000.

A Workflow Policy can contain more than one Workflow Policy condition. These conditions must be met before an action can occur. For example, a service request with a 1-High severity and a two hour duration might be important only if another comparison is also valid, such as if the service request status is Open. The condition is the combination of the following comparison operations:

```
SR Severity = 1-Critical AND SR Status = Open
```

A Workflow Policy supports only the AND operand between Workflow Policy conditions. It does not support the OR operand. To monitor the service request severity to be 1-Critical or 2-High and the status to be Open, you can use the IN operand to evaluate the OR logic of the severity:

```
SR Severity IN ('1-Critical','2-High') AND SR Status = Open
```

To implement OR operand logic, you can define multiple Workflow Policies for each Workflow Policy condition. For more information, see *Examples of Configuring Workflow Policies*.

# Workflow Policy Action

A *Workflow Policy action* is an event that occurs if the Workflow Policy conditions are met. It includes the following items:

- An *action*, which is a type of request, such as Send an Urgent Page
- The *action parameters*, which are the arguments, such as the name of the recipient of the page and the alphanumeric text that Siebel CRM transmits with the page

You can define multiple actions for one Workflow Policy, such as sending a text message to one person and an email to another person. You can reuse an action in multiple Workflow Policies. For more information, see *Customizing Workflow Policy Objects*

You can also use a Workflow Policy action to start a Workflow Process.

You can pass a constant from a Workflow Policy action to a Workflow Process. For more information, see *Passing a Constant from a Workflow Policy Action to a Workflow Process*.

# Workflow Policy Program

A *Workflow Policy program* is a predefined program that provides arguments for a Workflow Policy action. It identifies the action that Siebel CRM runs if the Workflow Policy conditions are met. A Workflow Process can run the action that a Workflow Policy program defines.

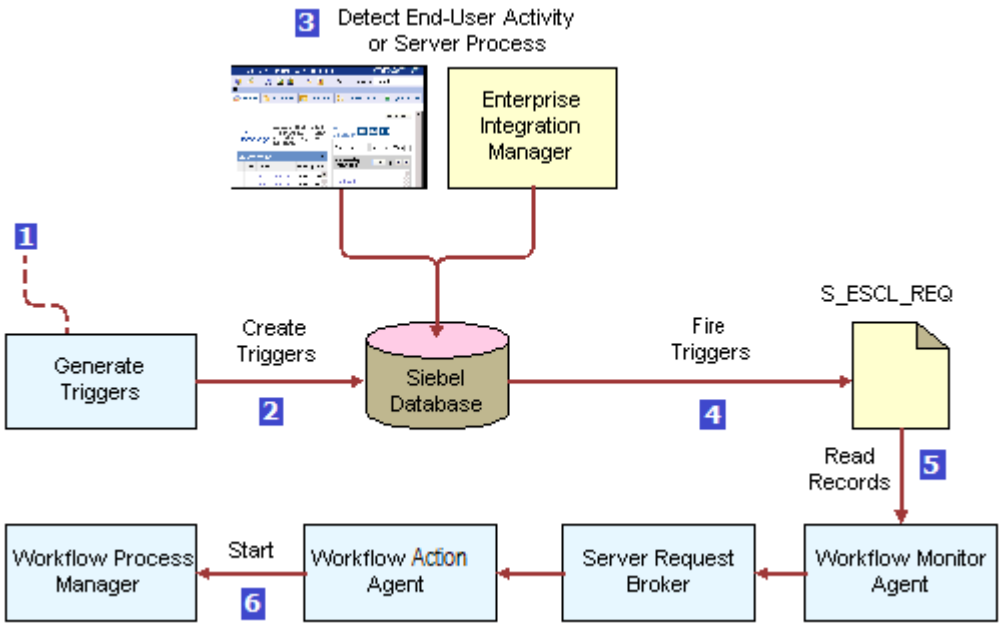
You can use a Workflow Policy program in multiple actions and you can use actions in multiple Workflow Policies.

An example of a Workflow Policy program includes *Send Message* (sends an email to one or more recipients) and *Generic Request Server* (sends a server request to the server). For more information about the different types of Workflow Policy programs, see *Types of Predefined Workflow Policy Programs*.

For more information about Workflow Policy programs and argument object types, see *Siebel Object Types Reference* .

# Sequence That a Workflow Policy Uses

The following figure illustrates the sequence of steps that a typical Workflow Policy follows.




As shown in this figure, the sequence for a typical Workflow Policy is as follows:

1. **Define Workflow Policies.** Administrator defines Workflow Policies to be monitored.
2. **Create Triggers.** To enforce Workflow Policies, Siebel CRM uses database triggers at the data layer. If the conditions for a Workflow Policy are met, then these database triggers capture the database event and provide this information to the Workflow Policy Manager server component.
3. **Detect End-User Activity or Server Process.** An end-user activity or server process occurs.
4. **Fire Triggers.** Fire triggers populating the S_ESCL_REQ table with information about the policy being violated.

5. **Read Records.** The Workflow Monitor Agent reads records in the S_ESCL_REQ table. To process requests, it then runs the actions that are defined for the Workflow Policy. The Workflow Monitor Agent is a component of the Workflow Policy Manager.
6. **Start Workflow Actions.** Start the Workflow Actions that have been defined, such as, invoking a Workflow Process, sending a text message, or sending an email.

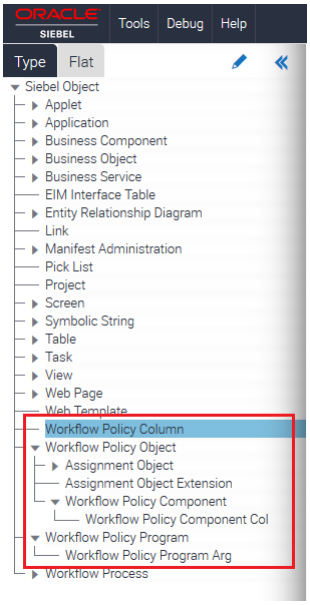
For more information about:

- Workflow Action Agent, see *Running a Workflow Policy with the Workflow Action Agent*.
- Generate Triggers, Workflow Process Manager and Workflow Monitor Agent, see *Siebel Workflow Process Runtime Architecture* .
- The S_ESCL_REQ table, see *Fixing Problems in the S_ESCL_REQ Table*.

# Hierarchy of Workflow Policy Objects

You can modify an existing Workflow Policy and create a new Workflow Policy to meet your business requirements. If Workflow Policy objects are not shown in the Object Explorer, then you can add them. For more information, see *Object Hierarchy That Workflow Processes Use*.

The following figure shows the Workflow Policy object type hierarchy in the Object Explorer.




As shown in this figure, the following Workflow Policy objects are available under Siebel Object:

1. Workflow Policy Column
2. Workflow Policy Object
    - Workflow Policy Component

> **Note:** Siebel Assignment Manager is closely coupled with Siebel Workflow Policies, but details are not included here. For more information on Assignment Manager, see *Siebel Assignment Manager Administration Guide* .

3. Workflow Policy Program
    - Workflow Policy Program Arg

## How Siebel Indicates the Primary Workflow Policy Component

The Primary property provides a visual indication of the Workflow Policy component that is the primary Workflow Policy component and the Workflow Policy components that are not primary. For example, the Primary property for the Account Workflow Policy component includes a check mark, which defines Account as the primary Workflow Policy component. Other Workflow Policy components that are shown in the Workflow Policy Components list are nonprimary Workflow Policy components of the Account Workflow Policy object.

# Types of Predefined Workflow Policy Programs

A Workflow Policy uses Workflow Policy actions that reference Workflow Policy programs that are predefined. This topic describes the different types of predefined Workflow Policy programs you can use. It includes the following subtopics:

- *Overview of Predefined Workflow Policy Programs*
- *Workflow Policy Program That Sends a Page*
- *Workflow Policy Program That Sends an Email*
- *Workflow Policy Program That Sends a Notification*
- *Workflow Policy Programs That Run a Database Operation*
- *Workflow Policy Program That Runs an External Program*

Workflow Policies come with a number of predefined Workflow Policy programs that you can use to meet specific business requirements. You can modify a predefined Workflow Policy program that already exists rather than creating a new Workflow Policy program.

## Overview of Predefined Workflow Policy Programs

A Workflow Policy program is a generic event that determines an action. It defines the actions that Siebel CRM runs if the Workflow Policy conditions are met. You can view a Workflow Policy program as follows:

- **To view a Workflow Policy program.** Open the Siebel client, navigate to the Administration - Business Process screen, and then the Actions view.
- **To view object definitions of a Workflow Policy program.** In the Object Explorer, click Workflow Policy Programs.

| Program Type | Workflow Policy Program | Description |
|---|---|---|
| Send Page | Send Page | Sends a generic page. |
| Send Page | Send Opportunity Page | Sends a page regarding an opportunity. |

| Program Type | Workflow Policy Program | Description |
|---|---|---|
| Send Page | Send Quote Page | Sends a page regarding a quote. |
| Send Page | Send SR Page | Sends a page regarding a service request. |
| Send Email | Send Email | Sends a generic email message. |
| Send Email | Send Opportunity Email | Sends an email regarding an opportunity. |
| Send Email | Send Quote Email | Sends an email regarding an opportunity quote. |
| Send Email | Send SR Email | Sends an email regarding a service request. |
| Send Broadcast Message | Send Message Broadcast | Sends a generic alert. |
| Send Broadcast Message | Send SR Message Broadcast | Sends an alert regarding a service request. |
| Send Broadcast Message | Send Opportunity Message Broadcast | Sends an alert regarding an opportunity. |
| Run External Program | Run External Program | Runs an external program. |
| Database Operation | Change SR Close Date to Today | Closes the SR with today's date. |
| Database Operation | Change SR Owner | Changes the owner of the service request. |
| Database Operation | Change SR Group | Changes the group of the service request. |
| Database Operation | Change SR Owner to Manager | Changes the owner of the service request to the current manager for the owner. |
| Database Operation | Change SR Priority | Changes the priority of the service request to a new value. |
| Database Operation | Change SR Severity | Changes the severity of the service request to a new value. |
| Database Operation | Change SR Status | Changes the status of the service request to a new value. |
| Database Operation | Change SR Sub-Status | Changes the Sub-Status of the service request to a new value. |
| Database Operation | Create SR Activity | Creates an activity for a service request. |
| Database Operation | Create Opportunity Activity | Creates an activity for an opportunity. |

# Workflow Policy Program That Sends a Page

The following table describes arguments and values for the Send Page Workflow Policy program type. If you choose a Send Page type of Workflow Policy program in the Actions view in the Siebel client, such as Send SR Page, then Siebel CRM displays the Send Page Arguments form and the Recipients list. The Send Page Arguments form allows you to define a template that Siebel CRM uses to create the message it sends to the recipient that you add in the Recipients list.

| Argument | Description |
|---|---|
| Numeric Message Template | Specifies the numeric message if the pager is numeric. |
| Alpha Message Template | Specifies the text message if the pager is alphanumeric. The following phrase is reserved in Siebel so do not use this phrase in a message: *Workflow Policy Object*. |
| Available Substitutions | Specifies the dynamic fields that you can use in the Alpha Message Template.<br><br>If the action runs, then Siebel updates the substitution value with the value from the record that meets the Workflow Policy conditions. |
| Request Key | A string indicating which Page Manager should execute the action. You use this when multiple Page Managers are running. When you specify a request key string, it should match the Request Key parameter of the Page Manager that you want to execute the action. Leave this argument blank when you are running one Page Manager or when the Page Manager that executes the action is not important. |

Note the following:

- Workflow Policies determine how to format the message depending on the type of pager that the person who receives the page uses.
- If none of the message arguments contains a value, then Workflow Policies logs an error message and Siebel does not finish the action.
- Siebel can send a page only to an employee. It stores pager information for an employee in the Employee Administration view. The Siebel database does not store pager information for contacts.
- Siebel can substitute values in a message that originates from the Available Substitutions field.

# Using Numeric Paging

Numeric paging is inherently not reliable because of insufficient computer protocols that exist to send a numeric page. If Siebel must send a numeric page, then you can use the Pager Pin field in the employee table to control the delay that occurs between dialing the paging phone number and sending the numeric message. You can add commas to the Pager Pin field. Two commas is equal to approximately a one second delay. It is recommended that you avoid using numeric paging in a Siebel application that is mission critical.

# Workflow Policy Program That Sends an Email

The following table describes arguments and values for the Send Email Workflow Policy program type. If you choose a Send Email type of Workflow Policy program in the Actions view in the Siebel client, such as Send SR Email, then Siebel displays the Send Message Arguments form and the Recipients list. The Send Message Arguments form allows you to define a template that Siebel CRM uses to create the message it sends to the recipient that you add in the Recipients list.

| Argument | Description |
|---|---|
| Subject | Specifies the Subject line of the email message. |
| Message Template | Specifies the message text. The maximum length is 2000 characters, including variable substitutions. The following word is reserved in Siebel so do not use this word in a message: *Current*. |
| Repeating Message | Specifies the message that Siebel CRM repeats if the Consolidate field contains a check mark on the Workflow Policies view. The following word is reserved in Siebel so do not use this word in a message: *Current*. |
| Available Substitutions | Specifies a dynamic field that you can use in the Subject, Message Template, or Repeating Message. If Siebel CRM runs the action, then it updates the substitution value with the value from the record that meets the Workflow Policy conditions. |
| Request Key | A string indicating which Email Manager should execute the action. You use this when multiple Email Managers are running. When you specify a request key string, it should match the Request Key parameter of the Email Manager that you want to execute the action. Leave this argument blank when you are running one Email Manager or when the Email Manager that executes the action is not important. |

# Sending an Email to an Address Stored in a Custom Field

You can configure Workflow Manager to send an email to an email address that is defined in a custom field, including to an address that Siebel stores in a column other than S_CONTACT.EMAIL_ADDR.

In the example shown in the following procedure, a new Workflow Policy Program based on the Activity workflow object is defined which will allow an alternative email address to be used when emailing an employee. In this example, an alternative email address has been stored in an extension column of the S_EMPLOYEE (for Siebel 6.x) or S_CONTACT (for Siebel 7.x and later) table.

## To send an email to an address stored in a custom field

1. Create a new Workflow Policy Program with the values shown in the following table.

| Name | Workflow Object |
|---|---|
| Send Activity Email | Activity |

| Name | Workflow Object |
|---|---|
| | |

The Workflow Policy Program must include a relative recipient, which references the database column that contains the email address. The relative recipient is defined as arguments to the Workflow Policy Program.

2. Add the following program arguments:

   a. These program arguments are the standard fields required for a send email program:

| Name | Required | Visible |
|---|---|---|
| Email Message | N | Y |
| Email Message Repeated | N | Y |
| Email Request Key | N | Y |
| Email Subject | N | Y |
| Primary Id | N | Y |

   b. These program arguments are used to obtain the values of the relative Employee ID and the alternative email address from an extension column in the S_EMPLOYEE (for Siebel 6.x) or S_CONTACT (for Siebel 7.x and later) table. The literal `'Send to Address'` is required when sending an email to an address stored in an alternative column.

| Name | Default Value | Picklist |
|---|---|---|
| Address Main | `Select ACT.OWNER_PER_ID, 'Send to Employee' from &Table_Owner.S_EVT_ACT ACT where ACT.ROW_ID = ?` | Workflow Relative Type Picklist |
| Address Alternative | `Select EMP.X_EMAIL_ALTERNATIVE, 'Send to Address' from &Table_Owner.S_CONTACT EMP,&Table_Owner.S_EVT_ACT ACT where ACT.ROW_ID = ? and EMP.ROW_ID = ACT.OWNER_PER_IN` | Workflow Relative Type Picklist |

c. These program arguments provide the mechanism for defining the recipients in the Workflow Action Arguments view in the Siebel client. The `Address Main` and `Address Alternative` values will be available when the Recipients are defined for the Action, and `Send To Relative` is selected as the Recipient Type.

| Name | Visible | Picklist | Source | Applet |
|---|---|---|---|---|
| Send to Address | Y | Picklist SR Owner | N/A | N/A |
| Send to Employee | Y | N/A | Id | Employee Notification Pick Applet |
| Send to Relative | Y | Workflow Relative Type Picklist | Name | N/A |

(Optional) Define other program arguments, as required, to provide suitable substitution values from other columns in the database.

# Workflow Policy Program That Sends a Notification

The following table describes the arguments and values for the Send Broadcast Message program type. If you chose a Send Message Broadcast type of Workflow Policy program in the Actions view, then Siebel CRM opens the Send Message Broadcast Arguments form.

| Argument | Description |
|---|---|
| Activation | Specifies the date and time for which the notification is active. You can use the CURRENT variable when you define the activation date. For more information, see *Using Date Calculations in the Conditions List*. |
| Expiration | Specifies the date and time when the notification expires. You can use the CURRENT variable when you define the activation date. For more information, see *Using Date Calculations in the Conditions List*. |
| Abstract | A short description of the notification. |
| Message Template | Specifies the notification text. The maximum length is 2000 characters, including variable substitutions. The following word is reserved in Siebel so do not use this word in a message: *Current*. |
| Severity | Specifies the severity of the notification. |
| Available Substitutions | Specifies the dynamic fields that you can use in the Abstract and Message Template arguments. If Siebel CRM runs the action, then it updates the substitution value with the value from the record that meets the Workflow Policy conditions. |

# Activating the Check New Broadcasted Message Workflow Policy

The Check New Broadcasted Message Workflow Policy monitors the S_BRDCST_MSG table. If Siebel adds a record to this table, then it starts the Notify Broadcasted Message Workflow Process to send a new notification. If the Type property of a Workflow Policy program is set to Send Broadcast Message, and if your implementation uses caching for notifications on an Object Manager server component, then you must activate the Check New Broadcasted Message Workflow Policy. This policy belongs to the Siebel Messaging policy group. For more information about:

- Activating a Workflow Policy, see *Overview of Creating Database Triggers*.
- Configuring caching for notifications, see *Siebel Fundamentals Guide* .

# Workflow Policy Programs That Run a Database Operation

The following table describes arguments and values for the Database Operation Workflow Policy program type. Siebel CRM comes predefined with a number of database operation programs. If you choose a Database Operation type of Workflow Policy program in the Actions list, such as Create Opportunity Activity, then Siebel CRM displays the Arguments list. You can add these arguments.

| Argument | Description |
|---|---|
| Name | Specifies the name of the column that Siebel CRM updates. |
| Required | Indicates that the argument is required. |
| Value | Specifies the updated value of the column. If you define a substitution in the program, then you can use a substitution in the value. To add a substitution to the value, enclose the variable in brackets, for example, as follows:<br><br>`[SR Num]` |

# Workflow Policy Program That Runs an External Program

The following table describes arguments and values for the Run External Workflow Policy program type. If you choose a Run External Workflow Policy program type in the Actions view, then Siebel CRM displays the External Programs Arguments list. For more information, see *Configuring a Workflow Policy Action That Runs an External Program*.

| Argument | Description |
|---|---|
| Executable Name | Specifies the path and name of the executable to run. For example, the executable starts from the Siebel Server. The executable can be a batch program.<br><br>If you do not provide a path, then Siebel assumes the executable is located in the directory where you install the Siebel Server. For example, if you install the Siebel Server in the `c:\Siebel\ses\siebsrvr` directory, then Siebel uses the following default path for the executable: |

| Argument | Description |
|---|---|
| | `c:\Siebel\ses\siebsrvr\bin`<br>The external program cannot be an interactive program that requires a user interface or that accesses the Windows desktop. |
| Command Line | Include the parameters that Siebel must send to the executable. To send multiple substitution parameters, you must include a space between each parameter. For example:<br>`"[FirstName]"^"[LastName]"`<br>where ^ (circumflex) is a space<br>You must use double quotes to enclose a substitution parameter that includes a space. Siebel sends these quotes as part of the parameter.<br>For more information, see the topic about the Workflow Policy Program object type in *Siebel Object Types Reference* . |
| Execute Type | You can use one of the following values for the Execute Type:<br>• **Wait.** Siebel waits for the external program to finish, and then examines the code that the external program returns. If the return code is not 0, then an error occurs.<br>• **No Wait.** Siebel runs the external program in the background and then continues processing. It does not examine the return code.<br>If the external program is a Visual Basic program that creates a file, then you must set the Execute Type argument to Wait. If you set Execute Type to No Wait, then Visual Basic attempts to write the file (twice), thus corrupting the data. |
| Available Substitutions | Specifies dynamic fields that you can use as command line parameters. If the action runs, then Siebel updates the substitution value with the value from the record that caused it to run the Workflow Policy program. |

# Using Predefined Workflow Policy Programs

The following examples use predefined Workflow Policy programs. To view these programs, click Workflow Policy Program in the Object Explorer.

- *Using a Workflow Policy Program to Manage the Service Request Close Date*
- *Using a Workflow Policy Program to Assign a Service Request Owner*
- *Using a Workflow Policy Program to Escalate a Service Request*
- *Using a Workflow Policy Program to Send a Quote Through a Pager*
- *Using Predefined Workflow Policy Programs for Siebel Marketing*.

