Siebel Business Process Designer Administration Guide

Version 7.7, Rev. A

June 2004

Siebel Systems, Inc., 2207 Bridgepointe Parkway, San Mateo, CA 94404

Printed in the United States of America

PRODUCT MODULES AND OPTIONS. This guide contains descriptions of modules that are optional and for which you may not have purchased a license. Siebel's Sample Database also includes data related to these optional modules. As a result, your software implementation may differ from descriptions in this guide. To find out more about the modules your organization has purchased, see your corporate purchasing agent or your Siebel sales representative.

Proprietary Information

Siebel Systems, Inc. considers information included in this documentation and in Siebel eBusiness Applications Online Help to be Confidential Information. Your access to and use of this Confidential Information are subject to the terms and conditions of: (1) the applicable Siebel Systems software license agreement, which has been executed and with which you agree to comply; and (2) the proprietary and restricted rights notices included in this documentation.

Contents

Chapter 5: For Developers: Basics of Building Workflow Processes

Chapter 6: For Developers: Workflow Process Steps

Chapter 8: For Developers: Testing Workflow Processes

Chapter 9: For Administrators: Administering Workflow Processes

Chapter 10: Workflow Policies

Index

1 What’s New in This Release

What’s New in Siebel Business Process Designer Administration Guide, Version 7.7, Rev. A

Table 1 lists changes in this version of the documentation to support release 7.7 of the software.

Table 1. What’s New in Siebel Business Process Designer Administration Guide, Version 7.7, Rev. A

Topic	Description
“Run-Time Architecture of Workflow” on page 32	Revised this topic by adding subsections titled Administration and Monitoring, Recovery.
“Siebel Tools and Workflow Processes” on page 44	Added this topic.
“Process Designer Palette Items” on page 51	Revised this topic by adding a graphic and modifying descriptions in Table 5.
“Modifying Existing Process Definitions” on page 55	Revised this topic by adding a sequence of steps for this task and modifying the surrounding text.
“Tutorial: Using Process Designer in Siebel Tools” on page 56	Added this topic.
“About Process Properties” on page 69	In “Object ID and Non-7.0 Workflow Processes,” added a note explaining the need to update the Object ID process property when you change the active row in a step of a workflow.
“Field Descriptions: WF Step Branch Applet” on page 75	Added a note to the Description for the Event Cancel Flag field in Table 7.
“Field Descriptions: Compose Condition Criteria Dialog Box” on page 79	Revised this topic by adding a graphic and a note.
“Defining Conditions and Values for Branches, Decision Steps, and User Interact Steps” on page 82	Added a note to this topic.
“Field Descriptions: WF Step Recipients Applet” on page 92	Revised this topic by adding field descriptions to Table 11.
“Field Descriptions: Subprocess Applet” on page 93	Added a note to the Possible Value for the Subprocess field in Table 12.
“Defining a Stop Step” on page 103	Added a note to this topic.
“For Developers: Understanding How Workflow Processes Are Designed” on page 107	Added this chapter, which contains content that was previously in other chapters.

Table 1. What’s New in Siebel Business Process Designer Administration Guide, Version 7.7, Rev. A

Topic	Description
“Generating User Events with the User Event Business Service” on page 121	Revised this topic and added a note to the task.
“Configuring Long-Running Workflow Processes to Wait for User Events” on page 122	Added this topic.
“Using the Validate Tool to Correct Errors in Workflow Processes” on page 139	Revised this topic by adding a list of possible errors and fixes.
“Activating Workflow Processes” on page 149	Added this topic.
“Siebel Application Response Management (Siebel ARM)” on page 155	Added this topic.
“Siebel Flight Data Recorder (FDR) Files” on page 156	Added this topic.
“Defining Workflow Policy Actions” on page 164	Added a note to this topic.
“About the Recipients Applet” on page 171	Added a note regarding Recipient Name in Table 41.
“About the Conditions Applet” on page 183	Revised a note in this topic.
“Passing Parameters to and from Workflow with the Workflow Process Manager Business Service” on page 261	Revised this topic by adding information on passing inputs to Workflow, and passing outputs from Workflow.

What’s New in Siebel Business Process Designer Administration Guide, Version 7.7

Table 2 lists changes described in this version of the documentation to support release 7.7 of the software.

Table 2. New Product Features in Siebel Business Process Designer Administration Guide, Version 7.7

Topic	Description
Workflow Process Designer See “Using Process Designer in Siebel Tools” on page 46.	The Workflow Process Designer has been moved into Siebel Tools. The Process Designer in the run-time client has been made read-only to facilitate upgrade of pre-7.7 workflow processes. In the next release, the Process Designer will be removed from the run-time client. The Workflow Process object type is a new top-level object type in Siebel Tools.
Workflow Process Simulator See “Testing Workflow Processes with the Process Simulator” on page 137.	The Workflow Process Simulator has been removed from the run-time client and is now hosted in Siebel Tools.
End-to-end business process framework See “About Workflow Persistence” on page 117 and “About Workflow Process Monitoring” on page 151.	Using enhanced persistence and monitoring, you can create a single workflow process to handle an entire business process transaction and coordinate between multiple short and long-running subprocesses. For example, rather than use short workflow processes that trigger every time a service request is updated and which together over time accomplish a larger business process, you can maintain the state in the service request record, and create a longer workflow with wait steps and persistence. This provides a comprehensive view from which you can maintain the workflow state and monitor the progress of the workflow at the instance level.
New deployment scheme See “Deploying Workflow Processes” on page 144.	This release includes a new deployment scheme that is used to move Workflow objects from the repository to the run-time environment. This does not require recompiling the SRF.
Workflow modes See “About Workflow Processing Modes” on page 107.	Workflow processes are now categorized into four different modes, each offering a specific set of capabilities and a specific quality of service: ■ 7.0 Flow ■ Long-Running Flow ■ Interactive Flow ■ Service Flow

Table 2. New Product Features in Siebel Business Process Designer Administration Guide, Version 7.7

Topic	Description
Interactive workflow processes See “About Interactive Workflow Processes” on page 108.	You can configure your workflow processes to allow parking of interactive workflow processes in the user's Inbox. (Inbox is a single screen in Siebel eBusiness applications that shows all approval and notification items assigned end users regardless of the screen where the item originated.) You can configure free-flow navigation through the use of synthetic events, allowing the user to navigate backward and forward between views and to suspend and resume a process.
User Interact step See “About User Interact Steps” on page 100.	The User Interact step has been enhanced to take process properties as input arguments. In this way, you can dynamically set view names as you design your interactive workflow processes.
Workflow User Event business service See “About the Workflow User Event Business Service” on page 120.	A high-level event mechanism has been introduced for use by Siebel Workflow. This allows interaction as well as data transfer into workflow processes and instances. User events can be generated anywhere in the Siebel enterprise (wherever a Siebel business service is used) by calling the Workflow User Event business service. The event model also introduces the capability of correlation to ease communication with external entities.
Long-Running workflow processes See “About Long-Running Workflow Processes” on page 108.	You can now define collaborative workflows (such as approval processes) that route tasks across users. Long-running workflow processes use user events.
Workflow persistence See “About Workflow Persistence” on page 117.	Workflow persistence is now a quality of service and is not related to workflow monitoring. Workflow persistence is available to all workflow modes (as a configurable option) and is automatically enabled for long-running workflows. The 7.0 workflow persistence level and frequency settings are now replaced with a single flag: the Auto Persist flag. Monitoring of workflow processes is now a separate feature.
Administration of workflow processes See “Administering Workflow Processes in the Run-Time Client” on page 148.	There is now increased accountability of executed processes using the new Workflow Admin view. The Workflow Admin view allows the administrator to see the persisted status of workflow processes and provides recovery features.

Table 2. New Product Features in Siebel Business Process Designer Administration Guide, Version 7.7

Topic	Description
Monitoring of workflow processes See “About Workflow Process Monitoring” on page 151.	There is now extended process monitoring, including enhanced error notification and execution tracing, using the new Workflow Instance Monitor view. The Workflow Instance Monitor view allows monitoring of all workflows (regardless of persistence setting). Monitoring level is a deployment parameter, not a design-time parameter.
Recovery of process instances See “Recovering Workflow Processes” on page 127.	Workflow processes marked as recoverable can be automatically and manually recovered after server failures.
Palette Designer edit features See “Using Process Designer in Siebel Tools” on page 46.	You can copy and paste Palette Designer shapes within a process. The Palette Designer also features: ■ Properties window and Object List editor to enter process properties as well as input and output arguments ■ Pop-up dialog box to enter conditions ■ Zoom, copy drawing, print ■ Show and hide labels and exception branches ■ Define shape colors, lines, fill colors, fonts and sizes
Seeded workflow processes See “Upgrading Siebel Workflow” on page 42**.**	All seeded workflow processes have been moved from run-time tables to repository tables.

2 Overview of Siebel Business Process Designer

This chapter provides a conceptual overview of Siebel Business Process Designer, also referred to as Siebel Workflow: a customizable business application that allows you to define, manage, and enforce your business processes.

Using Siebel Business Process Designer to manage business processes in your organization, you can address such challenges as:

- Automating escalation of events and notification of appropriate parties
- Routing and assigning work
- Processing work
- Enforcing authorization and transition rules

This chapter is organized as follows:

Siebel Business Process Designer

Siebel Business Process Designer is the configuration interface and the administrative interface for Siebel Workflow. These interfaces bring together Workflow Processes and all other repository configuration objects, including the Workflow Policies module, for creating a comprehensive workflow design.

- **Workflow Processes.** Allows you to define your company's business processes using a familiar flowcharting interface. A workflow process consists of one or more process steps such as start steps, subprocesses, decision points, and tasks.
- **Workflow Policies.** Allows you to define policies that can act as triggers to execute a process. A policy consists of conditions and actions. When policy conditions are met, the policy action executes the relevant process.

General Principles of Workflow

In theory, businesses are managed according to policies and procedures that allow efficiency, quality service, adherence to contractual agreements, and profitability. These policies enforce business processes such as:

- Allowing that response time objectives are met for customer callbacks and open service requests

- Specifying review policies for important processes like contracts, quotes, or product shipments
- Monitoring service requests or opportunities over time

In practice, the benefits of policies often are not realized because policies are not consistently enforced. This may be because of the large number of processes or because of the dynamic nature of the information being monitored.

The management of important events is central to the enforcement of business workflow. *Workflow* is the timely management of an event to allow proper handling. For example, service departments have procedures for managing an open service request or making sure that response times are met. A workflow can increase the visibility of these processes within an organization and check that they are correctly handled.

Service departments have sets of defined rules that match their policies and service agreements such as:

- **Standards for processing calls.** For example, when a Severity 1 call is assigned, the new owner is automatically paged.
- **Contracted service agreements that must be adhered to.** For example, customers may purchase a support agreement guaranteeing a callback in two hours and problem resolution in four hours.

Sales departments also have rules to enforce desired business practices, such as:

- **Discount authority.** If a sales representative quotes a discount exceeding the maximum discount allowed, it requires the approval of the district sales manager or VP of Sales.
- **Pipeline management.** Each sales representative manages his or her pipeline to ensure sufficient levels of prospects at each stage of the sales cycle. If an area of the pipeline needs attention, the representative or manager should be alerted.
- **Forecasting accuracy.** Opportunities that are forecasted but never closed or forecasts having wide discrepancies with the actual revenue need to be flagged.

Understanding the Workflow Processes Module

Workflow Processes is the module in Siebel Business Process Designer that you use to create and administer workflow processes.

Workflow Processes Configuration Overview

Workflow Processes allows you to define your company's business processes using the Process Designer in Siebel Tools. Using Workflow Processes, you can define a process that consists of process steps such as start steps, decision points, subprocesses, or business services steps to complete tasks.

A task can be completed with either a predefined business service or a custom business service. Predefined tasks include updates to the Siebel database, notifications (such as an email or page), integration messages to external systems, and calls to invoke server tasks. Custom tasks can be defined by using Siebel VB or Siebel eScript.

Workflow Processes Administration Overview

Workflow Processes can vary from a simple process such as entering a product order to a complex process such as managing call center workflow. Complex processes can comprise multiple smaller processes.

Workflow Processes are administered through the Administration - Business Process views in the Siebel Client. Instructions for accessing and using the Administration - Business Process views are in Chapter 9, "For Administrators: Administering Workflow Processes."

Using Workflow Processes

Workflow processes can be invoked from events in the Siebel application or from external systems. Within the Siebel application, a process can be invoked from a workflow policy, an event (such as an insert of a record or a button click), or a server component.

From an external system, processes can be invoked using COM or CORBA. For information on invoking a workflow process, see "About Invoking a Workflow Process" on page 129.

Sample Workflow Process Scenario

To help you understand how the Workflow Processes module works, see the usage scenario "Scenario: New Service Request" on page 21. You can view sample workflow processes in detail by selecting the project Workflow - Samples in Siebel Tools.

Scenario: New Service Request

Prior to implementing Siebel Call Center, ABC Computing found itself unable to resolve many customer issues in a timely manner. To better track and manage service requests, ABC implements the Service Request module and automates the company's service-request management process.

The goal is to meet a service-level agreement (SLA) commitment by making sure that all newly-logged service requests (SRs) are resolved within a specific amount of time. ABC Computing wants the SRs to be assigned by the system to the best representative based on availability and matching skills. If the SR needs immediate attention, the company wants to notify the owner of the SR.

This automation is achieved using Siebel Business Process Designer. When an SR is logged, a workflow process is triggered. The workflow process calls Siebel Assignment Manager to assign the SR to the best available service representative. Based on the severity of the SR, Workflow might then send email notification to the representative, using Siebel Communications Server. Automating this process helps ABC Computing achieve faster turnaround time to resolve SRs and meet the company's SLA commitment.

ABC Computing defines its business process for a new service request with the Process Designer. Figure 1 on page 22 illustrates a diagram of the process as drawn in the Process Designer.

The diagram demonstrates the steps and decision points involved when a new service request comes into the organization. The steps and decision points are displayed in the diagram in such a way that the flow of the work is clear.

NOTE: The steps explained below are not generic; they refer specifically to the workflow process illustrated in Figure 1 on page 22.

Each step is interpreted as follows:

- **Start.** This is the start step initiating the process instance. The work item is the new service request.
- **Assign Service Request.** This is a subprocess task. The service request is assigned to the appropriate agent based on the assignment rules defined in the Assign Service Request subprocess.
- **Severity.** This is a decision step. The service request severity determines the next step in the process instance of the three possible paths: Critical, High, or Medium.
- **Send Email.** This is an automated business service task. If the service request priority is critical, an email is sent to the assigned agent. This task calls the Outbound Communications Manager business service.
- **Priority High.** This is a Siebel Operation update task. This step updates the service request priority to High.
- **Substatus Assigned.** This is a Siebel Operation update task. This step updates the sub status to Assigned.
- **Email Error Activity.** This is a Siebel Operation insert task. This task is triggered if an error is returned in the Send Email task.
- **Priority Very High and Dispatch.** This is a Siebel Operation update task. This step changes the service request priority to Very High and the sub status to Dispatch.
- **End.** This step defines the completion of the process.

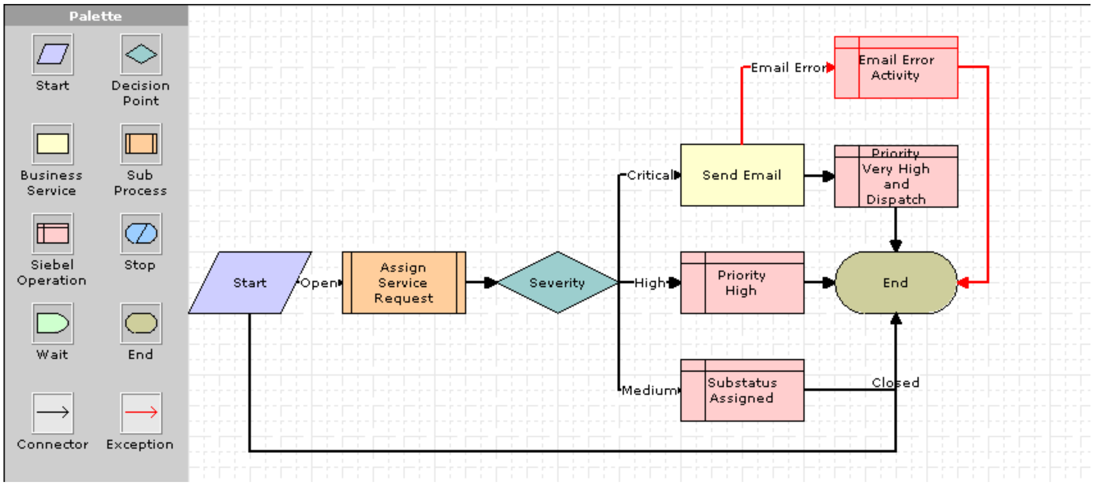


Figure 1. New Service Request Workflow Process

Understanding the Workflow Policies Module

The Workflow Policies module allows you to define policies that can act as triggers to execute a workflow process.

NOTE: The name Workflow Policies replaces the name Workflow Manager, which was used to refer to the Siebel Business Process automation tool in earlier releases.

A policy consists of one or more policy conditions. When the policy conditions are met, the policy action is executed.

NOTE: A number of the functions available with Workflow Policies can be supported using Workflow Processes. It is recommended that Workflow Policies be used to define conditions for invoking workflow processes. Use Workflow Processes for defining the actions.

Workflow Policies Structure

The basic underlying construct of Workflow Policies is the *rule*. The structure of a rule is: if all conditions are true, then an action occurs. The rule contains a policy condition and a policy action. This means when the conditions of the workflow policy are met, an action occurs.

A *workflow policy* represents the rules the database monitors. A workflow policy, based on the Workflow Policies rule structure, is composed of conditions and actions. A *workflow policy condition* is a trigger—a circumstance or situation that causes something to happen. A *workflow policy action* is an action invoked by a policy condition being fulfilled. You can also have a *duration,* which is the period of time for which all policy conditions exist for the conditions of the policy to be met.

Workflow Policy Conditions

A policy condition expresses an object/attribute relationship to a value. For example, a policy condition may target data such as Service Request Severity. The policy condition compares that data to a value, such as 1-Critical. The combination of the data element (Service Request Severity), a comparison operation (=), and the value (1-Critical) make up the policy condition.

The fact that a Service Request Severity is 1-Critical may be an issue only if the policy condition remains valid for some extended period of time, such as two hours. If this is the case, a duration can be set for two hours on the workflow policy. The duration becomes part of the policy condition. The policy actions are not executed until the policy conditions are met for the specified duration.

Policy actions can also occur when time duration is not set. For example, email is automatically sent to a sales manager each time a sales representative quotes a discount rate exceeding 25 percent on revenue less than $100,000.

Policies frequently have more than one condition. All the conditions of the policy must be met before an action can occur. A service request with a severity of 1-High and a duration of two hours may be important only if another comparison is also valid, such as the Service Request Status is Open. The policy condition becomes the combination of these two comparison operations:

```
    SR Severity = 1-Critical AND SR Status = Open
```

Siebel Workflow Policies supports only AND linkages between policy conditions, not OR linkages. If you need to monitor the SR Severity to be 1-Critical or 2-High and the SR Status is Open, you can use the IN operand to evaluate the OR of the SR Severity Condition.

```
SR Severity IN ('1-Critical', '2-High') AND SR Status = Open
```

Alternatively, OR linkages can be simulated by creating multiple policies for each key policy condition. The combination of workflow policies will act like an OR linkage. For more discussion on comparisons, see "Using Comparison Values in the Conditions Applet" on page 183.

Workflow Policy Actions

A workflow policy action contains two parts: the action and the action parameters. An *action* is a type of request, such as "Send an Urgent Page." *Action parameters* are the arguments, such as the name of the recipient of the page and the alphanumeric text transmitted with the page.

You can specify several actions for one workflow policy, such as sending a page to one person and an email to another. You can reuse actions in multiple workflow policies. See "About Customizing Workflow Policies with Siebel Tools" on page 189 for a discussion of actions and their parameters.

NOTE: In most cases, use workflow policy actions to run a workflow process.

Workflow Policy Action Program Types

Workflow policy actions are based on underlying predefined programs in Siebel Tools and inherit all the arguments of the program. Workflow policy programs can be one of the following types:

- **Send Message.** A program of this type sends an email to one or more recipients.
- **Send Page.** A program of this type sends a page to one or more recipients.
- **Send Message Broadcast.** A program of this type inserts a message broadcast for one or more recipients.
- **DB Operation.** A program of this type either inserts or updates the data records of a Siebel database table for selected workflow policy components.
- **External Program.** A program of this type allows you to run an executable.
- **Assignment Request.** For internal use only.
- **Generic Request Server.** A program of this type submits a server request to a designated server component.

NOTE: Most functionality included in workflow policy action programs can be executed using Workflow Processes.

You can use programs in multiple action definitions and you can use action definitions in multiple workflow policies. "Predefined Programs" on page 247 contains a list of the predefined programs.

Workflow Policy Groups

Workflow policies are organized into groups. A *workflow policy group* is a collection of workflow policies to facilitate load balancing on the servers. Workflow policy groups allow you to manage and optimize Workflow Agent process performance by grouping similar policies to run under one Workflow Agent process.

Workflow Policies Administration Overview

The key elements of the Workflow Policies module are workflow policy object creation in Siebel Tools, workflow policy creation in Siebel Business Process Designer, and policy execution by the Siebel Server Workflow Components.

The Workflow Policies module is administered through the Siebel Business Process Designer in the Siebel client. Instructions for accessing and using the Workflow Policies views are in "About Customizing Workflow Policies with Siebel Tools" on page 189.

Workflow Roles

The job roles associated with Siebel Business Process Designer are the following:

- The *Workflow Configurator* uses Siebel Tools to develop workflow processes and to define objects, business services, and programs.

  Your organization can use the predefined objects, business services, or programs provided in the application; or, the Workflow Configurator can define customized objects, business services, and programs in Siebel Tools.

  NOTE: Business services can also be defined in the Siebel client. For more information, see *Integration Platform Technologies: Siebel eBusiness Application Integration Volume II*.

- The *Workflow Administrator* monitors workflow processes in the Siebel client using Siebel Business Process Designer. The Workflow Administrator also activates workflow policies by generating database triggers in a script and creating them in the Siebel database. The Workflow Administrator then starts Siebel Server processes that execute workflow processes and policies. This person is typically a system administrator, database administrator, or someone from the Information Services department.
- The *End User* uses the system and executes workflow processes and policies. This person is typically an employee of your organization, and can also be a customer.

3 Introduction to Workflow Processes

Siebel Workflow is an interactive software tool that lets you automate how your organization handles workflow processes. It uses as its basic model the same processes that organizations use in their sales, marketing, and service departments that determine business workflow. You can use Siebel Workflow to promote consistency and adherence to processes through the automatic enforcement of business policies and procedures.

The Siebel Workflow product is administered through the Siebel Business Process Designer, a graphical user interface featuring a familiar flowcharting methodology for designing workflow processes.

This chapter is organized as follows:

Overview of the Workflow Architecture

Siebel Business Process Designer works with all Siebel eBusiness Applications and involves the following architectural components:

- **Siebel Tools.** Siebel Tools is an integrated environment for configuring all aspects of a Siebel application. In this release, the Workflow Process Designer resides in Siebel Tools. You use the Process Designer within Siebel Tools to build your workflow processes.
- **Siebel Client.** Workflow Processes and Workflow Policies are administered through the Administration - Business Process views in the Siebel client (Mobile Web Client).
- **Siebel Server.** Siebel Server manages the Siebel Workflow components that automate business policies.
- **Siebel Database.** A relational database containing the set of data that Workflow Policies act against.

Siebel Tools provides the design interface of Siebel Business Process Designer. The Mobile Web Client (the run-time client) provides the debugging interface of Siebel Business Process Designer. After workflow processes are designed and debugged, they are written to repository tables for deployment from the administrative interface in the run-time client.

Design-Time Architecture of Workflow

Workflow components and definitions are defined as Siebel Tools objects and are stored in the Siebel Tools repository. Before you can run a workflow process, you must first deploy the workflow process from Siebel Tools, then activate the workflow process from the Siebel Web Client.

The Workflow Process repository object is a top-level object in the Object Explorer of Siebel Tools. You use the Object List Editor (OBLE) to create Workflow processes. Workflow processes belong to a project. There is no SRF compile required for deployment of workflow processes. There is no merge required. There is independent versioning of workflow processes in Siebel Tools and in the run-time client.

Configuration data is available at design time, but run-time data is not available at design time. You use process properties to create workflow definitions, or alternatively, you can enter data through unbounded picklists.

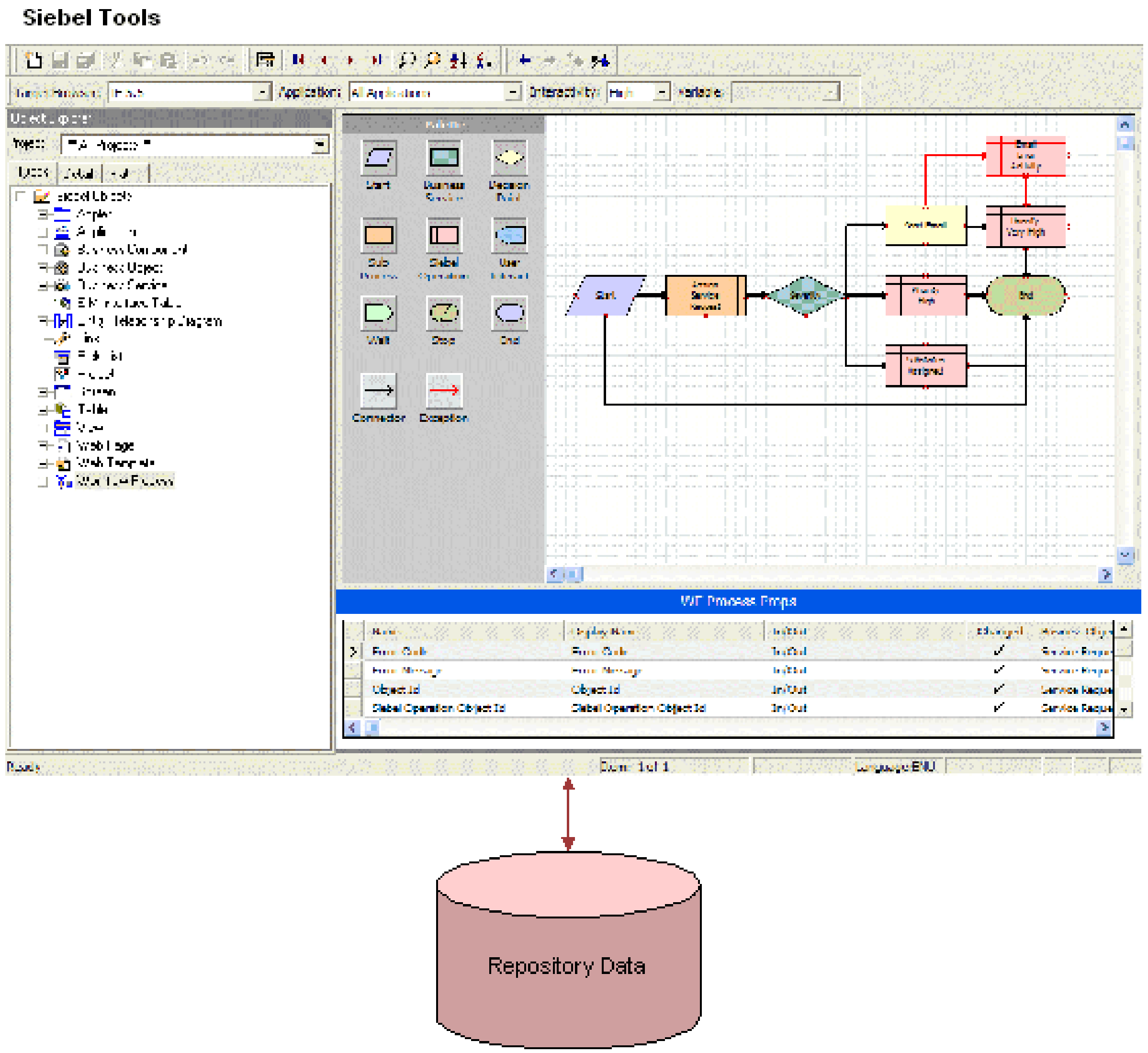
The following Siebel Tools features are not applicable to Workflow objects:

- SIF export and import
- Object Compare
- Three-way merge during upgrades

Because Siebel Tools excludes Workflow objects from these features, it is important to use the Workflow import and export feature for backing up and restoring workflow definitions. For example, if you archive a project in Siebel Tools, the Workflow objects within that project are not archived.

CAUTION: If you delete all the objects from a project expecting that they can be restored from the SIF, it is important to keep in mind that Workflow objects are an exception, and cannot be restored from the SIF. Use the Workflow import and export feature to back up and restore workflow definitions.

Figure 2 shows the design-time architecture of Workflow.



Name	Display Name	In/Out	Changed	Business Obje
Error Code	Error Code	In/Out	✓	Service Reque
Error Message	Error Message	In/Out	✓	Service Reque
Object Id	Object Id	In/Out	✓	Service Reque
Siebel Operation Object Id	Siebel Operation Object Id	In/Out	✓	Service Reque

Figure 2. Use the Process Designer in Siebel Tools to develop workflow processes. Workflow processes in development are stored in the Siebel repository.

Simulation Architecture of Workflow

After designing your workflow processes, you test them using the Process Simulator. Testing your workflow processes before migrating them to your production environment verifies that resulting actions are accurate and useful and the results are exactly what you want.

Figure 3 shows the simulation architecture of Workflow.

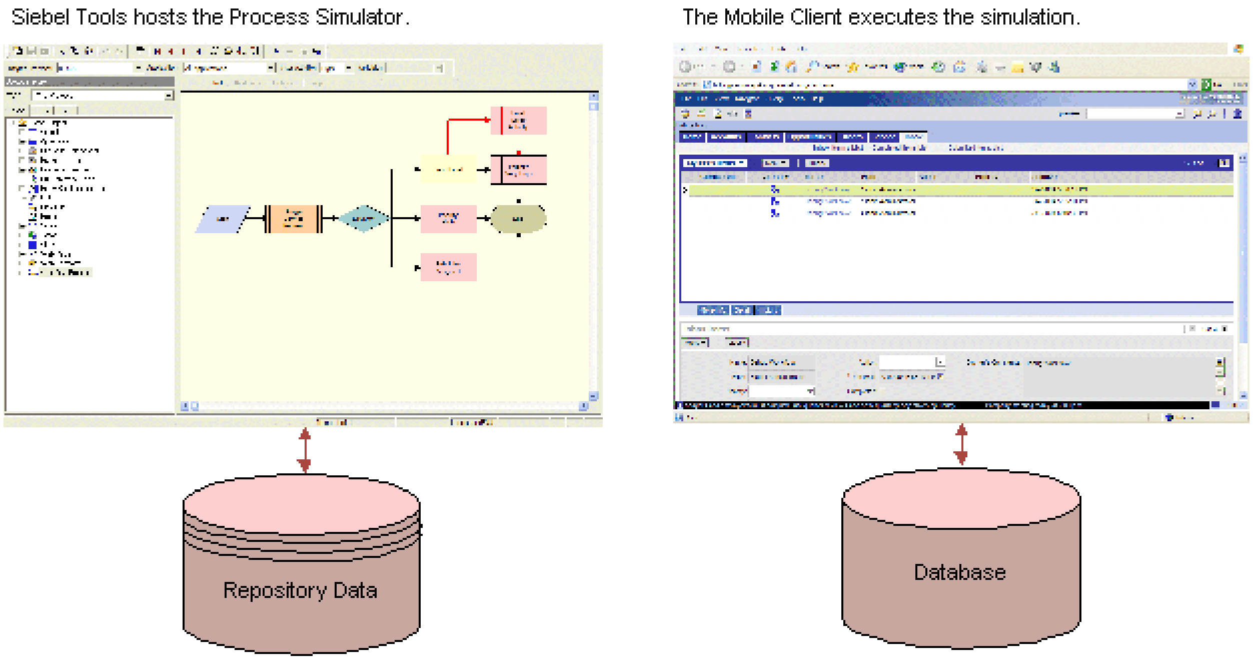

Siebel Tools hosts the Process Simulator.
The Mobile Client executes the simulation.
Repository Data
Database

Figure 3. Simulation architecture of Workflow

Deployment Architecture of Workflow

After designing your workflow processes and testing them, it is time for deployment. Figure 4 shows the relationship of Siebel Tools and the run-time client in the deploying of workflow processes.

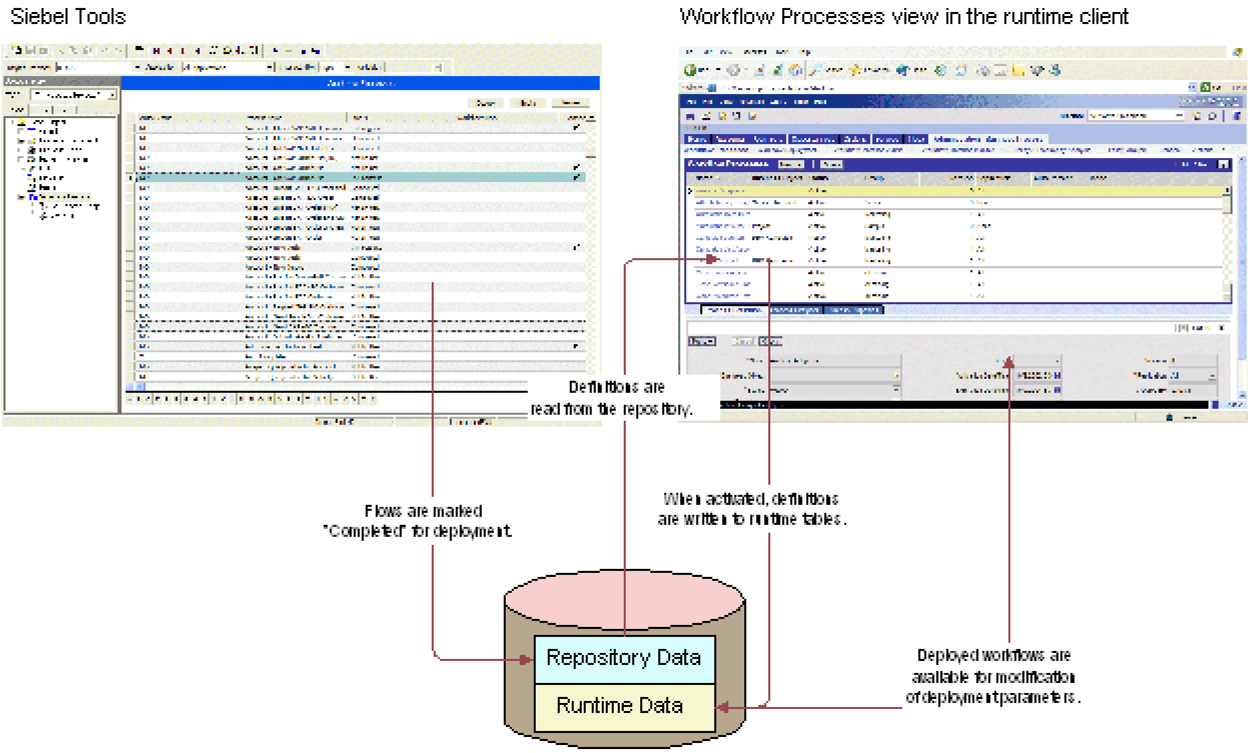

Siebel Tools
Workflow Processes view in the runtime client
Definitions are read from the repository.
Flows are marked "Completed" for deployment.
When activated, definitions are written to runtime tables.
Repository Data
Runtime Data
Deployed workflows are available for modification of deployment parameters.

Figure 4. Deployment of Workflow Processes. Workflow definitions are read from the repository. When a workflow is activated, its definitions are written to run-time tables.

Run-Time Architecture of Workflow

The Workflow run-time architecture is based on the Siebel Object Manager layer and the server infrastructure layer of the Siebel eBusiness applications architecture. The run-time environment is available both as a business service and as a server component. The run-time architecture supports three invocation modes for invoking and resuming workflow processes: Local Synchronous, Remote Synchronous, and Remote Asynchronous. Figure 5 shows the run-time architecture of Workflow.

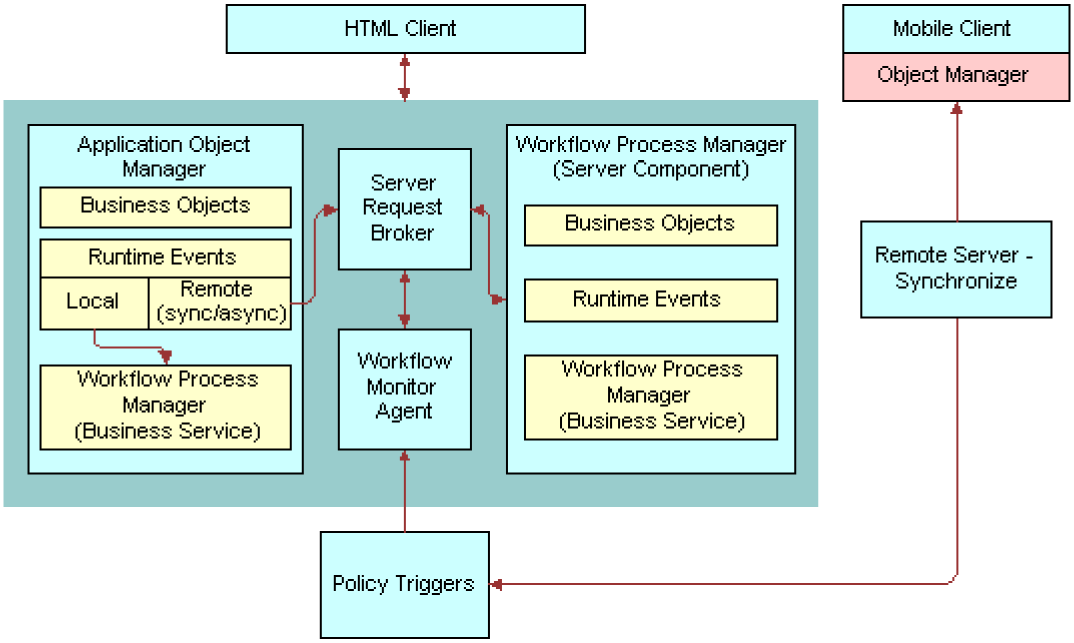

HTML Client
Mobile Client
Object Manager
Application Object Manager
Business Objects
Runtime Events
Local
Remote (sync/async)
Workflow Process Manager (Business Service)
Server Request Broker
Workflow Monitor Agent
Workflow Process Manager (Server Component)
Business Objects
Runtime Events
Workflow Process Manager (Business Service)
Remote Server - Synchronize
Policy Triggers

Figure 5. Run-time Architecture of Workflow

Workflow Process Types

Siebel Business Process Designer has four types of workflow processes that characterize run-time behavior. The processing type is set in the Workflow Processes list editor of Siebel Tools, using the Workflow Mode field. The workflow process types are as follows:

- **7.0 Flow.** A 7.0 workflow process provides backward compatibility for existing Siebel 7 (pre-7.7) workflows. For more information, see "About 7.0 Workflow Processes" on page 108.
- **Long Running Flow.** A long-running workflow process is a persistent workflow that can last for hours, days, or months. For more information, see "About Long-Running Workflow Processes" on page 108.
- **Interactive Flow.** An interactive workflow process navigates the user across Siebel views. For more information, see "About Interactive Workflow Processes" on page 108.
- **Service Flow.** A service workflow process executes a set of operations upon event invocation. For more information, see "About Service Workflow Processes" on page 109.

Workflow Process Manager

Workflow processes can be executed in any application object manager as a business service. Workflow processes can also be executed in the Workflow Process Manager server component.

Running Workflow as a Business Service

Workflow execution in an application object manager is invoked as a business service. The Workflow business services are called Workflow Process Manager. The Workflow Process Manager business services are also referred to as the Workflow engine. As a business service, the Workflow engine takes input arguments and returns output arguments.

The two Workflow business services are:

- Workflow Process Manager
- Workflow Process Manager (Server Request)

When the Workflow Process Manager business service is called, the workflow process is run in the object manager of the application called. When the Workflow Process Manager (Server Request) business service is called, the workflow process is run in the server component called Workflow Process Manager.

Running Workflow in the Workflow Process Manager Server Component

Workflow processes can be executed in the background using the Workflow Process Manager server component. The Workflow Process Manager server component is configured and optimized to run the Workflow Process Manager business service. The Workflow Process Manager server component acts as the object manager to run workflow processes, including all application logic within the workflow process.

Workflow Management Server Components

The Workflow Management server component group includes the following server components:

- Workflow Process Manager (WfProcMgr)
- Workflow Process Batch Manager (WfProcBatchMgr)
- Workflow Monitor Agent (WorkMon)
- Workflow Action Agent (WorkActn)
- Workflow Recovery Manager (WfRecvMgr)
- Generate Triggers (GenTrig)

Workflow Process Manager and Workflow Process Batch Manager

The Workflow Process Manager server components act as the application object manager to run workflows. The Workflow Process Manager server components are specialized server components configured and tuned to run workflow processes. Like all server components, the Workflow Process Manager server components provide a multi-threaded environment.

The Workflow Process Manager uses the Siebel Object Manager framework and runs workflows as a business service. The Workflow Process Manager hosts the Business Object layer and the Data Object layer. It is a scalable architecture with the ability to run multiple object managers and multiple tasks for each object manager.

NOTE: The name Workflow Process Manager refers to both the Workflow business service (referred to as the Workflow engine) and the workflow server components.

Workflow Monitor Agent

The Workflow Monitor Agent executes Workflow policies. Workflow Monitor Agent monitors policies, and executes actions once the policy conditions are met.

Workflow Action Agent

The Workflow Action Agent processes requests logged in the action request table (S_ESCL_ACTN_REQ) for a policy group and invokes all actions linked with the Workflow policy being processed.

Workflow Recovery Manager

The Workflow Recovery Manager polls the Workflow engine to check workflow instances running on the server. The Workflow Recovery Manager recovers crashed instances and resumes instances that have been waiting beyond a due date.

Generate Triggers

Generate Triggers allows you to create database triggers. The Workflow Policies module of Siebel Business Process Designer uses these database triggers to identify which records may match policy conditions. The Generate Triggers server component needs to be rerun whenever new policies are created or deleted. You can run the Generate Triggers server component from either the Server Manager graphical user interface, or command line mode.

Invocation Mechanisms

Siebel Workflow can be invoked in three ways:

- **Workflow Policies.** Workflow Policies are database triggers that invoke a workflow process when a policy violation occurs. The violations are written to a database table. The Workflow Monitor Agent reads from the table and invokes the Workflow Process Manager server component to execute the workflow process. Processing invoked by Workflow Policies is not real time.
- **Events:**
  - **Run-time events (Personalization events).** Run-time events are based in the object manager and they occur when a change is encountered by the user interface or the business component layer. Processing invoked by run-time events is real time.
  - **User events.** User events are unique Workflow-internal events that trigger or resume a long-running workflow process. User events are generated by the User Event business service.

- **Script.** Scripts can call Siebel Workflow programmatically as a business service. Using scripts, you can invoke Workflow from an external system. The Workflow Process Manager server component provides APIs for such programmatic invocation.

For more information on invoking Siebel Workflow, see "Invoking Workflow Processes" on page 128.

Administration and Monitoring

Administration and monitoring of your workflow processes happens in the Administration - Business Process views in the run-time client. You use the Administration - Business Process views to stop workflow processes, to delete process instances and to purge process instances from the log, to monitor workflow processes that are running, and to recover workflow processes that have been interrupted.

For more information, see "Administering Workflow Processes in the Run-Time Client" on page 148.

Recovery

If the Workflow Process Manager server component fails, Siebel Workflow automatically resumes the interrupted workflow instances when the server restarts. Recovery is performed by the Recovery Manager based on the process instance's state information that is saved by the Workflow engine.

To manually recover process instances, you use the Workflow Instance Admin view. From the application-level menu in the run-time client, choose Administration - Business Process > Workflow Instance Admin. See "Workflow Instance Admin View" on page 152.

For more information, see "Recovering Workflow Processes" on page 127.

Workflow Interaction with Other Siebel Components

In automating your organization's business processes, Siebel Workflow interacts with various components of the Siebel eBusiness architecture.

Siebel Server Components

The Workflow engine interacts with other server components through the Server Request Broker. Working as a business service, the Workflow engine calls server components.

To invoke server components that are exposed as specialized services, the Workflow engine calls them by their respective signature. For example, to send emails, the Workflow engine calls the Communications Server as the Outbound Communications Manager business service. To assign objects to users, it calls the Assignment Manager component as the Synchronous/Asynchronous Assignment Request business service.

To invoke server components that are not exposed as specialized services, the Workflow engine uses the predefined business service called Server Request. The Server Request business service sends a generic request to the Server Request Broker. For more information on the Server Request business service, see "Predefined Business Services" on page 253.

Server Request Broker

The Server Request Broker (SRBroker) acts as a request broker for the Siebel application server. The Workflow engine sends requests to SRBroker, synchronously or asynchronously, and SRBroker brokers the request to the appropriate component. The messaging involves:

- Sending asynchronous messages from an interactive-mode server component to the Workflow engine.
- Communication (synchronous and asynchronous) between the Workflow engine and batch components.
- Scheduling repeated tasks that are to be executed periodically in the Workflow engine.

Another job performed by SRBroker is load balancing. When SRBroker receives a request, it routes it to the server component in the current server. If the component is not available in the current server, SRBroker then sends it to other servers on a round-robin basis.

Siebel Workflow also uses SRBroker to resume waiting processes. SRBroker pools a database table on a regular basis to see all tasks that needs to be resumed.

Personalization Engine

The Personalization engine handles run-time events (application events, applet events, and business component events). It is through integration with the Personalization engine that Siebel Workflow processes run-time events. A workflow process triggered or resumed by run-time events registers itself with the Personalization engine at the time of the process’s activation. When a run-time event occurs in a user session, the Personalization engine calls Workflow in the local object manager. If the workflow process is persisted, it is resumed in the Workflow Process Manager server component.

Inbox

Inbox is a single screen in Siebel eBusiness applications that shows all approval and notification items assigned end users regardless of the screen where the item originated. Inbox shows enough detailed information about the item so that the end users can act on the item from the Inbox and not have to navigate to other screens for more information. See *Applications Administration Guide* for more information on Inbox.

b

4 Planning Workflow Processes

This chapter describes the steps involved in planning workflow processes. To plan your workflow processes, perform the following tasks:

1. "Gathering Information for Workflow Process Planning" on page 37
2. "Understanding Workflow Process Requirements" on page 38
3. "Considering Business Objects and Business Services When Planning Workflow Processes" on page 39
4. "Defining a Test and Migration Strategy for Workflow Processes" on page 40
5. "Verifying Workflow Policies Installation" on page 41

For information about upgrading Siebel Workflow, see "Upgrading Siebel Workflow" on page 42 and *Upgrade Guide* for the operating system you are using.

Gathering Information for Workflow Process Planning

Start gathering information by looking at how your organization currently handles workflow issues, business processes, and overall workflow. These current processes are the basis of what you will create using Siebel Business Process Designer.

If you currently have an automated system, you need to gather information on the processes handled by that system. It is also important to understand the limitations or problems that the current system has that you want to address with Siebel Workflow Processes.

There are two primary areas you may want to research for information on your current workflow processes: existing process information and areas for improvement or new process requirements.

Researching Existing Process Information

Existing process information can come from a variety of sources:

- Current automated processes
- Management guidelines
- Written guidelines of process rules or approval paths
- Internal procedures, written or unwritten

An example of gathering information about an existing process would be to document each step that a new work item, such as a service request, takes from the moment it is initiated to the moment it is complete. Include information about decision points in the process, such as when a service request should be escalated or which approval path an order takes when it is high priority versus low priority.

Researching New Processes and Areas for Improvement

After you have gathered as much information as you can about existing processes, review the information you have to see if there are areas for improvement in the process or whether a new process might be useful. Think of the following possibilities:

- New management guidelines or business requirements
- Current problems that need to be solved
- Areas that you would like to make more visible
- Customer satisfaction issues
- Workflow processes that you would like to automate

Understanding Workflow Process Requirements

A workflow process operates on business objects and business components. Usually, each workflow process is associated with a business object.

A workflow process consists of various tasks. There are many predefined tasks that can be used when you define a process. Some examples of the predefined tasks are:

- **Notifications.** Sending an email, page, or fax.
- **Siebel Operations.** Inserting or updating information in the Siebel database.
- **Integration Messages.** Requesting to send or receive data from an external system.
- **Assignment.** Requesting Assignment Manager to assign an object.
- **Navigation.** Navigating a user to a specific view.
- **Server Request.** Requesting the Siebel Server Request Broker to run a server process.

Except for Siebel Operations, all of the above tasks are invoked by calling a method on a business service. Siebel has predefined these business services so they can be used in workflow processes.

You may determine a specialized task that you are interested in calling in your workflow, such as “calculate credit risk.” Specialized tasks can be added by defining a custom business service. Workflow Processes can call both predefined and custom business services. For more information on defining custom business services, see *Integration Platform Technologies: Siebel eBusiness Application Integration Volume II*.

Seeded Workflow Processes

Many applications include seeded workflow processes for product functionality. By default, these seeded workflow processes are in a Completed state with a version number of 0. If you want to customize these seeded workflow processes, you need to revise them in Siebel Tools. After revising a seeded workflow, its version changes from 0 to 1. You then modify this version to suit your business needs, deploy it, and activate it. Activating a workflow process creates the appropriate run-time events defined within the process.

For information on activating workflow processes, see "Activating Workflow Processes" on page 149 and "To deploy a workflow process" on page 144.

Considering Business Objects and Business Services When Planning Workflow Processes

When planning a workflow process, please be aware of the following issues:

- If your workflow process is associated with a business object, the business object should have a primary component defined in Siebel Tools. For more information, see "Defining a Primary Business Component for a Business Object."
- If your business requirements require specialized functions, you may want to create a custom business service for the specific task. Business services can be defined in Siebel Tools or in the Siebel client administration screens. See *Integration Platform Technologies: Siebel eBusiness Application Integration Volume II* for information about defining a business service with Siebel Tools.
- If your workflow process involves a Siebel-defined business service or a custom business service, you must enable the business service in order to use it. See "Enabling a Business Service for Workflow Processes" on page 40.