# Using a Workflow Policy Program to Manage the Service Request Close Date

The example in this topic uses a Workflow Policy program to close service requests that are marked as resolved but not closed. You can define a Workflow Policy so that if a service request includes an activity of type Resolution, and if the service request is open for more than five days, then Siebel CRM changes the service request close date to the date for today. If this Workflow Policy calls the Workflow Policy program, then the Workflow Policy program changes the current system date to the value that the Close Date field of the service request contains.

| Argument Name | Description |
|---|---|
| Primary ID | Contains the Row Id of the service request that meets the conditions for the Workflow Policy. |
| Primary Table<br><br>Operation Type | Specifies the S_SRV_REQ table and the operation that Siebel CRM runs to do the Update operation. |
| Sql Statement | Contains a SQL statement to get the current date. It uses the following format:<br><br>`select sys_extract_utc(current_timestamp)`<br>`from &Table_Owner.S_DUAL`<br><br>To get the current date, this statement calls the following Siebel function: **now**.<br><br>The SQL can include a math function. For example, the following SQL statement returns the current date plus seven days:<br><br>`select {fn now()}+7 from &Table_Owner.S_DUAL`<br><br>Different RDBMS databases use different formats for the same function. For example, in MSSQL, the GetDate function returns the current date.<br><br>The column name length must be less than 30 characters. For example, the following statement is not valid because it is more than 30 characters in length:<br><br>`TO_CHAR(CREATED,'dd month yyyy','NLS_DATE_LANGUAGE=FRENCH')` |
| Sql Statement Outputs | Siebel CRM gets the value for the Today variable from the SQL statement. |
| New Close Date (Column) | Specifies to update the ACT_CLOSE_DT column in the record. |
| New Close Date | Specifies to update the field to the value of Today. |
| Update Row Id | Specifies the Row Id of the record you must update. This argument is the same as the value of the Primary ID. |

# Using a Workflow Policy Program to Assign a Service Request Owner

The example in this topic uses a predefined Workflow Policy program to assign a service request that is not yet assigned. If an open service request is not assigned for a certain amount of time, then this Workflow Policy program assigns it to a person who is considered an expert handling this type of the service request. This configuration allows the appropriate person to view the incoming service request, choose a new owner from a drop-down list, and then place this owner in the service request field that matches the Workflow Policy condition.

| Argument Name | Description |
|---|---|
| Primary ID | Contains the Row Id of the service request that meets the Workflow Policy condition. |
| Primary Table<br>Operation Type | Specifies the S_SRV_REQ table. Specifies to run the Update operation. |
| New Owner (Column) | Specifies to update the Owner_EMP_ID field in the record that Siebel CRM updates. |
| New Owner | Specifies to display a drop-down list that allows the user to assign a new owner. |
| Visible | If checked, then indicates that the drop-down list is visible to the user. |

# Using a Workflow Policy Program to Escalate a Service Request

The example in this topic uses a predefined Workflow Policy program to assign an open service request to a manager. If the service request is not closed within a certain amount of time, then Siebel assigns it to the manager of the service request owner. This Workflow Policy program does the following:

- Uses the Primary ID as input into a SQL statement
- Uses an SQL query statement to get the current value of the Manager field
- Sets the New Owner field to default to the current value of the Manager field
- Allows the user to use a drop-down list to update the New Owner field

| Argument Name | Description |
|---|---|
| Primary ID | Contains the Row Id of the service request that meets the Workflow Policy condition. |
| Primary Table | Specifies the S_SRV_REQ table. |
| Operation Type | Specifies to run the Update operation. |

| Argument Name | Description |
|---|---|
| | |
| New Owner (Column) | Specifies to update the Owner_EM_ID field in the record that Siebel CRM updates. |
| New Owner | Specifies to display a drop-down list that allows the user to assign a new owner. |
| Sql Statement | The Workflow Policy Monitor must use object definitions that are contained in the Workflow Policy object, Workflow Policy components, and Workflow Policy columns. If your configuration uses a Workflow Policy program, and if this configuration references Siebel tables, then you must use SQL code to explicitly join the base table.<br><br>The following example SQL statement joins four tables. These joins allow Siebel CRM to access the data that resides in these tables:<br><br>`SELECT MGRPOS.PR_EMP_ID FROM`<br>`&TABLE_OWNER.S_POSTN POS, &TABLE_OWNER.S_EMPLOYEE EMP,`<br>`&TABLE_OWNER.S_POSTN MGRPOS, &TABLE_OWNER.S_SRV_REQ SR`<br>`WHERE SR.ROW_ID = ?`<br>`AND SR.OWNER_EMP_ID = EMP.ROW_ID`<br>`AND EMP.PR_POSTN_ID = POS.ROW_ID`<br>`AND POS.PAR_POSTN_ID = MGRPOS.ROW_ID`<br><br>This SQL gets only one field.<br><br>This following SQL statement uses the question mark as a placeholder to input the value of the Primary ID. Siebel CRM replaces the question mark with the Primary ID:<br><br>`SR.ROW_ID = ?` |
| Sql Statement Inputs | Set to the value of Primary ID. |
| Sql Statement Outputs | Set to the value of Manager. |

# Using a Workflow Policy Program to Send a Quote Through a Pager

This example uses a predefined Workflow Policy program to send a quote depending on the relationship that exists between the value for the quote and the revenue value of the opportunity that the quote references. If a quote contains a value that is less than some percentage of the revenue for the opportunity, then Siebel CRM considers it as heavily discounted, and then sends a page to a designated employee.

# Using Predefined Workflow Policy Programs for Siebel Marketing

The following examples show how to use Workflow Policy programs to assist with a marketing campaign.

- *Using a Workflow Policy Program to Send Email for a Marketing Campaign*
- *Using a Workflow Policy Program to Create Activities for a Marketing Campaign*
- *Using a Workflow Policy Program to Assign a Contact to a Marketing Campaign*
- *Developing a Workflow Policy That Manages a Marketing Campaign*

## Using a Workflow Policy Program to Send Email for a Marketing Campaign

The Send Campaign Email Workflow Policy program allows a marketer to send an email to the targets of the marketing campaign, such as contacts or prospects. To personalize these emails, it uses Available Substitutions in the Send Message Arguments form, such as Prospect First Name. To choose the Recipient Type, you use the Recipients list in the Workflow Policy view in the Siebel client. The campaign contacts and prospects are visible in the Contacts/Prospects list in the Campaign Administration view.

### Modifying Available Substitutions

To add a new substitution to the Available Substitutions list, you modify the SQL Statement Outputs property of the Workflow Policy program.

#### To modify available substitutions

1. Query the Name property of the Workflow Policy Programs list for Send Campaign Email.
2. In the Workflow Policy Program Arguments list, choose the SQL Statement Outputs property.
3. In the Default Value property, add the substitution.

   The Default Value property contains a comma separated list of substitution variables. These variables hold the result of the query statement. Siebel CRM displays the list that is defined in the Default Value property in the Available Substitution field in the Send Message Argument form.

## Using a Workflow Policy Program to Create Activities for a Marketing Campaign

The Create Campaign Contact Activity Workflow Policy program creates an activity for the contacts or prospects who are the targets of the campaign. To define the data that Siebel CRM uses to update the Contact Activity record, you use the Workflow Policy Program Arguments list. The following table describes the arguments in the Workflow Policy Program Arguments list.

| Argument | Description |
|---|---|
| Name | The following values are available:<br>• **Description.** Use text to describe activity.<br>• **Status.** Choose the activity status, such as planned or active, from the drop-down list.<br>• **Type.** Choose the Activity type from the drop-down list. |
| Required | Indicates if the argument is required. |
| Value | Text or drop-down list. |

# Using a Workflow Policy Program to Assign a Contact to a Marketing Campaign

The Assign to Campaign Workflow Policy program adds a contact or prospect to the list of contacts or prospects that Siebel uses for the campaign.

| Argument | Description |
|---|---|
| New Campaign | Choose a campaign to which Siebel must assign the contact or prospect. |

# Developing a Workflow Policy That Manages a Marketing Campaign

The example in this topic describes how to develop a Workflow Policy to manage a marketing campaign. In this example, a marketer must run a campaign that depends on how the campaign recipient replies:

- Siebel sends an email that informs recipients that if they order a new product over the phone, then they can receive a discount. The marketer must keep track of the recipients and give them two weeks to reply.
- At the end of the two week period, Siebel assigns recipients who do not reply to a new campaign.

The following tasks are involved in developing a Workflow Policy that manages a marketing campaign:

1. *Defining Workflow Policy Actions for the Marketing Campaign*.
2. *Defining the Workflow Policy Group for the Marketing Campaign*.
3. *Defining the Email for the Marketing Campaign Policy*.
4. *Defining the Assign Non Respondents Policy*.

## Defining Workflow Policy Actions for the Marketing Campaign

This task is a step in *Developing a Workflow Policy That Manages a Marketing Campaign*.

This example uses the following Workflow Policy actions:

- **Send Campaign Email.** Sends the offer email to the campaign recipients.
- **Create Campaign Contact Activity.** Records the activity that Siebel associates with the contact.
- **Assign to Campaign.** Assigns nonrespondents to a new campaign.

## To define Workflow Policy actions for the marketing campaign

1. Define the send campaign email action:
   a. In the Siebel client, navigate to the Administration - Business Process screen, then the Actions view.
   b. In the Actions list, add a new record with the values shown in the following table:

| Field | Value |
|---|---|
| Name | Send First Campaign Contact |
| Program | Send Campaign Email |
| Workflow Object | Campaign Contact |
| Comment | Enter appropriate text, as required. |

   c. In the Send Message Arguments form, define the argument with the values shown in the following table:

| Field | Value |
|---|---|
| Subject | Enter text and dynamic fields, as required. |
| Message Template | Enter text and dynamic fields for sending email to contacts. |

   d. In the Recipients list, add a new record with the values shown in the following table:

| Field | Value |
|---|---|
| Type | Choose a predefined Recipient. |
| Name | Choose the appropriate recipient name. |

   This action is now available for use in a Workflow Policy.

2. Define a create campaign contact activity action:
   a. Navigate to the Administration - Business Process screen, then the Actions view.
   b. In the Actions list, add a new record with the values shown in the following table.

| Field | Value |
|---|---|
| Name | First Widget Campaign |
| Program | Create Campaign Contact Activity |
| Workflow Object | Campaign Contact |
| Comment | Enter appropriate text, as required. |

   c. In the Arguments list, define the Type argument with the values shown in the following table.

| Field | Value |
|---|---|
| Argument | Type |
| Value | Define the type of contact activity for this action, such as In Store Visit, or Demonstration. |

   The Type argument is required. You can also define additional optional arguments, such as Description or Status. For each additional argument you define, create a new record in the Arguments list, then define the field and value.
3. Define an assign to campaign email action:
   a. Navigate to the Administration - Business Process screen, then the Actions view.
   b. In the Actions list, add a new record with the values shown in the following table.

| Field | Value |
|---|---|
| Name | Assign to Campaign |
| Program | Assign to Campaign |
| Workflow Object | Campaign Contact |
| Comment | Enter appropriate text, as required. |

| Field | Value |
|---|---|
| | |

c. In the Arguments list, define the Type argument with the values shown in the following table.

| Field | Value |
|---|---|
| Argument | New Campaign |
| Value | Not applicable |

## Defining the Workflow Policy Group for the Marketing Campaign

This task is a step in *Developing a Workflow Policy That Manages a Marketing Campaign*.

In this example, you define a Workflow Policy group for the marketing campaign.

### To define the Workflow Policy group for the marketing campaign

1. Navigate to the Administration - Business Process screen, then the Policy Groups view.
2. In the Policy Groups list, add a new record with the values shown in the following table.

| Field | Value |
|---|---|
| Name | Widget Promotion |
| Comments | Group of policies for Widget marketing campaign |

## Defining the Email for the Marketing Campaign Policy

This task is a step in *Developing a Workflow Policy That Manages a Marketing Campaign*.

In this topic you define the email for the marketing campaign policy.

### To define the email for the marketing campaign policy

1. Navigate to the Administration - Business Process screen, then the Policies view.
2. In the Policies List, add a new record with the values shown in the following table.

| Field | Value |
|---|---|
| Name | Email for Widget campaign |
| Workflow Object | Campaign Contact |

| Field | Value |
|---|---|
| | |
| Policy Group | Widget Promotion |
| Duration | 0 |

The Policy Group you define in this step is the group you created in *Defining the Workflow Policy Group for the Marketing Campaign*.

3. In the Conditions list, create a new name with the values shown in the following table.

| Field | Value |
|---|---|
| Condition Field | Name |
| Operation | = (equals sign) |
| Value | 1st Widget Promotion |

4. In the Conditions list, define the Start Date with the values shown in the following table.

| Field | Value |
|---|---|
| Condition Field | Start Date |
| Operation | = (equals sign) |
| Value | (Enter the date where the campaign starts sending messages to the target audience.) |

5. In the Conditions list, to define the Campaign Status, create a new record with the values shown in the following table.

| Field | Value |
|---|---|
| Condition Field | Campaign Status |
| Operation | = (equals sign) |
| Value | Launched |

| Field | Value |
|---|---|
| | |

The Launched value starts the campaign.

## Defining the Assign Non Respondents Policy

This task is a step in *Developing a Workflow Policy That Manages a Marketing Campaign*.

The policy that you define in this topic reassigns nonrespondents to a new campaign.

### To define the assign non respondents policy

1. Navigate to the Administration - Business Process screen, then the Policies view.
2. In the Policies List, add a new record with the values shown in the following table.

| Field | Value |
|---|---|
| Name | Non-Respondents of Widget campaign |
| Workflow Object | Campaign Contact |
| Policy Group | Widget Promotion |
| Duration | 14 |

3. In the Conditions list, define the Name with the values shown in the following table.

| Field | Value |
|---|---|
| Condition Field | Name |
| Operation | = (equals sign) |
| Value | 1st Widget Promotion |

4. In the Conditions list, define the Campaign Status with the values shown in the following table.

| Field | Value |
|---|---|
| Condition Field | Campaign Status |

| Field | Value |
|---|---|
| Operation | = (equals sign) |
| Value | Launched |

5. In the Conditions list, define the Done Flag with the values shown in the following table.

| Field | Value |
|---|---|
| Condition Field | Done Flag |
| Operation | = (equals sign) |
| Value | N |

### Defining Logic That Calls the Assign Nonrespondents Policy

Setting the Done Flag to N identifies the activity for this recipient as requiring additional attention. If the Policy duration is 14 days, and if the Done Flag is N, then the policy runs in 14 days. Siebel assigns recipients who do not reply to the first campaign to a new campaign after 14 days.

# Configuring a Predefined Workflow Policy

This topic describes how to configure a predefined Workflow Policy and includes the following subtopics:

- *Viewing Groups of Predefined Workflow Policies*
- *Configuring a Predefined Workflow Policy for Messaging*
- *Identifying Objects That the Workflow Policy Must Modify*

## Viewing Groups of Predefined Workflow Policies

You can view groups of predefined Workflow Policies. In this example, you view groups of predefined Workflow Policies for messaging.

### To view groups of predefined Workflow Policies

1. In the Siebel client, navigate to the Administration - Business Process screen, then the Policy Groups view.
2. In the Name column, select Siebel Messaging.

   Siebel displays the messaging policies in the Policies list.

# Configuring a Predefined Workflow Policy for Messaging

You can use a predefined Workflow Policy to display a message in a dialog box in the Siebel client.

## To configure a predefined Workflow Policy for messaging

1. In the Siebel client, navigate to the Administration - Business Process screen, then the Policy Groups view.
2. Select Siebel Messaging in the Name column.
3. Locate the Workflow Policy you want to use, such as Messaging Policy Send Screen Pop for Activity.
4. Set the Expiration field to NULL.
5. Run the Generate Triggers server component with the values shown in the following table.

| Parameter | Value |
|---|---|
| EXEC | True |
| Remove | True |

   For more information, see *Configuring Database Triggers*.
6. Run Generate Triggers again. This time set the Remove parameter to False.
7. Navigate to the Messages screen, then the All Messages view.
8. Click New and set the following field values:
   a. In the Last Name field, enter the name of the contact who receives the message.
   b. In the Assigned To field, enter the name of the employee to whom this user is assigned.
   c. In the Message field, type the text message that Siebel must display in the dialog box.
   d. Set the Alert Type field to Screen Alert.
9. Click the Message Alert Setup tab, and then make sure a corresponding entry exists for the message recipient and that Alert Type is set to Screen Alert.
10. If the message recipient is already defined, then do the following:
    a. Click New.
    b. Set the Last Name field to the same name you set in step 8a.
    c. Set the Alert Type field to Screen Alert.
11. Start a Workflow Monitor Agent server task for the Workflow Policy group you activated in step 5.

    For more information, see *Running a Workflow Policy with Workflow Monitor Agent*.
12. Verify the configuration:
    a. Define a test user when assigning the contact in step 8a.
    b. In a second client session, log in to the Siebel client as the test user.
    c. In the first client session, insert a new activity for this user.
    d. In the second client session, verify that Siebel CRM displays the dialog box and that it contains the message you previously entered in step 8c.

# Identifying Objects That the Workflow Policy Must Modify

You might find that the predefined Workflow Policy objects do not reference the data that your business requires. This topic describes how to identify the objects that can provide this data. It uses account objects as an example.

## To identify objects that the Workflow Policy must modify

1. In the Siebel client, navigate to the view that displays the business data that Siebel must monitor.

   For example, if Siebel must monitor account activity, then navigate to the Accounts screen, Accounts List, and then the Activities view.
2. Select the Help menu, then the About View menu item.

   The About View dialog box identifies the business object, business components, and applets that this view uses. For example, if Siebel currently displays the Account Activities view, then this dialog box identifies Action as the business component that the Activities list uses.
3. In the Object Explorer, click Business Component.
4. In the Business Components list, query the Name property for Action.
5. Note the value in the Table property.

   In this example, the table name is S_EVT_ACT.
6. In the Object Explorer, click Business Object.
7. In the Business Objects list, query the Name property for Account.
8. In the Object Explorer, expand the Business Object, and then click Business Object Component.
9. In the Business Object Components list, query the BusComp property for Action, and then click the link in the Link Property.

   The Source Field property is empty and the Destination Field property contains Account Id. The value in the Link property identifies the link that defines the relationship between the Account business component and the Action business component. This Link defines the relationship between the parent business component and the child business component through the Source field and the Destination field.

   A Source Field property that is empty indicates that the join uses the ROW_ID column of the table that the Table property of the parent business component references. The Destination Field property identifies the field in the child business component that is a foreign key to the parent business component.
10. Click the link in the Child Business Component property.

    Siebel displays the Action business component in the Business Components list.
11. In the Object Explorer, expand the Business Component object, and then click Field.
12. In the Fields list, choose the Account Id field, and then note the value in the Column property.

    The Column property contains TARGET_OU_ID, which is the column in the table that the Account Id field references. You can use this information when you define the Workflow Policy component.
13. When you define a Workflow Policy component, you can use the table name (step 5) and column name (step 12) you previously noted.

# 15 Defining Custom Workflow Policies

## Defining Custom Workflow Policies

This chapter describes how to define a custom Workflow Policy. It includes the following topics:

- *Process of Planning a Workflow Policy*
- *Process of Defining a Workflow Policy*
- *Examples of Configuring Workflow Policies*
- *Customizing Workflow Policy Objects*
- *Defining Conditions for a Workflow Policy*

## Process of Planning a Workflow Policy

This topic describes how to plan a Workflow Policy. The following tasks are involved in planning a Workflow Policy:

1. *Creating a Plan for the Workflow Policy Group*
2. *Creating a Plan for the Workflow Policy*
3. *Identifying Objects That the Workflow Policy Monitors*
4. *Determining Settings for the Workflow Policy and Policy Conditions*
5. *Creating a Plan for the Workflow Policy Action*
6. *Examining Predefined Workflow Policies*
7. *Creating a Plan for the Test and Migration Strategy*

For more information, see *Examples of Planning a Workflow Policy*.

Before you plan a Workflow Policy, you must identify the solution that most closely meets your business requirements. For more information, see *Identifying an Automation Solution*.

### Creating a Plan for the Workflow Policy Group

This task is a step in *Process of Planning a Workflow Policy*.

A *Workflow Policy group* is a collection of Workflow Policies that allows Siebel to identify policies that include similar system requirements. It helps Siebel to load balance work on the Siebel Servers and to achieve scalability. A Workflow Policy group allows you to run similar Workflow Policies as a group in one Workflow Agent process, which helps to manage and optimize performance of the Workflow Agent.

You must define a Workflow Policy group before you define a Workflow Policy. Siebel CRM assigns each Workflow Policy Agent to one Workflow Policy group. If you run only one Workflow Policy Monitor Agent and one Workflow Policy Action Agent, then you must assign your policies to one policy group.

It is recommended that you use multiple Workflow Policy Agents for the following reasons:

- Shorten the lag time between when Siebel calls the Workflow Policy event and when Workflow Policies detects the event

- Spread the workload across multiple Siebel Servers
- Adjust the polling interval so that polling for a noncritical Workflow Policy does not prevent the timely processing of a critical Workflow Policy.

Defining Workflow Policy groups and using multiple Workflow Policy Agents is part of tuning your system to realize higher performance. You can do this work while you monitor system performance.

The MaxInt parameter and the S_ESCL_RULE table determines the maximum number of Workflow Policies that a single Workflow Policy group can contain.

### Grouping Workflow Policies That Include Similar Durations

It is recommended that you group together the Workflow Policies that include similar durations. This configuration allows you to assign the Workflow Policy group to a Workflow Policy Agent that includes a polling rate that matches the duration of the Workflow Policies in the group. This configuration results in a more efficient use of resources.

## Creating a Plan for the Workflow Policy

This task is a step in *Process of Planning a Workflow Policy*.

Many of the Workflow Policy objects and Workflow Policy programs that your Workflow Policies require come predefined. You can modify Workflow Policy programs, define more Workflow Policy objects, or to make more Workflow Policy columns available for monitoring. The planning phase is an appropriate time to review business process activities for your company. You must identify the work that a Workflow Policy can automate. For more information, see the following topics:

- *Process of Defining a Workflow Policy*.
- *Customizing Workflow Policy Objects*.

### Developing Workflow Policies in Small Groups

It is recommended that you define and implement a small group of Workflow Policies at a time. After you successfully implement the group, you can proceed to another small group of policies in a systematic manner. If the business environment requires you to change your plan, then you can assign a Workflow Policy to a different group. For more information, see *Moving a Workflow Policy to a Different Group*.

## Identifying Objects That the Workflow Policy Monitors

This task is a step in *Process of Planning a Workflow Policy*.

You can create a plan that identifies the objects to monitor. For example, assume the service department must send an email to the contact for the service request if a user creates a service request that includes a severity level of critical. The following table describes the information to monitor in this example.

| What to Monitor | Purpose of the Policy |
|---|---|
| Service request status<br><br>Service request severity | Send an email to the contact of a service request if a user creates a service request, and if the severity level for this service request is critical. |

# Determining Settings for the Workflow Policy and Policy Conditions

This task is a step in *Process of Planning a Workflow Policy*.

To determine the settings for the Workflow Policy and Workflow Policy conditions, you can identify the properties of the Workflow Policy and policy conditions to modify, identify the Workflow Policy object to monitor in the Siebel database, and determine the duration for the monitoring interval.

The following able describes an example plan that details the type of information to monitor for a Workflow Policy.

| Name | Workflow Policy Object | Workflow Policy Group | Duration |
|---|---|---|---|
| Email Confirmation of SR | Service Request | Medium Frequency<br><br>Allows you to put Workflow Policies in Groups. | Duration indicates the time that must elapse before Siebel performs an action. Each Workflow Policy includes one duration. If an action must occur after one hour, two hours, and six hours, then you must define a different policy for each duration. |

The following table describes an example plan for the Workflow Policy conditions.

| Field in the Workflow Policy Condition | Operation | Value |
|---|---|---|
| Service Request Severity | = | 1-Critical |
| Service Request Status | = | Open |

A Workflow Policy condition includes an expression. The Condition Field identifies the table column that Siebel monitors in the Siebel database. Siebel reads the value for a Workflow Policy condition from a drop-down list that it defines at the table column level. The value that the operation requires for the condition that you define must be available in this drop-down list. IN and NOT IN are example values. If these values are not available, then the Workflow Policy cannot call the Workflow Policy action.

# Creating a Plan for the Workflow Policy Action

This task is a step in *Process of Planning a Workflow Policy*.

If the Workflow Policy conditions are met, then Siebel CRM runs a Workflow Policy action. There are a number of possible Workflow Policy actions provided out-of-the-box. You can use these actions or define your own custom actions.

The following table includes an example plan that describes the type of information you must include for a Workflow Policy action.

| Action Name | Workflow Policy Program | Workflow Policy Object | Arguments |
|---|---|---|---|
| Send SR Email to Contact | Send SR Email | Service Request | Send to Contact |

# Examining Predefined Workflow Policies

This task is a step in *Process of Planning a Workflow Policy*.

Siebel comes with predefined Workflow Policies and policy programs. It is recommended that you examine these predefined policies to determine if a policy already exists that meets your business requirements. If necessary, you can modify the predefined objects rather than defining new objects. For more information, see *Types of Predefined Workflow Policy Programs* and the topic about the Workflow Policy Object type in *Siebel Object Types Reference* .

# Creating a Plan for the Test and Migration Strategy

This task is a step in *Process of Planning a Workflow Policy*.

You must make sure that any new Workflow Policy that you create works properly in a test environment. The test environment can include a sample Siebel CRM database and test data. Testing a new Workflow Policy, Workflow Policy condition, and Workflow Policy action makes sure that the policy you release to the production environment runs correctly and does not cause a conflict with another Workflow Policy.

## Guidelines for Creating a Plan for the Test and Migration Strategy

If you create a plan for the test and migration strategy, then it is recommended that you adhere to the following guidelines:

- Make sure your test environment and production environment contain identical software versions.
- To make sure the test environment includes realistic data, you can use a partial or full copy of the production database.

# Examples of Planning a Workflow Policy

The following examples will help you plan a Workflow Policy.

- *Planning a Workflow Policy That Sends a Discount Notice*
- *Planning a Workflow Policy That Notifies the User That Service Requests Require Attention*

## Planning a Workflow Policy That Sends a Discount Notice

In this example, if a sales representative quotes a discount that is greater than 30%, then Siebel CRM must notify the sales manager.

### To plan a Workflow Policy that sends a discount notice

1. Identify the items to monitor, as described in the following table.

| What to Monitor | Purpose of the Policy |
|---|---|
| A quote that contains a discount that is greater than 30% requires Sales Manager approval. | Notify Sales Manager to review and approve the quote. |

2. Plan the Workflow Policy action, as described in the following table.

| Name | Workflow Policy Object | Workflow Policy Group | Duration | Description |
|---|---|---|---|---|
| Notify Sales Manager upon Sales Approval | Quote | Low Frequency | 0 | Notify the manager if a user creates a quote that includes a discount that is greater than 30%. |

3. Plan the Workflow Policy conditions, as shown in the following table.

| Field (Column in the Siebel database) | Comparison | Value |
|---|---|---|
| Quote Item Discount Percent | > | 30 |

4. Plan the Workflow Policy actions that Siebel CRM runs if the Workflow Policy conditions are met.

   You can include dynamic values in the action arguments, such as the email subject and the message template. The following table describes how to configure the Send Email to Sales Manager action.

| Action Name | Workflow Policy Program | Workflow Policy Object | Arguments and Substitutions |
|---|---|---|---|
| Send Email to Sales Manager | Send Quote Email | Quote | Subject: `Please approve quote discount for [Account].`<br><br>Message Template: `Please approve the quote discount for quote [Quote Number] and notify [Last User First Name] [Last User Last Name].`<br><br>Repeating Message: `The following quote also requires approval: [Quote Number] .` |

## Planning a Workflow Policy That Notifies the User That Service Requests Require Attention

In this example, if the number of open service requests for a service representative reaches 20, then Siebel CRM must send a notification.

### To plan a Workflow Policy that notifies the user that service requests require attention

1. Identify the items to monitor, as described in the following table.

| What to Monitor? | Purpose of the Policy |
|---|---|
| Monitor open service requests when they reach a quantity of 20. | Use Send Broadcast Message to alert the service representative about the situation. |

2. Plan the Workflow Policy action, as described in the following table.

| Name | Workflow Policy Object | Workflow Policy Group | Quantity |
|---|---|---|---|
| Over 20 Open Service Requests | Service Request | High Frequency | 20 |

3. Plan the Workflow Policy conditions, as described in the following table.

| Field (Column in the Siebel database) | Comparison | Value |
|---|---|---|
| Service Request Status | = | Open |

4. Plan the Workflow Policy actions that Siebel CRM runs if the Workflow Policy conditions are met.

   You can define the action arguments, as shown in the following table.

| Action Name | Workflow Policy Program | Workflow Policy Object | Arguments and Substitutions |
|---|---|---|---|
| Alert Representative of Open SR | Send SR Message Broadcast | Service Request | Abstract: `You own over 20 service requests.`<br><br>Message Template: `You own over 20 service requests. Please review the queue for your service requests.` |

# Process of Defining a Workflow Policy

The following tasks are involved in defining a Workflow Policy:

1. *Defining a Workflow Policy Group*
2. *Defining a Workflow Policy Action*
3. *Defining a Workflow Policy*

For more information, see the following topics:

- *About Workflow Policies*
- *Examples of Configuring Workflow Policies*
- *Customizing Workflow Policy Objects*
- *Types of Predefined Workflow Policy Programs*

# Defining a Workflow Policy Group

This task is a step in *Process of Defining a Workflow Policy*.

To define a Workflow Policy, you begin by defining a Workflow Policy group. For more information, see *Creating a Plan for the Workflow Policy Group*.

## To define a Workflow Policy group

1. In the Siebel client, navigate to the Administration - Business Process screen, then the Policy Groups view.
2. In the Policy Groups list, create a new record and set the following values:
   - In the Name field, specify the name of the Workflow Policy group. A Workflow Policy can belong to only one policy group. The name can contain no more than 30 characters.
   - (Optional) Enter comments in the Comments field to describe the purpose or use of the policy group.
3. In the Policies list, create a new record with the values shown in the shown in the following table.

| Field | Usage |
|---|---|
| Name | Define the name of the Workflow Policy that is included in the Workflow Policy group. |
| Object | Define the Workflow Policy object for which the Workflow Policy is defined. For example, Service Request, Opportunity, or Quote. |
| Activation | Define the date and time when Siebel CRM must activate the Workflow Policy. If this field is null, then Siebel CRM activates the Workflow Policy immediately. For more information, see *Grouping Workflow Policies That Include Similar Durations*. |
| Expiration | Define the date and time that the Workflow Policy expires. If this field is null, then the Workflow Policy is active indefinitely. |

| Field | Usage |
|---|---|
| | |
| Comments | Define comments that describe the purpose or use of the Workflow Policy. |

4. Repeat step 5 for each Workflow Policy you want to include in the group.

   If, at some point in the future, you must reassign the Workflow Policy to another group, then see *Moving a Workflow Policy to a Different Group*.

# Defining a Workflow Policy Action

This task is a step in *Process of Defining a Workflow Policy*.

You must define the Workflow Policy action before you define the Workflow Policy that uses the action.

## To define a Workflow Policy action

1. In the Siebel client, navigate to the Administration - Business Process screen, then the Actions view.
2. In the Actions list, create a new record with the values shown in the following table.

| Field | Description | Value |
|---|---|---|
| Name | Define the name of the Workflow Policy action. Siebel CRM displays this name in the Actions Applet of the Workflow Policies view. | Enter a descriptive name that is consistent with your overall naming strategy and meaningful to the policy maker. |
| Program | Define the Workflow Policy program that is associated with the action. | Choose the program from a drop-down list. For more information, see *Types of Predefined Workflow Policy Programs*. |
| Workflow Object | Define the Workflow Policy object with which this action is associated. If you define a Workflow Policy object, then Siebel CRM displays this action only in the Actions Applet of the Workflow Policies view for Workflow Policies that are associated with this Workflow Policy object. | Choose the object from a drop-down list of Workflow Policy objects. |
| Comments | Define comments that describe the purpose or use of this Workflow Policy action. | Enter comments text. |

3. In the Arguments list, define each argument as required.

   The Arguments list changes depending on the Program you choose in the Actions list. For more information, see *Using the Arguments List*.

4. If you selected a program in the Actions list in step 2, then define a recipient in the Recipients list with the values shown in the following table. You only use the Recipients list if you choose certain programs in the Actions list. For more information, see *Using the Recipients List*.

| Field | Description |
|---|---|
| Type | The following choices are available:<br>○ **Send to Employee**. Allows you to pick an employee.<br>○ **Send to Position.** Allows you to pick a position, thereby sending to the primary employee of this position without having to know the name of the person. The employee must be ACTIVE.<br>○ **Send to Contact.** Allows you to pick a contact.<br>○ **Send to Relative.** For more information, see *Using the Send to Relative Recipient Type*.<br>○ **Send to Address**. Allows you to manually enter an email address. This option supports a program that sends an email. |
| Name | Defines the Name of the recipient according to the value you select for the Type field. Siebel CRM sends the message according to the position. If you choose an employee as the recipient, and if the same position is assigned to multiple employees, then Siebel CRM sends the message to every employee who is assigned to this position, even though you only choose one employee as the recipient. |

5. (Optional) Repeat step 2 to add additional recipients if required.

# Restrictions with Defining a Workflow Policy Action

This topic highlights the restrictions when defining a Workflow Policy action.

## Calling a DLL or External Function

You can use a Workflow Process to call a DLL or external function, however, note the following:

- You cannot call a DLL or external function through a Workflow Policy action.
- You cannot call a business service from a Workflow Policy action.

## Using an Insert Operation with a Workflow Policy Action

An insert operation in a Workflow Policy action cannot update the Primary Owner. For example, you cannot modify a Workflow Policy Program, such as Create SR Activity, to update Primary Activity Owner (OWNER_PER_ID) because Siebel CRM must also update the S_ACT_EMP intersection table. One Workflow Policy Program cannot update two tables in one database operation. To update OWNER_PER_ID, you must use a Workflow Process.

# Using the Arguments List

The Arguments list is context sensitive. Siebel CRM displays a different applet depending on the Program you choose in the Actions list. For example:

- If you choose Send Message Broadcast in the Program field of the Actions list, then it displays the Send Message Broadcast Arguments list.
- If you choose Send Email in the Program field of the Actions list, then it displays the Send Message Arguments form.

If you use the Arguments form, then note the following:

- A program argument is case-sensitive. You must use the correct case. It is recommended that you use the drop-down lists in the Arguments form when possible, instead of entering the arguments manually.
- Before you configure Siebel CRM to use email or paging, you must complete the setup procedures described in *Administering Workflow Policies*.
- If the Workflow Policy program is Send Email, Send Text or Send Broadcast Message, then you must use the Recipient List to enter the recipients of the action.

For a description of each Workflow Policy program type, Workflow Policy program arguments and valid values, see *Types of Predefined Workflow Policy Programs*.

## How Siebel CRM Refreshes the Applets You Use

The applets that Siebel CRM displays in the Workflow Policies view change automatically depending on the program type you choose for the Workflow Policy action. This behavior is different from most views that you use in the Siebel client. For example, if you navigate to the Administration - Business Process screen and then the Policies view, Siebel CRM displays the following applets:

- Policies List
- Conditions
- Actions
- Arguments

If you choose Send Page to Opportunity Sales Rep in the Action field of the Actions list, then it replaces the Arguments list with the Send Page Arguments form.

# Using the Recipients List

You use the Recipients list only if you choose Send Email, Send Page, or Send Message Broadcast in the Program field of the Actions list.

## Using the Send to Relative Recipient Type

The Send to Relative recipient type sends an email or page to an individual or to the user who is assigned to the position that is associated with the current record, such as Primary Sales Rep or Primary Sales Rep Manager. The choices that are available are context sensitive. They depend on the Workflow Object you choose in the Actions list.

If you are defining a custom Send to recipient, then use one of the following recipient types in the Type field of the Recipients list:

- Send to Employee
- Send to Position
- Send to Contact
- Send to Relative

Email Manager does not send an email to the same recipient twice for the same action. If it detects that it already sent an email to a specific email address, then it does not send another message. If the Send to Relative type returns more than one recipient, then it sends an email to each recipient only if each email address is unique.

## Sending an Email to Multiple Relative Recipients

You can define a Workflow Policy that sends the same email to multiple recipients.

### To send an email to multiple relative recipients

1. Define an action for the policy program.
2. In the Recipients list, add a new record, set the Type field to Send to Relative, then select the name of the first recipient in the Name field.
3. Repeat step 2 to add each additional recipient, updating the Name field for each recipient.
4. Associate the action with the Workflow Policy.

# Defining a Workflow Policy

This task is a step in *Process of Defining a Workflow Policy*.

You use the Workflow Policies view to define a Workflow Policy. For more information, see *Examples of Configuring a Workflow Policy*.

## To define a Workflow Policy

1. In the Siebel client, navigate to the Administration - Business Process screen, then the Policies view.
2. In the Policies List, create a new record with the values shown in the following table.

| Field | Description | Explanation |
|---|---|---|
| Name | Define the Workflow Policy name. | N/A |
| Workflow Object | Define the Workflow Policy object for which the Workflow Policy is defined. | For example, choose Service Request, Activity, or Accounts. This field is required. |
| Policy Group | Define the Workflow Policy group to which the Workflow Policy belongs. | You must assign each Workflow Policy to a Workflow Policy group. For more information, see *Moving a Workflow Policy to a Different Group*. |

| Field | Description | Explanation |
|---|---|---|
| Activation | Define the date and time when Siebel CRM must activate the Workflow Policy. | If this field is empty, then Siebel CRM activates the Workflow Policy immediately. |
| Expiration | Define the date and time when Siebel CRM must expire the Workflow Policy. | If this field is empty, then the Workflow Policy remains active indefinitely. |
| Duration | Define the duration fields in days, hours, or minutes that the Workflow Policy conditions must be true in order for Siebel CRM to run the Workflow Policy. | If the Workflow Policy runs in batch, then Siebel CRM ignores this field. |
| Qty | Define the number of records that must meet the Workflow Policy conditions before Siebel CRM runs the Workflow Policy action. | If you do not define a value in the Qty field, then Siebel CRM assumes a quantity of `1`. You can define a Workflow Policy condition according to the number (Qty) of records that meet the Workflow Policy condition. For example, you can define a policy that sends a notification if 20 or more (`Qty=20`) critical service requests are open. |
| Batch Mode | If the Batch Mode field contains a check mark, then this Workflow Policy evaluates records that meet the Workflow Policy conditions. | The Workflow Monitor Agent uses the Workflow Policy conditions to identify records to process.<br><br>If the Batch Mode field contains a check mark, then you must run the Workflow Monitor Agent with the Batch Mode field set to TRUE. |

3. In the Conditions list, create a new record with the values shown in the following table.

| Field | Usage | Example |
|---|---|---|
| Condition Field | Define the Workflow Policy component column in the Workflow Policy object that the Workflow Policy condition references. Values that Siebel CRM displays in the Condition Field are context sensitive, depending on the value you choose for the Workflow Object.<br><br>To allow Siebel CRM to call a Workflow Policy that contains a Condition Field that is not equal to [Value n], a value must be defined in this field that is different than the value that the Workflow Policy condition contains. If Condition Field contains no value, then Siebel CRM does not call the Workflow Policy and it uses the IS NULL operator. | Choose Service Request Area from the drop-down list. |

| Field | Usage | Example |
|---|---|---|
| Operation | Define the comparison that Siebel CRM must make between the value of the column for a Workflow Policy agent and the value you define. Choose the comparison from the drop-down list for the field. For more information, see *Examples of Configuring Workflow Policies*. | Choose equals (=) from the drop-down list. |
| Value | Define the value that Siebel CRM compares to the value that exists in the column of the Workflow Policy instance. If the Comparison field does not contain one of the following values, then the Value field is a required:<br>○ Is Null<br>○ Is Not Null<br>○ Is Updated<br>○ Is Deleted<br>○ Is Added<br>For more information, see *Examples of Configuring Workflow Policies*, and *Using Date Calculations in the Conditions List*. | Choose Printer from the drop-down list. |

4. In the Actions list, create a new record with the values shown in the following table.

| Field | Usage |
|---|---|
| Action | Pick the action that Siebel CRM runs with this Workflow Policy. |
| Sequence | Define the sequence of the action relative to other actions. |
| Consolidate | If more than one record meets the Workflow Policy conditions during the same action interval, then you can consolidate the action to one instance. To consolidate an action, make sure the consolidate field contains a check mark.<br>The consolidate field is not available with an action that sends a page. |

5. (Optional) Repeat step 4 to add more actions if required.

## Restrictions with Defining a Workflow Policy

If you define a Workflow Policy, then note the following restrictions:

- A Workflow Policy cannot reference the S_DOCK_TXN_LOG table. The unique index for this table is TXN_ID rather than the ROW_ID that other Siebel tables use. Siebel Workflow uses ROW_ID to do the comparison and run actions. If it makes comparisons to the S_DOCK_TXN_LOG table, then Siebel CRM creates an error.

- You cannot run a Business Service directly from a Workflow Policy. You can, however, call a Workflow Process that calls a Business Service.
- A Workflow Policy updates a database field directly through the data layer. It does not update a field through the business object layer. Siebel CRM can run a Workflow Process that includes a business component event that is related to the updated field.

# Examples of Configuring Workflow Policies

The examples in the following subtopics describe how to configure Workflow Policy actions and Workflow Policies:

- *Examples of Configuring a Workflow Policy Action*
- *Examples of Configuring a Workflow Policy*

## Examples of Configuring a Workflow Policy Action

The following examples describe how to configure a Workflow Policy action:

- *Configuring a Workflow Policy Action That Sends a Page*
- *Configuring a Workflow Policy Action That Sends an Email with a Repeating Message*
- *Configuring a Workflow Policy Action That Sends a Notification*
- *Configuring a Workflow Policy Action That Runs a Database Operation*
- *Configuring a Workflow Policy Action That Runs an External Program*

You can use these examples as the basis for defining your own Workflow Policy actions.

### Configuring a Workflow Policy Action That Sends a Page

The example in this topic includes a Workflow Policy action that sends a page. If the priority for a service request is very high and if the service request is not assigned to anyone, then Siebel CRM sends a page to a support manager.

#### To configure a Workflow Policy action that sends a page

1. In the Siebel client, navigate to the Administration - Business Process screen, then the Actions view.
2. In the Actions list, add a new record with the values shown in the following table.

| Field | Value |
|---|---|
| Name | Page Support Manager when SR request changes. |
| Program | Send SR Page |
| Workflow Object | Service Request |

If a Workflow Policy object is defined in the Workflow Policy Program field, then Siebel CRM updates the Workflow Object field. If necessary, you can choose a Workflow Policy object from the drop-down list.

3. In the Send Page Arguments form, define the argument with the values shown in the following table.

| Field | Value |
|---|---|
| Alpha Message Template | The [SR Status] of [SR Number] was changed. |

You use the Numeric Message Template for numeric paging and the Alphanumeric Message Template for alphanumeric paging. The pager type in the employee table determines the type of paging to use. You can copy and paste fields that are available from the Available Substitutions window.

4. In the Recipients list, add a new record with the values shown in the following table.

| Field | Value |
|---|---|
| Type | Send to Position |
| Name | Support Manager |

You can now use this action in a Workflow Policy.

5. Navigate to the Administration - Business Process screen, then the Policies view.
6. Query the Workflow Object field for Service Request.

This query returns a list of policies that you can use with the action that you defined in step 4.

7. In the Name column, choose a Workflow Policy such as Page SR Owner (Gold).
8. In the Actions list, create a new record and then locate the Page Support Manager When SR Request Changes action, which is the action you defined in step 2.
9. Examine the Send Page Arguments form.

Siebel CRM uses the information you define in this topic to update information in the Send Page Arguments form.

## Configuring a Workflow Policy Action That Sends an Email with a Repeating Message

The example in this topic uses a Workflow Policy action to send an email that includes a repeating message. Siebel CRM must notify the vice president of sales only after a specific number of deals fail to close. It uses this action with a Workflow Policy that uses the Batch feature. You enter relevant information in the Repeating Message field of the Send Message Arguments form. This configuration makes sure that the recipient receives one email that includes a consolidated list of information about each deal. If you do not configure a Repeating Message, then the email might not contain meaningful information.

### To configure a Workflow Policy action that sends an email with a repeating message

1. In the Siebel client, navigate to the Administration - Business Process screen, then the Actions view.

2. In the Actions list, create a new record with the values shown in the following table.

| Field | Value |
|---|---|
| Name | Excellent Quality Opportunity |
| Program | Send Opportunity Email |
| Workflow Object | Opportunity |
| Comment | Send an email to the VP of Sales when deals are not closing. |

3. In the Send Message Arguments form, define the argument with the values shown in the following table.

| Field | Value |
|---|---|
| Subject | Opportunities not Closing |
| Message Template | Meet with [Last User First Name] [Last User Last Name] about [Opportunity] for [Account] |
| Repeating Message | Meet with [Last User First Name] [Last User Last Name] about [Opportunity] for [Account] |

4. In the Recipients list, create a new record with the values shown in the following table.

| Field | Value |
|---|---|
| Type | Send to Position |
| Name | VP Sales |

You can now use this action in a Workflow Policy.

5. Navigate to the Administration - Business Process screen, then the Policies view.
6. Query the Workflow Object field for Opportunity.
   This query returns a list of policies that you can use with the action that you defined in step 4.
7. In the Name column, choose a Workflow Policy such as New Opportunity.
8. In the Actions list, create a new record and then locate the Excellent Quality Opportunity action, which is the action you defined in step 2.
9. Examine the Send Page Arguments form.
   Siebel CRM uses the information you define in this topic to update information in the Send Page Arguments form.

10. In the list for the Policies List, make sure the Batch Mode field contains a check mark.

## Configuring a Workflow Policy Action That Sends a Notification

The example in this topic configures a Workflow Policy action to send a notification. In this example, a service department must automate a notification policy for open service requests when there are at least 20 open requests for one service representative.

### To configure a Workflow Policy action that sends a notification

1. In the Siebel client, navigate to the Administration - Business Process screen, then the Actions view.
2. In the Actions list, add a new record with the values shown in the following table.

| Field | Value |
|---|---|
| Name | Alert Representative of Open SRs |
| Program | Send SR Message Broadcast |
| Workflow Object | Service Request |

3. In the Send Message Broadcast Arguments form, define the argument with the values shown in the following table.

| Field | Value |
|---|---|
| Abstract | You own over 20 Service Requests. |
| Message Template | You own over 20 service requests. Please review your service request queue. |

4. In the Recipients list, add a new record with the values shown in the following table.

| Field | Value |
|---|---|
| Type | Send to Relative |
| Name | SR Owner |

You can now use this action in a Workflow Policy. For information about how to add an action to a Workflow Policy, see *Configuring a Workflow Policy Action That Sends a Page*.

## Configuring a Workflow Policy Action That Runs a Database Operation

The example in this topic uses a Workflow Policy action to run a database operation. If the user changes the Severity of a service request to Critical, then Siebel CRM changes the value of the Priority field to Very High. A Workflow Policy can perform the following types of database operations:

- **Insert.** Inserts a table in the Siebel database.
- **Update.** Updates one or more columns in an existing record in the Siebel database.

### To configure a Workflow Policy action that runs a database operation

1. In the Siebel client, navigate to the Administration - Business Process screen, then the Actions view.
2. In the Actions list, add a new record with the values shown in the following table.

| Field | Value |
|---|---|
| Name | Update SR Priority to Very High |
| Program | Change SR Priority |
| Workflow Object | Service Request |

3. In the Arguments form, define the argument with the values shown in the following table.

| Field | Value |
|---|---|
| Name | New Priority |
| Value | 1-ASAP |

You can now use this action in a Workflow Policy. For information about how to add an action to a Workflow Policy, see *Configuring a Workflow Policy Action That Sends a Page*.

## Configuring a Workflow Policy Action That Runs an External Program

The example in this topic uses a Workflow Policy action to run an external program. To define an action that runs an external program, you can use Run External Program. For example, you can write a custom executable that calculates the quality of a new lead. If the parameters to calculate the lead changes, then a Workflow Process can call this executable. You can use a program named leadcalc.exe in the following directory and an Action to call and run Run External Program:

```
C:\bin
```

### To configure a Workflow Policy action that runs an external program

1. In the Siebel client, navigate to the Administration - Business Process screen, then the Actions view.
2. In the Actions list, add a new record with the values shown in the following table.

| Field | Value |
|---|---|
| Name | Run Lead Calculation Program |
| Program | Run External Program |
| Workflow Object | Define the Workflow Policy object involved in this action, such as Account. |

If a Workflow Policy object is already defined for the chosen Workflow Policy program, then Siebel CRM updates the Workflow Object field. If necessary, you can choose a Workflow Policy object. You cannot modify the Workflow Object field after you step off the record.