Defining a Primary Business Component for a Business Object

For a business object to be used with a workflow process, it must have a defined primary business component.

To designate a primary component for a business object

1. In Siebel Tools, navigate in the Object Explorer to the appropriate business object.
2. Select the business object.
3. In the properties applet, use the picklist in the Primary Business Component field to select the appropriate component name.

   Select a primary component by selecting the key component for the specific business object.

4 Compile the SRF.

Once a primary business component has been defined, the business object appears in the Workflow Processes list editor.

Enabling a Business Service for Workflow Processes

Siebel provides a number of predefined business services. (See "Predefined Business Services" on page 253 for a list of these services.) You can also define your own custom business services using Siebel Tools or the Administration - Business Service view in the Siebel client.

To be displayed in a Siebel Workflow picklist, the Hidden flag for the business service must be set to FALSE. Additionally, you must set the Hidden flag for the associated business service methods and method arguments.

NOTE: By default, business services defined in the Siebel client are not hidden. Also note that business services, business service methods, and business service method arguments all have a Display Name property. The value of this property is the value that appears in the picklists on the workflow administration views.

To enable business services for Workflow

1. In Siebel Tools, from the Object Explorer applet, select the business service object.

   This action displays a list of defined business services.

2. Select the business service you want to modify.
3. In the Properties applet, change the Hidden field to FALSE.
4. In the Object Explorer applet, select the business service method under the business service.
5. Select the method you want to modify and change the Hidden field to FALSE in the properties applet.
6. Repeat Step 5 for each method, if applicable.
7. In the Object Explorer applet, select the method argument under the business service method.
8. Select the argument you want to modify and change the Hidden field to FALSE in the Properties applet.
9. Repeat Step 8 for each method argument, if applicable.

Defining a Test and Migration Strategy for Workflow Processes

Before implementing new workflow processes, you must verify them in a test environment. Testing new processes verifies that the process you release into the production environment properly executes and does not cause conflicts with your existing workflow processes.

The following are some suggestions for setting up your test and migration policy:

- Make sure your test environment and production environment have identical versions of the software and that you are using realistic data in your database by using a partial or complete copy of the production database.
- Create a small group of workflow processes to implement as a first phase of implementation. After you have successfully implemented the first group, you can add more processes in a systematic manner.

For more information on migrating your test environment to your production environment, see "Migrating Policies to the Production Environment" on page 246.

Verifying Workflow Policies Installation

Workflow Policies is installed as part of the Siebel Server and client installation and is enabled by using your license key information. This section describes only how to verify the correct installation of Workflow Policies. For information about the installation process, see the installation guide for the operating system you are using.

To run Workflow Policies, make sure the Siebel Server components (including Workflow Management), as well as both Siebel Tools and the Siebel client (Service, Call Center, or Sales), are installed.

Verifying the Repository Setting for Workflow Policies Installation

In the Siebel client, the .cfg file is used to configure Workflow Policies. Check that the DockRepositoryName entry specifies the correct repository name.

Verifying the Workflow Setup for Workflow Policies Installation

You need to verify that your license key includes Siebel Workflow. Because Siebel Workflow runs as server components on the Siebel Server, you also need to verify the proper installation of the Siebel Server.

To do this, follow the procedure that follows to verify that you can access the Workflow Policies client screens and server screens.

To verify the Workflow setup

1. Log in to the Siebel client as the Siebel administrator.
2. Select View > Site Map > Administration - Business Process.

   Under the list of views, you should be able to see Policies, Policy Groups, and so forth. This indicates that your license key is correct.

3 From a client that is configured to manage the server component groups, select View > Site Map > Administration - Server Management > Servers > Component Groups.

Check that the Workflow Management component group is enabled.

Upgrading Siebel Workflow

There have been schema changes in this release. The main Siebel database tables that contain workflow definitions have changed. The new tables contain only the workflow definitions for active, deployed workflow processes. The definitions of workflows that are inactive or in progress are located in the Siebel Tools repository.

The upgrade process automatically copies or moves all workflow process definitions to the Siebel Tools repository. Workflow policy object and policy program data is upgraded normally; no data is changed or lost. Database triggers are not upgraded. After upgrading, you must regenerate database triggers.

For more information and to see the necessary pre-upgrade and post-upgrade actions you must take, see the *Upgrade Guide* for the operating system you are using.

5 For Developers: Basics of Building Workflow Processes

This chapter describes the steps involved in building workflow processes. This chapter is divided as follows:

- "Overview of Developing a Workflow Process" on page 43
- "Siebel Tools and Workflow Processes" on page 44
- "Using Process Designer in Siebel Tools" on page 46
- "Tutorial: Using Process Designer in Siebel Tools" on page 56

Overview of Developing a Workflow Process

You use Siebel Tools to create workflows. In a typical development cycle for creating workflows, your steps take the following sequence:

1. Review existing process definitions. See "Reviewing Existing Process Definitions" on page 53.
2. Use the Siebel Tools client to develop new workflow processes. See "Using Process Designer in Siebel Tools" on page 46.
3. Save the workflow processes to your local database.
4. Test the workflow processes using the Siebel Web Client. See Chapter 8, "For Developers: Testing Workflow Processes."
5. Debug the workflow processes using your local master or test database.

   NOTE: Long-running workflow processes cannot be debugged using the Process Simulator. For testing of long-running workflow processes, see "Testing Workflows That Involve Server Components" on page 141.

   Workflow definitions are checked out from the repository into the local database, where they are modified and debugged locally before being checked in to the master repository.

   NOTE: Debugging against a server or test database, instead of debugging locally, allows the workflow engine to access server components such as the Server Request Broker.

6. Run the workflow processes from the Siebel Web Client, using your local master or test database.
7. Using the Siebel Tools client:
   - Check the workflow processes into and out of your master database.
   - Export the workflow processes into an XML file for backup.
   - Import the workflow processes from the XML file to restore them (if necessary). See "Importing or Exporting a Process Definition" on page 147.

8 Deploy the workflow process definitions from your master database to your staging or production database. See "Deploying Workflow Processes" on page 144.

This development-cycle sequence is illustrated in Figure 6 on page 44.

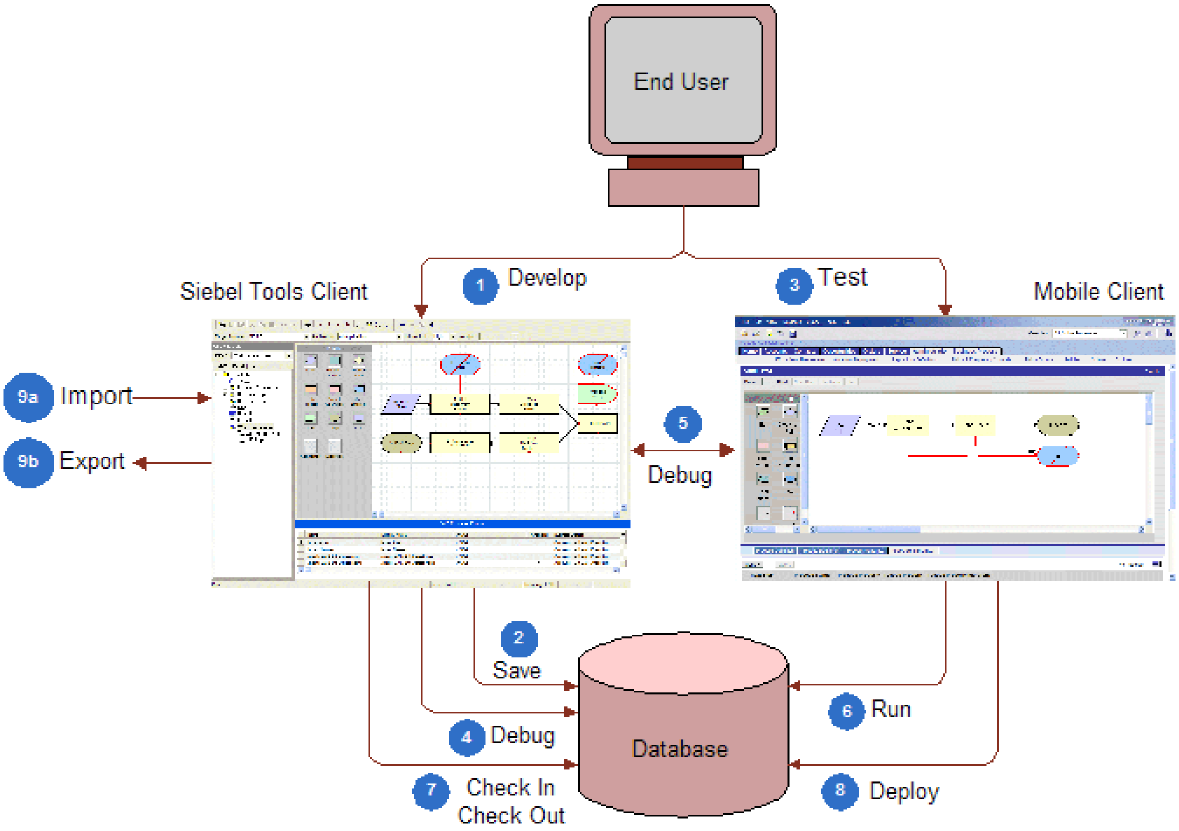


Figure 6. Typical sequence for developing workflows

Siebel Tools and Workflow Processes

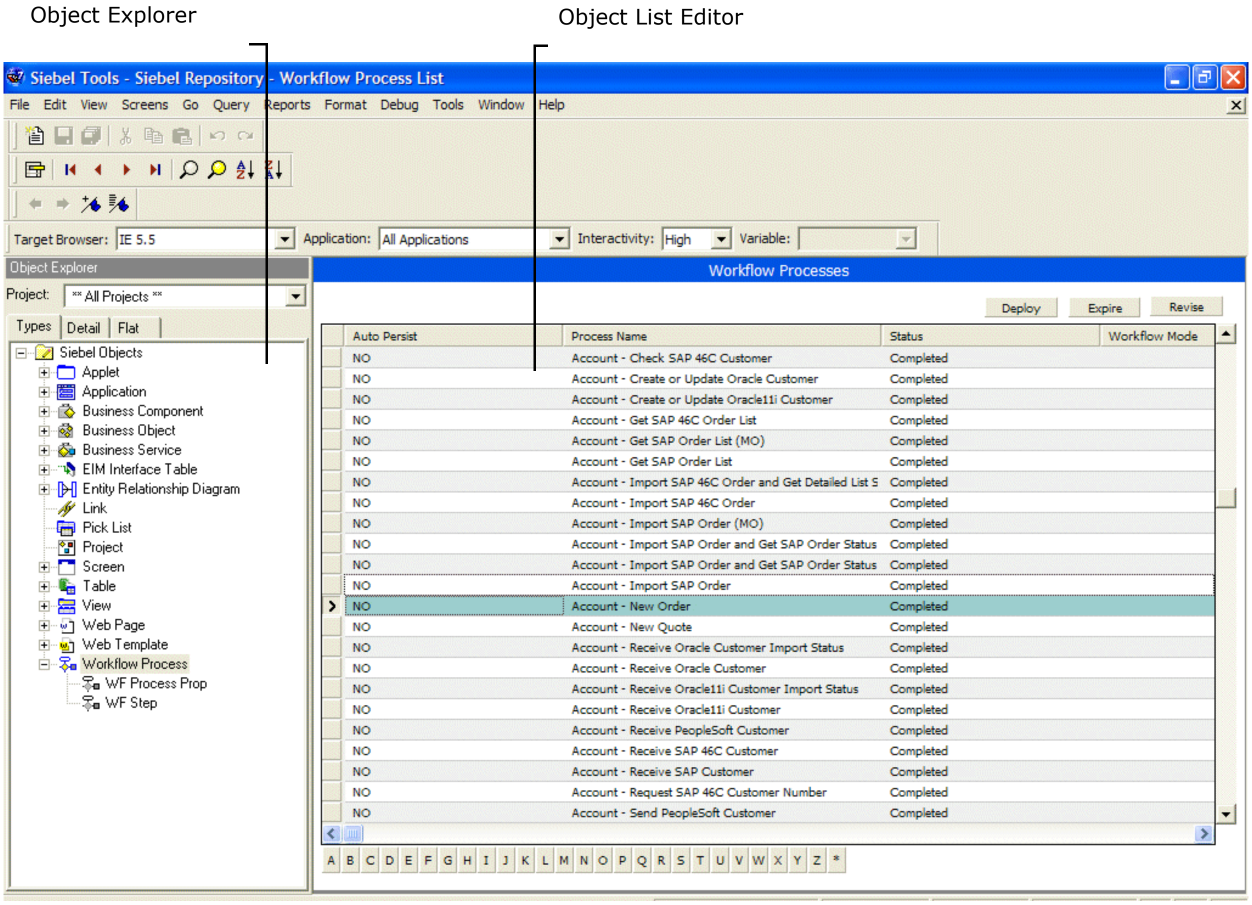
Siebel Tools is an integrated environment for configuring Siebel applications. You use Siebel Tools to modify standard Siebel objects and create new objects to meet your organization's business requirements. Just as you use Siebel Tools to extend the data model, modify business logic, and define the user interface, you also use Siebel Tools to configure the workflows that the Siebel application uses to automate your organization's business processes.

Siebel Tools consists of an Object Explorer window and one or more Object List Editor windows, as shown in Figure 7 on page 45. The Object List Editor window lists object definitions for each object type and allows you to edit object type properties. The Object Explorer provides navigation between each group of object definitions of a particular object type.

Object type is an entity that is displayed as a node on the Object Explorer. An object type is the template from which object definitions are created and have a predefined set of properties. Workflow Process is one object type.

An *object definition* implements one piece of the software such as Service Request or Contact. This object definition consists of *properties*, which are characteristics of that piece of the software.

Properties correspond to the columns in Object List Editor windows. The information entered under the columns is *values*. You can edit the properties of the currently selected object definition in an Object List Editor window by changing the values in the columns. You may change the property values in an object definition but not the set of properties to which values are assigned.



Target Browser: IE 5.5 Application: All Applications Interactivity: High Variable:

Project: ** All Projects **

Auto Persist	Process Name	Status	Workflow Mode
NO	Account - Check SAP 46C Customer	Completed	
NO	Account - Create or Update Oracle Customer	Completed	
NO	Account - Create or Update Oracle11i Customer	Completed	
NO	Account - Get SAP 46C Order List	Completed	
NO	Account - Get SAP Order List (MO)	Completed	
NO	Account - Get SAP Order List	Completed	
NO	Account - Import SAP 46C Order and Get Detailed List S	Completed	
NO	Account - Import SAP 46C Order	Completed	
NO	Account - Import SAP Order (MO)	Completed	
NO	Account - Import SAP Order and Get SAP Order Status	Completed	
NO	Account - Import SAP Order and Get SAP Order Status	Completed	
NO	Account - Import SAP Order	Completed	
NO	Account - New Order	Completed	
NO	Account - New Quote	Completed	
NO	Account - Receive Oracle Customer Import Status	Completed	
NO	Account - Receive Oracle Customer	Completed	
NO	Account - Receive Oracle11i Customer Import Status	Completed	
NO	Account - Receive Oracle11i Customer	Completed	
NO	Account - Receive PeopleSoft Customer	Completed	
NO	Account - Receive SAP 46C Customer	Completed	
NO	Account - Receive SAP Customer	Completed	
NO	Account - Request SAP 46C Customer Number	Completed	
NO	Account - Send PeopleSoft Customer	Completed	

Figure 7. Object Explorer and Object List Editor Windows

This guide assumes that you understand the basics of using Siebel Tools. The information you need to know includes the following topics:

- Siebel Tools application windows (the Object Explorer and the Object List Editor)
- Configuring object properties, applets, and applet controls
- Using the Siebel Tools menu bar

- Checking out and checking in projects
- Working with projects

For information on these and many other topics, see *Using Siebel Tools*. For further information, see *Configuring Siebel eBusiness Applications*.

Using Process Designer in Siebel Tools

You use the Process Designer in Siebel Tools to configure workflows to meet the specific needs of your business. In this release, the Process Designer resides in Siebel Tools and the Workflow Process object type is a new top-level object type in the Object Explorer within Siebel Tools.

Siebel Workflow is a configuration object. The Workflow Process object type belongs to a specific project, but it does not get compiled and it is not part of the repository merge that happens during the upgrade process.

Because the Process Designer is in Siebel Tools, data objects are available for use as you design your workflow processes. Changes to repository data are immediately available for use in a workflow process without the need for a compile of the SRF. You can use configuration data (business component fields and other repository information) while you are building your workflows. For example, if you have a List of Values (LOV) such as Account Status (with values of Gold, Silver, and Bronze) you can use a newly added LOV in the definition of your workflow process conditions as you are defining them. As another example, if you add a field to a business component, that new field is readily available for use in Process Designer.

The type of data that is not available for use in designing workflow processes is the run-time data such as an account name or a ZIP code and other transactional data. To use run-time data as you are building workflow processes, make the data a process property variable. For more information, see "About Process Properties" on page 69. You can also hardcode the run-time data in your workflow process, if you need this flexibility, by using unbounded picklists.

All business services are available for use at design time, both those business services defined in Siebel Tools and those defined in the run-time client.

More information on the Process Designer's design functions, applet fields, and palette items is included in the following sections:

- "About the Design Functions of the Process Designer" on page 46
- "Field Descriptions: Workflow Processes Applet" on page 47
- "Field Descriptions: WF Process Props Applet" on page 49
- "Field Descriptions: WF Steps Applet" on page 50
- "Process Designer Palette Items" on page 51

About the Design Functions of the Process Designer

You use the Process Designer in Siebel Tools to build your workflow processes. While drawing your workflow processes in the Process Designer, you can:

- **Copy and paste.** Copy and paste workflow palette items within a workflow as you are building it. When you copy a step, its associated branches are copied along with it.
- **Edit shape properties and layout.** Define shape colors and other attributes such as the look of the line, the fill pattern, and the font for labels. Create consistency by controlling alignment of shapes and by making shapes the same size as others in the workflow process.
- **Zoom.** Zoom in and out on the Process Designer Palette to view the workflow process you are drawing at various magnifications.
- **Copy drawings.** Copy workflow drawings into another application, such as a Microsoft Word document.
- **Print.** Print a drawing of the workflow process.
- **Show branch labels and exceptions.** You can choose to show or hide the names that label connectors within a workflow.

The copy-paste function works as Windows applications do with the CTRL+C and CTRL+V key combinations. The rest of the design functions are employed by right-clicking the Process Designer Palette.

For more information, see "Process Designer Palette Items" on page 51.

Field Descriptions: Workflow Processes Applet

Table 3 describes the fields in which you enter data to define workflow processes, in the Workflow Processes applet.

Table 3. Workflow Processes Applet Fields

Field	Description	Possible Value
Auto Persist	Sets workflow persistence.	Checkmark.
Process Name	The name of the process.	Required. A descriptive name that is: ■ Consistent with your overall naming strategy ■ Meaningful to the designer of the process ■ Unique
Status	The current status of the process.	■ Not In Use ■ In Progress ■ Completed

Table 3. Workflow Processes Applet Fields

Field	Description	Possible Value
Workflow Mode	The workflow process type.	The workflow process types are the following: ■ **7.0 Flow.** A 7.0 workflow process provides backward compatibility for existing Siebel 7 (pre-7.7) workflows. ■ **Long Running Flow.** A long-running workflow process is a persistent workflow that can last for hours, days, or months. ■ **Interactive Flow.** An interactive workflow process navigates the user across Siebel views. ■ **Service Flow.** A service workflow process executes a set of operations upon event invocation.
Group	This field applies to pre-7.7 workflows.	**NOTE:** Do not use this field for new workflows you create. Use the Project field instead.
Project	The name of the project to which the workflow process belongs.	Required.
Version	The version number of the process definition.	Read-only. The default version is 0. The version number increments by one when you use the Revise button to modify an existing process definition.
Business Object	The name of the associated business object.	(Optional) This value is selected from a picklist of business objects. Only business objects with a defined primary component appear in this picklist. See Chapter 4, "Planning Workflow Processes" for more information on defining the primary component for a business object.
Error Process Name	The name of the workflow process to call as the error process.	Choose from the picklist of predefined workflow processes.

Field Descriptions: WF Process Props Applet

Table 4 describes the fields in which you enter data to define workflow processes, in the WF Process Props applet.

Table 4. WF Process Props Applet Fields

Field	Description	Possible Value
Name	The name of the process property.	Required. A descriptive name that is: ■ Consistent with your overall naming strategy ■ Meaningful to the designer of the process ■ Unique
Display Name	A user-friendly version of the property name.	The Display Name can be the same as or different from the Name.
In/Out	Describes whether or not the process property is passed in or out of the process, passed into the process and returned, or used only within the process.	■ **In.** The process property is passed into the process. (Binary types cannot be assigned this value.) ■ **Out.** The process property is passed out of the process. (Binary types cannot be assigned this value.) ■ **In/Out.** The process property is passed into the process and returned. (Binary types cannot be assigned this value.) ■ **None.** The process property is used only within the process.
Business Object	The name of the associated business object.	Read only. See Chapter 4, "Planning Workflow Processes" for more information on defining the primary component for a business object.
Business Component	The name of the business component containing the virtual field.	(Optional) This value is selected from a picklist of business components belonging to the workflow process business object.
Virtual Field	The name of the business component field mapped to the workflow process property.	(Optional) This value is selected from a picklist of fields belonging to the business component. Only calculated fields with no calculated values appear in this picklist.
Default String	Initial value if the process property is a string type.	Free-form text. If you enter <Value>, the process property is initialized with the value in the Value field of the workflow input property set.

Table 4. WF Process Props Applet Fields

Field	Description	Possible Value
Default Date	Initial value if the process property is a date type.	Calendar widget.
Data Type	The type of data that can be stored in the process property.	■ **String.** For alphanumeric data. ■ **Date.** For dates. ■ **Number.** For numeric data. ■ **Hierarchy.** Data type used by eBusiness Application Integration to store data from a property set. For more information, see Overview: Siebel eBusiness Application Integration Volume I. ■ **Binary.** For variant or binary information. Binary types must be assigned the None value in the In/Out field. ■ **Integration Object.** For storing integration objects. Integration objects have the Hierarchy data type.
Default Number	Initial value if the process property is a numeric type.	Numeric widget.
Integration Object	Data type used by eBusiness Application Integration to store data from an integration object.	Example: "Account - Get Oracle Customer (Oracle)"
Correlator Flag	Makes the process property ready for use as a correlator (a piece of business data that identifies the recipient of the incoming message). See "About the Workflow User Event Business Service" on page 120.	Check mark.
Access Mode	Controls whether a process property is read-only or read-write.	Read-only or Read-write.

Field Descriptions: WF Steps Applet

For descriptions of the fields in the WF Steps applet, see "Field Descriptions: WF Steps Applet" on page 73.

Process Designer Palette Items

Figure 8 shows the Process Designer Palette.

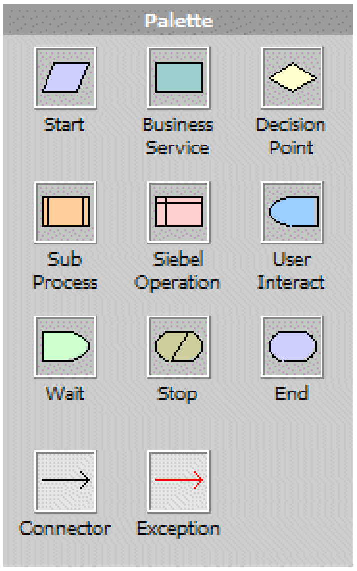


Figure 8. Drag and drop steps and connectors from the palette to the work space in the Process Designer.

Table 5 describes the items available on the Process Designer Palette. Each item represents a type of step or connector.

Table 5. Process Designer Palette Items

Item	Description	Possible Value
Start	Represents the input conditions that must be met to execute an instance of a business process. For more information, see "About Start Steps" on page 81.	Every process must begin with a Start step. There can be only one Start step in a process.
Business Service	Represents an activity within a business process. Business service steps call business services that allow you to execute predefined or custom actions in a workflow process. For more information, see "About Business Service Steps" on page 86.	A process can have one or more Business Service steps.

Table 5. Process Designer Palette Items

Item	Description	Possible Value
Decision Point	Represents a step in the process definition where the work item branches off to different steps depending on a set of conditions. A Decision Point evaluates the defined conditions to determine the next step of the process instance. For more information, see "About Decision Steps" on page 84.	A process can have one or more Decision Point steps.
Subprocess	Represents a workflow process embedded into another workflow process. A subprocess has its own process definition. For more information, see "About Subprocess Steps" on page 90.	A process can have one or more Subprocess steps.
Siebel Operation	Represents a type of action. Siebel Operation steps perform Siebel operations, such as Insert, Update, or Query, on business components. For more information, see "About Siebel Operation Steps" on page 94.	Business object logic applies to all Siebel operations. A process can have one or more Siebel Operation steps.
User Interact	Represents end-user view navigation. User Interact steps control the flow of Siebel views within an application. A workflow process containing User Interact steps guides end users through a specified flow of Siebel views based on the users' actions, or executes a specified set of actions. For more information, see "About User Interact Steps" on page 100.	A process can have one or more User Interact steps.
Wait	Represents a pause in workflow process execution. Wait steps suspend execution for a specific period of time or until a specific event occurs. For more information, see "About Wait Steps" on page 99.	A process can have one or more Wait steps.
Stop	Represents an end to a workflow process and the presentation of an error to the user. Stop steps terminate the workflow process instance. For more information, see "About Stop Steps" on page 103.	A process can have one or more Stop steps.

Table 5. Process Designer Palette Items

Item	Description	Possible Value
End	Indicates when process execution is complete. An End step allows you to save output arguments as process properties. For more information, see "About End Steps" on page 105.	A process can have one or more end steps.
Connector	Represents the direction of flow between steps in a business process.	A process can have one or more connectors.
Exception	Represents a deviation from normal processing. An exception can be a system error or a user-defined error.	A process can have one or more Exception branches.

About Defining Workflow Process Parameters and Steps

The first part of designing a workflow process is working with workflow process definitions to set the general process parameters and lay out the flow of the process steps. This involves:

- "Reviewing Existing Process Definitions" on page 53
- "Defining a New Workflow Process" on page 54
- "Modifying Existing Process Definitions" on page 55

Reviewing Existing Process Definitions

Review your existing workflow processes to see if the process you need is already available or if a similar process exists that you can copy and modify to meet your requirements.

The Workflow Processes Object List Editor (OBLE) provides a list of all the current process definitions. You can access the Workflow Processes OBLE in Siebel Tools.

To review existing workflow process definitions

1. In Siebel Tools, in the Object Explorer Types tab, choose the Workflow Process object type.

   The right pane shows the Workflow Processes OBLE, which lists all the workflow processes.

2. If you find a process you want to copy as the basis for a new process definition, select the record, then right-click and choose Copy Record.

   See "Copying a Workflow Process" on page 68 for more details on copying an existing process definition.

Defining a New Workflow Process

Once you have reviewed your existing workflow process definitions, you are ready to create new workflow processes for the processes you have identified as missing.

To design a workflow process

1. From within Siebel Tools, check out (for local use) or lock (for server use) the project which contains the Workflow objects.

   NOTE: You must check out the Workflow objects only if you are using a local data source. You check out the Workflow objects just like any other repository objects. If you are logged on directly to the server data source, check-out is not needed.

2. In the Object Explorer Types tab, choose the Workflow Process type.

   The right pane shows an Object List Editor (OBLE) window with a list applet containing all the workflow processes.

3. In the Workflow Process OBLE, right-click and select New Record to create a new workflow process record.
4. In the Process Name field, give the workflow a name that is short but meaningful.
5. Your new workflow process must belong to a project. Scroll to the Project field and from the picklist, choose the project to which you want the workflow to belong.
6. In the Description field, enter a description.

   Use this field to describe the purpose of the process and any special requirements or notes.

7. In the Business Object field, choose the business object that the workflow process involves.
8. In the Workflow Mode field, choose the type of workflow process: long-running, interactive, or service.
9. Enter other relevant details.
10. Right-click the record and choose Edit Workflow Process.
11. Drag and drop shapes (palette items) from the palette to the design canvas to build the workflow diagram. For information on workflow steps, see Chapter 6, "For Developers: Workflow Process Steps." For more information on diagramming a workflow using the Palette Designer, see "Diagramming a Workflow Process" on page 66.

Naming Conventions for Workflow Processes and Process Properties

When naming workflow processes, the combination of workflow process name and version must be unique. That is, you can have two workflow processes of the same name as long as their version numbers are different.

Names given to workflow process steps and process properties must be unique as well, within a workflow process. When you create a new process step, in its name field it is automatically given a unique number that you can change. If you change the name, the new name must be unique from the names of the other steps in the process.

Names for workflow processes, steps, and properties should not contain any characters that can form parts of expressions or that have special meaning within the Siebel application or the underlying database platform.

Reserved symbols that cannot be used in naming of workflow processes, steps, and properties are the following:

Ampersand	&
Period	.
Back Slash	\
Forward Slash	**/**
Colon	**:**
Asterisk	*****
Question Mark	**?**
Quotation Marks	**”**
Brackets	**<>**
Slash	**|**
At	**@**

Modifying Existing Process Definitions

You can modify active workflow process definitions without restarting the Workflow Process Manager. The server parameter called Workflow Version Checking Interval controls how often the server component checks for new active versions of each process definition.

After a new process definition is activated, all incoming process instances after the Workflow Version Checking Interval will use the new definition. Process instances initiated before this activation will continue using the previous process definition.

To modify an existing workflow process definition

1. In Siebel Tools, in the Workflow Processes OBLE, select the record for the workflow process you want to modify.
2. Click the Revise button.

   A copy of the workflow process, with the same name and a version number incremented by one, appears in the OBLE with a status of In Progress.
3. With this new version selected, right-click and choose Edit Workflow Process.
4. Make changes to this new version, as necessary.

5 When you have finished modifying the workflow process, validate it by right-clicking in the Palette Designer. For more information, see "Using the Validate Tool to Correct Errors in Workflow Processes" on page 139.

6 Deploy the workflow process to make it active. For more information, see "Deploying Workflow Processes" on page 144.

Tutorial: Using Process Designer in Siebel Tools

This tutorial takes you through the steps involved in designing, simulating, and deploying a workflow process using Siebel Business Process Designer.

This tutorial is organized as follows:

- "Designing Your Workflow Process" on page 56
- "Testing Your Workflow Process" on page 62
- "Deploying Your Workflow Process" on page 63

Designing Your Workflow Process

In this tutorial, you complete the following sequence of tasks to design your workflow process:

1 Create a new workflow process record. See "To create a new workflow process" on page 56.

2 Draw the workflow process in the Palette Designer. See "To design the workflow process" on page 57.

3 Define properties for the steps of the workflow process. See "To set properties for the workflow process" on page 59.

4 Create a trigger for the workflow process. See "To define the run-time event that triggers the workflow process" on page 59.

5 Define condition criteria for a Decision branch of the workflow process. See "To define condition criteria for the workflow's Decision step" on page 60.

6 Validate the workflow process. See "To validate the workflow process" on page 61.

NOTE: The login information (SADMIN/SADMIN) as well as the other settings listed in this tutorial are provided as examples. You should modify these example settings as appropriate for your own setup.

You create a workflow process and set its properties using the Process Designer in Siebel Tools.

To create a new workflow process

1 Log in to Siebel Tools as SADMIN/SADMIN connecting to the Sample database.

2 In the Object Explorer Types tab, choose the Project object type.

3 In the Projects OBLE, right-click and select New Record to create a new project.

- **a** Name the project “77 Workflow.”
- **b** Lock the project.

4 In the Object Explorer Types tab, choose the Workflow Process object type.

NOTE: If the Workflow Process object type does not appear in the Object Explorer by default, you can add it. Choose View > Options, and on the Object Explorer tab, select the Workflow Process type.

5 In the Workflow Process OBLE, right-click and select New Record to create a new workflow process record.

6 Fill in the record’s fields to give the workflow process the following properties:

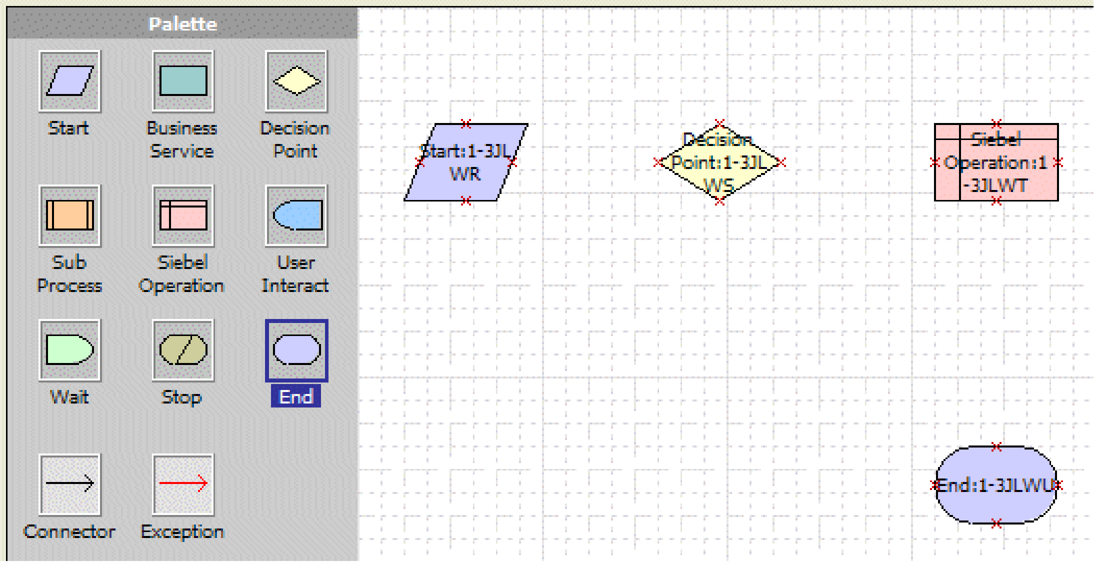
Property	Value
Process Name	77 Workflow Large Opportunity Creates Activity
Workflow Mode	Service Flow
Project	77 Workflow
Business Object	Opportunity

This workflow is to be invoked by a run-time event. Once a user creates a new Opportunity record, the revenue of the opportunity will be evaluated. If the value of the opportunity is over $10,000, an Activity record will be created to have the sales representative follow up on the deal.

To design the workflow process

1 Right-click on the new workflow process record you created in Step 5 on page 57 and select Edit Workflow Process.

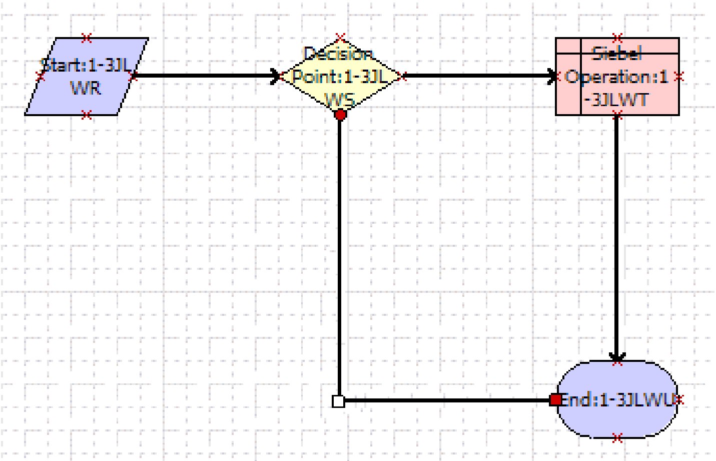
2 From the Palette Designer, drag and drop step icons to the design canvas to match the following graphic. Add one Start step, one Decision Point, one Siebel Operation step, and one End step.



3 Drag and drop Connector branches to connect all the steps, as depicted in the following graphic.

Make sure that all of the connectors are correctly attached. A connector is correctly attached when red dots and squares appear at each of its ends. If you see a white square at one end, that end is not correctly attached.

NOTE: To create an elbow on a Connector branch, as in the branch connecting the Decision step to the End step in the following graphic, right-click the branch, choose Edit > Add Point. Then select the point on the branch and drag it to the appropriate place in the design canvas.



To set properties for the workflow process

1 Select the Start step.

The corresponding Start step record is selected in the WF Steps list applet below the Palette Designer. When you select a step icon in the design canvas, the corresponding step is selected in the list applet so that you can define its properties.

2 For the Start step, set the Name property to "Start."

3 Select the Decision step.

4 For the Decision step, set the Name property to "Is opportunity over 10K?"

5 Select the Siebel Operation step.

6 For the Siebel Operation step, set the following properties:

Property	Value
Name	Insert Activity to Follow Up
Business Component	Action
Operation	Insert

7 In the design canvas, right-click the Siebel Operation step and choose Show Fields.

The list applet below the design canvas toggles to the Input Arguments list applet.

8 In the Input Arguments list applet, right-click and choose New Record to add two new records with the following properties:

Field Name	Type	Value
Description	Literal	Please call the customer ASAP since this is a large opportunity.
Type	Literal	To Do.

9 Select the End step, and set its Name property to "End."

To define the run-time event that triggers the workflow process

1 In the design canvas, select the Connector branch that connects the Start step to the Decision step.

The list applet below the design canvas toggles to the WF Step Branch list applet.

2 Set the branch properties as follows:

Property	Value
Name	Evaluate Oppty
Type	Default
Event Object Type	BusComp
Event Object	Opportunity
Event	WriteRecord

The business scenario for this tutorial dictates that when an opportunity’s revenue is greater than $10,000, an activity is inserted for the sales representative to follow up with the customer. To determine whether this activity is inserted or not, you set condition criteria on the workflow’s Decision step.

To define condition criteria for the workflow’s Decision step

1 In the design canvas, double-click the Connector branch that connects the Decision step to the Siebel Operation step.

   The Compose Condition Criteria dialog box appears.

2 In the Compare To dropdown picklist, select Business Component.

3 In the Operation dropdown picklist, select Greater Than.

4 In the Object dropdown picklist, select Opportunity.

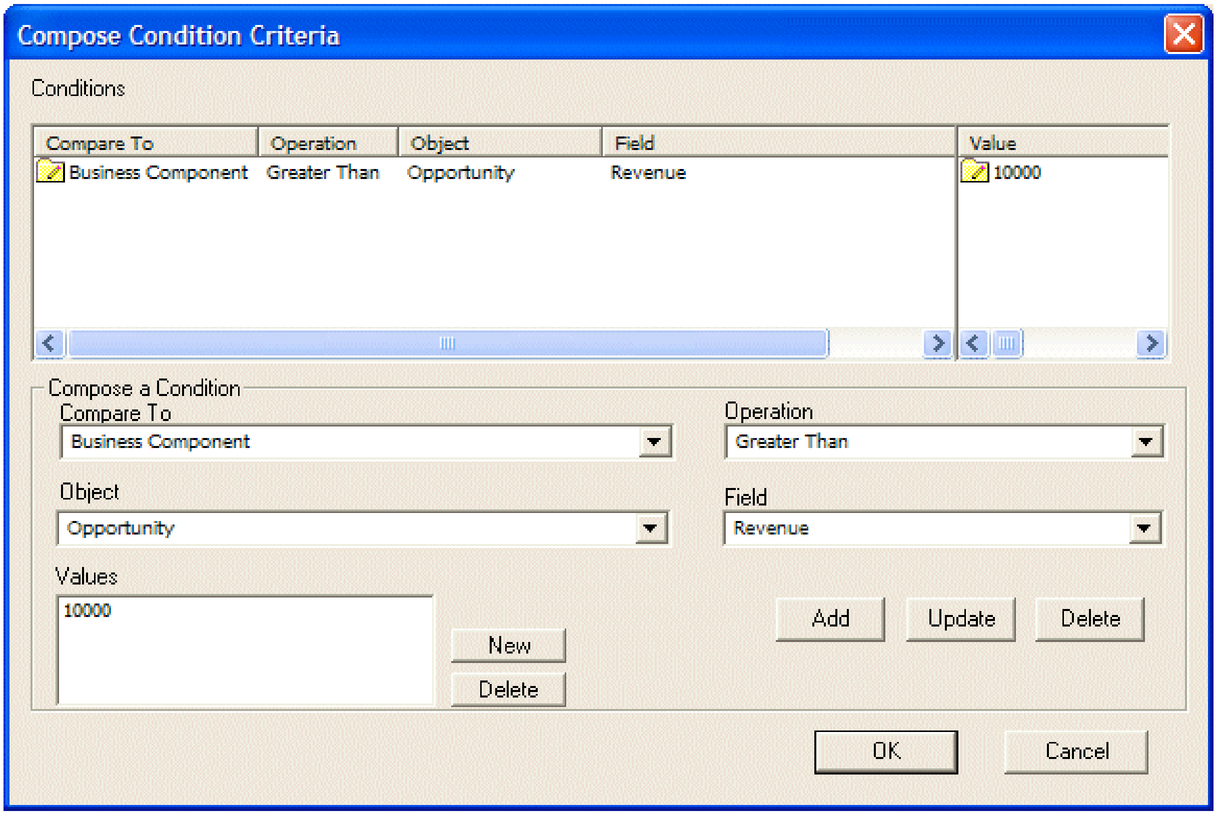
5 In the Field dropdown picklist, select Revenue.

6 In the Values box, click the New button to add a new value.

7 In the Add Value dialog box, type “10000” and click OK.

8. In the Compose Condition Criteria dialog box, click the Add button.

   The conditions you have set in the Compose Condition Criteria dialog box appear as shown in Figure 9.



Compose Condition Criteria

Conditions

Compare To	Operation	Object	Field	Value
Business Component	Greater Than	Opportunity	Revenue	10000

Compose a Condition

Compare To: Business Component

Operation: Greater Than

Object: Opportunity

Field: Revenue

Values: 10000

Figure 9. Condition Criteria for “Insert Activity to Follow Up” Siebel Operation Step

9. Click OK.

To validate the workflow process

1. In the Palette Designer, right-click and select Validate.

   The Validate dialog box appears.

2. Click Start.

   “Starting validation...” appears in the bottom-left corner of the Validate dialog box.

3. If the validation is successful, there are no errors to report and in the bottom-left corner of the dialog box, a message reads, “Total tests failed: 0.”

   NOTE: See “Using the Validate Tool to Correct Errors in Workflow Processes” on page 139 for a list of possible errors.

4. Click Cancel to exit the Validate dialog box.

Testing Your Workflow Process

In this section of the tutorial, you simulate the workflow process you created in "Designing Your Workflow Process" on page 56.

In this tutorial, you complete the following sequence of tasks to test your workflow process:

1. Before you can test your workflow process, you must create an opportunity that matches your test criteria. You create this opportunity, and make note of its row ID. This row ID is then used in the workflow process properties to run the test. See "To prepare for testing the workflow process" on page 62.
2. You start the simulation process with the Simulate command, and then you set the Debug properties for the run-time client. See "To begin your test of the workflow process" on page 62.
3. You run the Process Simulator in Siebel Tools by using the Start, Next Step, and Stop buttons. See "To run the Process Simulator" on page 63.

To prepare for testing the workflow process

1. Launch Siebel Call Center logging in as SADMIN/SADMIN to the Sample database.
2. From the application-level menu, choose Navigate > Site Map > Opportunities.
3. Add an opportunity with the following properties:

Name	77 Workflow Large Test Opportunity
Revenue	$400,000

4. From the applet-level menu, choose About Record to find the row ID for the opportunity in the Row # field. Make a note of this row ID.
5. In Siebel Tools, in the Palette Designer, right-click and select Show Process Properties.
6. In the WF Process Props list applet, select the Object Id record.
7. In the Default String field, insert the row ID value you noted in Step 4.

To begin your test of the workflow process

1. In the Palette Designer, right-click and select Simulate.

   A read-only version of the workflow process diagram now shows where the Palette Designer appeared previously. The simulation buttons (Start, Next Step, Continue, and Stop) appear above the workflow process diagram.

   Because the simulator in Siebel Tools begins by launching an instance of the run-time client, you must first set the appropriate Debug properties.

2. Set the Debug properties for your run-time client, as follows:

   a. In Siebel Tools, navigate to View > Options > Debug.

b Set the following properties for your run-time client:

Executable	`C:\sea77\client\BIN\siebel.exe`
CFG file	`C:\sea77\client\BIN\ENU\uagent.cfg`
Browser	`C:\Program Files\Internet Explorer\iexplore.exe`
Working directory	`C:\sea77\client\BIN`
Arguments	/h
User name	SADMIN
Password	SADMIN
Data source	Sample

c Click OK.

To run the Process Simulator

1. In the read-only version of the workflow process diagram, click the Start button.

   The simulation process begins, and a debug instance of your run-time client is launched. This can take several seconds.

   Once the debug instance is launched, you are automatically navigated to your Inbox in the run-time client.

2. In your Inbox, navigate to Submitted Items List to find the Debug Workflow record.
3. Click the Debug Workflow hyperlink.
4. In Siebel Tools, view the read-only version of the workflow process diagram.

   The Start step icon is selected, the Start simulation button is unavailable, and the other simulation buttons are active.

5. (Optional) In order to see additional information as you simulate your workflow process, in the Palette Designer, right-click and select Watch Window.

   You can hide the Watch window at any time by right-clicking and selecting Hide Watch Window.

6. Click the Next Step button to continue simulating the workflow process.

   In the read-only version of the workflow process, the Siebel Operation step icon is selected.

7. Click the Next Step button again to continue simulating the workflow process.
8. Click the Stop button to end the simulation.
9. From the application-level menu in the run-time client, choose Navigate > Site Map > Opportunities > Activities to see your test opportunity.

   If an Activity record has been generated, the simulation of your workflow process was completed successfully.

Deploying Your Workflow Process

When you have finished a successful simulation of your workflow process, you are ready to deploy it.

To deploy the workflow process

1. In Siebel Tools, in the WF Process Props list applet, scroll to the Object ID process property for your workflow process.
2. In the Default String field for the Object ID process property, delete the row ID.
3. In the Palette Designer, right-click and select All Processes.
4. Select your workflow process record and click the Deploy button.

   The Status of the workflow process changes to Completed.
5. From the application-level menu in the run-time client, choose Navigate > Site Map > Administration - Business Process > Workflow Deployment.
6. In the parent applet, select your workflow process and click the Activate button.

   In the child applet, your workflow process record shows a status of Active. You may need to refresh the list of records to see this newly updated status. To refresh the list, perform a null query in the child applet.

6 For Developers: Workflow Process Steps

This chapter describes the various steps you use to build workflow processes and how to work with them. When you add a step to build a workflow process, you also define the branches, conditions, and values associated with the step. This chapter is organized as follows:

About the Workflow Processes OBLE in Siebel Tools

You use the Workflow Processes list editor in Siebel Tools to do the following:

- Create and edit a workflow process. See "Diagramming a Workflow Process" on page 66.
- Define step details for a workflow process. See "Defining Step Details for a Workflow Process" on page 67.
- Copy a workflow process. See "Copying a Workflow Process" on page 68.
- Delete a workflow step. See "Deleting a Workflow Step" on page 68.
- Delete a workflow process. See "Deleting a Workflow Process" on page 68.
- Revise an existing workflow process. See "Modifying Existing Process Definitions" on page 55.
- Import and export a workflow process. See "Importing or Exporting a Process Definition" on page 147.

Diagramming a Workflow Process

Diagramming the process steps is an important part of creating a functional process. The flowchart interface of the Process Designer allows you to build a visual representation of the entire process flow, including decision points and decision branches. From this design, you then access and fill in step details by clicking a step icon.

You can choose to define the details for each step as you create them in the Designer applet, or you can finish the entire flowchart of the process and then enter the details for each step.

You define workflow step details by clicking on a step icon or connector arrows in the flowchart workspace.

Based on your planning results, use the following procedure to diagram the steps of the process.

To diagram the steps of a workflow process

1. From Step 11 on page 54, add a Start step to the design canvas.

   All processes must have one and only one Start step. Details on defining a Start step are in "Defining a Start Step" on page 81.

2. Add one or more middle steps to the design canvas. Processes can have one or more of any of the action step types, such as Business Service, Decision, Subprocesses, Stop, Wait, Exceptions, and Siebel Operation. There can be multiples of each type of step. For details on each type of step, see:

   - "Defining a Decision Step" on page 84
   - "Defining a Business Service Step" on page 89
   - "Defining a Subprocess Step" on page 91
   - "Defining a Siebel Operation Step" on page 94
   - "Defining a Wait Step" on page 99
   - "Defining a User Interact Step" on page 101
   - "Defining a Stop Step" on page 103

3. Add an End step to the diagram area.

   All processes must have at least one End step. Details on defining an End step are in "Defining an End Step" on page 105.

4 Illustrate the flow and paths of the process by dragging and dropping connector arrows between the steps. Position one end of the arrow on one of the steps and drag the handles to connect the other end to the next step in the flow.

NOTE: An end point on a connector is white if it is not successfully connected to a step. Be sure that both ends of all connectors are red, indicating that it is successfully connecting two steps.

Connecting an arrow to a Decision step creates a decision branch for that specific Decision. See "Defining Decision Branches" on page 85 and "Defining Conditions and Values for Branches, Decision Steps, and User Interact Steps" on page 82 for information about defining decision branches.

To add or remove a point in a connector, use the following steps:

a Select the connector or exception.

b Right-click and select one of the following:

- ❏ Edit > Add Point
- ❏ Edit > Remove Point

Defining Step Details for a Workflow Process

A workflow step's details include input and output arguments, branch parameters, and conditions. You define each step's details using the WF Steps applet in the Object List Editor (OBLE), one at a time by selecting each step in the workflow.

To define step details

1 Single-click on each step palette item to select the step, and enter the step's details in the WF Steps applet of the OBLE.

2 Enter input arguments, output arguments, branch parameters, and condition criteria using the OBLE.

- To enter input and output arguments (for business service steps, subprocess steps, and wait steps), right-click each step and choose Show Input Arguments or Show Output Arguments.
- To enter branch parameters (for start steps, decision points, and user interact steps), right-click each step and choose Show Branches.
- To enter conditions (for branches), double-click the branch (or right-click the branch and choose Edit Conditions).