3. In the External Program Arguments form, define the fields as shown in the following table.

| Field | Value |
|---|---|
| Executable Name | leadcalc.exe |
| Command Line | Enter the command line parameters that Siebel CRM must pass to the executable. |
| Execute Type | Choose an execute type. |
| Available Substitutions | Choose the appropriate dynamic fields. |

4. In the Recipients list, create a new record with the values shown in the following table.

| Field | Value |
|---|---|
| Type | Send to Position |
| Name | Support manager |

You can now use this action in a Workflow Policy. For information about how to add an action to a Workflow Policy, see *Configuring a Workflow Policy Action That Sends a Page*.

## Configuring an External Program to Run on UNIX

This topic describes how to configure an external program to run on UNIX. The predefined Run External Program Workflow Policy program does not support UNIX.

### To configure an external program to run on UNIX

1. Define a business service that runs an external program:
   a. Navigate to the Business Service Administration screen, then the Business Service Methods view.
   b. Add a new business service.

      For example, Run Program.
   c. Add a new method.

      For example, Run.
   d. Add a new method argument.

      For example, Program.
   e. Select the Proc: Service_PreInvokeMethod.
   f. Call Clib.system in the function.

      For example:

```
var program = Inputs.GetProperty("...")
if (program != "")
 Clib.system(program);
return(CancelOperation);
```

   > **Note:** For more information, review the use of the Clib.system method in the *Siebel eScript Language Reference* .

2. Define a Workflow Process that calls the business service that you defined in *Configuring an External Program to Run on UNIX*:
   a. Add a start step, a business service step, and an end step.
   b. Connect the steps.
   c. For the business service step, define Run Program and Run.
   d. For the input argument for Program, define the external program you must run.

      For example, you can define letter.txt that exists in the following directory:

```
/bin/mail hkim@pcs.com /home/users/hkim/letter.txt
```

3. Invoke your Workflow Process as a Workflow Action.

## Examples of Configuring a Workflow Policy

The following examples describe how to configure a Workflow Policy:

- *Configuring a Workflow Policy That Sends Paging Notification of Service Requests That Are Not Assigned*
- *Configuring a Workflow Policy That Sends an Email Notification of Open Opportunities*

## Configuring a Workflow Policy That Sends Paging Notification of Service Requests That Are Not Assigned

The example in this topic uses a Workflow Policy to send a page. If the priority for a service request is Very High, and if no one is assigned to this service request, then Siebel CRM sends a page to the support manager. Assume the Send Page action is already defined.

### To configure a Workflow Policy that sends paging notification of service requests that are not assigned

1. Navigate to the Administration - Business Process screen, then the Policies view.
2. In the Policies List, create a new record with the values shown in the following table.

| Field | Value |
|---|---|
| Name | Page Support Manager |
| Workflow Object | Service Request |
| Policy Group | High Frequency |
| Duration | 2 hours |

3. In the Conditions list, add two new records with the values shown in the following table.

| Condition Field | Operation | Value |
|---|---|---|
| Service Request Priority | = (equals) | High |
| Service Request Owner | `IS NULL` | (Leave this field empty.) |

   This step defines the action that Siebel CRM runs if the conditions are met.
4. In the Actions list, add an Action.
5. Select the appropriate notification action in the Action field.

   For example, select Page for Critical SR Gold Support.

## Configuring a Workflow Policy That Sends an Email Notification of Open Opportunities

The example in this topic uses a Workflow Policy to send an email notification. If the number of deals that are not closed reaches a designated level, then Siebel CRM must notify the vice president of sales. Assume you already defined a Workflow Policy action that batches information on the deals and sends an email that includes information about the number of designated deals.

## To configure a Workflow Policy that sends an email notification of open opportunities

1. Navigate to the Administration - Business Process screen, then the Policies view.
2. In the Policies List, create a new record with the values shown in the following table.

| Field | Value |
|---|---|
| Name | Very High Value Opportunity |
| Workflow Object | Opportunity |
| Policy Group | Medium Frequency |
| Quantity | 5 |
| Batch | Checked |

   It is not necessary for you to configure the Quantity field to display a repeating message.
3. In the Conditions list, add two new records with the values shown in the following table.

| Condition Field | Operation | Value |
|---|---|---|
| Forecast Probability | <= (less than or equal to) | 50 |
| Forecast Revenue | > (greater than) | 400000 |

4. In the Actions list, add a new action record with the values shown in the following table.

| Action | Consolidate |
|---|---|
| Excellent Quality Opportunity Email | Checked |

5. Go to and examine the Send Message Arguments form.

   Siebel CRM updates the values in this form with information from the action that you defined previously.
6. In the Actions applet, click the Excellent Quality Opportunity E-mail link in the Action applet.
7. Examine values in the Actions applet and Send Message Arguments.

   You can modify these values to meet the specific requirements of your implementation.

### Aligning the Timing of the Workflow Policy Monitor with Email Creation

This topic describes how to make sure the email that Siebel CRM includes in a list of opportunities meets the Workflow Policy conditions. If the Workflow Policy Action agent runs too fast, then Siebel CRM might insert an individual email each time the Workflow Policy conditions are met.

To align the timing of the Workflow Policy monitor with email creation

1. Restart the Workflow Policy Monitor agent.
2. Restart the Workflow Policy Action agent.

   Set the Sleep parameter on the Workflow Policy Action to at least 5 minutes.

   For more information about starting a server process, see *Administering Workflow Policies*.

# Customizing Workflow Policy Objects

This topic describes how to make changes in the repository to modify predefined Workflow Policy objects or to configure custom Workflow Policy objects. It includes the following subtopics:

- *Displaying Workflow Policy Object Types*
- *Configuring a Custom Workflow Policy Column*
- *Configuring a Custom Workflow Policy Object*
- *Configuring a Custom Workflow Policy Component*
- *Configuring a Custom Workflow Policy Component Column*
- *Configuring a Custom Workflow Policy Program*
- *Configuring a Workflow Policy Program Argument*

For more information about:

- Using the Siebel client to configure a Workflow Policy that references compiled objects, see *Process of Defining a Workflow Policy*
- Using predefined policies instead of creating new objects, see *Types of Predefined Workflow Policy Programs*
- Reference information, see the topic about the Workflow Policy object type in *Siebel Object Types Reference* .

## Displaying Workflow Policy Object Types

Predefined or custom Workflow Policies and programs can be modified in the Siebel Repository. This topic describes how Workflow Policy objects appear in the Siebel client. For more information, see *Hierarchy of Workflow Policy Objects*.

### Objects that Siebel CRM Displays in the Workflow Policy View

The following table shows the Workflow Policy objects and describes how they appear in the Workflow Policy view in the Siebel client.

| Object | In the Siebel Client, the Workflow Policy Object Appears in the: |
|---|---|
| Workflow Policy Object | Workflow Object drop-down list in the Policies List. |

| Object | In the Siebel Client, the Workflow Policy Object Appears in the: |
|---|---|
| | |
| Workflow Policy Component Column | Condition Field drop-down list in the Conditions list.<br><br>Values that Siebel CRM displays in the drop-down list of the Condition field are context sensitive, depending on the value that the Workflow Object field in the Policies List contains. |

## Objects That Siebel CRM Displays in the Actions View

The following table shows the Workflow Policy objects and describes how they appear in the Actions view of the Administration - Business Process screen in the Siebel client.

| Object | In the Siebel Client, the Workflow Policy Object Appears in the: |
|---|---|
| Workflow Policy Program | Program field drop-down list in the Actions list. |
| Workflow Policy Program Argument | Arguments field drop-down list in the Arguments list.<br><br>Values that Siebel CRM displays in the drop-down list of the Arguments field are context sensitive, depending on the value that the Program field in the Policies List contains. |

# Configuring a Custom Workflow Policy Column

This topic describes how to configure a custom Workflow Policy column. Before you can add a Workflow Policy column to a Workflow Policy component, you must configure the Workflow Policy column in the Workflow Policy Columns list.

**CAUTION:** If you delete a Workflow Policy column, then you must remove all references to this column that exist in other Workflow Policy objects. Even if an active Workflow Policy does not currently reference a Workflow Policy column, the Workflow Monitor Agent might reference this column in the repository, which might allow Siebel CRM to activate a new Workflow Policy in the future. To avoid a potential conflict, it is recommended that you remove all old references to a column that you delete.

## To configure a custom Workflow Policy column

1. Identify the business object, business component, and applet that use the new Workflow Policy column:
   a. In the Siebel client, navigate to the view that will use the new policy column.

      For example, from the Accounts List, drill down on an account record and then choose the Activities view tab.
   b. Select About View from the Help menu.
   c. Note the business object, business components, and applets that this view uses.
   d. In the Object Explorer in the Business Components list, select one of the business components that you identified in step c and note the value in the Table property.

      The Table property displays the table name for the Siebel database table that this business component references.
2. Configure the new Workflow Policy column:
   a. In the Object Explorer, click Workflow Policy Column.
   b. In the Workflow Policy Columns list, create a new record with the values you identified earlier to configure properties for the new object.

      The combination of the table name and the column name must be unique. If this combination is already defined for another object, then you cannot save the object.

# Configuring a Workflow Policy Condition That References a Foreign Key

You can configure a Workflow Policy condition that references a foreign key that resides in the primary table of the Workflow Policy object. For example:

```
S_OPTY.CURR_STG_ID
```

where:

- S_OPTY is the primary table of the Opportunity workflow object
- CURR_STG_ID is a foreign key from the NAME column of the S_STG table

## To configure a Workflow Policy condition that references a foreign key

1. In the Object Explorer in the Workflow Policy Columns list, add a new Workflow Policy column with values shown in the following table.

| Property | Value |
|---|---|
| Name | S_STG.NAME |

2. Add a new Workflow Policy component to the Opportunity Workflow Policy object that references the S_STG table, with the values shown in the following table.

| Property | Value |
|---|---|
| Name | Enter a name of your choice |

| Property | Value |
|---|---|
| | |
| Source Table Name | S_STG |
| Source Column Name | ROW_ID |
| Target Component Name | Opportunity |
| Target Column Name | CURR_STG_ID |

3. Add the new column (step 1) to the new Workflow Process component that you configured in step 2.

   You can now configure a Workflow Policy condition that references the new Workflow Policy column.

# Configuring a Custom Workflow Policy Object

The following procedure describes how to configure a custom Workflow Policy object.

## To configure a custom Workflow Policy object

1. In the Object Explorer, navigate to the Workflow Policy Objects list.
2. Create a new record and configure the properties for the new record, as required.
3. In the Object Explorer, click Workflow Policy Components.
4. Create a new record, making sure that the Primary property contains a check mark.

   **CAUTION:** You must define only one record in the Workflow Policy Components list as the primary Workflow Policy component. You can add a check mark to the Primary property for only one record. For more information, see *How Siebel Indicates the Primary Workflow Policy Component*.

5. Add more Workflow Policy components, configuring relations to the primary Workflow Policy component, as required.
6. In the Object Explorer, click Workflow Policy Component Col.
7. In the Workflow Policy Component Columns list, add a Column object for each of the Workflow Policy components that you configured in step 5.

# Configuring a Custom Workflow Policy Component

The following procedure describes how to configure a custom Workflow Policy component.

## To configure a custom Workflow Policy component

1. In the Object Explorer, click Workflow Policy Object.
2. In the Workflow Policy Objects list, query the Name property for Account.

3. In the Object Explorer, expand Workflow Policy Object and then click Workflow Policy Component.
4. In the Workflow Policy Components list, add a new record:
    - Enter a value in the Name property.
    - Enter a value in the Source Table Name property.
    - Enter a value in the Source Column Name property.
5. The Source Column Name property creates a relationship between the Workflow Policy component you are currently configuring and the primary policy component.
6. Identify the set of columns that you want to monitor for the Workflow Policy component that you created in step 4.
    a. Navigate to the Workflow Policy Column list.
    b. Identify predefined Workflow Policy columns that you can use.

# Configuring a Custom Workflow Policy Component Column

This topic describes how to configure a custom Workflow Policy component column. Your might require terminology that more clearly describes your business environment than the labels that come predefined with Siebel CRM. Siebel CRM uses the Alias property to update the values that it displays in the Condition Field drop-down list of the Conditions list. It displays this list in the Workflow Policies view in the Siebel client. To configure a custom label, you can use the Alias property in the Workflow Component Columns list in the Siebel Repository.

## To configure a custom Workflow Policy component column

1. In the Object Explorer, navigate to the Workflow Policy Columns list.
2. Query the Name property for the name of the column that you want.

   If the query returns no results, then add a new Workflow Policy column that references the data table and data table column that you require.
3. Navigate to the Workflow Policy Objects list and query the Name property for the Workflow Policy object that you want.
4. Navigate to the Workflow Policy Components list and query the Name property for the Workflow Policy component that you want.
5. Navigate to the Workflow Policy Component Columns list and configure your customization.

   To configure the Workflow Column Name property, use the drop-down list to choose a column from the current set of columns that are available for this Workflow Policy Component.

   If the Workflow Policy component column that you want to monitor is not in the list, then use the Workflow Policy Columns list to configure a new object for it.
6. In the Object Explorer, click Workflow Policy Object.

   By default, the Workflow Policy Object with the name of the object you modified in step 5 is the active record in the Workflow Policy Objects list.
7. In the Siebel client, navigate to the Administration - Business Process screen, then the Policies view.
8. In the Policies List, click Query.
9. Query the Workflow Object field for the object you modified (in step 5) and then click Go.

   Siebel CRM displays the custom alias in the Condition Field drop-down list in the Conditions list.

## Modifying a Workflow Policy Component Column to Display a Custom Label

The example in this topic modifies a Workflow Policy component column. The sales department in your company must refer to the person who manages an opportunity as the Opportunity Primary Sales Associate.

### To modify a Workflow Policy component column to display a custom label

1. In the Workflow Policy Objects list, query the Name property for Opportunity.
2. In the Object Explorer, expand the Workflow Policy Object, click Workflow Policy Component, and then query the Name property of the Workflow Policy Components list for Opportunity.
3. In the Object Explorer, expand the Workflow Policy Component, and then click Workflow Policy Component Col.
4. In the Workflow Policy Component Columns list, query the Workflow Column Name property for Opportunity Primary Sales Rep Position, and then change the Alias property to Opportunity Primary Sales Associate.
5. In the Object Explorer, click Workflow Policy Object.

   By default, the Workflow Policy Object named Opportunity is the active record in the Workflow Policy Objects list.
6. In the Siebel client, navigate to the Administration - Business Process screen, then the Policies view.
7. In the Policies List, click Query, enter New Opportunity in the Name field, and then click Go.
8. In the Conditions list, add a new record:
   - Set the Condition Field to Opportunity Primary Sales Rep Position.
   - Set the Operation field to IS ADDED

   Siebel CRM displays the custom alias in the Condition Field drop-down list.

# Configuring a Custom Workflow Policy Program

This topic describes how to configure a custom *Workflow Policy program*, which is a generic action that can be invoked by a Workflow Policy.

> **CAUTION:** Make sure you thoroughly test any SQL query that you use with a custom Workflow Policy program. If this SQL query fails to find a row, then the Workflow Policies action cannot process a token.

## To configure a custom Workflow Policy program

1. In the Workflow Policy Programs list, locate a Workflow Policy program that provides some or most of the functionality that you require.
2. Copy the existing Workflow Policy program you identified in the previous step.

   This will copy the entire program, including the program arguments. For more information, see *Copying an Existing Workflow Policy Program*.

   > **CAUTION:** Do not change the name of an existing Workflow Policy program. If you rename an existing Workflow Policy program, then Siebel CRM might not be able to locate the actions that it creates for the Workflow Policy program.
3. In the new Workflow Policy program, modify the properties as necessary to meet the requirements of the custom program, such as the Workflow Object property.
4. Configure the Workflow Policy program arguments.

5. To avoid Workflow Monitor Agent errors, delete any arguments not used by your new Workflow Policy program..

## Copying an Existing Workflow Policy Program

If you configure a new Workflow Policy program, then it is recommended that you copy an existing Workflow Policy program that provides functionality that is similar to the functionality that you require. You can then modify this copy. If something goes wrong with your custom Workflow Policy program, then you can start over with the original, unmodified Workflow Policy program. If you modify a copy, then you can often reuse a significant portion of the existing configuration, which leads to fewer errors than if you configure an entirely new Workflow Policy program.

## Using a Carriage Return in a Workflow Policy Program

Do not use the carriage return character. If you configure a Workflow Policy program argument, and if a carriage return character exists in SQL Statement or SQL Statement Outputs, then this might result in unexpected behavior. Siebel CRM might not substitute the substitution value with the intended value.

## Configuring Required Values

The following table describes the values you must provide if you configure a Workflow Policy program that inserts a new record. For more information, see *Siebel Data Model Reference* on My Oracle Support.

| Required Values | Values Not Required |
|---|---|
| If you configure a Workflow Policy program that inserts a new record, then you must provide a value for each field that Siebel CRM requires to add this record.<br><br>If you configure a default value for a column, and if the Workflow Policy program does not specify a value, then Siebel CRM uses that default value on the insert. For example, the S_EVT_ACT table includes the following required columns:<br><br>• NAME<br>• ROW_STATUS<br><br>Siebel CRM sets ROW_STATUS to Y by default. | Siebel CRM does not require a value for a system column. For example, the following columns are system columns:<br><br>• CREATED<br>• CREATED BY<br>• LAST_UPD<br>• LAST_UPD_BY<br>• ROW_ID<br>• MODIFICTION_NUM<br>• CONFLICT_ID |

# Configuring a Workflow Policy Program Argument

The example in this topic uses a Workflow Policy program argument to specify to whom we are sending the email. It limits the current recipients of type relative to the Primary Sales representative. You add a relative for Primary Contact, which allows a user to configure an action that sends an email to the Primary Contact of the opportunity.

## To configure a Workflow Policy program argument

1. In the Workflow Policy Program list, query the Name property for Send Opportunity Email.
2. In the Object Explorer, expand the Workflow Policy Program object, then click Workflow Policy Program Arg.

3. In the Workflow Policy Program Arguments list, make sure the Send to Relative Workflow Policy program argument exists.
4. Create a new Workflow Policy program argument and set the Name property to Primary Contact.

   A new Workflow Policy program argument cannot contain the same name as an SQL Statement Output. If you attempt to add a new Workflow Policy program argument with the same name as an SQL Statement Output, then the Monitor Agent server task pauses and then displays the following message:

   ```
   Examining request for policy
   ```

5. Add the following SQL statement to the Default Value property.

   ```
   select O.PR_CON_ID, 'Send to Contact'
   from &Table_Owner.S_OPTY O
   where O.ROW_ID=?
   ```

   Siebel passes the ROW_ID of the violating row. An SQL query must use this same ROW_ID. In this example, the WHERE clause uses the ROW_ID of the opportunity row that meets the policy. For more information, see *Guidelines for Using an SQL Statement with a Workflow Policy Program Argument*.
6. Set the PickList property to Workflow Relative Type Picklist.

   This drop-down list identifies the argument as a relative.

## Guidelines for Configuring a Workflow Policy Program Argument

If you configure a Workflow Policy program argument, then make sure you adhere to the following:

- Make sure capitalization, punctuation and spelling are correct.
- Enter the configuration in the Name column exactly as described in the Workflow Policy Program Argument object type in *Siebel Object Types Reference* . The following items must contain one space between each word and you must capitalize each word correctly:
  - Primary ID
  - Primary Table
  - Operation Type
  - SQL Statement
  - SQL Statement Outputs

  For example, `Primary Id` must include one space between Primary and Id, a capital "P", a capital "I", and a lowercase "d".
- Do not use a carriage return. For more information, see *Using a Carriage Return in a Workflow Policy Program*.
- If you use an SQL statement in a program argument, then make sure the RDBMS you use supports this statement.
  - To enter the names of the column pairs, use the following format
  - Include one space between each word.
  - Use identical capitalization.
  - Include one space before the start of parenthesis.
  - Do not include spaces in the column.
- The order of the rows is not important.

### Guidelines for Using an SQL Statement with a Workflow Policy Program Argument

If you use an SQL statement with a Workflow Policy program argument, then it is recommended that you adhere the following guidelines:

- Make sure the table name and column name are in upper case.
- You must prefix the case-sensitive table name with the following value:

  ```
  &Table_Owner
  ```

- Make sure the RDBMS you use supports the SQL statement.
- It is recommended that the SQL statement return only one record.
- Siebel CRM can use only one Workflow Policy program argument that uses an SQL statement for a Workflow Policy program. Do not use two or more Workflow Policy program arguments that use SQL statements for a given Workflow Policy program.
- Use an SQL tool that is external to Siebel CRM to test your statement.

**CAUTION:** It is recommended that you thoroughly test your SQL statement in the context of the Workflow Policy program before you implement it in a production environment.

# Defining Conditions for a Workflow Policy

This topic describes how to define conditional logic for a Workflow Policy. It includes the following subtopics:

- *Using Standard Comparisons in the Conditions List*
- *Using Specialized Operators in the Conditions List*
- *Using Date Calculations in the Conditions List*
- *How Siebel CRM Interprets a Field That Is Not Known*

## Using Standard Comparisons in the Conditions List

The following table describes the standard operators that the Comparison field supports. An example, showing how each operator might be used for a typical database, is also included. The format requirements for a particular RDBMS might vary.

| Operator | Description | Examples |
|---|---|---|
| >= | Greater than or equal to | >=5 |
| <= | Less than or equal to | <=5 |
| = | Equals | =A |
| LIKE | LIKE | LIKE Abc% |

| Operator | Description | Examples |
|---|---|---|
| | | |
| IN | IN | `IN (1, 2, 3)` |
| NOT IN | NOT IN | `NOT IN ('A', 'B', 'C')` |
| BETWEEN | BETWEEN | `BETWEEN 1 and 2` |
| BETWEEN | BETWEEN | `BETWEEN 'A' and 'B'` |
| IS NULL | IS NULL | `B IS NULL` |
| IS NOT NULL | IS NOT NULL | `B IS NOT NULL` |

You must use the following requirements:

- If you use LIKE, IN, NOT IN, or BETWEEN with a character field, then you must enclose the value with single quotes.
- If you use IN or NOT IN, then you must enclose the value with parentheses.
- Siebel CRM implies an AND operator between multiple conditions that use these comparison values. AND means that all the conditions must be met for the statement to evaluate to true.
- LIKE and NOT LIKE allow you to use wildcards. For example, `LIKE Smith%`, or `NOT LIKE Sm%th%`.