  The Compose Condition Criteria dialog box appears.

  NOTE: The values you find listed in the Compose Condition Criteria dialog box are constrained by the business object for your workflow process, which is specified at the process level. For more information, see "Using Process Designer in Siebel Tools" on page 46 and "Field Descriptions: Workflow Processes Applet" on page 47.

  - ❏ Enter your condition criteria and click Add to add the condition.
  - ❏ To update a condition, select the record in the Conditions control, make modifications as necessary in the other controls, and click Update.

- To delete a condition, select the record from the Conditions control and then click Delete.
- Click OK.

Deleting a Workflow Step

You can delete a process step in the Palette Designer. When you delete a step, its associated branches are deleted along with it.

To delete a step

1. From the flowchart diagram, choose the icon for the step you would like to delete.
2. Right-click the icon, then select Edit > Delete.

Deleting a Workflow Process

You can delete a workflow process in the Object List Editor in Siebel Tools.

To delete a workflow process

1. In the Object Explorer Types tab within Siebel Tools, choose Workflow Process.
2. In the OBLE, select the process you want to delete.
3. Right-click and select Delete Record.

Copying a Workflow Process

You can copy a workflow process in the Object List Editor in Siebel Tools.

To copy a process

1. In the Object Explorer Types tab within Siebel Tools, choose Workflow Process.
2. In the OBLE, select the process you want to copy.
3. Right-click and select Copy Record.
4. In the Process Name field, enter a new name for the process.
5. Modify the other definition fields as necessary for the new process.

About Process Properties

Process properties are fields for storing values that you can use in steps, either as input and output arguments, or for performing evaluations. Process properties store values that the workflow process retrieves from the database or derives before or during processing. You can base decision branches on the values in a process property and pass process properties as step arguments. When a workflow process completes, the final results of the process properties are available as output arguments. You can also use process property values in expressions.

Some default process properties are automatically defined for each process. They are:

- **Object ID.** The Siebel row ID of the work item being processed. For more information, see "Object ID and Non-7.0 Workflow Processes" on page 69.
- **Error Code.** An error symbol of the process instance if a step returns an error. This process property is automatically populated when an error occurs.
- **Error Message.** An error message text of the process instance if a step returns an error. This process property is automatically populated when an error occurs.
- **Siebel Operation Object ID.** The object identification of an object that is updated, created, or queried on during a Siebel Operation step. This process property is automatically populated when a Siebel Operation step is executed.
- **Process Instance ID.** The object identification of the process instance. This process property is automatically populated when a process is executed and persistence is enabled.

The following sections provide further information on process properties:

- "Object ID and Non-7.0 Workflow Processes" on page 69
- "Process Properties Versus Property Sets" on page 70
- "Defining Process Properties" on page 70
- "Concatenating Process Properties" on page 71

Object ID and Non-7.0 Workflow Processes

In non-7.0 Flows, that is, in long-running, interactive, and service workflow processes:

- The Object ID must match the Row ID of the primary business component's active row. The Workflow engine will not allow the active row of the primary business component (BC) to be different from the Object ID process property. If the Object ID process property is different from the active row, the primary BC is re-executed to make the active row the same as the Object ID.

  NOTE: If you want to change the active row in a step of a workflow, you may do so (using an appropriate Business Service step or Siebel Operation step), but you must promptly update the Object ID process property to the new active Row ID, in the output arguments of the step that changes the active row.

  Once a step completes and output arguments have been evaluated, Workflow checks to make sure the Object ID matches the active row. So changes to the active row must be reflected in the Object ID property within the affected step.

- Run-time events are always received on the row specified by the Object ID parameter.

- It is possible to change the active row by assigning a new Row ID to the Object ID parameter. When Workflow detects that an assignment is made to the Object ID process property, Workflow re-executes the BC and makes the new Row ID the active row.
- You can set the Object ID to an empty string and Workflow will no longer enforce the must-match rule. However, the parts of Workflow that require an Object ID (such as run-time events and Siebel Operation steps) cannot be used until the Object ID is set to a new Row ID.

Process Properties Versus Property Sets

Siebel business services use a structure known as the property set to represent input and output data for a method call. Property sets are hierarchical structures containing name/value pairs, known as properties, at each level in the hierarchy. For a detailed description of property sets, see *Integration Platform Technologies: Siebel eBusiness Application Integration Volume II*.

Siebel Business Process Designer provides capabilities to store property sets as process properties. You can use process properties to pass property sets to and from business service steps. Such process properties have a data type of hierarchy, and can be used as input and output arguments for any business service method arguments that have a data type of hierarchy.

When you want to call a workflow as a business service, you can map the data contained in the input and output property sets to and from process properties. This is useful when you want to run a workflow within a script.

When a workflow process is started, any process properties of type string, number, or date will be initialized to the input property with the same name, if one exists. Any hierarchy process properties will be initialized with any child input property sets that have a matching name (type). Any process properties with their Default String set to "<Value>" will be initialized with the value in the Value field of the input property set.

When a workflow process completes, all process properties of type string, number, or date will be stored as properties in the output property set. Any hierarchy process properties will be stored as child property sets. If a process property with the name <Value> is defined, its value will be stored in the Value field of the output property set.

For more information, see "Passing Parameters to and from Workflow with the Workflow Process Manager Business Service" on page 261.

Defining Process Properties

Process properties are the variable data you enter as you define workflow processes in the Process Designer within Siebel Tools. You use the WF Process Props applet in the Object List Editor (OBLE). You can define process properties step-by-step as you diagram a workflow process, or you can draw the workflow process diagram and then later define properties for the steps that make up the workflow process.

To define process properties

1. Depending on your situation, do one of the following:

- If in the process of creating a new workflow process, begin from Step 11 on page 54.
- If defining process properties for an existing workflow process, navigate to the Palette Designer for the workflow process. From within an existing workflow process, having the process selected in the OBLE’s Workflow Processes list applet, right-click and choose Edit Workflow Process.

2. In the Palette Designer, right-click and choose Show Process Properties.

   The WF Process Props applet appears below the Palette Designer.

3. In the WF Process Props applet, right-click and choose New Record.
4. Enter a name for the process property.

   NOTE: See “Naming Conventions for Workflow Processes and Process Properties” on page 54 for reserved symbols, such as the period (“.”) character, that cannot be used in a process property name.

5. Select a data type code from the picklist.

   The choices are:

   - **String.** If the property holds a character value.
   - **Number.** If the property holds a numeric value.
   - **Date.** If the property holds a date value.
   - **Hierarchy.** If the property holds hierarchical data (a property set).
   - **Binary.** If the property holds a binary value.
   - **Integration Object.** If the property holds an integration object.
   - **Alias.** If the property holds an XPath notation for pointing to a child in a hierarchical process property.

   NOTE: The default data type is String. Once a data type has been selected, it cannot be modified.

6. Enter a default string value, date value, or number value, if applicable.

   This is the value of the process property at the start of process execution.

7. Repeat Step 3 through Step 6 to define additional properties as necessary.

Concatenating Process Properties

You can use process property values in your expressions by concatenating workflow process properties with other process properties or with text. For example, you want to concatenate these process properties so that they appear as “Welcome to SupportWeb.”

- ProcessProperty1= "Welcome"
- ProcessProperty2="to"
- ProcessProperty3="SupportWeb"

You must create a ProcessProperty4 = “Welcome to SupportWeb.”

To concatenate process properties

- Define a Wait step with these values:
  - Output Argument = ProcessProperty4
  - Type = Expression
  - Value = [&ProcessProperty1]+' '+[&ProcessProperty2]+' '+[&ProcessProperty3]

    The ampersand (“&”) indicates that this is a process property. The process property you indicate can also be the name of a business component field. The process property cannot hold any binary data.

Field Descriptions for Defining Workflow Process Steps

Information on field descriptions is organized as follows:

Field Descriptions: WF Steps Applet

Table 6 describes the fields in which you enter data to define workflow process steps, in the WF Steps applet.

Table 6. WF Steps Applet Fields

Field	Type of Step	Description	Possible Value
Name	All	The name of the step.	A descriptive name that is: ■ Consistent with your overall naming strategy ■ Meaningful to the designer of the process ■ Unique **NOTE:** When you create a new step, the step is automatically assigned a unique name, with a unique number. You can change this name or leave it as is.
Type	All	The type of step.	This value is automatically entered when you create the step in the Process Designer view. This is a read-only field.
Business Component	Siebel operation step	Required. The business component that will perform the action you specify.	Choose from a list of business components that have been defined for the selected business object.
Business Service Name	Business service step	The name of the service to invoke.	The picklist displays business services existing in Siebel Tools with the Hidden flag set to FALSE. See "Enabling a Business Service for Workflow Processes" on page 40 for more information.
Business Service Method	Business service step	The name of the method to invoke on the service.	The picklist displays methods defined for the selected business service.
Subprocess Name	Subprocess step	The name of the subprocess step.	A descriptive name that is: ■ Consistent with your overall naming strategy ■ Meaningful to the designer of the process

Table 6. WF Steps Applet Fields

Field	Type of Step	Description	Possible Value
Error Code	Stop step	A number associated with a string in the database that comprises the error message.	Numeric value.
Error Message	Stop step	A string in the database that comprises the error message.	Text string.
User Interact View	User interact step	The name of the view for the user interact step.	Choose from a picklist containing predefined view names.
Operation	Siebel operation step	The type of operation.	Insert, Query, or Update.
Maximum Iterations	Wait step	The maximum number of times you can execute this step within a process instance.	When the maximum number of iterations is reached, an Object Manager error will be generated and the workflow process will return an In Error status. If you want the process to run to completion, you need to use a Workflow exception mechanism (such as an error process or exception branch) to catch and handle the error. For more information, see "Using Exceptions to Handle Errors" on page 126.
Service Hours	Wait step	The name of the schedule used in calculating the wait end time.	This value is selected from a picklist of service calendars.
Description	All	A text narrative describing the purpose of the step.	Free-form text.

Table 6. WF Steps Applet Fields

Field	Type of Step	Description	Possible Value
Update Snapshot	All	This parameter is used for recovery. Update Snapshot indicates that when the process reaches this step, Workflow takes a snapshot of the process state, so that if there is a crash, you can get the state back.	Check mark.
Processing Mode	Wait step	The mode in which the process will be run when triggered by run-time events.	(Optional) **Local Synchronous.** Executes the process in the application object manager. This is the default. **Remote Synchronous.** Submits a synchronous request to the Workflow Process Manager server component to execute the process. **Remote Asynchronous.** Submits an asynchronous request to the Workflow Process Manager server component to execute the process.

Field Descriptions: WF Step Branch Applet

NOTE: A branch can wait for multiple run-time events or a single user event. A branch cannot wait for a mixture of run-time events and user events. Specify only one event for a branch. Only long-running workflows can wait for user events. Never use run-time events in long-running workflows.

Table 7 describes the fields in the WF Step Branch applet.

Table 7. WF Step Branch Applet Fields

Field	Description	Possible Value
Name	The name of the branch.	The name of the branch must be unique or you will not be able to commit the record.
Type	The type of branch.	The value can be one of the following choices: **Default.** This value indicates that if nothing else is satisfied, this branch will be followed. Additionally, if this value is used, any conditions defined for the branch are ignored. **Condition.** This value indicates that a condition is defined for the branch. **Connector.** Use this value when there is no condition branching involved. **Error Exception.** Use this value to define exception handling. This connector type captures system errors, such as an error noting that the Assignment Manager server component is not available. For more information, see "Using Exceptions to Handle Errors" on page 126. **User Defined Exception.** Use this value to define exception handling. This connector type captures user-defined errors, such as an error noting that an order being submitted is incomplete. For more information, see "Using Exceptions to Handle Errors" on page 126.
Event Object Type	This field is used when defining a run-time event. The type of object to which the event occurs: an application, an applet, or a business component. **NOTE:** Event Object must be selected if Event Object Type exists.	(Optional) Application. Applet. BusComp.
Event Object	The name of the object (the application, applet, or business component) to which the event occurs.	Required if Event Object Type is specified. This is the name as defined in Tools. The set of objects is different for different object types.

Table 7. WF Step Branch Applet Fields

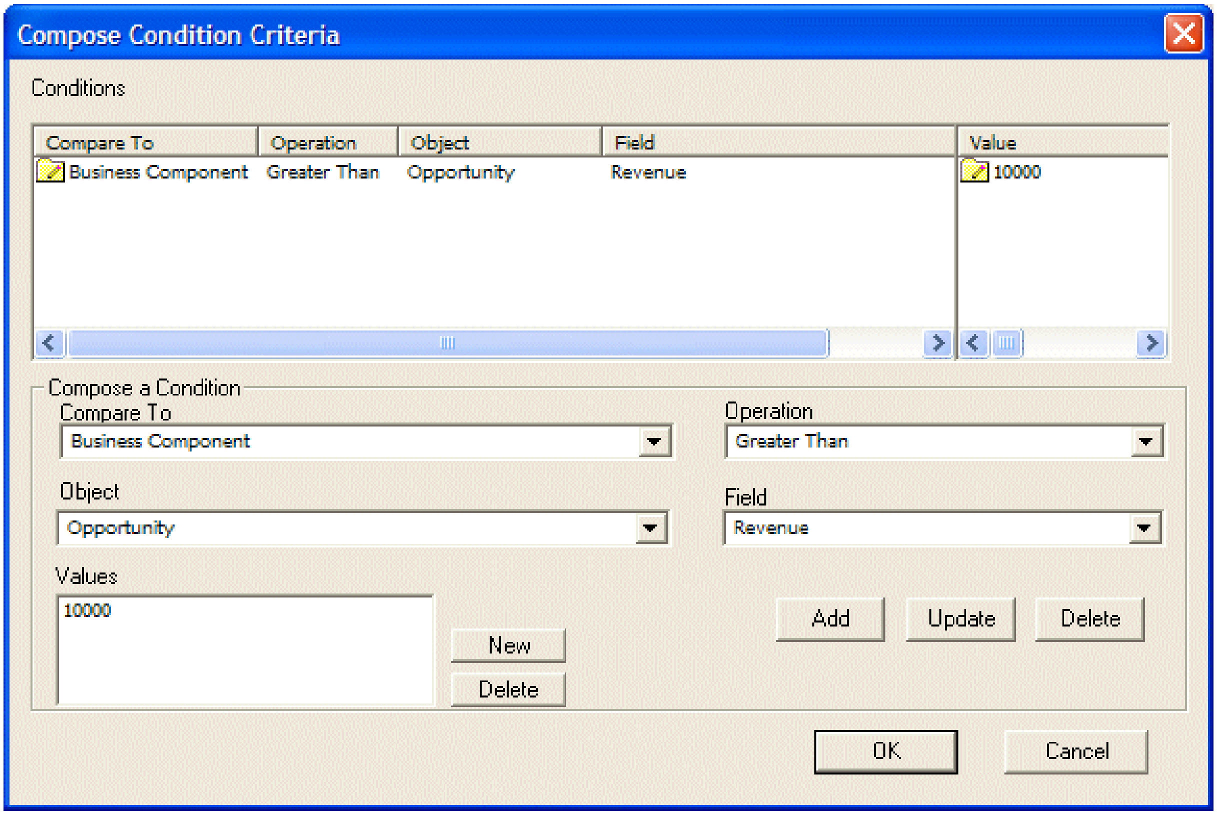
Field	Description	Possible Value
Event	The specific event that happens to the object.	Required if Event Object Type is specified. The set of available events is different for different object types.
Sub Event	An options parameter for the event, used when the object type is BusComp or Applet and the event is InvokeMethod or SetFieldValue. The subevent is the name of the method or business component field to be monitored.	(Optional) For InvokeMethod, the name of the method being invoked. For SetFieldValue, the name of the field being set.
Event Cancel Flag	Aborts the run-time event after executing the process. **NOTE:** If this flag is not checked, the error "The specialized method <Method Name> is not supported on this business component" will result when running the workflow process.	(Optional) This flag only applies to events that are cancelable. This flag functions like CancelOperation in scripting.
Expression	User-friendly text for the condition defined for the branch	Read-only
Event Visibility	Controls whether the workflow process waits for run-time events generated locally (that is, within the current session) or by any session.	If the workflow process is persistent, the visibility can be set to Local or Enterprise. If the workflow process is not persistent, visibility should be set to Local. **NOTE:** Setting Event Visibility to Enterprise means the run-time event occurring anywhere will search for any matching instance. This can cause a negative performance impact, so the Enterprise setting should be used sparingly.
User Event Name	An arbitrary string that denotes the name of a user event.	This can be any string and must be unique within the Siebel enterprise. **NOTE:** Make sure to give the User Event Name field a name that is unique and that is long enough to remain unique across the Siebel enterprise. Example: "Order Placed - Begin Processing Event for Service Request Automation - Version 2"

Table 7. WF Step Branch Applet Fields

Field	Description	Possible Value
User Event Storage	The process property that serves as the destination for the payload on the incoming user event.	This value can be any process property except the process property marked as the correlator.
User Event Timeout (Days)	The amount of time, in days, before the event times out.	Numeric value **NOTE:** If the user event is on a wait step, use this parameter. Do not specify a wait duration on the wait step.
Comments	Additional statements relative to the branch.	Free-form text.

Field Descriptions: Compose Condition Criteria Dialog Box

You set condition criteria for branches, decision steps, and user interact steps using the Compose Condition Criteria dialog box. Figure 10 shows an example of the Compose Condition Criteria dialog box.



Compose Condition Criteria

Conditions

Compare To	Operation	Object	Field	Value
Business Component	Greater Than	Opportunity	Revenue	10000

Compose a Condition

Compare To: Business Component

Operation: Greater Than

Object: Opportunity

Field: Revenue

Values: 10000

Figure 10. Compose Condition Criteria Dialog Box

Table 8 describes the fields that you use to define criteria for condition connectors, in the Compose Condition Criteria dialog box.

NOTE: The values you find listed in the Compose Condition Criteria dialog box are constrained by the business object for your workflow process, which is specified at the process level. For more information, see "Using Process Designer in Siebel Tools" on page 46 and "Field Descriptions: Workflow Processes Applet" on page 47.

Table 8. Compose Condition Criteria Fields

Field	Description	Possible Value
Compare To	Indicates where the comparison value is coming from.	This is a required field, with the following choices: ■ Business Component ■ Process Property ■ Expression ■ Applet
Operation	Identifies the comparison operation.	**This Must Match.** The current value must match exactly, including case. **One Must Match.** One or more values must match exactly, including case. **All Must Match.** All of the values must match exactly, including case. **None Can Match.** None of the values can match exactly, including case. **This Must Match (ignore case).** The current value must match without regard to case. **One Must Match (ignore case).** One or more values must match without regard to case. **All Must Match (ignore case).** All of the values must match without regard to case. **None Can Match (ignore case).** None of the values can match without regard to case. **Greater Than.** Value must be greater than the comparison value. **Less Than.** Value must be less than the comparison value. **Between.** Value must be between a range of values. **Not Between.** Value cannot be between a range of values. **Is Null.** Value must be null. **Is Not Null.** Value cannot be null.

Table 8. Compose Condition Criteria Fields

Field	Description	Possible Value
Object	The name of the associated business object.	This value is selected from a picklist of business objects.
Field	This is a required field when Applet is the Compare To value.	The name of the field within the named applet.
Values		The Values field is dynamic based on the Compare To field. The Values field is for storing data to be used in the condition evaluation.

About Start Steps

Start steps identify the input conditions that must be met for a process to execute. For example, to handle open service requests, you could define a start condition of “Status = Open.”

The main parts of defining Start steps for a workflow process are:

- “Defining a Start Step” on page 81
  - See “Field Descriptions: WF Steps Applet” on page 73 for descriptions of all step field values.
- Define the start step branches, conditions, and values.
  - See “Field Descriptions: WF Step Branch Applet” on page 75.
  - See “Field Descriptions: Compose Condition Criteria Dialog Box” on page 79.
  - See “Defining Conditions and Values for Branches, Decision Steps, and User Interact Steps” on page 82.

For more information on workflow process field descriptions, see “Field Descriptions: Workflow Processes Applet” on page 47 and “Field Descriptions: WF Process Props Applet” on page 49.

Defining a Start Step

You define a Start step in the Process Designer in Siebel Tools.

To define a Start step

1. In the Workflow Processes OBLE in Siebel Tools, select the workflow process for which you would like to define a Start step or create a new workflow process.
2. Right-click and choose Edit Workflow Process.
3. Drag and drop a Start icon from the palette to the workspace.

4 In the WF Steps applet, enter or modify the step name, and then enter a description of the purpose of the Start step.

5 See "Defining Next Step Branches for Start Steps" on page 82 for instructions on setting up branches for the Start step.

Defining Next Step Branches for Start Steps

Start steps can have multiple next step branches. Use the following procedure to define each branch.

NOTE: In this release, Workflow processes do not support parallel processing. Make sure that you define your conditions so that only one branch is valid. If an object matches the conditions in multiple branches, Workflow will take the first condition that passes and execute it. Workflow will assume that the other conditions fail.

To define a next step branch

1 In the Process Designer, drag and drop the appropriate step icon for the next step in the process.

   NOTE: If you have already designed the entire workflow process, including connector arrows, click the connector arrow attached to the Start step, then proceed to Step 4.

2 Drag and drop a connector arrow to the workspace, connecting the Start step with the new next step.

3 Click the connector arrow to access the WF Step Branch applet.

4 Enter or modify the branch name.

   NOTE: The name of the branch must be unique or you will not be able to commit the record.

5 Choose a branch type. Type values are described in Table 7 on page 76.

   CAUTION: Always define a Default branch step in case some work items do not meet any of the conditions you define.

6 Enter any comments.

7 See "Defining Conditions and Values for Branches, Decision Steps, and User Interact Steps" on page 82 to define the conditions that apply to each branch.

Defining Conditions and Values for Branches, Decision Steps, and User Interact Steps

Conditions and values affect the flow of your process execution. For example, you can define a condition based on the value of a priority field. If the priority is "high," the process follows a branch that sends an email to a vice president. If the priority is "medium," the email is sent to an engineer.

You define conditions and values for branches, decision steps, and user interact steps.

To define conditions and values

1 Right-click the appropriate branch in the Process Designer palette workspace and choose Edit Conditions.

The Compose Condition Criteria dialog box appears.

NOTE: The values you find listed in the Compose Condition Criteria dialog box are constrained by the business object for your workflow process, which is specified at the process level. For more information, see "Using Process Designer in Siebel Tools" on page 46 and "Field Descriptions: Workflow Processes Applet" on page 47.

2 In the Compare To picklist, choose a Compare To value:

- **Process Property.** Compares a process instance's process property value to a specified value.
- **Expression.** Uses an expression to evaluate a specific value.
- **Business Component.** Uses the value in a business component field for the condition comparison or when you are defining an expression.
- **Applet.** Uses the value in an applet field for the condition comparison.

3 Choose the operation to use for evaluating the values:

- **This Must Match.** The current value must match exactly, including case.
- **One Must Match.** One or more values must match exactly, including case.
- **All Must Match.** All of the values must match exactly, including case.

  NOTE: The All Must Match operation is only applicable in a scenario where a workflow branch must make a decision based on a field of a child business component. The name of the child business component is specified by the Business Component Name field of the branch. The child business component is different from the primary business component of the workflow process. It is possible for the child business component to contain multiple records. If the workflow branch uses the All Must Match comparison criterion, the branch will pass only when the field values of all child business component records match branch criteria value(s) of the branch.

- **None Can Match.** None of the values can match exactly, including case.
- **This Must Match (ignore case).** The current value must match without regard to case.
- **One Must Match (ignore case).** One or more values must match without regard to case.
- **All Must Match (ignore case).** All of the values must match without regard to case.
- **None Can Match (ignore case).** None of the values can match without regard to case.
- **Greater Than.** Value must be greater than the comparison value.
- **Less Than.** Value must be less than the comparison value.
- **Between.** Value must be between a range of values.
- **Not Between.** Value cannot be between a range of values.
- **Is Null.** Value must be null.

- **Is Not Null.** Value cannot be null.

4 Enter an object name and field, if applicable.

5 Enter any appropriate values in the Values form.

   You can enter multiple records in the Values form. Workflow Processes assumes an OR condition between values.

If you selected Expression in the Compare To field, enter your expression in the Values form. The syntax is the same as the syntax used in Siebel Tools. For more information about expressions, see *Siebel Developer's Reference*.

NOTE: You can define multiple conditions for each branch. Workflow Processes treats multiple conditions with the AND operator. To define multiple OR conditions, use expressions.

The following example shows an expression comparing a business component field with today's date, using the OR operator, which allows you to compare multiple conditions.

```
([Close Date] <= Today()) OR ([Name] = 'Opportunity test1')
```

About Decision Steps

Decision steps are a type of step that evaluate one or more defined conditions to determine the next step of a process instance.

The main parts of creating Decision steps for a workflow process are:

- "Defining a Decision Step" on page 84.
- "Defining Decision Branches" on page 85.
- Defining the conditions and values. See "About Conditions and Values for Decision Steps" on page 86 and "Defining Conditions and Values for Branches, Decision Steps, and User Interact Steps" on page 82.

Defining a Decision Step

You define a Decision step in the Process Designer in Siebel Tools.

To define a decision step

1 In the Workflow Processes OBLE in Siebel Tools, select the workflow process for which you would like to define a Decision step.

2 Right-click and choose Edit Workflow Process.

3 Drag and drop a Decision Point icon from the palette to the workspace.

4 In the WF Steps applet, enter or modify the step name, and then enter a description of the purpose of the Decision step.

   NOTE: See "Defining Decision Branches" on page 85 to create the branches for the decision step.

Defining Decision Branches

If you connected the decision steps to the next steps in the process with connector arrows, as described in "Diagramming a Workflow Process" on page 66, branches are automatically created and appear in the Next Step applet. If this is the case, modify the branch fields as necessary and then go to "About Conditions and Values for Decision Steps" on page 86 for the procedure on defining conditions and values for each branch.

To define a decision branch

1. Drag and drop a connector arrow to the workspace, connecting the Decision step with the new next step.
2. Click the connector arrow to access the WF Step Branch applet.
3. Enter or modify the branch name.

   NOTE: The name of the branch must be unique or you will not be able to commit the record.

4. Choose a branch type. Type values are described in Table 7 on page 76.

   CAUTION: Always define a Default branch step in case some work items do not meet any of the conditions you define.

5. Enter any comments.
6. See "Defining Conditions and Values for Branches, Decision Steps, and User Interact Steps" on page 82 to define the conditions that apply to each branch.
7. Verify in the Process Designer that the decision point branch is connected with a connector arrow to the appropriate next step.

   The handles of the connector arrow will be red if they are correctly attached to the steps.

   NOTE: Decision points can have multiple decision point branches. Repeat Step 1 on page 85 through Step 7 for additional branches.

8. Go to the next procedure, "Defining Conditions and Values for Branches, Decision Steps, and User Interact Steps" on page 82 to define the conditions that apply to each branch.

For more information, see "About Conditions and Values for Decision Steps" on page 86.

About Conditions and Values for Decision Steps

Conditions and values affect the flow of your process execution. Different actions may occur depending on which path is followed. For example, you can define a condition based on the value of a priority field, so that if the priority is equal to "high," the process execution follows a branch leading to an action that sends an email to a vice president. However, if the priority is equal to "medium," the email is sent to an engineer.

NOTE: A decision step evaluates a record's business component value at the time the workflow process is executed. If the workflow process is triggered by a workflow policy and multiple violations of a policy condition occur within the Workflow Monitor Agent's action interval, then at the time the workflow process is executed, the decision step determines which branch to take based on the current value of the business component field. If the decision branch criterion is moved from the workflow process level to the workflow policy level, then the policy generates unique events within the defined action interval. In this way, the workflow process is triggered for all violations.

For more information, see "Defining Conditions and Values for Branches, Decision Steps, and User Interact Steps" on page 82.

About Business Service Steps

Business services allow you to execute predefined or custom actions in a workflow process. Some examples of predefined business services include:

- **Notifications.** Notifications can be sent to employees or contacts using the Outbound Communication Server business service.
- **Assignment.** Assignment Manager can assign an object in a workflow process by calling the Synchronous Assignment Manager Request business service.
- **Server tasks.** You can run a server component task using the Asynchronous or Synchronous Server Requests business service.

For a list of some of the most commonly-used predefined business services, see "Predefined Business Services" on page 253.

You can use Siebel VB or Siebel eScript to define your own custom business services that you can invoke from a workflow process. You can define business services by navigating to Site Map > Administration - Business Service, or by selecting the business service object in Siebel Tools. The methods and arguments you define in your business service appear in the picklists in the Arguments list applets for the business service.

The main parts of creating Business Service steps for a workflow process are:

- "Defining a Business Service Step" on page 89
- "Defining Input Arguments for Business Service Steps" on page 90
- "Defining Output Arguments for Business Service Steps" on page 90

CAUTION: Business services invoked from workflow processes cannot include browser scripts; they only work with server scripts. A business service with browser scripts will fail if it is executed from a workflow process on the Siebel Server.

Field Descriptions: Input Arguments for Business Service Steps, Subprocess Steps, and Wait Steps

Input arguments allow you to define values that you want to pass to a service method. Many methods require input arguments. Table 9 describes the fields in the Input Arguments applet.

Table 9. Input Arguments Applet Fields

Field	Description	Possible Value
Input Argument	*For business service steps only:* the name of the input argument.	This field is required. The picklist displays input arguments existing for the selected business service method. A method argument appears in this picklist if it has been defined as a business service method argument, the Hidden flag is set to FALSE, and the type is input or input/output.
Subprocess Input	*For subprocess steps only:* the name of the input argument.	This field is required.
Type	The type of argument.	This is a required field. The picklist contains the following choices: ■ Literal ■ Process Property ■ Business Component ■ Expression
Value	A string value.	For Literal and Expression type input arguments. This could be a picklist, depending on the argument selected. String values can only be a maximum of 32,767 characters.
Property Name	The name of the business process property.	For Process Property–type input arguments.
Business Component Name	The name of a business component within the business object of the business process.	For Business Component–type input arguments.
Business Component Field	The name of a field within the business component.	For Business Component Field–type input arguments.
Changed		Checkmark.

Field Descriptions: Output Arguments for Business Service Steps, Subprocess Steps, and Siebel Operation Steps

Output arguments are the result of a business service method. Output arguments should be stored in process properties.

Table 10 describes the fields in the Output Arguments applet.

NOTE: Calculated fields are unavailable as values for input or output arguments. If you want to use a calculated value, use an expression.

Table 10. Output Arguments Applet Fields

Field	Description	Possible Value
Property Name	The name of the Process Property to store the results.	This is a required field. This is a picklist of properties that have been defined for the process. For more information about defining process properties, see "Defining Process Properties" on page 70.
Type	The type or argument.	This is a required field. The picklist contains the following choices: - Literal - Output Argument - Business Component - Expression
Value	A string value.	For Literal or Expression arguments. Note that string values can only be a maximum of 32,767 characters.
Output Argument	*For business service steps only:* the name of the output argument.	For Output Arguments type. This is a picklist of output arguments for the selected method. An argument appears in this picklist if it has been defined as a business service method argument, the Hidden flag is set to FALSE, and the type is Output or Input/Output.
Subprocess Output	*For subprocess steps only:* the name of the output argument.	For Output Arguments type.

Table 10. Output Arguments Applet Fields

Field	Description	Possible Value
Business Component Name	The name of the business component within the business object of the business process.	For Business Component type.
Business Component Field	The name of a field within the business component.	For Business Component Field type. **NOTE:** Business component fields based on multi-value groups cannot be selected as values for input or output arguments. If you want to use a field based on a multi-value group, you need to define a business component for the field and link it to the appropriate business object. See *Configuring Siebel eBusiness Applications* for more information.

Defining a Business Service Step

You define a business service step in the Process Designer in Siebel Tools.

To define a business service step

1 In the Workflow Processes OBLE in Siebel Tools, select the workflow process for which you would like to define a Business Service step.

2 Right-click and choose Edit Workflow Process.

3 Drag and drop a Business Service icon from the palette to the workspace.

4 In the WF Steps applet, enter or modify the step name, and then enter a description of the purpose of the Business Service step.

5 In the Business Service Name field, select the name of the service to be invoked from the picklist.

   The picklist contains the business services defined in Siebel Tools or the Siebel client.

   See *Integration Platform Technologies: Siebel eBusiness Application Integration Volume II* for information on creating customer-defined services.

6 In the Business Service Method field, select the method for invoking the service. The choices available for this field depend on the service you select in Step 5.

7 If you need to define input arguments for this task, continue to "Defining Input Arguments for Business Service Steps."

8 If you need to define output arguments for this task, continue to "Defining Output Arguments for Business Service Steps" on page 90.

Defining Input Arguments for Business Service Steps

You define input arguments in the Process Designer in Siebel Tools.

To define input arguments for a business service step

1. With the appropriate business service step selected in the Process Designer palette workspace, right-click and choose Show Input Arguments.
2. In the Input Arguments applet, right-click and choose New Record.
3. Complete the fields. For field descriptions, see "Field Descriptions: Input Arguments for Business Service Steps, Subprocess Steps, and Wait Steps" on page 87.

Defining Output Arguments for Business Service Steps

Output arguments allow you to store a resulting value in a process property.

To define output arguments for a business service step

1. With the appropriate business service step selected in the Process Designer palette workspace, right-click and choose Show Output Arguments.
2. In the Output Arguments applet, right-click and choose New Record.
3. Complete the fields. For field descriptions, see "Field Descriptions: Output Arguments for Business Service Steps, Subprocess Steps, and Siebel Operation Steps" on page 88.

NOTE: Business services, methods, and arguments all have Display Name and Hidden properties in Siebel Tools. For a business service, method, or argument to be displayed on any picklist, the Hidden flag for the object must be set to FALSE. For more information, see "Considering Business Objects and Business Services When Planning Workflow Processes" on page 39.

About Subprocess Steps

A subprocess step allows you to invoke a separate process within a process. A process definition can have one or more subprocess steps.

The main parts of creating a subprocess step for a workflow process are:

- "Defining a Subprocess Step" on page 91
- "Defining Input Arguments for Subprocess Steps" on page 91
- "Defining Output Arguments for Subprocess Steps" on page 91
- "Defining Recipients for Subprocess Steps" on page 92

Defining a Subprocess Step

Before you define a subprocess step, you must define the workflow process you will call with the step.

To define a subprocess step

1. In the Workflow Processes OBLE, verify that the workflow process that is to be called as a subprocess is defined, or create it.
2. Select the workflow process to which you want to add a subprocess step.
3. Right-click and select Edit Workflow Process.
4. Drag and drop a Subprocess icon from the palette to the workspace.
5. In the WF Steps applet, enter or modify the step name, and then enter a description of the purpose of the subprocess step.
6. In the Subprocess Name field, select the process that the subprocess step will call.
7. If you need to define input arguments for this subprocess, continue to "Defining Input Arguments for Subprocess Steps" on page 91. If you need to define output arguments for this subprocess, continue to "Defining Output Arguments for Subprocess Steps" on page 91.

Defining Input Arguments for Subprocess Steps

Input arguments allow you to populate process properties in the subprocess. For example, if you want to pass the object ID from the main process to the subprocess, you would do this through input arguments. If the subprocess is based on a different business object, you must pass the relevant row ID of the target object as the subprocess Object ID Process Property.

For field descriptions, see "Field Descriptions: Input Arguments for Business Service Steps, Subprocess Steps, and Wait Steps" on page 87.

To define input arguments for a subprocess step

1. With the appropriate subprocess step selected in the Process Designer palette workspace, right-click and choose Show Input Arguments.
2. In the Input Arguments applet, right-click and choose New Record.
3. Complete the fields. For field descriptions, see "Field Descriptions: Input Arguments for Business Service Steps, Subprocess Steps, and Wait Steps" on page 87.

Defining Output Arguments for Subprocess Steps

Output arguments allow you to store a resulting value in a process property.

To define output arguments for a subprocess step

1. With the appropriate subprocess step selected in the Process Designer palette workspace, right-click and choose Show Output Arguments.
2. In the Output Arguments applet, right-click and choose New Record.
3. Complete the fields. For field descriptions, see "Field Descriptions: Output Arguments for Business Service Steps, Subprocess Steps, and Siebel Operation Steps" on page 88.

Defining Recipients for Subprocess Steps

1. With the appropriate subprocess step selected in the Process Designer palette workspace, right-click and choose Show Recipients.
2. In the WF Step Recipients applet, right-click and choose New Record.
3. Complete the fields. For field descriptions, see "Field Descriptions: WF Step Recipients Applet" on page 92.

Field Descriptions: WF Step Recipients Applet

Table 11 describes the fields in the WF Step Recipients applet.

Table 11. WF Step Recipients Applet Fields

Field	Description	Possible Value
Recipient Type Code	The type of recipient.	This value is fixed as "User" and cannot be changed.
Value Type Code	The source from which the recipient value comes.	This field is a picklist with the following values: ■ Name ■ Business Component ■ Process Property ■ Expression
Recipient Name	The name of the recipient. This field is a pick applet which displays the first name, last name, and login name of all users in the database. **NOTE:** Complete this field if the Value Type Code is set to Name.	Choose one name from the list of all users available in the database.

Table 11. WF Step Recipients Applet Fields

Field	Description	Possible Value
Business Component Name	The name of the business component. **NOTE:** Complete this field if the Value Type Code is set to Business Component.	Choose one business component.
Business Component Field	The name of the business component field. **NOTE:** Complete this field if the Value Type Code is set to Business Component.	Choose one business component field.
Process Property Name	The name of the process property. **NOTE:** Complete this field if the Value Type Code is set to Process Property.	Choose one process property.
Expression	If the recipient value is derived from an expression, the expression is entered in this field. **NOTE:** Complete this field if the Value Type Code is set to Expression.	The expression from which the recipient value is derived.

Field Descriptions: Subprocess Applet

Table 12 describes the fields in which you enter data in the Subprocess applet.

Table 12. Subprocess Applet Fields

Field	Description	Possible Value
Step	The name of the subprocess step.	A descriptive name that is: ■ Consistent with your overall naming strategy ■ Meaningful to the process designer
Type	The type of step.	This value is automatically entered when you create the step on the Process Designer view. Read-only.
Description	A text narrative describing the purpose of the subprocess.	Free-form text.

Table 12. Subprocess Applet Fields

Field	Description	Possible Value
Subprocess	The name of the process to run.	This value is selected from a picklist of defined workflow processes. **NOTE:** In order for the subprocess to appear in this picklist of defined workflow processes, the subprocess must have a status of Complete. See "Activating Workflow Processes" on page 149.
Created By	The name of the person who creates the step.	This value is automatically entered based on the log in name of the user.
Created	The date that the step is created.	This value is automatically entered.

About Siebel Operation Steps

Siebel Operation steps include operations such as Insert, Update, or Query. These steps are performed on business components.

After you define a Siebel Operation step, you can use the Fields applet to define any field values for the step. For the Update step, you can use the Search Specifications applet to define which records you want to update.

Information about Siebel Operation steps is organized as follows:

- "Defining a Siebel Operation Step" on page 94.
- "Defining Fields for a Siebel Operation Step" on page 95
- "Defining Siebel Operation Search Specifications" on page 96
- "Defining Siebel Operation Step Output Arguments" on page 97
- "Field Descriptions: Search Specifications" on page 98
- "Updating a Field Based on a Multi-Value Group" on page 98

Defining a Siebel Operation Step

NOTE: After executing an Insert step, the Siebel Operation object ID process property automatically stores the row ID of the record that was created.

You can define Siebel Operation steps for any business component associated with the business object selected for the process. If you want to update a business component not associated with the business object, you can either invoke a subprocess or associate the business component to the business object using Siebel Tools.

All fields are available for update and insert except fields based on multi-value groups and calculated fields. If you want to update a field based on a multi-value group, you can define a business component for the field and link the business component to the object using Siebel Tools. An example is an update to an Account Team. Account Team is based on a multi-value group, so it cannot be updated by selecting the Account business component. However, you can create a business component called "Account Team" and then associate it with the Account business object using Siebel Tools. You could then select Account Team as the business component to update with the Siebel Operation step.

To define a Siebel operation

1. Select the appropriate workflow process in the Workflow Processes OBLE.
2. Right-click and choose Edit Workflow Process.
3. Drag and drop the Siebel Operation icon from the palette to the workspace.
4. In the WF Steps applet, enter a name for the step, and a description of the purpose of the step.
5. Select the type of operation. The available choices are:
   - Insert
   - Update
   - Query

   NOTE: Verify that updates or inserts of fields that have dependencies are valid fields. For example, if you have a service request process and your process is updating the area and sub-area fields, you will need to make sure that the values selected for the subarea field are valid for that associated area.
6. Select the name of the business component.
7. If you need to define fields for this Siebel operation, continue to "Defining Fields for a Siebel Operation Step" on page 95.
8. If you need to define search specifications for this Siebel operation, continue to "Defining Siebel Operation Search Specifications" on page 96.
9. If you need to define output arguments for this Siebel operation, continue to "Defining Siebel Operation Step Output Arguments" on page 97.

Defining Fields for a Siebel Operation Step

NOTE: If the Siebel Operation step will perform an insert operation, make sure that all required fields have been added to the Siebel Operation step. System fields and predefaulted fields are automatically populated.

To define fields for a Siebel operation step

1. With the appropriate Siebel operation step selected in the Process Designer, right-click and choose Show Fields.

2 In the Input Arguments applet, right-click and choose New Record.

3 In the Field Name field, select the name of the field to be updated.

4 In the Type field, choose an input argument type. The choices available are:

   - Literal
   - Process Property
   - Business Component
   - Expression

5 If the field type selected is Literal, enter a value.

6 If the field type is a Process Property, select a property name.

7 If the field type is a Business Component, select the applicable business component name and business component field.

8 If the field type is an Expression, enter the expression in the value field.

9 Enter any appropriate comments.

Defining Siebel Operation Search Specifications

You can define search specifications to identify the specific data on which to perform the operation. Search specifications are used when the business component has multiple records and you want to perform the operation on only some of the records. For example, if you have a process for the Account object and you want to update only those Opportunities with a lead quality of Poor, you would define search specifications to access only those Opportunities.

To define Siebel operation search specifications

1 With the appropriate Siebel operation step selected in the Process Designer, right-click and choose Show SearchSpec.

2 In the Search Specifications applet, right-click and choose New Record.

3 In the Type field, select a search specification type. The choices available are:

   - Literal
   - Expression

4 In the Search Specification field, enter search specifications.

   CAUTION: Define your Siebel operation search specification as efficiently as possible, so that only the smallest necessary set of rows will match. Search specifications that select a large set of rows could cause severe performance degradation.

5 If the search specification type is expression, select the applicable business component name.

NOTE: A search specification of type Literal is executed as written. For example, [Status] LIKE ‘*Open*’. A search specification of type Expression allows you to construct a search specification dynamically. For example, “[Contact ID] = ‘ ” + [&New ID] + “ ‘ ” will be evaluated to [Contact ID] = ‘1 - ABC’ if the New ID process property is 1 - ABC at run time.

Defining Siebel Operation Step Output Arguments

Output arguments allow you to store a resulting value in a process property. This value can then be passed to other processes.

To define output arguments for an operation step

1. With the appropriate Siebel operation step selected in the Process Designer, right-click and choose Show Output Arguments.
2. In the Output Arguments applet, right-click and choose New Record.
3. In the Property Name field, select the property name from the picklist.
4. In the Type field, choose an output argument type. The choices available are:
   - Literal
   - Output Argument
   - Expression
   - Business Component
5. If the output argument type you chose is
   - Literal, then enter a value.
   - Output Argument, then enter the argument.
   - Business Component, then select the applicable business component name and business component field.
6. Enter any appropriate comments.

Field Descriptions: Search Specifications

Table 13 describes the fields in the Search Specifications list. Use these fields to define Siebel operation search specifications.

Table 13. Search Specification Fields

Field	Description
Expression Business Component	If you entered Expression in the Type field, enter the name of the business component that will evaluate the expression. For example, in the Search Specification field, you can enter: "[Due Date] < '" + [Order Date] + "'" The Expression business component evaluates Order Date so that the search specification becomes: [Due Date] < '07/04/2001 18:51:26'
Filter Business Component	Enter the name of the business component that will provide the group of records on which you will perform your search.
Search Specification	If you entered Literal in the Type field, enter a literal value in the form of an expression. For example, "= 100". If you entered Expression in the Type field, enter an expression such as [Status] LIKE '*Open*'. The expression will be evaluated by the Expression business component you specify.
Type	Required. Choose the type of value on which to base your search: Literal or Expression.
Comments	Enter a text description of the purpose of the search.
Changed	Checkmark indicates a changed search specification.

Updating a Field Based on a Multi-Value Group

Calculated fields cannot be updated using Siebel Operation steps because typically they require values from other business component fields. Use expressions to perform calculations.

The object ID for the process is automatically passed to Siebel Operation steps. Because this automatic passing occurs, you do not need to enter a search specification value unless you are updating child records. For example, if you have a process based on the service request object and you want to update the service request, you do not need to enter a search specification. However, if you want to update activities for the service request, you may want to enter a search specification to query the specific activity that you want to update. Otherwise, the update step updates all activities for the service request.

The object ID cannot be null if you are executing a Siebel Operation, unless you are inserting into the primary object ID. If the process has no object ID, the Siebel Operation step returns an error.

When performing a query operation for child records, the Siebel Operation Object ID process property field will return the row ID if one record matches, a "*" if multiple records match, and Null/ no value if no records match.

NOTE: The *only* ability provided is to return a row ID of a matching row.

The insert/update operation updates the Siebel Operation Object ID process property field of the record's row ID.

NOTE: The Workflow Policy programs and Siebel Operation steps use different object layers to update data. For example, you may have a Workflow Policy that calls a Workflow Policy Program to update a Service Request Record. This method goes through the Data Layer in which State Model does not apply.

Conversely, if you have a Workflow Policy that calls a Workflow Process Action and in the Workflow Process, you have defined a Siebel Operation step to update a Service Request Record, this method goes through the Object Layer in which the State Model does apply.

About Wait Steps

Wait steps allow you to suspend process execution for a specific period of time or until a specific event occurs. Workflow administrators can specify to pause a process instance in units of seconds, minutes, hours, or days. In addition, administrators can specify a service calendar to account for business hours and days when waiting a specified duration.

If a workflow process includes a wait step, by default it is persisted.

The main parts of creating a wait step for a workflow process are:

- "Defining a Wait Step" on page 99

Defining a Wait Step

You can define a wait step to pause a process instance.

To define a wait step

1. Select the appropriate workflow process in the Workflow Processes OBLE.
2. Right-click and choose Edit Workflow Process.
3. Drag and drop the Wait icon from the palette to the workspace.
4. In the WF Steps applet, enter a name for the step, and a description of the purpose of the step.
5. Select the Wait icon, right-click and choose Show Input Arguments.

6. In the Input Arguments applet, enter input arguments. See "To define input arguments for a wait step" on page 100.

   NOTE: For durations greater than 60 seconds, specify minutes or a greater unit of measure so that business component data is refreshed. Workflow is resumed from Workflow Process Manager when units of minutes or higher are specified. Wait steps with durations measured in anything other than seconds are automatically persisted.

To define input arguments for a wait step

1. In the Input Arguments applet, right-click and choose New Record.
2. Select the property name from the picklist.
3. Choose an input argument type. The choices available are:
   - Literal
   - Input Argument
   - Expression
   - Business Component
4. If the input argument type you chose is
   - Literal, then enter a value.
   - Input Argument, then enter the argument.
   - Business Component, then select the applicable business component name and business component field.
5. Enter any appropriate comments.

About User Interact Steps

The user interact step allows application designers to design and configure the flow of Siebel views within an application. Siebel Workflow guides end users through a specified flow of Siebel views based on the users' actions, or executes a specified set of actions. This flow can be modified as business rules change.

The user interact step has the following behaviors:

- The user interact step brings up the required view. The user interact step runs in the user session. It sends a request to the Siebel Web Engine to build the view.

  NOTE: Only one view can be built at a time. You cannot combine a user interact step with another action, such as bringing up a message box or building another view simultaneously.

- The user interact step waits for a run-time event to resume processing. It waits in the memory of the user session. In cases where there is no run-time event defined, the workflow process will continue to the end.

- If, after a user interact step, the user manually navigates out of the view, the workflow process remains in the memory of the user session. The process is deleted when the user session is terminated or when another workflow process is instantiated in the same user session.

NOTE: Workflows running in the Workflow Process Manager server component should not contain user interact steps. That is, if the workflow is running in background mode or in batch mode, it cannot include user interact steps. If the Workflow Process Manager encounters a user interact step, an error will result.

CAUTION: If using the workflow persistence feature with user interact steps, note that workflows with user interact steps should be persisted only for debugging purposes. A persisted workflow that contains a user interact step can resume in a different user session based on which user generated the event, causing unintended behavior. For more information on persistence, see "About Workflow Persistence" on page 117.

The User Interact step has been enhanced to take process properties as input arguments. In this way, you can dynamically set view names as you design your interactive workflow processes. The view name property is set in the view field (an unbounded picklist) of the User Interact step.

For more information on user interact steps, see the following sections:

- "Defining a User Interact Step" on page 101
- "Defining User Interact Next Step Branches" on page 102
- "Defining Conditions and Values for Branches, Decision Steps, and User Interact Steps" on page 82

Defining a User Interact Step

You define a user interact step in the Process Designer in Siebel Tools.

To define a user interact step

1. Select the appropriate workflow process in the Workflow Processes OBLE.
2. Right-click and choose Edit Workflow Process.
3. Drag and drop the User Interact icon from the palette to the workspace.
4. In the WF Steps applet, enter a name for the step, and a description of the purpose of the step.
5. From the User Interact View picklist, select the view name to which you would like to navigate the user. Only views associated with the business object will be available in the picklist.
6. Enter a description of the purpose of the user interact step.

NOTE: The user interact step is only supported if the process is invoked through a script or run-time event and the process is run locally in the application object manager.

Defining User Interact Next Step Branches

User interact steps can have multiple next step branches. Use the following procedure to define each branch.

NOTE: In this release, Workflow processes do not support parallel processing. Make sure that you define your conditions so that only one branch is valid. If an object matches the conditions in multiple branches, it will try to take all branches one at a time in a random order until the first End step is reached.