You must use the following requirements for the database:

- Any value that you define for a LIKE, IN, NOT IN, or BETWEEN operator in the Value field of the Conditions list of the Workflow Policies view must be in a format that the RDBMS supports.
- The IN, NOT IN, and BETWEEN operators require you to use the format that the RDBMS supports. For example:
  - `IN ('a', 'b', 'c')` or `IN (1, 2, 3, 4)`
  - BETWEEN 'A' and 'B'
  - BETWEEN 1 and 10
- If your implementation uses an MS SQL Server database, and if you define a Workflow Policy condition on a LONG column, then you can use only the following operators on this column:
  - IS NULL
  - IS NOT NULL
  - LIKE
  - NOT LIKE

You must manually make sure you use the correct format. The Process Designer passes the BETWEEN clause to the RDBMS. It does not confirm format, except for date and time. For date and time fields, the Process Designer converts the date and time columns to the following format:

```
month/day/year
hour:minute:second
```

# Using Specialized Operators in the Conditions List

The Comparison field supports the IS ADDED, IS UPDATED, and IS DELETED specialized operators. Siebel CRM uses these operators for special conditions that it uses with Dynamic mode when it calls rows to look up a regular condition. If you define a Workflow Policy that runs in batch, then you must use a specialized operator in conjunction with a regular Workflow Policy condition.

Siebel CRM uses the following operators only at the Workflow Policy component level:

- **IS ADDED.** If Siebel CRM adds a new row for the Workflow Policy component, then it calls the Workflow Policy that it must examine. If you use IS ADDED in conjunction with a standard comparison, then you can use IS ADDED to update a record.
- **IS DELETED.** If Siebel CRM deletes a row from the Workflow Policy component, then it calls the Workflow Policy that it must examine.

Siebel CRM uses the following operators only at the field level:

- **IS UPDATED.** If Siebel CRM modifies the field in an existing record, or if it adds a new record that includes this field, then it calls the Workflow Policy that it must examine. Note the following:
  - To determine if Siebel CRM updates a field for a particular table, you can use the Workflow Policy component column that represents the LAST_UPD column for this table.
  - To determine if Siebel CRM modified a field in the Workflow Policy component, you can use the field that is named after the Workflow Policy component.

## Guidelines for Using Specialized Operators

It is recommended that you adhere to the guidelines described in the following table if you use specialized operators.

| Comparison | Value | Explanation |
|---|---|---|
| IS ADDED | If you define a Workflow Policy component column in the Condition field, and if you define nothing in the Condition value, then use IS ADDED. | If an instance of the Workflow Policy component is added, then the Workflow Policy condition is met. For example, if you define the Workflow Policy component column for the service request in the Condition field, and if you use IS ADDED in the comparison, then the condition is met when Siebel CRM creates a new service request. |
| IS UPDATED | If you define a field in the Condition field, and if you define nothing in the Condition value, then use IS UPDATED. The condition is met when the field changes. | For example, if you define the status of a service request in the Condition field, and if you use IS UPDATED in the comparison, then the Workflow Policy condition is met when Siebel CRM changes the status of the service request. For more information, see *Using IS UPDATED in the Conditions List*. |
| IS DELETED | If you define a child Workflow Policy component in the Condition field, and if you define nothing in the Condition value, then use IS DELETED. | A *child Workflow Policy component* is a Workflow Policy component that Siebel CRM associates with a parent Workflow Policy component. For example, a parent Workflow Policy component might be Service Request. A child Workflow Policy component might be Service Request Activity.<br><br>If you use IS DELETED in conjunction with another Workflow Policy condition, then the other condition must reference the parent Workflow Policy component. |

| Comparison | Value | Explanation |
|---|---|---|
| | | For more information, see *Using IS DELETED in the Conditions List*. |

## Generating Database Triggers with Specialized and Standard Operators

Siebel CRM implies an OR between specialized operators, where one or more of the Workflow Policy conditions must be met before the action occurs. For example, a service representative can receive an email if Siebel CRM adds an activity to an open service request. The following conditions in the policy implement this example:

- `Service Request Status = 'Open'`
- `Service Request Activity Component IS ADDED`

If a Workflow Policy condition is IS ADDED or IS UPDATED, then the database triggers that Siebel CRM creates do not represent every condition defined in the policy. The policy ignores any database triggers that are not represented. For details, you can examine the entries in the trigger.sql file that Siebel CRM creates as a result of doing the comparison. This behavior is expected.

If you modify a condition, then you must run Generate Triggers so that Siebel CRM implements this condition. For more information, see *Overview of Creating Database Triggers*.

If you use a Workflow Policy condition with a standard operator, then the database triggers that Siebel CRM creates encompass the condition. If you use a specialized operator, then Workflow Monitor Agent evaluates this condition at runtime.

## Using IS UPDATED in the Conditions List

Siebel CRM joins Workflow Policy conditions when it runs an IS UPDATED statement, but the format of the trigger.sql statement that it creates for the condition does not include an AND operator in the SQL format.

If a Workflow Policy condition is met, and if IS UPDATED does not occur, then the Workflow Monitor Agent calls the policy. If an IS UPDATED operator is included as criteria on a field in this condition, then Siebel CRM does not check any of the other fields that the condition references.

## Using IS DELETED in the Conditions List

The example in this topic uses IS DELETED in the Conditions list. If a user deletes an activity from a service request that includes a Sub-Status of In Process, then Siebel CRM must notify the service request owner. The following table shows an example of the IS DELETED configuration in the Conditions list.

| Policy | First Condition | Second Condition | Action |
|---|---|---|---|
| References the Service Request Workflow Policy object. | The first condition depends on each of the following situations being true:<br>• Field is Activity Component<br>• Comparison is IS DELETED<br>• Value is empty | The second condition depends on each of the following situations being true:<br>• Field is Service Request Sub-Status<br>• Comparison is equal (=)<br>• Value is In Process | Send an email to the service request owner. |

If you use IS DELETED, then Siebel CRM cannot track the ROW_ID of the record that it deletes from the child Workflow Policy component. It cannot log this record in the S_ESCL_REQ table and the Workflow Monitor Agent cannot identify the deleted row. If you must use Siebel Workflow to capture the deleted row, then you must use a Workflow Process

that a runtime event starts. The runtime event is the BusComp_PreWriteRecord event. For more information about the BusComp_PreDeleteRecord event, see *Siebel Object Interfaces Reference* . For more information, see *Tables That Workflow Monitor Agent Uses*.

# Using Date Calculations in the Conditions List

The Workflow Monitor considers date and time when it evaluates a Workflow Policy condition that performs a date comparison. You can use the CURRENT operator for a date comparison, in the following format:

```
CURRENT [+/-] d:h:m
```

where:

- *d* is the day
- *h* is the hour
- *m* is the minutes

For example, the following value is the value in CURRENT plus one day, two hours, and three minutes:

```
CURRENT + 01:02:03
```

Other examples of where you can use CURRENT are:

- You can use CURRENT in the comparison value for a date field.
- You can use CURRENT to define the activation date and expiration date for a notification action.

For more information, see *Predefined Business Services*.

# How Siebel CRM Interprets a Field That Is Not Known

If a field value is *not known* rather than having no value, then Siebel CRM interprets it as null. For example, assume a condition includes the following logic:

```
FieldA IS UPDATED
FieldB <> "MyValue"
```

In this situation, if FieldB is null, then the following condition is false:

```
<> "MyValue"
```

Siebel CRM interprets a Null field value as not known. It assumes this field could be set to anything, including Y. Siebel CRM cannot conclude that the field is definitely not set to Y, so it returns false and does not run the Workflow Monitor Agent. To fix this, use one of the following:

```
FieldB IS NULL
```

OR,

```
FieldB <> "MyValue"
```

# 16 Administering Workflow Policies

## Administering Workflow Policies

This topic describes how to administer Workflow Policies. It includes the following topics:



## Confirming Workflow Policies Installation

Make sure that Workflow Policies are installed correctly. You install the Workflow Policies module when you install the Siebel Server and the Siebel client. You use your license key to activate Workflow Policies. For information about the installation process, see the installation guide for the operating system you are using.

For Workflow Policies to work, the Server Component Group *Workflow Management* and the Components *Workflow Monitor Agent* and *Workflow Action Agent* must be enabled on at least one Siebel Server in the Siebel Enterprise.

### Confirming Siebel Workflow is Set Up for Workflow Policies

Make sure that your license key includes Siebel Workflow. You must also make sure you installed the Siebel Server correctly. Siebel Workflow runs as a server component on the Siebel Server.

#### To confirm Siebel CRM is set up for Workflow Policies

1. Log in to the Siebel client as the Siebel administrator.
2. Navigate to the Administration - Business Process screen.
3. In the list of views, verify that the Policies, Policy Groups, and so on, are visible. This visibility indicates that your license key is correct.
4. Log in to the Siebel application that is configured to manage the server component groups, navigate to the Administration - Server Management screen, Servers view, then the Component Groups view.
5. Make sure the Workflow Management component group is active.

# Administering Database Triggers on the Workflow Policy Server

This topic describes how to administer database triggers on the Workflow Policy server. It includes the following topics:



## Overview of Creating Database Triggers

The Generate Trigger (GenTrig) server component on the Siebel Server allows you to create database triggers. Workflow Policies use these triggers to identify the records that match Workflow Policy conditions. You must run Generate Triggers if you do any of the following:

- Define new Workflow Policies, including assignment policies, except for Workflow Policies with the Batch Flag property set to TRUE.
- Delete Workflow Policies, including assignment policies.
- Amend Workflow Policy conditions or policy criteria.
- Change activation or expiration dates of policies, including assignment policies.

### How Generate Triggers Works with a Workflow Policy That Contains Multiple Conditions

If a Workflow Policy includes two or more Workflow Policy conditions, then Generate Triggers uses OR logic instead of AND logic. The following table shows some example Workflow Policy conditions that Siebel CRM uses to create a Workflow Policy that references the Account object.

| Property | Condition 1 | Condition 2 |
|---|---|---|
| Condition Field | Account Modification Num | Account Last Update By |
| Operation | > (greater than) | <> (not equal to) |
| Value | 0 | 0-1 |

Siebel CRM can use multiple database triggers for multiple Workflow Policy conditions in one Workflow Policy. This configuration keeps Generate Triggers functionality and Workflow Monitor Agent functionality separate:

- Generate Triggers monitors changes that Siebel CRM makes to database records and inserts records in tables that are specific to a Workflow Policy.
- Workflow Monitor Agent evaluates conditions, determines if the conditions that are associated with the rule are met, and runs the actions that are associated with the Workflow Policy.

### Using an AND Condition with Multiple Database Triggers

If multiple Workflow Policy conditions exist in a Workflow Policy, then you cannot use an AND condition between database triggers. Generate Triggers can only monitor database changes. Database changes that meet different conditions might not be concurrent. Using an AND condition can cause Generate Triggers to miss many conditions because the conditions might be on different tables.

For example, assume a Workflow Policy contains the following Workflow Policy conditions:

- SR area is Network
- Activity Priority is 1-ASAP

In this situation, Siebel CRM creates the following database triggers:

- One database trigger monitors a service request that Siebel CRM creates or updates, and then determines if the area equals Network.
- One database trigger monitors an activity that Siebel CRM creates or updates, and then determines if the Priority equals 1-ASAP.

If you use an AND database trigger, and if a user creates a service request that does not include an activity, then Siebel CRM does not run the database trigger because the activity does not exist. If the user then adds an activity to the service request, then no database trigger runs because the service request does not change. The AND condition causes Siebel CRM to miss this violation. If you use an OR condition, and if Workflow Monitor Agent evaluates the Workflow Policy condition even though multiple violations exist in the S_ESCL_REQ table, then the Workflow Monitor Agent only processes one request because the other requests do not evaluate to TRUE. For more information, see *Tables That Workflow Monitor Agent Uses*.

## Configuring Database Triggers

This topic describes how to configure the Generate Triggers server component. You can configure this server component from the Siebel client or from the command line. The Siebel client and the command line use the same parameters.

> **CAUTION:** If you incorrectly define a Workflow Policy condition, then running Generate Triggers can result in an invalid database trigger. An invalid database trigger can prevent Siebel CRM from processing normal user transactions. It is recommended that you thoroughly test your Workflow Policies in a test environment before you migrate them to a production environment.

### To configure database triggers

1. Navigate to the Administration - Server Management screen, then the Jobs view.
2. In the Jobs list, create a new record.
3. In the drop-down list for the Component/Job field, choose Generate Triggers.

4. In the Job Parameters list, click New to modify component-specific parameters, using values from the following table.

| Name | Value | Description |
|---|---|---|
| Remove | TRUE or FALSE (default) | Set to TRUE to create DROP TRIGGER statements to clean up the database triggers. Remove does not create CREATE TRIGGER statements. |
| Trigger File Name | Valid filename on the Siebel Server | Defines the name and output location for the SQL script file. The default value is TRIGGER.SQL. Siebel CRM creates this file and then places it in the root directory of the Siebel Server during installation. |
| EXEC | TRUE or FALSE | Set to TRUE to run the SQL script file automatically. Set to FALSE to run the SQL script manually. For more information, see *Using the EXEC Parameter*. |
| Mode | ALL, WORK, or ASGN | Select one of the following values:<br>○ **ALL.** Creates database triggers for Workflow Policies and Assignment Manager.<br>○ **WORK.** Creates database triggers only for Workflow Policies.<br>○ **ASGN.** Creates database triggers only for Assignment Manager. |
| Privileged User Name and Privileged User Password | Assigned Privileged User name and password | To make sure the user enters a Privileged User name and password, you can define the username and password. Siebel CRM considers the Table Owner (Oracle/DB2) or Database Owner (MSSQL) as a Privileged User. You can enter the Table Owner or Database Owner username and password in the Privileged User name and password fields. |

   For a description of generic and enterprise parameters, see *Siebel System Administration Guide*
5. Enter the Privileged User (Table or Database Owner) username and password.
6. In the Job Detail form, choose the applet-level menu, and then choose Start Job.
7. To view changes to the state, choose the applet-level menu, and then click Run Query to refresh the screen.

   The Status field contains Success or Error after Siebel CRM refreshes the screen. You can view the log details.
8. If EXEC equals FALSE, then you must manually run the SQL script file.

   For more information, see *Manually Running the SQL Script File*.

## Using the EXEC Parameter

The EXEC parameter specifies how to run the SQL script file automatically according to one of the following values:

- **TRUE.** Generate Triggers automatically, creates the SQL script and applies it to your database.
- **FALSE.** You must manually run the SQL script file

Set the EXEC parameter to FALSE if you want to apply the triggers manually. Reasons for applying the triggers manually are:

- The DBA (database administrator) has not provided Siebel administrators the privilege to make schema changes, including the ability to modify or add triggers. This is very common in environments where the DBA is not directly associated with the Siebel CRM team and the DBA wants to manage database changes directly, analyzing the possible negative database performance impacts.
- You want to add the triggers during a down time, so use `EXEC=False` to generate the triggers *now* but apply them later, for exmaple, at 2:00 AM on a Saturday night.

## Manually Running the SQL Script File

After Generate Triggers finishes, if the EXEC parameter is FALSE, then you must run the SQL script file.

### To manually run the SQL script file

1. Connect to the server that contains the database.
   Connect as the Siebel table owner (Oracle/DB2) or database owner (MSSQL). Use the SQL tool that your RDBMS vendor provides.
   For example, use SQL*Plus for Oracle or the Microsoft SQL Server Studio for MSSQL.
2. Run the SQL script file that the Trigger File Name parameter references.
   The default file name is trigger.sql. The default location of this file is the root directory of where you installed the Siebel Server. For example:

   ```
   ...\ses\siebsrvr\trigger.sql
   ```

   If your environment uses an MS SQL server database, then you must run the trigger.sql script as the owner login, which is also known as the dbo login.
3. After executing the trigger.sql script, ensure that there are no errors thrown. If so, examine the logs and determine the source of the issue and correct the policies as necessary. If the source of the issue is unclear, log a Service Request with Oracle Support.

# Guidelines for Configuring Database Triggers

If you configure the Generate Triggers server component, then it is recommended that you adhere to the following guidelines:

- If you delete a Workflow Policy, then you must run Generate Triggers with the remove parameter set to TRUE, which removes every database trigger. You must then rerun Generate Triggers to reset the database triggers for the remaining Workflow Policies.
- If you drop or recreate database task triggers, then you must start a new Workflow Monitor Agent, which refreshes the cache for this agent.
- If you change a Workflow Policy condition or a Workflow Policy group, then you must rerun Generate Triggers. It is not necessary for you to rerun Generate Triggers if you change a Workflow Policy action. For more information, see *Moving a Workflow Policy to a Different Group*.
- If your configuration uses a SQL Server, then make sure you set your default database correctly. To determine your default database, start the SQL Server Enterprise Manager, and then navigate to the SQL Server Machine name. Next, click Security, and then click LOGIN. The default database is listed.

- If a table name exceeds the **<maximum length>** , then Generate Triggers fails with an error that is similar to the following:

  ```
  # character limit, table_name trigger fail
  ```

  To prevent this from happening: set **EXEC=FALSE** when you generate the triggers with the GenTrig component, manually edit the trigger name in the generated SQL file, and then run the SQL script file manually.
- If you run Generate Triggers, then the limit on table names that DB2 SQL uses results in limiting the database trigger name to **<maximum length>** . Siebel CRM derives the database trigger name from the **(length of table name)** plus a **(suffix)**, such as U, I, D, U1, I1, D1, and so on.

# Using Database Triggers with Remote Users

Database triggers supporting Workflow Policies are only created on the Siebel server database; they are not created in Siebel Remote or Siebel Replication databases. For this reason, database changes associated with a Workflow Policy will only trigger when a Siebel Remote or Replication node synchronizes to the primary server database. For example, if a trigger is due to an update to Account information in the S_ORG_EXT table, the trigger will not fire until that change is synchronized to the primary database. If you configure a Workflow Policy that creates database triggers that compare changes in the Siebel database to a Workflow Policy condition, and if these changes affect this condition during synchronization, then the database triggers fire and Siebel CRM writes rows to the S_ESCL_REQ table. For more information, see *Tables That Workflow Monitor Agent Uses*.

# Managing Database Triggers and Database Administration

It is important to make sure your database administrators are informed of database triggers that are active for a Workflow Policy. A database update or insert event causes the database trigger to react, regardless of how the event runs. The mechanism by which a create, update, or delete operation occurs does not change the behavior of the trigger. For example, all the following would cause a trigger to fire if there is an associated trigger on the underlying table:

- A user commits a change in the user interface.
- An EAI or REST call updates a record through the business object layer.
- EIM updates a record through the database layer.

# Fixing Problems in the S_ESCL_REQ Table

If a database trigger runs on a Workflow Policy condition, then Siebel CRM inserts a record in the S_ESCL_REQ (escalation request) table. This table contains the rows that can cause a Workflow Policy to run. After the Workflow Monitor Agent processes a request, it removes the row from this table.

A database trigger does not include the logic that you define in a Workflow Policy condition. The conditions in the database trigger file might not be indicative of the Workflow Policies that are met. When Workflow Monitor Agent runs, the records in the S_ESCL_REQ table causes Siebel Workflow to evaluate the related Workflow Policy conditions. The database triggers exist only to trigger the Workflow Engine to examine the Workflow Policy conditions.

To fix problems in the S_ESCL_REQ table

- Use your database tools to monitor the size and efficiency of the table:
  - If the table is very large, then this situation indicates that Siebel CRM is monitoring too many Workflow Policies. To fix this problem, redefine the Workflow Policy Groups so that they share the load.
  - If Siebel CRM monitors rows but does not remove them after the time interval finishes, then this situation might indicate that you deactivated a Workflow Policy but did not remove the database triggers that are associated with this policy. The database triggers continue to send data that no Workflow Policy works on. To fix this problem, remove the database triggers that are associated with the deactivated Workflow Policy.
- If you create a new business component, then do not set the Table property to the S_ESCL_REQ table. A business component cannot reference the S_ESCL_REQ table.
- Expire, rather than delete a Workflow Policy. For more information, see *Expiring a Workflow Policy*.
- Remove rows that belong to obsolete Workflow Policies. For more information, see *Deleting an Obsolete Workflow Policy*.

# Administering Email Manager and Page Manager

For more detailed information on how to administer Email Manager and Page Manager, including information on all of the following, see *Siebel Email Administration Guide* :

- Configuring Communications Drivers and Profiles for Email
- Administering Siebel Communications Server for Siebel Email Response
- Managing Workflows for Siebel Email Response
- Page Manager and Sending Pages
- Troubleshooting Email Manager and Page Manager

## How Siebel Workflow Sends Email

The Outbound Communications Manager uses the following sequence to send email messages:

1. If the Workflow Policy is met, then the Workflow Monitor Agent inserts a record in the S_APSRVR_REQ table for workflow actions that call the Send Email Workflow Policy programs.
2. Email Manager picks up records from the S_APSRVR_REQ table, setting the status for each record from QUEUED to ACTIVE.
3. Siebel CRM calls the Outbound Communications Manager to log onto the *profile_name* profile.
4. To send emails to recipients, the Outbound Communications Manager uses the Send Message method of the Outbound Communications Manager business service.
5. After successfully sending the email, the status for the record in S_APSRVR_REQ is set to SUCCEEDED.

# Running a Workflow Policy with the Workflow Action Agent

If Siebel CRM runs an action, then the Workflow Action Agent sends a request to the Email Manager and Page Manager. You must define a component for each Workflow Action Agent server task. The Workflow Action Agent does the following:

- Processes requests that Siebel CRM logs in the S_ESCL_ACTN_REQ (escalation action request) table for a single group. For more information, see *Tables That Workflow Monitor Agent Uses*.
- Calls actions that are linked with the Workflow Policy that Siebel CRM processes for Siebel Workflow.
- Logs email and page actions in the S_APPSRVR_REQ table so that Email Manager and Page Manager can run them.
- Purges requests from the S_ESCL_ACTN_REQ table after Siebel CRM finishes processing.

  If the Use Action Agent parameter is set to TRUE in the Monitor Agent process, then you must do the work described in *Starting the Workflow Action Agent*.

## Starting the Workflow Action Agent

You can start the Workflow Action Agent in the same way that you start the Workflow Monitor Agent. For more information, see *Starting the Workflow Monitor Agent*.

If you start the Workflow Action Agent, then it is recommended that you adhere to the following guidelines:

- A Workflow Action Agent exists for each Workflow Monitor Agent. You must start only one Workflow Action Agent for each Workflow Monitor Agent.
- If your environment uses email consolidation, and if you set the Use Action Agent parameter to TRUE on the Workflow Monitor Agent, then you must start Workflow Action Agent separately.

For more information about starting Workflow Action Agent, see *Siebel System Administration Guide* .

## Shutting Down the Workflow Action Agent

You shut down the Workflow Action Agent in the same way that you shut down the Workflow Monitor Agent.

If you restart a Workflow Action Agent, it will immediately begin tracking relevant activities that occurred since it was shut down.

## Using Workflow Action Agent for Batch Policies

You can run Workflow Action Agent separately with batch policies. Workflow Action Agent does not typically improve how Siebel CRM runs a batch policy, but if the action persists for a long time or is intensive, then the Workflow Action Agent might improve this processing.

### To use Workflow Action Agent for batch policies

1. Start Workflow Monitor Agent with the following parameters:
   - Group Name
   - Processes the batch policies is TRUE
   - Use Action Agent is TRUE
2. Start Workflow Action Agent with the Group Name parameter.

   The Workflow Monitor Agent continues processing and enters the qualified rows in the S_ESCL_ACTN_REQ table. The Workflow Action Agent runs actions for rows in the S_ESCL_ACTN_REQ table, and then deletes rows from the S_ESCL_ACTN_REQ table. For more information, see *Tables That Workflow Monitor Agent Uses*.