To define a next step branch

1. From the Process Designer palette workspace, drag and drop the appropriate step icon for the next step in the process.

   NOTE: If you have already designed the entire workflow process, including connector arrows, click the connector arrow attached to the user interact step, then proceed to step 4.

2. Drag and drop a connector arrow to the workspace, connecting the user interact step with the new next step.

3. Enter or modify the branch name.

   NOTE: The name of the branch must be unique or you will not be able to commit the record.

4. Choose a branch type. Type values are described in Table 7 on page 76.

   CAUTION: Always define a Default branch step in case some work items do not meet any of the conditions you define.

5. Enter comments, if applicable.

6. See "Defining Conditions and Values for Branches, Decision Steps, and User Interact Steps," to define the conditions that apply to each branch.

About Conditions and Values for User Interact Next Step Branches

Conditions and values affect the flow of your process execution. Different actions may occur depending on which path is followed. For example, you can define a condition based on the value of a priority field, so that if the priority is equal to High, the process execution follows a branch leading to an action that sends an email to a vice president. However, if the priority is equal to Medium, the email is sent to an engineer.

For more information, see "Defining Conditions and Values for Branches, Decision Steps, and User Interact Steps" on page 82.

Creating Substitute View Names with Process Properties

You can associate view names with process properties so that they can be set dynamically at run time. You do this by assigning a view name to the run-time value of a process property.

To associate a view name with a process property to set view names dynamically at run time

- In the Interact Navigate View field of a user interact step, type the following string:
  - [&ProcessPropertyName]

The Workflow engine recognizes this string and assigns the view name at run time.

About Stop Steps

Stop steps are used to raise an error to the user and terminate the workflow process instance.

The main parts of creating a stop step for a workflow process are:

- Define a stop step. See "Defining a Stop Step" on page 103.
- Define input arguments for the stop step. See "Defining Stop Step Input Arguments" on page 104.

Table 14 describes how the Stop step is handled, depending on how it is called and in which object manager it is running.

Table 14. How Workflow Process Manager Handles Stop Steps

Stop Step Conditions		Results
Workflow policy calls a process that contains a Stop step.		Workflow Process Manager: ■ Exits ■ Writes an error message to the log file.
A script or run-time event calls a process that contains a Stop step.	■ Process is running in the Workflow Process Manager object manager.	Workflow Process Manager: ■ Writes an error message to the log file.
	■ Process is running in the application object manager.	Workflow Process Manager: ■ Flags an error message to the user.

Defining a Stop Step

It is recommended that the Stop step be used only in Workflow Processes invoked from a script.

To define a stop step

1 Select the appropriate workflow process in the Workflow Processes OBLE.

2 Right-click and choose Edit Workflow Process.

3 Drag and drop the Stop icon from the palette to the workspace.

4 In the WF Steps applet, enter a name for the step, and a description of the purpose of the step.

5 Select an error code.

   NOTE: To define a custom error message, select an error code starting with WF_ERR_CUSTOM. The error message displayed will be %1. To define the text of the custom error message, enter an input parameter with the name %1, and then enter the text of the message in the value field for input arguments.

6 Enter an error message.

7 If you need to define input arguments for this step, continue to "Defining Stop Step Input Arguments" on page 104.

Defining Stop Step Input Arguments

NOTE: No picklist is available for Input Argument Name. The input arguments for a Stop step are the substitution variables in the error message. Substitution variables are identified by a "%". To define the substitution value, enter the substitution variable in the input argument name, such as "%1".

To define input arguments for a stop step

1 Select the stop step, right-click and choose Show Input Arguments.

2 In the Input Arguments field, right-click and choose New Record.

3 Enter a name for the input argument.

   This should be the substitution variable appearing in the error message.

4 Choose an input argument type. The choices available are:

   - Literal
   - Process Property
   - Expression
   - Business Component

5 If the input argument type selected is Literal, enter a value.

6 If the input argument type is Process Property, select a property name and a property data type.

7 If the input argument type is Business Component, select the applicable business component name and business component field.

8 If the input argument type is Expression, enter the expression in the value field.

9 Enter any appropriate comments.

About End Steps

An End step specifies when a process instance is finished. It also provides one last chance to store output arguments to a process property. Each workflow process definition must have at least one end step.

For more information, see "Defining an End Step" on page 105.

NOTE: An important difference between the Stop step and the End step is that the Stop step sets the workflow state to In Error while the End step sets the workflow state to Completed. This is important to keep in mind when calling a workflow process using Workflow Monitor Agent. If the WorkMon parameter Ignore Errors is set to False, a workflow process that encounters a Stop step will cause the WorkMon to exit with error. If the workflow process encounters an End step, WorkMon will not exit with error.

Defining an End Step

You define an end step in the Process Designer in Siebel Tools.

To define an end step

1. Select the appropriate workflow process in the Workflow Processes OBLE.
2. Right-click and choose Edit Workflow Process.
3. Drag and drop the End icon from the palette to the workspace.
4. In the WF Steps applet, enter a name for the step, and a description of the purpose of the step.
5. See "To define output arguments for an end step" on page 105 to enter output arguments.

   Output arguments allow you to store a resulting value in a process property. This value can then be passed to other processes.

To define output arguments for an end step

1. Select the End step, right-click and choose Show Output Arguments.
2. In the Output Arguments applet, right-click and choose New Record.
3. Select the property name from the picklist.
4. Choose an output argument type. The choices available are:
   - Literal
   - Expression
   - Business Component
   - Output Argument

5. If the output argument type selected:

   - Literal, then enter a value.
   - Expression, then enter the expression in the Value field.
   - Business Component, then select the applicable business component name and business component field.
   - Output Argument, then enter the argument.

6. Enter any appropriate comments.

7 For Developers: Understanding How Workflow Processes Are Designed

This chapter describes the ways in which workflow processes operate. This chapter is divided as follows:

About Workflow Processing Modes

Siebel Business Process Designer has four processing mode types that characterize run-time behavior. The processing mode is set in the Workflow Processes list editor of Siebel Tools, using the Workflow Mode field. The processing modes are as follows:

- **7.0 Flow.** The 7.0 workflow process provides backward compatibility for existing Siebel 7 (pre-7.7) workflows. If you have existing workflows and you upgrade to Siebel 7.7, these existing workflows become 7.0 workflows by default. For more information, see "About 7.0 Workflow Processes" on page 108.
- **Long Running Flow.** The long-running workflow process is a persistent workflow that can last for hours, days, or months. For more information, see "About Long-Running Workflow Processes" on page 108.
- **Interactive Flow.** The interactive workflow process navigates the user across Siebel views and runs in the user session. For more information, see "About Interactive Workflow Processes" on page 108.
- **Service Flow.** The service workflow process executes a set of operations upon event invocation. A service workflow completes a unit of work from start to finish. For more information, see "About Service Workflow Processes" on page 109.

NOTE: Existing workflow processes are set to a workflow mode of 7.0 Flow by default. All new workflows (that is, all workflows not existing in a prior release) should be categorized as long-running, interactive, or service flows.

A workflow process can be run within the Workflow Process Manager or in the application object manager, depending on the invocation method.

About 7.0 Workflow Processes

The 7.0 Flow workflow process mode provides backward compatibility for existing Siebel 7 (pre-7.7) workflows. If you have existing workflows and you upgrade to Siebel 7.7, these existing workflows become 7.0 workflows by default. You should categorize all *new* workflows as service, interactive, or long-running.

NOTE: If no mode is specified for a workflow process, the mode is assumed to be 7.0 Flow. It is strongly recommended that you not use the 7.0 Flow mode for new workflow processes you create. As you create new workflow processes, make sure to specify a workflow process mode (other than 7.0 Flow) so that 7.0 Flow is not assumed as the default mode.

About Long-Running Workflow Processes

A long-running workflow process is a persistent workflow that can last for hours, days, or months. One example of a long-running workflow is Send Order to SAP. In this example, the workflow sends an order to an external system and waits for a response.

You can use the long-running workflow processing mode to create a single workflow to handle an entire business process transaction (for example, the Quote to Cash business process), and to coordinate between multiple subprocesses.

You can build long-running workflow processes that are collaborative, by assigning subprocesses to end users. You do this by employing the Workflow User Event business service, which generates user events that can span from one user or session to another user or session. For more information, see "About the Workflow User Event Business Service" on page 120.

NOTE: You cannot build User Interact steps into long-running workflows, but you can build interactive workflows into long-running workflows as subprocesses.

For more information, see:

- "Building Long-Running Workflow Processes" on page 109
  - "Assigning Subprocesses to End Users to Create Collaborative Long-Running Workflows" on page 110

About Interactive Workflow Processes

An interactive workflow is used for controlling user navigation between screens and across views. An interactive workflow is comprised primarily of a set of user interact steps, and usually includes a run-time event.

NOTE: An interactive workflow can run only in the context of a user session; it cannot run in the Workflow Process Manager server component.

Interactive workflow processes can be controlled through the use of a synthetic event attached to explicit user interface buttons. A *synthetic event* is a specialized run-time event that is dedicated to controlling workflow navigation.

Examples of synthetic events include Suspend, Resume, Next, and Back. Associated with buttons on the user interface, these synthetic events are interpreted by the Workflow engine to control workflow navigation by moving the user back or forward, and by suspending or resuming a workflow process.

For more information, see:

- "Building Interactive Workflow Processes" on page 110
  - "Creating Synthetic Event Buttons to Control User Navigation" on page 111
  - "About Suspension and Resumption of Interactive Workflow Processes" on page 115
    - "In-memory Cache of Suspended Interactive Workflows" on page 116
    - "Events Handling of Suspended Interactive Workflows" on page 116
    - "Detection and Handling of the User Logout Event for Suspended Interactive Workflows" on page 116
  - "About Forward and Backward Navigation between Views" on page 117

About Service Workflow Processes

A service workflow process is a transient workflow. That is, it runs to completion in a short period of time, all at once without stopping or pausing for any other event or activity. A service workflow process cannot wait, not for run-time events, and not by pausing for time. A service workflow process cannot have user interact steps. A service workflow process can be part of any other type of workflow process as a subprocess. A service workflow process executes a set of operations upon invocation of an event. One example of a service workflow is a workflow for sending an email.

Building Long-Running Workflow Processes

A long-running workflow process is a persistent workflow that can last for hours, days, or months. An example of a long-running workflow process is an approval process that sends an order to an external system such as SAP, and then waits for a response. For more information, see "About Long-Running Workflow Processes" on page 108.

NOTE: When building long-running workflow processes, use user events and *not* run-time events to trigger processes and resume instances.

For more information, see "Assigning Subprocesses to End Users to Create Collaborative Long-Running Workflows" on page 110

Assigning Subprocesses to End Users to Create Collaborative Long-Running Workflows

Using the subprocess step, you can configure workflows that assign interactive subprocess workflows to end users to create collaborative workflow processes. An example of a collaborative workflow is one that includes a requirement for approvals; the route the workflow takes as tasks is a route across multiple users.

Use the Step Recipient field on a subprocess step to create collaborative workflows. Assignment occurs based on the login name, not on the Position or User ID. This login name may be a literal value, it may be held in a process property or a buscomp field, or it may be the result of an expression.

NOTE: The Process Designer cannot validate the data supplied to make sure that it represents a valid login name at design time.

To assign subprocesses to end users

1. Create a subprocess step.
2. Right-click the subprocess step and choose Show Recipients.
3. In the WF Step Recipients applet, set the Recipient Name field to the login name of the assignee (the end user who will be assigned the subprocess).

Building Interactive Workflow Processes

When you are building interactive workflow processes, do the following:

- Set the mode of interactive workflows to Interactive Flow.
- Set the Auto Persist flag for interactive flows that need to be persisted.
- Configure business components and applets by adding buttons to applets to make use of the synthetic events that control user navigation of workflow processes. See "Creating Synthetic Event Buttons to Control User Navigation" on page 111.

This topic is organized as follows:

Creating Synthetic Event Buttons to Control User Navigation

A *synthetic event* is a specialized run-time event that is dedicated to controlling workflow navigation. To control the way a user navigates through the Siebel application, you can create buttons on applets within views and then associate synthetic events with the buttons. For example, in the Account Note view of the Siebel application you are configuring, there is an Account Entry Applet on which you want to include buttons for Back, Next, and SaveWorkflow synthetic events so that the user can move forward or backward from the Account Note view, or suspend an interactive workflow process, and then return later to resume the workflow.

After you have created the buttons, you associate methods within the interactive workflow process—for example, with Back and Next synthetic events, you associate methods to outgoing branches on the User Interact step. You set the synthetic event method name in the MethodInvoked field of the button controls.

The name of the synthetic event method depends on the synthetic event, taking one of the following formats:

- **FrameEventMethodWFNext *and* EventMethodWFNext.** This event moves the user forward in the interactive workflow.

  NOTE: You can also give the method name a prefix of “BF,” as in “FrameEventMethodWFBFxxxx.” Use this optional “BF” prefix to define backward and forward behavior of the synthetic event. A synthetic event with this prefix can be used to resume a workflow process from a step that is different from the current step at which the workflow process is waiting.

- **FrameEventMethodWFBack *and* EventMethodWFBack.** This event moves the user backward in the interactive workflow.
- **SaveWorkflow.** This event suspends (saves) the interactive workflow and makes it appear in the user’s Inbox.
- **ResumeLastIntFlow.** This event resumes the last executed interactive workflow.

  NOTE: ResumeLastIntFlow is different from the other events in that it is not tied to any specific workflow process and can be invoked from anywhere in the Siebel application. That is, the button corresponding to this event can be put in any applet, including the task bar where the Site Map icon is located (the recommended place for this button).

This topic is organized as follows:

- “To create synthetic event buttons for Next and Back events” on page 111
- “To create synthetic event buttons for the SaveWorkflow event” on page 112
- “To create synthetic event buttons for the ResumeLastIntFlow event” on page 114

To create synthetic event buttons for Next and Back events

1. In Siebel Tools, select a view to which a User Interact step navigates the user.
2. Configure a Next button or a Back button on an applet where the event is to be triggered. For more information, see *Using Siebel Tools*.

3. Specify the MethodInvoked property of the button control as the name of the associated event, for example, FrameEventMethodWFBack for backward navigation.
4. In the Palette Designer, associate the applet type run-time event, for example, FrameEventMethodWFBack, to the outgoing branches of the User Interact step in the workflow process that will receive the event. Assign the event the following properties:
   - Event Type = Applet
   - Event Obj = AppletName
   - Event = InvokeMethod
   - Sub Event = [method name, for example, FrameEventMethodWFBack]

NOTE: You do not have to manually create buttons for each applet. You can copy any button you have created to other applets by using the Applet Comparison capability in Siebel Tools. Also, if you add the applet button controls to the HTML Model Controls applet, when you create new applets with the New Applet Wizard or the conversion process, you can then select all the Workflow-related method buttons.

To create synthetic event buttons for the SaveWorkflow event

1. In Siebel Tools, select a view to which a User Interact step navigates the user.
2. Configure a Save button on an applet where the event is to be triggered. For more information, see *Using Siebel Tools*.
3. Specify the MethodInvoked property of the button control as the name of the associated event, SaveWorkflow.
4. Use the script that follows to invoke the Workflow event handler to handle the button-click event, and passes the Workflow event handler the event's contextual information, that is, the name of the view where the event occurs.

   NOTE: The event does not need to be defined in the workflow process definition.

```
function WebApplet_InvokeMethod (MethodName)

{

  return (ContinueOperation);

}


function WebApplet_PreCanInvokeMethod (MethodName, &CanInvoke)

{

// Recognize SaveWorkflow event, which is

// used to save Interactive flow

if (MethodName == "SaveWorkflow")

{
```

```
CanInvoke = "TRUE";
return (CancelOperation);
}

return (ContinueOperation);
}

function WebApplet_PreInvokeMethod (MethodName)
{
// Handle SaveWorkflow event.
// Call Workflow Process Manager to save the interactive
// flow(s) that is waiting in the current view.
if (MethodName == "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 = BusComp ();
var bcId = bc.GetFieldValue ("Id");
Inputs.SetProperty("RowId", bcId);

var workflowSvc= TheApplication().GetService("Workflow Process Manager");
workflowSvc.InvokeMethod("_HandleSpecialEvent", Inputs, Outputs);
```

```
return (CancelOperation);


}


return (ContinueOperation);

}
```

To create synthetic event buttons for the ResumeLastIntFlow event

1. In Siebel Tools, select a view to which a User Interact step navigates the user.
2. Configure a Resume button on an applet where the event is to be triggered. For more information, see *Using Siebel Tools*.
3. Specify the MethodInvoked property of the button control as the name of the associated event, ResumeLastIntFlow.
4. Use the script that follows to invoke the Workflow event handler to handle the button-click event, and passes the Workflow event handler the event's contextual information, that is, the name of the view where the event occurs.

   NOTE: The event does not need to be defined in the workflow process definition.

   ```
   function WebApplet_InvokeMethod (MethodName)

   {

   return (ContinueOperation);

   }


   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-executed 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);
}
```

About Suspension and Resumption of Interactive Workflow Processes

Interactive workflow processes that have been suspended can be resumed from within the user's Inbox. The user can navigate out of an interactive process, then navigate back to the process and pick up where the user left off.

Suspension and resumption of interactive workflow processes can be used in a situation such as the following: a transaction involving the user (an insurance agent) cannot be completed because of missing information, such as a spouse's social security number required for the entry of an insurance policy quote. In this example, when the insurance agent has been able to obtain the social security number after suspending the interactive workflow process, she can resume the process from within her Inbox and enter the number to complete the entry of the quote. Once the process is complete, the Workflow engine removes the interactive workflow process from her Inbox.

NOTE: In-session interactive workflow processes are cached. The maximum number of interactive workflow processes in the cache is eight.

Suspended interactive workflows are placed in the workflow owner's Inbox for tracking and explicit resumption.

A suspended interactive workflow is placed in the Inbox under the following two conditions:

- When the workflow is explicitly suspended through use of the Suspend button (this is called *explicit suspension*).
- When a suspended workflow must be removed from the in-memory cache (such as when the user logs out), *and if* the suspended workflow has its Auto Persist flag checked (this is called *implicit suspension*).

A suspended interactive workflow in the Inbox is resumed under the following two conditions:

- When a workflow is explicitly resumed from within the Inbox when the user clicks on it (this is called *explicit resumption*).
- When the events that the workflow is waiting for are triggered by some user action (this is called *implicit resumption*).

A suspended interactive workflow in the Inbox is removed from the Inbox:

- When the workflow has run to its end and terminates.

NOTE: With explicit resumption, when a user clicks on a workflow in the Inbox, the Inbox will invoke the Workflow engine or business service and pass it the row ID of the workflow instance for resumption.

In-memory Cache of Suspended Interactive Workflows

Users often navigate out of structured interactive workflows because the workflows have been set up in this manner to address the specific needs of your business. When this happens, the interactive workflows remain in memory so that they can be resumed later in the same user session. As it is uncommon for users to have a large number of unfinished tasks at hand, there can be a maximum of eight suspended interactive workflows in the memory cache.

Events Handling of Suspended Interactive Workflows

Workflow handles events in the following sequence:

1. Checking of the in-memory cache to see if any workflow instances there can receive these events, using the matching criteria specified by the events.
2. Checking of the database to see if any persisted workflows can receive these events.
3. Resumption of all instances found in Step 1 and Step 2.

Detection and Handling of the User Logout Event for Suspended Interactive Workflows

Upon receiving the user logout event, the Workflow engine goes through suspended interactive workflows in the in-memory cache. Workflows with the Auto Persist flag checked are saved as Inbox items. Other workflows are deleted.

About Forward and Backward Navigation between Views

You can use synthetic events to define an interactive workflow process so that when the user clicks the Next and Back buttons, the user is taken to the next and previous views in the sequence without losing the context of the process instance.

NOTE: Use synthetic events to allow the user to navigate backward through views. Run-time events allow forward navigation, but not backward navigation.

Workflows can navigate back and forth if the following conditions are met:

- The workflow being resumed is either an interactive workflow process or a 7.0 workflow process.
- The triggering event is a workflow navigation event, that is, an event with a name such as InvokeMethod, and a sub-event with a name such as FrameEventMethodWFBFxxxx or EventMethodWFBFxxxx, where "xxxx" is the name of the event, such as Next.

NOTE: Free-flow (backward-forward) navigation is possible with interactive workflow processes and 7.0 workflow processes. Free-flow navigation is not possible with service workflows and long-running workflows. When considering whether to allow backward navigation in your workflow process, be aware of the following:

1. The backward navigation feature does not undo the effect of the workflow process; it only modifies the current step counter to point to a previous step.

2. The workflow configuration must make sure that the segment of the workflow that can be repeated by the backward navigation feature is idempotent.

For more information, see "Creating Synthetic Event Buttons to Control User Navigation" on page 111.

Using Workflow Persistence

Workflow persistence is a feature used to store the state of a workflow process instance and its steps, as well as its process properties. The workflow process state and process properties are saved in the WF_PROC_VAL_CD table.

By using workflow persistence to store the state of a workflow process, you can build end-to-end workflows which include wait steps, subprocesses, and other interruptions, and which maintain the active state of a process over short or long periods of time, with activity occurring in various parts of your enterprise. When persistence is set to TRUE, a user can continue with a workflow process that has been suspended. The suspended workflow appears in the user's Inbox.

For more information, see "About Workflow Persistence" on page 117.

About Workflow Persistence

Workflow persistence is an attribute of a workflow process that supports long-lived transactions within a single workflow process. Workflow persistence allows process resumption after a pause or a server crash. Workflow persistence saves and restores data when the process is resumed.

The workflow persistence setting applies to long-running workflow processes, interactive workflow processes, and 7.0 workflow processes. You cannot use workflow persistence with service workflow processes. The persistence property is a YES/NO setting. For long-running workflows, persistence is automatically set by the server at execution time. For interactive workflows, you set the persistence property using the Auto Persist flag in the Workflow Processes list editor of Siebel Tools. Persistence behaves as follows:

- Long-running workflows are automatically persisted
- Interactive workflows are persisted on pause if the Auto Persist flag is set
  - You control workflow persistence by setting the Auto Persist flag. When a session times out or a user logs out of a Web session, workflow processes with the Auto Persist flag set to YES are persisted and can be resumed from the Universal Inbox.
- 7.0 workflows with persistence set are marked as Auto-Persist and are persisted

NOTE: In the previous release, workflow persistence was used for monitoring workflow processes, and it was controlled by adjusting two settings that could apply to individual steps of a workflow: frequency and level. In this release, process monitoring is separate from workflow persistence, and persistence does not need to be set for long-running processes, because for long-running processes it is set by default. For 7.0 workflow processes that had persistence set, the Auto Persist flag is automatically set to TRUE (YES) during upgrade and import.

For more information on workflow persistence, see "Enabling Workflow Persistence" on page 118.

Enabling Workflow Persistence

For long-running workflow processes, persistence occurs by default. For interactive workflow processes, you set the Auto Persist flag in the Process Designer within Siebel Tools.

To set workflow persistence for interactive workflow processes

1. In Siebel Tools, from the Object Explorer applet, select the Workflow Process object.
2. In the Workflow Process list editor, select the process you want to work with.
3. In the Auto Persist field, choose YES using the drop-down picklist.

NOTE: Persisted workflow processes are automatically purged once they complete running.

Handling Events

Information on events handling is organized as follows:

- "Using Run-Time Events" on page 119
- "Using User Events" on page 120

Using Run-Time Events

Run-time events allow the Siebel application to respond in real time to user actions. Run-time events can be specified in the branches for Start, Wait, or User Interact steps to start or resume a workflow process. The fields in the WF Step Branch applet that are used to define a run-time event are described in "Field Descriptions: WF Step Branch Applet" on page 75 and are the following:

- Event Object Type
- Event Object
- Event
- Sub Event
- Event Cancel Flag

NOTE: When defining run-time events within a workflow process, a run-time event should not be repeated. You cannot use the same event more than once within a workflow process.

Run-time events can be used for workflows that run within a user session. For workflows that span across multiple users—long-running workflows—use user events instead. For more information, see "Using User Events" on page 120.

NOTE: Run-time events should not be used to trigger long-running workflows because a run-time event is specifically attached to a single user and a single session. A run-time event is only for that single user, as it stems from Personalization functionality. Instead use an interactive workflow or a service workflow to handle the run-time event, then after processing it and validating it, generate a user event to notify a long-running workflow.

For further information on run-time events, see *Siebel Personalization Administration Guide*.

Run-time Events and User Interact Steps

The following events are not supported with the User Interact step:

- All application events.
- The DisplayRecord event.
- The DisplayApplet event.
- The SetFieldValue event for a field that has the Immediate Post Changes property set to TRUE.
- The Login event. Use the WebSessionStart event instead.

Choosing Between Using Run-Time Events and Using Workflow Policies

In cases when it is necessary to detect database events, use workflow policies, not run-time events, for defining integrations that occur on data change or write. For example, when using the UI, use run-time events to trigger a workflow process. When using the Siebel EAI Adapter, which performs numerous WriteRecord events, use workflow policies.

Using User Events

While run-time events act on workflow processes from within the application object manager, user events are events internal to Siebel Workflow. User events initiate and resume long-running workflow processes in the Workflow Process Manager server component (WFProcMgr).