# Running a Workflow Policy with Workflow Monitor Agent

The *Workflow Monitor Agent* (WorkMon) is a server component that determines if the Workflow Policy conditions are met and runs actions. To run Workflow Policies, you must start the Workflow Monitor Agent. You can start and stop the Workflow Monitor Agent server task in the Administration - Server views.

This topic describes how to run a Workflow Policy using the Workflow Monitor Agent. It includes the following topics:

- *Tables That Workflow Monitor Agent Uses*
- *How the Workflow Monitor Agent Runs Workflow Policies*
- *How Workflow Monitor Agent Calculates the Duration*
- *Using Replication with the Workflow Monitor Agent*
- *Starting the Workflow Monitor Agent*
- *Stopping or Restarting a Component of the Workflow Monitor Agent*
- *Keeping Definitions for Workflow Monitor Agent Current*
- *Setting the Parameters of the Workflow Monitor Agent*

## Tables That Workflow Monitor Agent Uses

The following table describes some of the key Siebel database tables that Workflow Monitor Agent uses.

| Table Name | Description |
|---|---|
| S_ESCL_REQ | **Escalation request.** Contains the potential matching requests that an application causes. For more information, see *Fixing Problems in the S_ESCL_REQ Table*. |
| S_ESCL_STATE | **Escalation state.** Contains the policy matches that describe time. |
| S_ESCL_ACTN_REQ | **Escalation action request.** Contains the requests to run actions, which Siebel CRM uses only if Use Action Agent is TRUE. |

| Table Name | Description |
|---|---|
| | |
| S_ESCL_LOG | **Escalation log.** Contains a history of rows in the base table that contains matched policies. |

# How the Workflow Monitor Agent Runs Workflow Policies

The Workflow Monitor Agent runs several server processes that monitor the Siebel database. In summary, the Workflow Monitor Agent does the following:

- Examines the S_ESCL_REQ table to determine when the Workflow Policy conditions are met.
- Monitors Workflow Policies in a single group. You can run only one process of the Workflow Monitor Agent on one group at one time. If you run two processes on the same group, then a deadlock can occur. You can run multiple processes at the same time but you must run each process on a different group.
- If the Action Agent parameter is True, then Workflow Monitor Agent creates requests for the Workflow Action Agent in the S_ESCL_ACTN_REQ table.
- Purges requests from the S_ESCL_REQ table after processing finishes. If Siebel CRM activates a database trigger because a Workflow Policy condition is met, then it inserts a record in the S_ESCL_REQ table. The Workflow Monitor Agent evaluates the request on the rules that the policies in the Workflow Policy group establish.

The Workflow Monitor Agent handles a request in one of the following ways:

- **Scenario 1:** The Action Agent is True and the request in the S_ESCL_REQ table does not include a duration in the Workflow Policy. The Workflow Monitor Agent logs an entry in the S_ESCL_LOG table and sends the request to the S_ESCL_ACTN_REQ table.
- **Scenario 2:** The Action Agent is True and the request in the S_ESCL_REQ table does not include a duration in the Workflow Policy. The request remains in the S_ESCL_STATE table until the duration is met or until Siebel CRM removes the request because the Workflow Policy conditions are no longer met. Workflow Monitor Agent evaluates each request that remains in the S_ESCL_STATE table for a duration or to determine if the condition still matches values in the S_ESCL_STATE table. As each match occurs, Workflow Monitor Agent logs an entry in the S_ESCL_LOG table or sends it to the S_ESCL_ACTN_REQ table.

# How Workflow Monitor Agent Calculates the Duration

If a Workflow Policy includes a duration, then Siebel CRM calculates the duration starting with the time that the Workflow Monitor Agent detects that the row is in violation of the policy. It does not start the duration from when Siebel CRM inserts the row in the S_ESCL_REQ table. For example, assume you define a Workflow Policy and set the duration as one week, but you do not start the Workflow Monitor Agent until several days after Generate Triggers runs. In this example, the Workflow Policy action runs one week from when Workflow Monitor Agent starts. It does not run one week from when you create the Workflow Policy or when Generate Triggers runs.

# Using Replication with the Workflow Monitor Agent

Only one Workflow Monitor Agent and Workflow Action Agent can monitor a Workflow Policy group in a Siebel Enterprise. For example, assume a regional node runs a Workflow Monitor Agent that monitors a group named Group 1. Another Workflow Monitor Agent that resides in the headquarters node monitors a group named Group 2. This configuration can run the Workflow Policies that it requires and meet the restriction of one Workflow Monitor Agent for one group. Multiple Workflow Monitor Agent and Workflow Action Agent processes can monitor different groups that run at the same time.

# Starting the Workflow Monitor Agent

You use the Server Manager command line to start the Workflow Monitor Agent.

> **CAUTION:** If a Workflow Process Manager server task fails, and if more server tasks start that also fail, then Workflow Monitor Agent exits with error after only one violation. This situation can result in Workflow Monitor Agent starting more server tasks that are not required when the Workflow Process Manager server task fails. To avoid this situation, it is recommended that you stop, and then restart Workflow Process Manager.

For more information, see *Setting the Parameters of the Workflow Monitor Agent*. For more information about using the Siebel Server Manager Command Line Interface, see *Siebel System Administration Guide* .

## To start the Workflow Monitor Agent

1. Start the Server Manager. Enter the following command in the Server Manager command line:

   ```
   srvrmgr /g gateway_server_address /s Siebel_Server_name /e enterprise_server_name
   /u server_administrator_user_name /p server_administrator_password
   ```

2. Start a new Workflow Monitor Agent server task in background mode. Enter the following command:

   ```
   start task for component WorkMon with SleepTime=time,GroupName=group_name
   ```

3. (Optional) Start a new Workflow Monitor Agent server task to run in batch mode to process a Batch policy. Enter the following command:

   ```
   start task for component WorkMon with GroupName=group_name, BatchMode=Y
   ```

   You must create a separate server component instance for each Workflow Monitor Agent server task.

# Stopping or Restarting a Component of the Workflow Monitor Agent

You can stop or restart a Workflow Monitor Agent component.

### To stop or restart a component of the Workflow Monitor agent

1. In the Siebel client, navigate to the Administration - Server Management screen, then the Components view.
2. Choose the component, and then click Shutdown or Startup.

If you make changes to component parameters, then you must stop, and then restart the component. Also, you must define a separate instance of the Workflow Monitor Agent for each Workflow Policy group.

## Keeping Definitions for Workflow Monitor Agent Current

Workflow Monitor Agent and Workflow Action Agent read Workflow Policy configurations from the Siebel database. If you modify a Workflow Policy object, column, or program, then you must restart the Workflow Monitor Agent and Workflow Action Agent in the following situations:

- After modifying an object, column, or program in the Siebel Repository
- After using the Workflow Administration views in the Siebel client to modify a Workflow Action

If you do not restart Workflow Monitor Agent and Workflow Action Agent, then the modifications do not go into effect until Siebel CRM reloads the Workflow Policies, as defined in the Reload Policy parameter. Because the Reload Policy parameter is set to 600 seconds by default, in most situations 10 minutes elapses before the modifications go into effect if you do not perform a restart.

If you modify a Workflow Process, then it is not necessary to restart the Workflow Process Manager. For more information about checking in and checking out objects, see *Using Siebel Tools* .

### To keep definitions for Workflow Monitor Agent current

1. Deliver the modifications done to a Workflow Policy object, column or program.
2. If necessary, restart Workflow Monitor Agent and Workflow Action Agent.

   For more information, see *Stopping or Restarting a Component of the Workflow Monitor Agent*.

## Setting the Parameters of the Workflow Monitor Agent

The following table describes the parameters to set for the Workflow Monitor Agent in the Server Manager command line interface.

| Parameter Name | Display Name | Description | Default Value |
|---|---|---|---|
| ActionAgent | Use Action Agent | Determines if the Action Agent runs with Monitor Agent.<br><br>If set to FALSE (the default value), then the Workflow Action Agent server component starts in the Workflow Monitor Agent and the Workflow Monitor Agent runs the actions.<br><br>Several factors apply when configuring Use Action Agent. For more information, see *Setting the Use Action Agent Parameter*. | FALSE |
| ActionInterval | Action Interval | Specifies the action execution interval in seconds. This parameter determines when Siebel CRM runs actions for a Workflow Policy on | 3600 |

| Parameter Name | Display Name | Description | Default Value |
|---|---|---|---|
| | | the row of a base table. This parameter limits the number of times that Siebel CRM runs an action if a row continues to go in and out of a matching state.<br><br>If the same record repeatedly meets the same Workflow Policy before the action interval expires, then Siebel CRM removes the record from the S_ESCL_REQ table and does not perform the action again. For more information, see *Tables That Workflow Monitor Agent Uses*.<br><br>The default value is 3600 seconds. If you use this parameter, then you must set the parameter to a value greater than 0 (zero) or unexpected behavior might result. | |
| BatchMode | Processes the batch policies | Determines if Monitor Agent runs in batch mode. If the BatchMode parameter is set to:<br><br>• TRUE, then Siebel CRM evaluates only those policies with the Batch Flag set to TRUE. Workflow Monitor Agent runs one time. It processes records in the table, and then exits.<br>• FALSE, then Siebel CRM evaluates only those policies with the Batch Flag set to FALSE. | FALSE |
| CheckLogCacheSz | Cache size of Policy violations | Specifies the number of Workflow Policies that Siebel CRM stores in the cache. To determine if an action already started, Siebel CRM uses the ROW_ID and RULE_ID in the map or cache.<br><br>The Workflow Monitor examines the S_ESCL_LOG table for policy violations for the same BT_ROW_ID and RULE_ID. | 100 |
| DeleteSize | Request delete size | Specifies the number of records to commit at a time. The minimum value is 1. If the Workflow Monitor encounters a deadlock, then you can reduce the default value from 500 to 125 with minimal performance degradation. To avoid a call stack error, do not set the DeleteSize parameter to zero. | 500 |
| GenReqRetry | Number of seconds to retry | Specifies the number of seconds to retry sending a Generic Request message. | 120 |
| GroupName | Group Name | (Required) Specifies the Workflow Policy group on which Monitor Agent works. | Not applicable |
| IgnoreError | Ignore errors | Determines if Siebel CRM ignores errors while it processes a request. By default, the Workflow Monitor Agent and Workflow Action Agent do not ignore errors that occur while processing a request. If an error occurs, and if the IgnoreError parameter is set to:<br><br>• TRUE., then the Workflow Monitor Agent logs the error, deletes the request, and continues processing the next request.<br>• FALSE, then the Workflow Monitor Agent exits and sends an email to the mail Id that you define in the Mailing Address parameter.<br>It is recommended that you set the IgnoreError parameter to FALSE. | FALSE |

| Parameter Name | Display Name | Description | Default Value |
|---|---|---|---|
| | | | |
| KeepLogDays | Number of days to keep violation information | Specifies the number of days of violation information that Siebel CRM retains in the log. It removes log information that is older than the value in the KeepLogDays parameter. To prevent Siebel CRM from removing this log information, you can set this value to 0 (zero). For more information, see *Setting the Keep Log Days Parameter*. | 30 |
| LastUsrCacheSz | Cache size of last user information | Specifies the number of last user information items that Siebel CRM caches on the Siebel Server.<br><br>If running an action, then the information about the last user who modified the row of the base table is available as a token substitution in the program arguments. To improve throughput performance of actions that Siebel CRM runs, it can cache this information. | 100 |
| MailServer | Mail Server | Specifies the name of the email server where Siebel CRM sends an email to notify you that an abnormal termination occurred. | Not applicable |
| MailTo | Mailing Address | Specifies the email address where Siebel CRM sends an email if an abnormal termination occurred.<br><br>If a Workflow Agent process exits with an error, then Siebel CRM sends an email to the address that you specify. An error can result if an action fails to run, an object definition is not valid, and so on. | Not applicable |
| NumRetries | Number of Retries | Specifies the number of times Siebel CRM attempts a recovery.<br><br>If Siebel CRM loses connectivity with the Siebel database, then the NumRetries parameter works in conjunction with the Retry Interval and Retry Up Time parameters to reconnect MTS or server components to the Siebel database. | 10000 |
| ReloadPolicy | Reload Policy | Specifies the policy reload interval in seconds. It determines how frequently Siebel CRM reloads Workflow Policies. Reloading allows Siebel CRM to change the information that it displays and to incorporate changes that Generate Triggers creates. The default value is 600 seconds. | 600 |
| Requests | Requests Per Iteration | Specifies the maximum number of requests that Siebel CRM reads for each iteration. It controls the maximum number of requests that Workflow Monitor Agent reads from the requests queue in one iteration. Between iterations, Workflow Monitor Agent does the following:<br><br>• Deletes processed requests and commits from the requests queue<br>• Optionally reloads policies from the Siebel database<br>• Determines if a shutdown request exists<br>• Optionally sleeps | 5000 |
| Sleep Time | Sleep Time | Specifies the time in seconds that the Workflow Agent server task sleeps after it polls for events and fulfills obligations to notify. This | 60 |

| Parameter Name | Display Name | Description | Default Value |
|---|---|---|---|
| | | task waits for more requests when it sleeps. The default setting is 60 seconds so this task sleeps every one minute. It wakes up, processes requests if there are any, and then sleeps. It does this work repeatedly.<br><br>After the Workflow Agent process finishes, the Workflow Agent process stops polling until the time period that the Sleep Time parameter expires. This parameter affects the performance of the Workflow Agent process and the responsiveness of the Siebel Server. | |

## Setting the Use Action Agent Parameter

The following table describes when to set the Use Action Agent parameter to True or False.

| When to Set Use Action Agent to False | When to Set Use Action Agent to True |
|---|---|
| In most situations, it is recommended that you start Workflow Monitor Agent with Use Action Agent set to False, which is the default setting.<br><br>If Use Action Agent is False, then Workflow Monitor Agent monitors the situation and runs actions. Some possible actions that run include Workflow Processes, Workflow Policy programs, and email programs.<br><br>If you start Workflow Action Agent to run actions when Use Action Agent is True, and if the action is running a Workflow Process, then Siebel CRM might encounter an access violation error and Workflow Action Agent might fail. | If Siebel CRM runs an action that includes email consolidation, then it is recommended that you set Use Action Agent to True.<br><br>If you use email consolidation, and if Use Action Agent is True, then you must start Workflow Action Agent separately. |

## Setting the Keep Log Days Parameter

If you set the KeepLogDays parameter to 0 (zero), then the Workflow Monitor Agent does not remove data from the S_ESCL_LOG table. If you set this parameter to 0, then it is recommended that you monitor the size of the S_ESCL_LOG table. The size of this table affects performance. Use this parameter judiciously.

Infrequent cleaning of the S_ESCL_LOG table can cause slow performance. You must determine how often to start Workflow Monitor Agent using KeepLogDays, and then restart Workflow Monitor Agent with a setting of 0 for this parameter.

# Configuring a Workflow Policy to Run in Batch Mode

Running a Workflow Policy in batch mode allows you to evaluate more data in the Siebel database and not only the records that call a Workflow Policy. This configuration is useful if you create a new Workflow Policy that Siebel CRM must apply to historical data, and for enforcing a Workflow Policy that includes a date condition.

## To configure a Workflow Policy to run in batch mode

1. In the Siebel client, navigate to the Administration - Business Process screen, then the Policy Groups view.
2. Create a new Policy Group.
3. In the Name and Comments fields, enter a descriptive name and comment.
4. Navigate to the Administration - Business Process screen, then the Policies view.
5. In the Policies List, query the Name field for the Workflow Policy you want to run in batch mode.
6. In the Policies List, make sure the Batch Mode field contains a check mark.

   If you start Workflow Monitor Agent in batch mode, then it determines if Workflow Policies exist with the Batch Mode check box marked. To identify the records that meet the Workflow Policy conditions, each policy causes Siebel CRM to send an SQL statement. It then processes these records and performs the appropriate actions.
7. In the Policy Group field, associate the Workflow Policy with the Workflow Policy group you created in step 2.
8. (Optional) Consolidate email messages:

   a. Add a condition in the Conditions list and a corresponding action in the Actions list.
   b. Consolidate email messages. In the Actions list, click the Consolidate field.

   For more information, see *Consolidating Email Messages*.
9. When you start the Workflow Monitor Agent and Workflow Action Agent server components, use the Workflow Policy group name you created in step 2 and set the Processes the Batch Policies parameter to TRUE.

   For more information, see *Setting the Parameters of the Workflow Monitor Agent*.

If you create a batch Workflow Policy, then you must use an IS ADDED, IS UPDATED, or IS DELETED comparison operator in conjunction with regular Workflow Policy conditions. These comparison operators are special conditions that Siebel CRM uses for Dynamic mode when it triggers rows to look up regular conditions.

# Consolidating Email Messages

If you configure Siebel CRM to consolidate messages, then it sends one email to the recipient that includes information about multiple actions rather than sending information about multiple actions in multiple emails. For example, you can define a Workflow Policy that sends an email to a sales director each time a sales representative sends a quote that includes a discount over 30%. If 20 sales representatives send quotes that include the 30% discount, and if the Batch Mode field is:

- **Checked** (selected), then the sales director receives one email that lists the 20 quotes.
- **Not checked** (not selected), then the sales director receives 20 email messages.

## To consolidate email messages

1. In the Siebel client, navigate to the Administration - Business Process screen, then the Actions view.
2. Make sure the Consolidate field in the Actions list contains a check mark.

# Configuring Batch Mode to Ensure Siebel CRM Calls a Workflow Policy Correctly

You can configure batch mode to make sure Siebel CRM calls a Workflow Policy correctly.

## To configure batch mode to ensure Siebel CRM calls a Workflow Policy correctly

1. Make sure the Batch Mode field is checked for the Workflow Policy.
2. Set the Batch Mode parameter to TRUE for the Workflow Monitor Agent.

## Example of Configuring Batch Mode to Ensure Siebel CRM Calls a Workflow Policy Correctly

The example configuration shown in the following table demonstrate why a previously existing record that meets the Workflow Policy conditions does not call the Workflow Policy.

| Field | Value |
|---|---|
| Workflow Policy Name | Account Active |
| Workflow Object | Account |
| Group | Test |

Using this example, to monitor a field, Siebel CRM starts a Workflow Policy that references a business object when the following situations occur:

- Workflow Condition, Account Status is Active.
- Workflow Action, Action: Send Email to Employee.
- Remove and create database triggers, and then start a Workflow Monitor Agent server task for the Test Workflow Policy group.

Note the following:

- If a user adds a new account that meets these requirements, then Siebel CRM calls the Workflow Policy.
- If a user updates an existing account in such a way that these requirements are met, then Siebel CRM calls the Workflow Policy.

If an account existed in the Siebel database before you created the Workflow Policy, and if the account meets the requirements that you define for this Workflow Policy, then this Workflow Policy does not start a Workflow Process for this account.

If Siebel CRM creates database triggers for this Workflow Policy, then the Workflow Monitor Agent evaluates a record only if the user updates it or if the user creates a new record. Old records are not affected.

To evaluate an account record that existed before you created the Workflow Policy, make sure the Batch Mode field is checked for the Workflow Policy, and that the Workflow Monitor Agent runs with Batch Mode set to TRUE.

# Moving a Workflow Policy to a Different Group

To reply to changes in the business environment or to your configuration, it might be necessary to move a Workflow Policy to a different Workflow Policy group. Siebel CRM must finish requests that are associated with the Workflow Policy

before you can move a Workflow Policy from one group to another group. If you move a Workflow Policy to a different group while a request is pending, then Workflow Monitor Agent fails with a Rule Not Found error. In this situation, you must move the Workflow Policy back to the original group and wait for the request to finish.

If you move a Workflow Policy to a different Workflow Policy group, then you must also update database triggers. These triggers contain the RULE_ID and GROUP_ID for the Workflow Policy. If you do not update them, then rows in the S_ESCL_REQ table reference the original RULE_ID and GROUP_ID. For more information, see *Tables That Workflow Monitor Agent Uses*.

## To move a Workflow Policy to a different group

1. To drop database triggers, run a Generate Triggers server task using the following parameters:

   ```
   EXEC=TRUE
   Remove=TRUE
   ```

   Dropping database triggers makes sure Siebel CRM does not insert any more rows for Workflow Policies in the group you change. To avoid starting more database triggers, it is recommended that you make these changes only if no users or only a minimal number of users are currently using Siebel CRM. For more information, see *Configuring Database Triggers*.
2. If a Workflow Monitor Agent server task is already running, then allow the Siebel Server task to finish.

   This step makes sure the Siebel Server task processes the remaining rows in the S_ESCL_REQ table for the Workflow Policy group.
3. To make sure no more rows occur for GROUP_ID, query the S_ESCL_REQ table for the GROUP_ID.
4. If no server task for the monitor agent is running for the group, then start a server task for the group with the proper group name.
5. Wait for Siebel CRM to finish processing the remaining rows in the S_ESCL_REQ table for the group.
6. Stop the Siebel Server task of the Workflow Monitor Agent for the group.
7. To reassign the Workflow Policy to the group, change the field of the group name for the Workflow Policy to the new group name.
8. Generate database triggers again, using the following parameters:

   ```
   EXEC=TRUE
   Remove=FALSE
   ```

9. Restart the Workflow Monitor Agent for the old group.
10. Start a new Workflow Monitor Agent for the new group.
11. Restart the Workflow Monitor Agent for the old group.
12. Start a new Workflow Monitor Agent for the new group.

# Converting a Workflow Policy to a Workflow Process

It is recommended that you keep a Workflow Policy simple and place most business logic in the Workflow Process. You can convert a Workflow Policy to a Workflow Process. If you convert a Workflow Policy to a Workflow Process, then it is recommended that you adhere to the guidelines described in the following table describes.

| Configuration Used in a Workflow Process | Configuration Used in a Workflow Policy |
|---|---|
| A runtime event. | A specialized operator, such as IS UPDATED or IS ADDED. |
| A branch or decision step. | A standard operator, such as = (equals) or <> (no equal to). |
| A Siebel operation step at the object layer uses logic that modifies data at the data layer.<br><br>A Workflow Process cannot call a Workflow Policy program, but a Siebel operation step can perform an operation in the data layer. | A Workflow Policy program performs an operation at the Siebel database layer. |
| A Workflow Process uses the communication outbound manager to send email. It can use a recipient group and the exact email address. | Not applicable |

# Configuring a Workflow Policy to Reference Multiple Tables

A Workflow Policy references a database trigger and the Workflow Policy can run on a database record only after Siebel CRM commits the record. If a Workflow Policy references multiple database tables, then it goes into effect only after Siebel CRM commits the records on these tables.

For example, Siebel CRM stores opportunity revenue in the S_OPTY_POSTN table and lead quality in the S_OPTY table. A Workflow Policy that applies each of the following Workflow Policy conditions goes into effect only after Siebel CRM commits the records on each table:

- Opportunity Revenue is greater than 10M
- Lead Quality is High

To define multiple business components for the same database table, you can use a search specification. If you configure a Workflow Policy component to monitor a business component, then make sure you configure Siebel CRM to reference fields in the search specification as Workflow Policy columns. To enforce the required behavior, you can then use the Workflow Policy column in the Workflow Policy conditions.

# Expiring a Workflow Policy

When you expire a Workflow Policy, you can remove rows that belong to the expired or Workflow Policy in the S_ESCL_REQ table, which is preferable to deleting the Workflow Policy.

# To expire a Workflow Policy

1. Make a backup copy of the Siebel database or the effected tables.

   If necessary, you can use this backup later to restore the tables to their state that existed before you deleted rows.
2. Use SQL to manually delete the rows that reference expired or deleted Workflow Policies. Delete these Workflow Policies according to RULE_ID, for example, as follows:

   ```
   delete from table name where RULE_ID = 'rule_id'
   ```

   Siebel CRM does not support direct delete or update statements on the Siebel database. Also, the following tables are the only tables that do not contain dependencies. Make sure you perform this procedure in accordance with Siebel Support guidelines:

   - S_ESCL_REQ
   - S_ESCL_ACTN_REQ
   - S_ESCL_STATE
3. Expire the Workflow Policy:

   a. Log in to the Siebel client and navigate to the Administration - Business Process screen.
   b. Locate the Workflow Policy you want to expire in the Policies view.
   c. Enter a date and time in the Expiration field that has already occurred.
4. To delete database triggers that reference expired Workflow Policies, run the Generate Triggers server task with the following parameters:

   Drop Triggers:

   ```
   EXEC = True
   Remove = True
   ```

   Generate Triggers:

   ```
   EXEC = True
   Remove = False
   ```

   If some Workflow Policies are expired, and if these policies did not drop and recreate database triggers, then the database triggers for these policies can still cause Siebel CRM to insert records in the tables for Siebel Workflow. Dropping these database triggers prevents Siebel CRM from creating rows in the S_ESCL_REQ table that are related to expired policies.
5. If necessary, recreate database triggers.

   For more information, see *Configuring Database Triggers*.
6. If the errors continue, then contact Oracle Technical Support.

   Include a description of the steps you performed to resolve the error, along with a copy of the Workflow Monitor Agent trace files with the following traces set:

   ```
   Error Flags = 2
   SQL Trace Flags = 2
   ```

# Resolving a Problem That Fails to Load Parent Rows

If Siebel CRM displays a Failed to Load Parent Rows error message, then it might be due to the presence of rows in the S_ESCL_REQ, S_ESCL_STATE table and the S_ESCL_ACTN_REQ table that reference an expired Workflow Policy. This topic describes how to use SQL queries to determine if rows exist in these tables that reference a policy that no longer exists in the S_ESCL_RULE table.

## To resolve a problem that fails to load parent rows

1. Look for the following error message in the WorkMon_file_number.log trace file:

   ```
   [DBG33] 2000-01-24 08:49:30 Message: Rule not found
   [DBG33] 2000-01-24 08:49:30 Message: Failed to load parent rows
   ```

   Siebel CRM logs this error message if Workflow Monitor agent attempts to process records from the S_ESCL_REQ, S_ESCL_STATE table or the S_ESCL_ACTN_REQ table but the Workflow Policy that it is trying to review is expired or deleted.
2. To determine if records exist that reference an expired Workflow Policy, run the following query:

   ```
   select row_id, name, expire_dt from s_escl_rule
   where expire_dt is not null and expire_dt <= getdate()
   ```

   Sysdate is an Oracle date time function. The corresponding function for SQL Server is getdate.
3. Note the ROW_ID for the expired Workflow Policy identified in step 2.
4. To determine the number of records that exist in the S_ESCL_REQ table that reference an expired Workflow Policy, run the following query:

   ```
   select * from s_escl_req where rule_id in (select row_id from s_escl_rule
   where expire_dt is not null and expire_dt <= getdate())
   ```

5. Repeat the last step for the S_ESCL_STATE and S_ESCL_ACTN_REQ tables.
6. If records exist in the S_ESCL_REQ, S_ESCL_STATE table, or the S_ESCL_ACTN_REQ table that reference an expired Workflow Policy, then log in to the Siebel client and start the Workflow Monitor Agent with Ignore Errors set to TRUE.
   Workflow Monitor Agent can bypass the error and clean the S_ESCL_REQ table.

   > **CAUTION:** To clean up the tables, you can temporarily set Ignore Errors to True. It is strongly recommended that you do not permanently set Ignore Errors to True.

7. Stop the Siebel Server task.
8. Restart the Siebel Server task with Ignore Errors set to FALSE.
   This step makes sure you do not overlook other types of errors.
9. If the errors continue, then delete rows by RULE_ID, expire old Workflow Policies, and delete database triggers.

# Deleting an Obsolete Workflow Policy

You delete an obsolete Workflow Policy for the following reasons:

- To avoid the ESC-00054 error when you start Workflow Monitor.

- To avoid the unnecessary use of storage space. Normally, Workflow Monitor does not process rows that are associated with an obsolete Workflow Policy. The rows remain unused in the S_ESCL_REQ table.

Database triggers are still in effect when you expire or delete a Workflow Policy and shut down the Workflow Monitor. Database triggers that are related to an expired or deleted policy continue to work until you delete them from the Siebel database. Siebel CRM might continue to trigger rows in the S_ESCL_REQ table between the time you expire a Workflow Policy and the time you finish dropping and redefining database triggers.

You might encounter the ESC-00054 error when you start the Workflow Monitor because Workflow Monitor Agent finds Workflow Policies in the S_ESCL_REQ table that are not active or are not present. The workflow policies that Siebel CRM stores in the S_ESCL_RULE table must remain in the S_ESCL_REQ table and Siebel CRM must not process them.

If Use Action Agent is set to FALSE for the Workflow Monitor, then the Workflow Monitor Agent runs for the Action Agent, bypasses the S_ESCL_ACTN_REQ table, and uses only the S_ESCL_REQ table. If rows in the S_ESCL_REQ table exist for a Workflow Policy that is expired or deleted, then you must delete them before you start the Workflow Monitor. For more information, see *Expiring a Workflow Policy*.

For more information, see *Tables That Workflow Monitor Agent Uses*.

# Locating Workflow Policies That Are Deleted or Expired

The examples in this topic can help you determine if rows in the S_ESCL_REQ table exist that are related to a deleted or expired Workflow Policy. For more information, see *Tables That Workflow Monitor Agent Uses*.

You can enter a query that locates deleted Workflow Policies, for example, as shown in the following procedure.

## To locate deleted Workflow Policies

- Run the following query:

```
select RULE_ID, count(RULE_ID) from S_ESCL_REQ a
where not exists (select row_id from S_ESCL_RULE b where a.RULE_ID = b.ROW_ID)
group by RULE_ID
```

You can enter a query that locates expired Workflow Policies, for example, as shown in the following procedure.

## To locate expired Workflow Policies

1. Run the following query:

```
select a.RULE_ID, b.NAME, count (a.RULE_ID), b.EXPIRE_DT, b.SUB_TYPE_CD
from S_ESCL_REQ a, S_ESCL_RULE b
where a.RULE_ID = b.ROW_ID
and b.EXPIRE_DT < sysdate
group by a.RULE_ID, b.NAME, b.EXPIRE_DT, b.SUB_TYPE_CD
```

   Note that *sysdate* is the Oracle current date and time function. Replace *sysdate* with the appropriate date and time and time identifier for your database platform.
2. Determine if the Workflow Policies unintentionally expired.

   An example of a Workflow Policy that unintentionally expires is someone forgetting to extend the dates but nobody changed the database triggers since the policy expired.

3. If you determine a Workflow Policy unintentionally expired, and if Siebel CRM must process the rows, then do the following:

   - If the Workflow Monitor is running, and if it is not necessary for you to shut it down, then reverse the expiration of the Workflow Policy. To do this, enter a new date and time in the Expiration Date field for policy. Make sure this date and time is greater than the current date and time.

     The Workflow Policy goes into effect after 10 minutes. This delay occurs because the default for the Reload Policy parameter of Workflow Monitor is 600 seconds.
   - If you require the Workflow Policy to go into effect immediately, then shut down Workflow Monitor if it is running, reverse expiration of the policy, and then start Workflow Monitor.

   You can also do this with a Workflow Policy that includes a duration.

# 17 Monitoring, Testing, and Migrating Workflow Policies

## Monitoring, Testing, and Migrating Workflow Policies

This chapter describes how to monitor, test, troubleshoot, and migrate Workflow Policies. It includes the following topics:



## Tracing a Workflow Policy

When Siebel CRM starts a Workflow Policies server process, it creates a trace file for the following Siebel Server tasks:

- Generate Triggers
- Page Manager
- Email Manager
- Workflow Monitor Agent
- Workflow Action Agent

You can view trace files for Workflow Policies and Siebel Server tasks and examine them for error messages and other information.

### To view trace files

1. To view trace files for Workflow Policies and Siebel Server tasks:
   - In the Siebel client, navigate to the Administration - Server Management screen, Tasks view, then the Log view.

     The Tasks list displays the status of the Siebel Server tasks as running or started.
2. To view trace files on the Siebel Server, do one of the following:
   - Navigate to the following directory, and then choose the name for the Siebel Server. You can examine a file that lists the trace files for each server process:

`siebsrvr\log`

- Double-click the Trace File icon to access the trace file. You can view the trace file for application server tasks.

For more information about using trace files, see *Siebel System Administration Guide* .

# Log Levels for Tracing Events for Workflow Policies

Siebel CRM uses the General Events event to log a Workflow Policy. To view informational messages, set the log level to 3. To view debugging information, set the log level to 4. For more information, see *Tracing a Workflow Policy*.

The following table describes the events that Workflow Policy-related Server Components use for logging. Setting a trace level over the default parameter affects performance. It is recommended that you reset the trace level to the default parameter after you finish troubleshooting.

| Event | Level | Description |
|---|---|---|
| SqlParseAndExecute | 4 | Traces SQL statements and execution times. |
| Object Assignment | 3 | Traces Workflow Monitor Agent while it performs Dynamic Assignment. Used in conjunction with Rules Evaluation. |
| Rules Evaluation | 4 | Traces Workflow Monitor Agent while it performs Dynamic Assignment. Used in conjunction with Object Assignment. |
| Task Configuration | 4 | Parameter configuration of the Siebel Server task. |

# Using Charts and Reports for Workflow Policies

You can use charts to analyze how frequently a Workflow Policy condition is met and the total number of policy instances that occur during a particular time period. Reports also summarize Workflow Policy and log information.

The Policy Frequency Analysis view displays information about the number of times a Workflow Policy runs. The Policy Trend Analysis view displays information about Workflow Policy trends.

## To use charts and reports for Workflow Policies

1. Display and review frequency and trend information about a Workflow Policy as follows:
   a. In the Siebel client, navigate to the Administration - Business Process screen, then the Policy Frequency Analysis view.
   b. Examine the information in the view. The following table describes the information available in the Policy Frequency Analysis view.

| List or View | Description |
|---|---|
| Monitor Log | Lists the Workflow Policies. |
| Workflow Policy Frequency Analysis | Displays a chart that illustrates how frequently a Workflow Policy runs.<br><br>To toggle between the Workflow Policy Frequency Analysis view and the Workflow Policy Trend Analysis view, you can use the toggle feature on the chart view. |
| Workflow Policy Trend Analysis | Displays the total number of Workflow Policy conditions that are met over a particular time period. |

2. Display a report for a Workflow Policy as follows:
   a. In the Siebel client, navigate to the Administration - Business Process screen, then the Policies view.
   b. In the menu bar, click the Reports icon, and then choose Workflow Policy.
      Siebel CRM opens a separate Siebel Report Viewer window that displays summary information for the Workflow Policy.

   You can also display a similar report for the Log views.

# Using the Workflow Policy Log to Monitor a Workflow Policy

The Workflow Policy Log view displays a log of the records that meet a Workflow Policy condition that the Workflow Monitor Agent process tracks.

## To use the Workflow Policy log to monitor a Workflow Policy

1. In the Siebel client, navigate to the Administration-Business Process screen, then the Policy Frequency Analysis view.
2. Query the Policy field for the Workflow Policy you want to monitor.
3. Examine the log. The following table describes the information available in the Log.

| Field | Description |
|---|---|
| Policy | The name of the Workflow Policy. |
| Workflow Object | The name of the Workflow Policy object. |
| Object Identifier | The Id of the Workflow Policy object that meets the Workflow Policy condition. |

| Field | Description |
|---|---|
| | |
| Object Values | Identifying information for the row that met the Workflow Policy condition. |
| Event | The date and time that the Workflow Policy condition is met. |

# Testing a Workflow Policy

Testing a Workflow Policy before you implement it in a production environment increases the likelihood that the recipient of an action receives accurate and useful information, and that the overall results meet the business requirements for the Workflow Process. It is critical that you thoroughly test your Workflow Policy and fix problems before you implement the Workflow Policy in a production environment.

**CAUTION:** Your test environment and production environment must use identical software versions.

## To test a Workflow Policy

1. Develop a testing and migration strategy for introducing changes in the production environment.

   For more information, see *Creating a Plan for the Test and Migration Strategy*.
2. Make sure you installed the Workflow Policy components on the Siebel Server.

   For more information, see *Siebel System Administration Guide* .
3. Make sure the Siebel Server processes that your Workflow Policy requires for email and pager are running.
4. Make sure the Workflow Agent processes are running.
5. Make sure each Workflow Policy, Workflow Policy condition, and Workflow Policy action runs as expected:
   - To test your Workflow Policy, enter data that meets the Workflow Policy conditions.
   - Make sure the policies, conditions, and actions are correctly defined.
   - Make sure the policies, conditions, and actions accurately define the transactions.
6. Make sure the Workflow Policy actions perform as expected and occur as expected:
   - Make sure Siebel CRM creates the required database triggers.
   - Make sure the action interval and sleep times are correctly defined.
   - Make sure the proper action runs. For example, you can determine if the email arrives or that the pager notification activates. To monitor progress of the Workflow Agent, you can use the Workflow Policies Log view. For more information, see *Using the Workflow Policy Log to Monitor a Workflow Policy*.
7. After you verify that the Workflow Policy runs as expected, you can migrate it to your production environment.

   For more information, see *Migrating Workflow Policies to the Production Environment*.

# Fixing a Workflow Policy That Does Not Trigger

This topic describes how to troubleshoot a Workflow Policy that does not trigger.

## To fix a Workflow Policy that does not trigger

1. Make sure your test record meets the Workflow Policy conditions.
2. Make sure the Siebel client configuration file references the correct Siebel Server.

   If the Siebel client configuration file does not reference the correct Siebel Server, then an error that is similar to the following might occur:

   ```
   ESC-00053 Error loading rule definitions
   ```
3. Examine the date and time that Siebel CRM activated the Workflow Policy.
4. Examine the Monitor task:
   a. Determine if the Monitor is awake and running on the correct group.
   b. Search the Task Information log for the ROW_ID of your test record:
      - If ROW_ID does not exist, then run GENERATE TRIGGERS.
      - Update your test record.
5. Examine the Action Agent task:
   a. Determine if the Action Agent is awake and running on the correct Workflow Policy group.
   b. Search the Task Information log for the ROW_ID of the test record.
6. Make sure Siebel CRM creates the database triggers.

# Migrating Workflow Policies to the Production Environment

To migrate Workflow Policies to a production environment, you use a procedure that is similar to the procedure you use to migrate policies to a test environment. To help prevent invalid database triggers in the production environment, you apply them in the test environment before you apply them in the production environment.

## To migrate Workflow Policies to the production environment

1. Back up your production environment database.
2. After successfully testing the Workflow Policies in a test environment, migrate the repository changes from the Development (DR) environment to the production environment.

   For more information, see the upgrade guide for the operating system you are using.
3. Redefine the following items that you defined in the test environment in the production environment:
   - Workflow Policy action types

- Workflow Policies
- Workflow Policy groups

When you redefine the object definitions in the production environment, make sure you redefine them exactly as you defined them in the test environment. It is not necessary to redefine information that you defined in the Siebel Repository.

4. In the Siebel client, navigate to the Administration - Server Management screen, then the Jobs view.
5. In the Jobs list, click New.
6. In the Component/Job field, choose Generate Triggers.

   This step creates a new line entry but does not start the Siebel Server task.
7. In the Job Parameters list, click New to modify parameter settings.

   For a description of the parameters that are specific to the Generate Triggers server component, see *Administering Database Triggers on the Workflow Policy Server*. For a description of generic and enterprise parameters, see *Siebel System Administration Guide* .
8. Click Submit.
9. Use trace files to monitor Siebel CRM, as required.

   For more information, see *Tracing a Workflow Policy* and *Siebel System Administration Guide* .

# Setting the Component Parameters Before Running Workflow Policies

When the changes for running the Workflow Policy are delivered to the Integration branch, set the WorkspaceBranchName to Integration Branch and run the following commands:

1. Component WfProcMgr

   For example, set the component parameter WfProcMgr as follows:

   ```
   Parameter WorkspaceBranchName="Name of the integration Branch"
   ```

2. Component GenTrig

   For example, set the component parameter GenTrig as follows:

   ```
   Start task for component GenTrig server <serverName> with exec=true
   privuserPass=DB Password
   privuser=DB Instance
   username=SADMIN
   password=<password>
   ```

3. Component WorkMon

   For example, set the component parameter WorkMon as follows:

   ```
   Start task for component WorkMon with SleepTime=10
   GroupName="Name of the Group Created"
   username=SADMIN
   password=<password>
   Pub Caret -1
   ```

# 18 Siebel Workflow Process and Workflow Policy Reference

## Siebel Workflow Process and Workflow Policy Reference

This chapter provides reference information about Workflow Processes and Workflow Policies. It includes the following topics:

- *Process Property Fields and Arguments*
- *Predefined Business Services*

## Process Property Fields and Arguments

This topic includes reference information for process properties and their arguments. It includes the following topics:

- *Input Argument Fields*
- *Output Argument Fields*
- *Recipient Argument Fields*
- *Search Specification Input Argument Fields*

### Input Argument Fields

The following table describes fields that you can define on an input argument for the business service step, sub process step, and wait step.

| Field | Description |
|---|---|
| Business Component Field | Specifies the name of a business component field. For Business Component Field input arguments. |
| Business Component Name | Specifies the name of a business component that the business object references. For Business Component input arguments. |
| Changed | N/A |
| Input Argument | (Required) Specifies the name of the input argument. For the business service step only.<br><br>The drop-down list displays input arguments that exist for the business service method that you choose.<br><br>If you define a method argument as a business service method argument, and if the Hidden field is set to FALSE, and if the type is Input or Input/Output, then the method argument displays in this drop-down list. |

| Field | Description |
|---|---|
| Preferred Sequence | For the business service step only. |
| Property Name | Specifies the name of the process property. For the Process Property input argument. |
| Subprocess Input | Specifies the name of the input argument. For the sub process step only. |
| Type | (Required) Specifies the type of argument. You can choose one of the following values:<br>• Literal<br>• Process Property<br>• Business Component<br>• Expression |
| Value | Specifies a string value. For Literal or Expression input arguments. This value can be in a drop-down list, depending on the argument that you choose. A string value can contain a maximum of 32,767 characters. |

# Output Argument Fields

The following table describes fields that you can define on an output argument for a business service step, sub process step, and Siebel operation step.

| Field | Description |
|---|---|
| Property Name | (Required) Specifies the name of the process property that stores the results.<br>If you click Property Name, then a drop-down list appears of process properties that are defined for the Workflow Process. For more information, see *Passing a Process Property In and Out of a Workflow Process Step*. |
| Type | (Required) Specifies the type or argument. You can choose one of the following values:<br>• Literal<br>• Output Argument<br>• Business Component<br>• Expression |
| Value | Specifies a string value. Use for Literal or Expression arguments. A string value can contain a maximum of 32,767 characters. |
| Output Argument | Specifies the name of the output argument. For business service steps only. For the Output Argument type of output argument.<br>If you define an argument as a business service method argument, and if the Hidden field is set to FALSE, and if the type is Output or Input/Output, then the argument appears in this drop-down list. |

| Field | Description |
|---|---|
| | |
| Subprocess Output | Specifies the name of the output argument. For sub process steps only. For the Output Argument type of output argument. |
| Business Component Name | Specifies the name of the business component that the business object references. For the Business Component type of output argument. |
| Business Component Field | Specifies the name of a business component field. For the Business Component Field type of output argument.<br><br>You cannot choose a field that references a multi-value group as the value for an input argument or an output argument. If you must use a field that references a multi-value group, then you must define a business component for the field, and then link it to the appropriate business object. For more information, see *Configuring Siebel Business Applications* . |

# Recipient Argument Fields

The following table describes fields that you can define on a recipient argument. For configuration information, see *Adding a Sub Process Step*.

| Field | Description |
|---|---|
| Recipient Type Code | Specifies the type of recipient. Siebel fixes this value as User. You cannot change it. |
| Value Type Code | Specifies the source that Siebel CRM uses to get the recipient value. You can choose one of the following values:<br><br>• Name<br>• Expression<br>• Process Property<br>• Business Component |
| Recipient Name | Specifies the name of the recipient. If you set the Value Type Code to Name, then you must define the Recipient Name field.<br><br>Siebel displays a drop-down list that includes the first name, last name, and login name of a user. You choose one name from this list of users. |
| Business Component Name | Specifies the name of the business component. If you set the Value Type Code to Business Component, then you must define the Business Component Name field. |
| Business Component Field | Specifies the name of the business component field. If you set the Value Type Code to Business Component, then you must define the Business Component Field field. |
| Process Property Name | Specifies the name of the process property. If you set the Value Type Code to Process Property, then you must define the Process Property Name field. |

| Field | Description |
|---|---|
| | |
| Expression | If Siebel CRM gets the recipient value from an expression, then you enter this expression in the Expression property. If you set the Value Type Code to Expression, then you must define the Expression field. |

# Search Specification Input Argument Fields

The following table describes fields that you can define for an input argument on a search specification on the Search Spec Input Arguments tab of the Multi Value Property Window (MVPW) pane.

| Field | Description |
|---|---|
| Expression Business Component | If you enter Expression in the Type field, then you must enter the name of the business component that evaluates the expression. For example, in the Search Specification field, you can enter the following code:<br>`"[Due Date] < '" + [Order Date] + "'"`<br>In this example, the expression business component evaluates Order Date so that Siebel CRM uses the following search specification:<br>`[Due Date] < '07/04/2001 18:51:26'` |
| Filter Business Component | Enter the name of the business component that provides the group of records that Siebel CRM uses to do the search. |
| Search Specification | The value you enter depends on the value that you select for the Type argument:<br>• **Literal.** Enter a literal value in the form of an expression. For example: `= 100`.<br>• **Expression.** Enter an expression. For example:<br>`[Status] LIKE '*Open*'`<br>To evaluate this expression, Siebel CRM uses the Expression business component that you define. |
| Type | (Required) Select either Literal or Expression. |
| Comments | Enter a text description of the purpose of the search. |
| Changed | Check mark indicates a changed search specification. |

# Predefined Business Services

This topic describes some of the predefined business services that you can reference from a Workflow Process. It includes the following topics:

> **Note:** Only a small subset of predefined business services are mentioned here but many more exist. Predefined business services are typically specific to individual Workflow Processes.

- *Server Requests Business Service*
- *Workflow User Event Service Business Service*
- *Workflow Utilities Business Service*
- *Workflow Admin Service Business Service*
- *Other Business Services That a Workflow Process Uses*

For more information about predefined business services, see *Business Processes and Rules: Siebel Enterprise Application Integration* .

## Server Requests Business Service

You can use the predefined Server Requests business service to send a generic request to the Server Request Broker. The Server Requests business service can send a request in the following modes: Synchronous, Asynchronous, and Schedule.

For more information about the Server Requests business service, see the following topics:

- *Using Synchronous and Asynchronous Modes with the Server Requests Business Service*
- *Server Request Broker Parameters*
- *Passing Parameters to Server Components*
- *Server Requests Business Service Methods*
  - *Submit Request Business Service Method Arguments*
  - *Cancel Request Business Service Method Arguments*
- *Examples That Use the Server Requests Business Service*

### Using Synchronous and Asynchronous Modes with the Server Requests Business Service

The Server Requests business service sends a request to the Server Request Broker, and then waits or does not wait for a reply depending on the following mode that it uses:

- **Synchronous.** Waits for a reply.
- **Asynchronous.** Does not wait for a reply.

It is recommended that you use the Server Requests business service in most situations rather than the Workflow Process Manager (Server Request) business service. It is more efficient to make an asynchronous call to a Workflow

Process. If Siebel CRM makes a synchronous call to a Workflow Process, then that Workflow Process runs in the Object Manager and the user must wait for the Workflow Process to end.

## Server Request Broker Parameters

If you configure Siebel CRM to call the Server Requests business service to send a component request, then you must also do the following:

- Define parameters for the Server Request Broker in the input property set.
- Define parameters that are specific to a component in the child property set.

Note the following:

- If Siebel CRM calls the Server Requests business service to call a server component, then the Alias Name property must reference the names of the component parameters. The Alias Name usually does not include spaces.
- You must define parameters for components with the Alias Name in the child property set.
- Siebel does not validate the Alias Name.
- Siebel CRM does not display these arguments in the drop-down list in the Administration - Business Process views.
- Failure to use the correct format results in an error message. The following table describes correct and incorrect formats.

| Correct Format | Incorrect Format |
|---|---|
| `ObjReq.SetProperty "BCName", "Account"` | `ObjReq.SetProperty "Buscomp Name", "Account"` |

## Passing Parameters to Server Components

If you must pass parameters that Siebel CRM does not display as available arguments, then you can define a custom business service that includes the parameters that you require. You can also define a component job that includes these parameters. For more information on component jobs, see *Siebel System Administration Guide* .

## Server Requests Business Service Methods

You can use the following methods with the Server Requests Business Service:

- **Submit Request.** Sends a request to the Server Request Broker. For more information, see *Submit Request Business Service Method Arguments*
- **Cancel Request.** Cancels a server request that is currently waiting to run. For more information, see *Cancel Request Business Service Method Arguments*.

### Submit Request Business Service Method Arguments

The following table describes the arguments for the Submit Request business service method of the Server Requests Business Service.

| Argument | Description |
|---|---|
| Component | Enter the name of the server component to run. Required if you do not specify the Component Job. |
| Component Job | Enter the name of the component job to run. Required if you do not specify the Component. |
| Delete After | (Optional) Specifies the number of iterations before deleting the request. Works with Delete After Units. The default value is 0 (zero). |
| Delete After Units | (Optional). Specifies the units to measure the iterations for the Delete After argument. The default value is:<br>• NoReq for synchronous mode where Siebel CRM does not save the request to the Siebel database<br>• Eon for asynchronous mode where Siebel CRM never deletes the request.<br>You can also use one of the following values:<br>• ASAP<br>• SECONDS<br>• MINUTES<br>• HOURS<br>• DAYS<br>• WEEKS<br>• MONTHS<br>• YEARS |
| Description | (Optional) Includes a description of the Siebel Server request. |
| Enterprise Server | N/A |
| Hold Flag | (Optional) Specifies the argument that indicates whether or not to hold the request. Required if you do not specify the Component. |
| Maximum Execution Time | For future use. |
| Method | (Optional) Specifies the business service method that Siebel CRM must call. You use this argument only with a server component that provides a service. For example, Workflow Process Manager or Communications Manager. |
| Mode of Server Request | (Required) Instructs the Server Request Broker how to handle the Siebel Server request. While in Auto mode, the Server Request Broker sets the mode to Synchronous or Schedule, depending on if the Siebel client is connected or mobile. You can use one of the following values:<br>• **DirectDb.** Asynchronous request, written directly to the S_SRM_REQUEST table. DirectDb is case-sensitive. If you use a case other than DirectDb, such as DirectDB, then Siebel CRM might create an error.<br>It is recommended that you use DirectDb to call an asynchronous request. If you use DirectDb, then Siebel CRM does not lose the data in the S_SRM_REQUEST table. If you use Async, and if the Workflow Process Manager reaches MaxTasks, then it cannot process every SRM request. |

| Argument | Description |
|---|---|
| | • **Async.** Asynchronous.<br>• **Sync.** Synchronous.<br>• **Schedule.** Schedule.<br>• **Auto.** Automatic configuration. |
| Number of Retries | Specifies the maximum number of times Siebel CRM retries the request in case of error. |
| Request ID Needed | (Optional) You use this argument only with asynchronous mode or schedule mode. If you set Request ID Needed to false, then Siebel CRM returns server requests more quickly. |
| Repeat Amount | Specifies the amount of time that Siebel CRM repeats units. |
| Repeat Interval | (Optional) Specifies the interval that Siebel CRM uses to repeat a request. |
| Repeat From | (Optional) You can use one of the following values:<br>• Scheduled Start<br>• Actual Start<br>• End |
| Repeat Interval Units | (Optional) Specifies the unit of intervals that Siebel CRM uses for a repeating request. |
| Repeat Number | (Optional) Specifies the number of repetitions that Siebel CRM uses for a repeating request. |
| Retry On Error | Used for resending the request if the request errors out. The default value is FALSE. |
| Repeat Unit | Specifies the unit of time that Siebel CRM uses to repeat. |
| Request ID | Used for request key routing. |
| Server Name | (Optional) Enter the server name where Siebel CRM runs this request. |
| Start Date | (Optional) Specifies the start date and time. |
| Storage Amount | (Optional) Enter the amount of time that Siebel CRM stores the Siebel Server request in the Siebel database if the Siebel Server is down. |
| Storage Units | (Optional) Enter the units that Siebel CRM uses to measure the iterations for the Storage Amount argument. The units are the same as Delete After Units. |
| Component | Enter the name of the server component to run. Required if you do not specify the Component Job. |

### Cancel Request Business Service Method Arguments

The following table describes the arguments for the Cancel Request business service method of the Server Requests Business Service.

| Argument | Description |
|---|---|
| Request ID | (Required) The Id of the Siebel Server request that Siebel CRM must cancel. |
| Repeat Number | (Optional) The number of repetitions of the repeating server requests that Siebel CRM must cancel. |

## Workflow User Event Service Business Service

Siebel CRM uses the Workflow User Event Service business service for one way communication from the Siebel client to the server component for the Workflow Process Manager. It sends notification that a user event occurred. Siebel CRM can call the Workflow User Event Service business service to create a user event in the Siebel Enterprise. This business service includes the GenerateEvent method. For more information, see *Configuring a User Event*.

The following table describes the arguments of the GenerateEvent method.

| Argument | Description |
|---|---|
| UserEventName | The name of a user event is an agreement between the creator, which is an external entity, and the recipient, which is the Workflow Process. It includes no special significance. To successfully communicate with each other, the incoming event name and the Workflow Process instance definition must use exactly the same user event name. The user event name must be unique.<br><br>It is recommended that you name the event so that it reflects the business purpose that it serves. For example:<br><br>`Event Transferring Send Order Confirmation from Vitira To Siebel V2` |
| CorrelationValue | Uses business data, such as an order number, to match an incoming message with a Workflow Process instance. If a Workflow Process communicates with an external entity, then the external entity might not be aware it is communicating with a Workflow Process. In this situation, it is difficult for the external entity to use a Siebel identifier to identify the recipient, such as the instance Id of the Workflow Process. It is more convenient to use business data to identify the recipient, such as an order number. The correlator serves this purpose. Correlation applies to a user event that reaches a long-running Workflow Process, which can define a process property as a correlator.<br><br>Siebel CRM can use only one process property as a correlator. |
| *Value* (Payload) | If Siebel CRM creates the user event, then it can use data as payload. It delivers this data to the Workflow Process instance that receives the event. If you configure this Workflow Process to wait for a user event, then a process property can receive this payload data. The payload is a single value. Siebel CRM can pass only one payload.<br><br>If you must configure Siebel CRM to send complex or structured data, then it is recommended that you use the XML converter to convert the data into an XML document, and then pass the resulting XML string as the payload of the event. The receiving Workflow Process can then call the XML converter again to recover the original data structure. |

| Argument | Description |
|---|---|
| | |

# Workflow Utilities Business Service

The Workflow Utilities business service includes generic utilities that you can use in a test environment.

## Methods

The following table describes the methods of the Workflow Utilities business service.

| Method Name | Description |
|---|---|
| Sleep | Sleeps for the number of seconds defined by *value*. |
| PropSetToText | Converts a hierarchical input property set to a single string. |
| TextToPropSet | Converts a single string to a hierarchical output property set. |
| DynamicDispatch | Calls a service. |
| Echo | Sends inputs to outputs. |

## Echo Method

The Echo method of the Workflow Utilities business service returns a mirror image of the input arguments. Echo copies the echo inputs to the echo outputs. For example, you can use Echo to create or format the body of an email message before calling the SendMessage method of the Outbound Communications Manager.

The following table describes the arguments of the Echo method of the Workflow Utilities business service.

| Argument | Description |
|---|---|
| Input Arguments | This method accepts input arguments. |
| Output Arguments | An exact copy of the input arguments. |

## Using Output Parameters with the Echo Method

When using the Echo Method of the Workflow Utilities business service, you define the output parameters that Siebel CRM updates with values from the active record of the business component that the Workflow Process modifies. For example, assume Siebel CRM must get the value for the Price List Id field from the active record in the Account business component. You can use the Echo method with no input argument and one output argument.

The following table describes an example output argument that you can use with the Echo Method to get a value. When the Workflow Process in this example runs, Siebel updates the Echo Variable process property with the value in the Price List Id field of the active record for the Account business component.

| Property | Value |
|---|---|
| Property Name | Echo Variable |
| Type | Business Component |
| Business Component Name | Account |
| Business Component Field | Price List Id |

# Workflow Admin Service Business Service

The Workflow Admin Service business service allows a Workflow Process to do administrative work on multiple Workflow Processes that a search specification defines. Example administrative work includes Activate, Deploy, Export and Import (Workflow Processes).

The following table describes the methods, including the corresponding method arguments, of the Workflow Admin Service business service.

**Note:** These methods do not apply to the Siebel client.

**Note:** The ability to import a Workflow Process from a non Workspace-enabled environment to a Workspace-enabled environment is currently not supported.

| Method Name | Argument Type | Argument Name | Description |
|---|---|---|---|
| Activate | Input | FlowSearchSpec | The search specification that Siebel CRM uses to identify the Workflow Processes that it must activate. It adds the following default search specification to FlowSearchSpec:<br>`[Status] = "COMPLETED"`<br>It is not necessary to explicitly define the search specification. |
| Activate | Output | NumFlowActivated | The number of Workflow Processes that Siebel CRM activated. |
| DeleteDeploy | Input | FlowSearchSpec | The search specification that Siebel CRM uses to identify the Workflow Process deployment records that it must delete. |
| DeleteDeploy | Output | NumFlowDeleted | The number of Workflow Process deployment records that Siebel CRM deleted. |

| Method Name | Argument Type | Argument Name | Description |
|---|---|---|---|
| | | | |
| Deploy | Input | FlowSearchSpec | The search specification that Siebel CRM uses to identify the Workflow Processes that it must deploy. It adds the following default search specification to FlowSearchSpec:<br>`[Status] = "In Progress"`<br>It is not necessary to explicitly define the search specification. |
| Deploy | Output | NumFlowDeployed | The number of Workflow Processes that Siebel CRM deployed. |
| Export | Input | ExportDir | The directory to which Siebel CRM writes the export files. For example:<br>`D:\workflows` |
| Export | Input | FlowSearchSpec | The search specification that Siebel CRM uses to identify the Workflow Processes that it exports. For example:<br>`[Process Name] like 'User Reg*'` |
| Export | Input | Repository | The repository from which Siebel CRM exports Workflow Processes. The default value is Siebel Repository. |
| Export | Output | NumFlowExported | The number of Workflow Processes that Siebel CRM exports. |
| Import | Input | ImportDir | The directory where the Workflow Process export files are located. |
| Import | Input | FileSearchSpec | The search specification that Siebel CRM uses to identify the export files for Workflow Processes that it must import. For example:<br>`User*.xml` |
| Import | Input | Repository | The repository from which Siebel CRM imports Workflow Processes. |
| Import | Input | ProjectName | The project from which Siebel CRM imports Workflow Processes. The project must be locked. |
| Import | Output | NumFlowImported | The number of Workflow Processes that Siebel CRM imported. |

# Other Business Services That a Workflow Process Uses

Other business services that are sometimes used with a Workflow Process include the following:

- *Synchronous Assignment Manager Requests Business Service*

- **Outbound Communications Manager Business Service.** For more information, see *Examples That Use the Outbound Communications Manager Business Service*.
- **Report Business Service.** The Report business service automates the work that is required to send, schedule, print, save, or email a report. It also automates administrative jobs, such as synchronizing a new user. For more information, see *Siebel Reports Guide* .
- **FINS Data Transfer Utilities Business Service.** The FINS Data Transfer Utilities business service allows you to transfer data from a source business component to a destination business component without using a script. For more information, see *Siebel Finance Guide* .
- **Data Validation Manager Business Service.** The Data Validation Manager business service allows you to validate data according to a predefined rule. You develop it through Application Administration, not through a script. The Data Validation Manager business service supports custom messages.
- **Dynamic UI Business Service.** The Dynamic UI business service and associated administration views allow you to define and render a view that contains a single, read only applet in the Siebel Financial Services application. For more information, see *Siebel Finance Guide* .

## Synchronous Assignment Manager Requests Business Service

The Synchronous Assignment Manager Requests business service can use Assignment Manager rules to assign an object. This service includes one method, named Assign. The Assign method sends a request to the assignment manager server component. For more information, see *Siebel Assignment Manager Administration Guide* .

### Arguments

The following table describes the arguments for the Assign method of the Synchronous Assignment Manager Requests Business Service.

| Argument | Description |
|---|---|
| Assignment Object Name | (Required) The object that Siebel CRM must assign. |
| Object Row ID | (Required) The Row Id of the object that Siebel CRM must assign. To assign the work item for the Workflow Process, you can set this argument to the Object Id process property. |
| Reply | An output argument of this method. |

### Avoiding Errors Due to Locked Records

The Synchronous Assignment Manager Requests business service attempts to assign records that meet the appropriate criteria, even if they are locked. To prevent an error in a Workflow Process that a locked record causes, you can create a condition in the Workflow Process or Workflow Policy that skips a record that does not meet the following condition:

# 19 Glossary

## 7.0 Flow

A Workflow Process that provides backward compatibility for a Workflow Process that is defined prior to version 7.7. Set in the Workflow Mode property on the Workflow Process.

## Arguments

Data that Siebel CRM passes to or receives from a Workflow Process or a Workflow Process step.

## Branch

A possible outcome of a Workflow Process step. A step follows a branch in a Workflow Process. If the branch conditions are met, then work flows to the next step that is downstream of the branch.

## Branch Connector

A connector that emanates from a Start step, Decision step, Wait step, or User Interact step. You implement conditional logic on the branch connector.

## Business Object

A group of one or more business components. A business object represents an entity in Siebel CRM that you must monitor. A Workflow Process references only one business object.

## Business Process

A process that is associated with operational objectives and business relationships. A business process is a set of one or more linked procedures, which collectively realize a business objective. An example of a business process is managing a new service request.

## Business Rule

The definition of how an organization must carry out a process in operations for the organization.

# Business Service

A type of Workflow Process step in which Siebel CRM makes a call to a business service, such as the Outbound Communications service that handles inbound and outbound messaging. A Workflow Process can include one or more business service steps.

# Connector

A definition of the relationship between two Workflow Process steps.

# Decision Condition

The principles that Siebel CRM uses to evaluate the logical flow that it must take on a branch in a Workflow Process. You define a decision condition on a connector that emanates from a decision step.

# Decision Point

A type of Workflow Process step in which the work item branches off to different steps, depending on a set of conditions. A decision step consists of possible branches for that point in the business process. Each branch consists of one or more conditions that must be met in order for a work item to follow that branch. A Workflow Process can include one or more decision steps.

# End Step

A type of Workflow Process step that specifies when a process instance is finished.

# Error Exception

A type of Workflow Process step that runs in reply to a deviation from normal processing. An error exception can be an error that Siebel CRM creates or that the user creates.

# Expression Business Component

The name of the business component to evaluate an expression that Siebel CRM uses as part the input argument for a search specification.

# Filter Business Component

The name of the business component that provides the group of records on which Siebel CRM performs a search in reply to a Siebel operation step. The filter business component is part of the input argument for a search specification.

# Input Argument

A mechanism for providing data and configuration information as input to a Workflow Process step.

# Interactive Workflow Process

A Workflow Process that assists the user in navigating across Siebel views. You can set the Workflow Mode property on the Workflow Process.

# Long-running Workflow Process

A persistent Workflow Process that can last for hours, days, or months. You can set the Workflow Mode property on the Workflow Process.

# Multi Value Property Window

An applet in the Workflow Process Editor that allows you to define multiple Process Properties for a Workflow Process and Input and Output Arguments for a Workflow Process step.

# Object Type

An entity that appears in the Object Explorer. For example, a Workflow Process is an example of an object type.

# Output Argument

A mechanism for providing data and configuration information as output from the step of a Workflow Process.

# Policy Action

An event that Siebel CRM runs when policy conditions are true and properties of the Workflow Policy are satisfied. A policy action references programs. You can define a policy action in the Action applet of the Workflow Policies view. After you define a policy action, Siebel CRM can use it in a Workflow Policy.

# Primary Business Component

The business component that the Primary Business Component property of the business object references. The Business Object property of the Workflow Process references this business object.

# Process Property

A storage field in a Workflow Process that Siebel CRM uses to hold values for use in steps as input and output arguments or for performing evaluations.

# Process Simulator

A graphical flowchart interface that you can use to debug a Workflow Process.

# Search Spec Input Argument

A type of input argument that allows you to restrict the records that Siebel CRM considers when a Siebel operation step runs.

# Service Workflow Process

A Workflow Process that runs a set of operations when Siebel CRM calls an event. You can set the Workflow Mode property on the Workflow Process.

# Siebel Operation

A type of Workflow Process step that performs a database operation on a business component record or business component field. Example operations include insert, query, and update.

# Start Step

A type of Workflow Process step that defines the conditions to start a Workflow Process instance. If the conditions are met, then Siebel CRM starts an instance of the process. A Workflow Process includes only one start step.

# Step Branch

An object that provides capabilities for branching logic in a Workflow Process.

# Step Instance

An instance of the step of a Workflow Process that is started. Siebel CRM starts a start step if the conditions defined for the start step are met. It starts a decision step if the conditions for a branch connector are met. It starts other steps if the previous step finishes.

# Step Recipient

The intended recipient of a sub process step.

# Stop Step

A type of Workflow Process step that specifies the conditions that cause a Workflow Process instance to terminate prior to completion.

# Sub Process

A Workflow Process that another Workflow Process calls. A sub process includes a separate Workflow Process object definition.

# Sub Process Step

A type of Workflow Process step that calls a sub process. A Workflow Process can include one or more sub process steps.

# Task

A type of Workflow Process step that calls a task in a business process. A task step is similar to a sub process step in that the task step represents a container for a further set of steps below the level of the steps in the Workflow Process in which the task step is defined.

# User Interact Step

A type of Workflow Process step that controls the flow of views in the Siebel client. A Workflow Process can include one or more user interact steps. These steps can guide a user through a defined flow of views depending on user action, or can run a defined set of actions.

# Wait Step

A type of Workflow Process step that specifies when the instance of the process pauses and the duration of the pause.

# Watch Window

**Note:** In Web Tools, the Watch Window is called Workflow Process Data.

A window that dynamically displays the record values of a business component and process property values of a Workflow Process. The Workflow Process being simulated uses and modifies these values.

# Workflow Mode

A property of a Workflow Processs that Siebel CRM uses to control runtime behavior for a Workflow Process as a Service Flow, Interactive Flow, Long Running Flow, or 7.0 Flow.

# Workflow Policy

A systematic expression of a rule. A Workflow Policy includes one or more Workflow Policy conditions and one or more Workflow Policy actions. If the Workflow Policy conditions for a Workflow Policy are true, then Siebel CRM runs the Workflow Policy action. A Workflow Policy is contained by one Workflow Policy group and is related to one Workflow Policy object.

# Workflow Policy Column

A column in a table in the Siebel database that you must monitor. You use a Workflow Policy column when you define aWorkflow Policy condition for a Workflow Policy. To use a Workflow Policy column in a Workflow Policy, you must associate it with a Workflow Policy component. A Workflow Policy column that is associated with a Workflow Policy component is known as a Workflow Policy component column.

# Workflow Policy Component

A component that defines the Siebel database tables that Siebel CRM must monitor. A Workflow Policy component also defines the relationship between tables. It includes Workflow Policy columns.

# Workflow Policy Component Column

A Workflow Policy column that is associated with a Workflow Policy component. A Workflow Policy component column defines the Siebel database column that a Workflow Policy condition of a Workflow Policy can use.

# Workflow Policy Condition

An expression that Siebel CRM compares to data in the Siebel database. The result of the comparison is true or false. To define a Workflow Policy condition, you define a Workflow Policy column and a comparison operator, and you then enter or choose a value, if appropriate. You can define a Workflow Policy condition in the Workflow Policies view.

# Workflow Policy Group

A group of one or more Workflow Policies. A Workflow Policy group allows you to group Workflow Policies that share common behavior. Siebel Server processes monitor Workflow Policy groups. For example, Workflow Policies that Siebel CRM must monitor hourly are different in a Workflow Policy from those that it must monitor weekly. You can define a Workflow Policy group in the Policy Groups view.

# Workflow Policy Object

A group of one or more Workflow Policy components. A Workflow Policy object represents an entity in Siebel CRM that you must monitor. A Workflow Policy references only one Workflow Policy object.

# Workflow Process

Oracle's Siebel Workflow representation of a business process. A Workflow Process includes one or more steps that indicate when a business process starts and ends. It includes information about individual work that is performed in the business process.

# Workflow Process Instance

A Workflow Process instance that is started. If the input conditions for the Workflow Process are met, then Siebel CRM starts a Workflow Process instance. A Workflow Process includes one or more step instances and one or more work items.

# Workflow Process Program

The definition of an event. Types of events include Send Email, Send Page, Database Operation, Send Broadcast Message, and Run External Program. Different properties are associated with a Workflow Process program depending on the event type. Some of the properties that you can define for a program include the fields that Siebel CRM can substitute into a message, the possible recipients of a message, and the Siebel database columns that it must update.

# Workflow Process Step

An activity in a Workflow Process. Steps are logically linked together to define a process.

# Work Item

The representation of the work that Siebel CRM processes in the context of a step in the Workflow Process instance.