NOTE: User events can be used only in long-running workflow processes.

While run-time events can be used in workflows that run within a single user session, user events are for use in long-running workflows that span multiple users. A user event can be used to trigger a workflow process (when it is attached to a Start step), or to resume a waiting workflow instance (when it is attached to any step that can receive input arguments). A user event can also bring data into a workflow instance, in the form of the user event's payload, which can contain user data.

User events require use of the Workflow User Event business service to communicate with the Workflow Process Manager.

More information on user events is provided in the following topics:

- "About the Workflow User Event Business Service" on page 120
- "Generating User Events with the User Event Business Service" on page 121
- "Configuring Long-Running Workflow Processes to Wait for User Events" on page 122

About the Workflow User Event Business Service

User events can be generated anywhere in the Siebel enterprise (wherever a Siebel business service is used) by calling the Workflow User Event business service. The Workflow User Event business service is used for one-way communication from the run-time client to the Workflow Process Manager server component. If a long-running workflow runs in the run-time user session, you can use a service workflow or an interactive workflow to trigger this long-running workflow. But to trigger a long-running flow to be run in WFProcMgr, you need to send a notification. The Workflow User Event business service sends this notification in the form of a user event.

Arguments

The following arguments define a Workflow User Event business service:

- **User Event Name.** The name of a user event is an agreement between the creator (an external entity) and the recipient (the workflow definition). It has no special significance, except that the incoming event name and the workflow instance definition must specify exactly the same user event name in order to successfully communicate with each other. User event names must be unique. It is best to logically name user events after the business purpose they serve (for example, "Event Transferring Send Order Confirmation from Vitira To Siebel - V2").

- **Correlation.** Used to match an incoming message with a workflow instance using business data such as an order number. Correlation is the process of matching an incoming message with a workflow instance using business data such as an order ID. In this release, correlation applies to user events reaching long-running workflows. It is often the case that Siebel Workflow communicates with an external entity and the external entity is unaware that it is in contact with a Siebel workflow. In such cases, it is difficult for the external entity to use a Siebel identifier (like the workflow process instance ID) to identify the recipient. It is more convenient to use a piece of business data (such as an order number) to identify the recipient. The correlator serves this purpose. A long-running workflow can specify a process property as a correlator.

  NOTE: Only one process property can be used as a correlator.

- **Payload.** When the user event is created, the user can specify any data as payload. This data is delivered to the workflow instance that receives the event. When the workflow is defined to wait for a user event, the definition can specify a process property to receive this payload data. The payload is a single value—only one payload can be passed.

  NOTE: If your situation calls for sending complex or structured data, you should convert the data into an XML document (using the XML converter) and then pass the resulting XML string as the payload of the event. The receiving workflow can then call the XML converter again to recover the original data structure.

Generating User Events with the User Event Business Service

User events are Workflow-internal events used to resume long-running workflow processes from the Workflow Process Manager. To create a user event, you invoke the Workflow User Event business service, specifying the payload and the correlator.

NOTE: Long-running workflow processes should use only user events, not run-time events.

The Workflow User Event business service is a standard Siebel business service that can be used everywhere a Siebel business service can be used. You invoke the Workflow User Event business service by configuring a business service step that calls it.

A common case is when a foreground workflow (that is, a 7.0 workflow, an interactive workflow, or a service workflow) can initiate a user event (by using a business service step configured to call the Workflow User Events business service) to communicate to a background workflow (that is, a long-running workflow). User events can be created by all supported Siebel mechanisms to invoke a business service (such as scripting, COM interfaces, and Java interfaces). This is the recommended way to externally communicate with a Siebel workflow.

NOTE: While any type of workflow process (or business service) can generate user events, only long-running workflow processes should be configured to receive user events.

NOTE: The following task outlines one of several ways of invoking the User Event business service to generate user events. The User Event business service is a standard Siebel business service and can be invoked by all supported mechanisms, such as scripting, COM interfaces, and Java interfaces.

To invoke the User Event business service for generating a user event

1. Add a business service step to a workflow process.
2. In the WF Steps applet, set the step properties as follows:
   - Business Service Name = Workflow User Event Service
   - Business Service Method = GenerateEvent
3. In the Palette Designer, right-click on the business service step and choose Show Input Arguments.
4. In the Input Arguments applet, right-click and select New Record.
5. In the Input Argument field, choose Payload from the picklist and fill in the fields as appropriate.
6. Repeat Step 4 to create an input argument by choosing Correlator Value from the picklist.
7. Repeat Step 4 to create an input argument by choosing User Event Name from the picklist.

Configuring Long-Running Workflow Processes to Wait for User Events

When using user events in your workflow processes, keep in mind that only long-running workflows can wait for user events. All other types of workflows cannot *wait* for user events, though they can *generate* user events. Long-running workflows should be configured with *user* events only, not run-time events.

To configure a long-running workflow definition to wait for a user event

1. Select the workflow process that you are setting to wait for the user event.
2. Set one of the process properties of the workflow as the correlator by checking its Correlator flag.
3. On the branch of the step that handles the event (a Start step or a Wait step), enter parameters to complete the following fields:
   - User Event = [the name of the workflow you selected for the Value field in Step 7 on page 122]
   - User Event Timeout = [the timeout period for the user event]

     User Event Timeout works in a similar way as the timeout setting for run-time events. Workflows are resumed after the timeout period if no user event is received during the timeout period.
   - User Event DestProp = [the name of the process property that will hold the payload passed in from the user event]

NOTE: User events are not queued. If no recipient is waiting to accept the user event with the specified correlator, the event is discarded.

Workflow and Global Implementations

Information on global implementation of Workflow provided as follows:

- "Configuring Workflows in a Multilingual Environment" on page 123
- "Defining Expressions for Workflows Running in a Multilingual Environment" on page 123
- "Wait Steps and Global Time Calculations in Workflow" on page 124

Configuring Workflows in a Multilingual Environment

In order to create workflows in a language other than the base language, you need to make sure that your database is enabled for multilingual lists of values (that is, you make sure it is MLOV-enabled) for the non-base language type. For example, if you are modifying workflows using language=FRA and the base language=ENU, then you need to make sure that List of Values entries exist for the FRA language type.

To verify that List of Values entries exist for the appropriate language type

1. From the application-level menu of the run-time client, choose Navigate > Site Map > Administration - Data > List of Values.
2. Run the following query: Type = "WF_*".
3. Enable the database for multilingual lists of values by running the MLOV upgrade utility. For information on running the MLOV upgrade utility, see *Configuring Siebel eBusiness Applications*.
4. In the List of Values applet, make sure the Active flag is set.

For more information on global deployments, see *Global Deployment Guide*.

Defining Expressions for Workflows Running in a Multilingual Environment

Workflows use the Display value to fetch records from tables. In a multilingual deployment, the data in the tables is stored in language-independent code (LIC). To run workflows in a multilingual environment, use the LookupValue function to fetch the LIC based on the Display value.

Example 1: You have a Decision step that compares Account Status to "Active". The Account Status field is bounded by the Account Status picklist. You can set the Compare To field to "Expression", and set the Expression field to:

```
[Account Status] = LookupValue ("ACCOUNT_STATUS", "Active")
```

Example 2: You have a Business Service step that calls the Outbound Communications Manager to send emails to "Expense Approver". The Recipient Group argument is bounded by the Comm Recipient Group picklist. You can set the Type field to "Expression", and set the Value field to:

```
LookupValue ("COMM_RECIP_SRC", "Comm Employee")
```

For more information on globalization, see *Global Deployment Guide*. For more information on configuring Siebel Workflow to use MLOV-enabled fields, see *Configuring Siebel eBusiness Applications*.

Wait Steps and Global Time Calculations in Workflow

An absolute wait is a wait period governed solely by the duration specified. For example, an absolute wait set for 30 minutes waits 30 minutes from the time the wait is initiated by a Wait step. A service calendar wait, on the other hand, is not absolute. For example, a service calendar wait could be set to begin at 6 P.M., but if the service hours for the organization are 9 A.M. to 6 P.M., the wait will not initiate until 9 the next morning. So it will run from 9 to 9:30 instead of 6 to 6:30.

When creating workflows with Wait steps, keep in mind that in this release, absolute waits are not affected by any time-zone settings, including server and user time-zone preferences. The database server should always use UTC. For more information, see *Global Deployment Guide*.

For a wait that is not absolute—that is, the workflow involves service calendar integration—the Wait step requires a time zone for delay computations. In this case, the current user's time zone is used.

NOTE: When a workflow process is executing as a server task, you must shut down and restart the Workflow Process Manager after making any changes to the Time Zone user preference for the SADMIN user. The changes will only take effect after restarting the Workflow Process Manager. This is important if you are implementing UTC, as you may need to set the Time Zone user preference.

Handling Errors

Information on error processing is organized as follows:

- "Using Error Processes to Handle Errors" on page 124
  - "Passing User-Defined Process Properties and Property Sets to Error Processes" on page 125
  - "Assigning Error Processes to Subprocesses" on page 126
- "Using Exceptions to Handle Errors" on page 126
  - "Defining Exceptions" on page 127

Using Error Processes to Handle Errors

You can use an error process for handling errors. An error process is a standard workflow process. It becomes an error process when you associate it to another (main) process using the Error Process Name field in the Workflow Processes list to choose the error process from the picklist. When an error occurs, the exception branch calls this error process

Like the subprocess step, the error process must have been predefined in order to be selected. When the selection button associated with the Error Process Name field is clicked, a pick applet of all available processes will be shown. Once an error process is selected, this process will be called when the current process reaches an error state. Processing of the current process will stop and end, and will instead start the error process.

If a workflow process that has an error process defined for it encounters an error, it stops processing and passes all system-defined process properties to the error process.

When a workflow process reaches an error state, one of the following events happens depending on whether or not an error process is defined for the workflow process:

- If no error process is defined for the workflow process, the process remains in the state called In Error. The original error code is returned to the caller of the process.
- If an error process is defined for the workflow process, one of the following events happens:
  - The error process handles the error successfully.

    The error handling is considered successful if the error process arrives at an End step. In this case, the error process will be in the Completed state. No error code will be returned to the caller of the workflow process. The workflow process terminates immediately with a Completed state.
  - The error process tries to handle the error, but fails with a different error.

    The error handling is considered failed if the error process arrives at a Stop step. In this case, the error process will be in the In Error state. A new error code, which you have selected in the Stop step, is returned to the caller of the workflow process. The workflow process remains in the In Error state.
  - The error process cannot handle the error.

    If no Start conditions are satisfied, the error process ends immediately. In this case, the error process will be in the In Error state. The original error code will be returned to the caller of the workflow process. The workflow process remains in the In Error state.

The following sections provide more information about error processes:

- "Passing User-Defined Process Properties and Property Sets to Error Processes" on page 125
- "Assigning Error Processes to Subprocesses" on page 126

Passing User-Defined Process Properties and Property Sets to Error Processes

If you want more than just the system-defined process properties to be passed to the error process, note the following:

- If you want the original process instance to pass any user-defined process properties to the error process, you have to explicitly recreate those user-defined process properties in your error process, giving them the same name and data type.

- If you want to pass a property set from the original process to the error process, you have to create a common user-defined hierarchy process property in the original workflow process and in the error process, and use this common hierarchy property to pass the property set.
- If you want the error process to get the name of the original workflow process, you have to create a common user-defined process property in the original workflow process and in the error process, and then pass the original process name through this common user-defined process property.

Assigning Error Processes to Subprocesses

If a subprocess encounters an error and there is an error process defined for the subprocess, the error process completes with one of the following outcomes:

- The error process handles the error successfully.

  The error handling is considered successful if the error process arrives at an End step. In this case, the error process is in the Completed state. The subprocess also terminates with a Completed state, and the control returns to the main process instance. The main process instance continues to execute from the next step.

- The error process tries to handle the error, but fails with a different error.

  The error handling is considered failed if the error process arrives at a Stop step. In this case, the error process will be in the In Error state. The new error code, which you have selected in the Stop step, is returned to the subprocess. Both the subprocess and the main process will terminate with the In Error state.

Using Exceptions to Handle Errors

Exceptions are a type of branch designed for handling system and user-defined errors. An example of a system generated error would be a failure when sending an email notification. A user-defined error would be trying to submit an order that was incomplete.

As a branch, an exception is a type of connector between two steps. Exceptions are illustrated in the Palette Designer as red connectors. When you click on an exception connector in the Palette Designer, the WF Step Branch applet appears showing the branch properties.

NOTE: Be aware that exceptions on a step are evaluated after the step has completed. If you want to evaluate an exception before executing a step, you must attach the exception to the previous step in the process.

The main parts of creating exceptions for a workflow process are:

- Define an exception
- Define the exception conditions
- Add the exception actions

For more information, see "Defining Exceptions" on page 127.

Defining Exceptions

Exceptions are defined in the WF Step Branch applet of the Process Designer.

To define an exception

1. In Siebel Tools, make the appropriate process active by selecting it in the Workflow Processes OBLE.
2. Right-click and select Edit Workflow Process.
3. In the Palette Designer, drag and drop an exception connector from the palette area and connect it to an existing shape on the workspace. Be sure that the end of the connector is attached to the step.
4. Click the exception arrow in the workflow diagram.
5. In the WF Step Branch applet, select the exception and enter a name for the exception.
6. Under Type, select Error Exception or User Defined Exception.
7. Define the conditions that apply to the exception.

Recovering Workflow Processes

You can recover interrupted workflow processes both automatically and manually.

Automatic Recovery of Workflow Process Instances

If the Workflow Process Manager server component fails because of an exogenous event (such as a server crash), Siebel Workflow automatically resumes the interrupted workflow instances when the server restarts.

For a workflow process instance that cannot be automatically recoverable, you can manually recover the process. For example, if the server crashes in the middle of a Siebel operation to update a record, then the workflow can't know whether the Siebel Operation has completed. You may have to manually verify that the update was completed before resuming the workflow execution. In another case, if the Siebel operation is to query a set of records, then even after the server crashes, the workflow can be resumed automatically by requerying.

NOTE: Automatic recovery of workflow processes applies to workflow processes that run in the server component. Workflow processes running on a local database are not recovered.

Recovery is performed by the Recovery Manager based on the process instance's state information that is saved by the Workflow engine. The state information is saved at recovery checkpoints. For performance optimization, the recovery checkpoints are determined by the Workflow engine based on the nature of the step and Allow Retry flag step parameter.

The application developer can minimize run-time overhead by using the Allow Retry flag to reduce the number of checkpoints. This parameter is set at the step level.

NOTE: Set the Auto Retry flag for Siebel Operation steps and business service steps. Based on this information, the Workflow engine determines the recovery checkpoints.

If the Recovery Manager cannot determine from which step the workflow instance should recover, then those instances are marked for manual recovery.

Manual Recovery of Workflow Process Instances

You can correct and resume a workflow process instance that has encountered errors. For example, if the Communications Server is not available, a workflow process sending an email notification will have a status of In Error. You can enable the Communication Server component, then resume the workflow process.

Instances marked for manual recovery are recoverable from the Workflow Instance Admin view. From the application-level menu in the run-time client, choose Navigate > Site Map > Administration - Business Process > Workflow Instance Admin > Related Instances and select an option from the applet menu. There are two options for manual recovery:

- **Resume Instance - Next Step.** The process instance skips the current step, and evaluates all the branching conditions coming from the current step to determine the next step to execute.
- **Resume Instance - Current Step.** The process instance resumes from the current step. The current step is retried. If the current step is a subprocess step, a new subprocess instance is started.

NOTE: A workflow process instance is resumable only if, among its related process instances, its Call Depth setting is the highest. That is, Resume Instance - Next Step and Resume Instance - Current Step are disabled if the instance is part of a set of related instances and one or more other instances are set at a higher level (so the instance is called from another instance). For example, if there are multiple related instances with Call Depth settings of level 0,1,2,3, and 4 and you select the record with level 3, then these Resume controls will be disabled because level 3 is not the highest Call Depth level set among the related instances.

Invoking Workflow Processes

This section describes the different ways to invoke and run a workflow process. The information provided in this section is organized as follows:

- "Running a Workflow Process in the Application Object Manager" on page 135
- "Running a Workflow Process in Batch Mode" on page 135

About Invoking a Workflow Process

A workflow process can be invoked in the following ways:

- **From the Process Simulator view.** See "Testing Workflow Processes with the Process Simulator" on page 137.
- **From a workflow policy.** See "Invoking a Workflow Process from a Workflow Policy" on page 129.
- **From a script.** See "Invoking a Workflow Process from a Script" on page 130.
- **From a run-time event.** See "Invoking a Workflow Process from a Run-Time Event" on page 132.
- **From a user event.** See "Using User Events" on page 120.
- **From a synthetic event.** See "Creating Synthetic Event Buttons to Control User Navigation" on page 111.
- **As a configured business service.** See "Invoking a Workflow Process as a Configured Business Service" on page 133.

You can use these methods to test workflow processes in your test environment before migrating them to the production environment. When testing, being able to invoke from a workflow policy is important because it tests invoking a workflow process on the server. You can also use these methods to invoke processes in your production environment.

In Chapter 8, "For Developers: Testing Workflow Processes," invoking a process from the Process Simulator view is described. *Overview: Siebel eBusiness Application Integration Volume I* and *Siebel eMail Response Administration Guide* discuss invoking a process from a server component. The other invocation methods are explained in this chapter.

NOTE: Starting with Siebel 7.0, Business Integration Manager and Business Integration Batch Manager have been deprecated, so if you were using either one in your business processes you need to replace them with Workflow Process Manager or Workflow Process Batch Manager respectively.

Invoking a Workflow Process from a Workflow Policy

To invoke a workflow process from a workflow policy, you define a policy action that uses the workflow policy program Run Workflow Process. Alternatively, you can create a custom workflow policy program by copying the Run Workflow Process program and then adding program arguments that correspond to workflow process properties. This way you can use the policy program to pass data to the workflow process properties.

NOTE: For complete information about defining workflow policies, see "About Creating Workflow Policies" on page 163.

To invoke a process from a workflow policy

1 From the application-level menu, choose Navigate > Site Map > Administration - Business Process > Actions.

2 In the Actions applet, click New to define a new action.

- In the Program field, use the picklist to select the Run Workflow Process program.

3 In the Arguments applet, in the Argument field, select ProcessName.

- In the Value field, enter the name of the workflow process you want to invoke as the argument value.

4 From the application-level menu, choose Navigate > Site Map > Administration - Business Process > Policy Groups.

5 In the Policy Groups applet, click New to define a new group and name it.

6 From the application-level menu, choose Navigate > Site Map > Administration - Business Process > Policies.

7 In the Policies List applet, click New to define a new policy.

   a In the Conditions applet, click New to define a condition for the policy that must be met to invoke the workflow process.

   b In the Actions applet, click New to define a new action and enter the name of the action you defined in Step 1.

8 Run Generate Triggers.

   See "Creating Database Triggers" on page 217 for more information about running the trigger generator.

9 If you are using the Run Workflow Process program, verify that the Workflow Process Manager server component is online.

10 Run Workflow Monitor Agent.

   See "Using Workflow Monitor Agent" on page 228 for more information about monitoring workflow policies.

11 Violate the policy. The action should invoke the workflow process.

Invoking a Workflow Process from a Script

Workflow processes can be invoked programmatically from a script using Siebel VB or Siebel eScript. By using scripts, workflow processes can be invoked from anywhere in the Siebel application or from external programs.

NOTE: Invoking a workflow process from a script is done in synchronous mode.

When invoking a process from a script, you can specify that the process run either on the server or in the object manager. To run a process on the server, call the service Workflow Process Manager (Server Request). To run a process in the application object manager, call the service Workflow Process Manager.

See the following sample scripts for examples of how to invoke a workflow process from a script:

- "Example: Invoking a Workflow from a Script in Object Manager" on page 131
- "Example: Invoking a Workflow from a Script to Pass Field Values to Process Properties" on page 131

Example: Invoking a Workflow from a Script in Object Manager

The following is a sample script that invokes a workflow process called My Account Process. In this example, the process is invoked in the object manager.

```
// Example:  Invoking a Workflow Process via scripting
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("Account");
var rowId = bc.GetFieldValue("Id");

Input.SetProperty("ProcessName", "My Account Process");
Input.SetProperty("Object Id", rowId);

svc.InvokeMethod("RunProcess", Input, Output);
}
```

Example: Invoking a Workflow from a Script to Pass Field Values to Process Properties

The following is a similar example script that invokes a workflow process called My Opportunity Process. In this example, the process is invoked in the object manager and additional field values are passed to process properties 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);
```

Invoking a Workflow Process from a Run-Time Event

Workflow Processes integrates with the Run-time Events engine to provide a simplified event mechanism for automating business processes. This mechanism:

- Allows real-time monitoring of events
- Minimizes the need for scripting and workflow policy invocation

The following types of events are possible:

- Application
- Business Component
- Applet

NOTE: For more information about run-time events, see *Siebel Personalization Administration Guide*.

In order to use a run-time event to invoke a workflow process, you use Siebel Tools to configure buttons on applets.

To create a button that invokes a workflow process

1. From Siebel Tools, configure a button on an applet and specify the MethodInvoked property. For more information, see *Using Siebel Tools*.
2. To enable the button, override the PreCanInvokeMethod property. Next, edit the server script and compile your changes to the Siebel repository file. The following example is specific to Siebel VB:

```
Function WebApplet_PreCanInvokeMethod (MethodName As String, CanInvoke As String)
As Integer

If MethodName = "<Name>" Then

   CanInvoke = "True"

   WebApplet_PreCanInvokeMethod = CancelOperation

Else

   WebApplet_PreCanInvokeMethod = ContinueOperation

End If

End Function
```

3 Define a workflow process. To invoke this workflow process from a button click, specify a run-time event in one of the following ways:

- To start this workflow process, specify the run-time event on the Start step on a Condition branch.
- To resume this workflow process (if it has been paused) from a button click, specify the run-time event in a User Interact step or a Wait step on a Condition branch.

The fields values are described in Table 15.

Table 15. Run-time Event Field Values

Field	Value
Event Object Type	BusComp
Event	PreInvokeMethod
Event Object	Name of the business component on which the applet that contains the button is based.
Sub Event	Name of the method set in step one. The Sub Event name must be unique.
Event Cancel Flag	True. If this flag is not checked, the error "The specialized method <Method Name> is not supported on this business component" will result when running the workflow process.

NOTE: If users click the button when there are no workflow process instances waiting, they will receive the error message "Cannot resume process <Process ID> for object <Object ID>". You can prevent this error by requiring users to run a workflow process in order to access this view. For example, you can set up a workflow that includes the User Interact step. You can handle the error by defining an Error Exception branch or error process.

4 After you create the workflow process, activate the process and reload personalization. To reload personalization:

a From the application-level menu, choose Navigate > Site Map > Administration - Runtime Events > Events.

b From the applet menu, select Reload Runtime Events.

You must reload personalization after activating a workflow process that registers any run-time events in order for the process to take effect.

Invoking a Workflow Process as a Configured Business Service

You can invoke a workflow by defining a configured business service. With a configured business service like Workflow Process Manager (Server Requests), you do not need to specify Server Request Broker (SRBroker) parameters.

NOTE: To use Workflow Process Manager, SRProc and SRBroker must be running.

When you invoke the Server Requests directly, you must specify SRBroker parameters.

To define a configured business service

1. In Siebel Tools, navigate to Siebel Objects > Business Service. Add a new record with values:
   - Name: (this is the name that you can reference in scripting)
   - Class: CSSSrmService
   - Display Name: (this is the name that you see in Workflow views)
2. Click Business Service User Prop. Add the following records:
   - Name: Component
   - Value: (internal [or short] name of the server component, for example, 'WfProcMgr')
   - Name: Mode
   - Value: (mode of the server request, for example, 'Async')
3. (Optional) Enter additional user properties pertaining to SRBroker (see "Predefined Business Services" on page 253).
4. Select Business Service Method. Add the following records:
   - Name: (this is the name that you can reference in scripting)
   - Display Name: (this is the name that you see in workflow views)
5. Click Business Service Method Arg. Add records specific to the component being invoked, for example, ‘ProcessName’ for WfProcMgr. Note that the name is the internal (or short) name of the server component parameter.

NOTE: SRBroker and SRProc are required to run any business service that invokes server components.

For more information on invoking business services, see *Siebel Object Interfaces Reference*.

Running a Workflow Process in the Workflow Process Manager

A workflow process can be run within the Workflow Process Manager server component. The methods of invoking a workflow process that support running a process on the server include:

- From a workflow policy that executes on the server
- From a script specifying the Server Request parameter
- From a run-time event with Processing Mode set to Remote Synchronous or Remote Asynchronous

NOTE: Invoking a workflow process from a script is done in synchronous mode.

If a user invokes a process to be run on the server within the Workflow Process Manager server component, the process executes only if the user is connected to the server. If the user is not connected to the server, the request is queued and executes when the user synchronizes or the server becomes available.

If you compiled a custom .srf file using Siebel Tools, this file needs to be added to the Objects directory on the Siebel Application Server. In addition, you must update the siebel.cfg file referenced in the Server parameters to reflect the custom .srf file (note that the siebel.cfg file is the default configuration file for Workflow Process Manager server components).

NOTE: Business services calling UI functions, including navigation functionality such as the User Interact step, are not supported when processes run on the server.

Running a Workflow Process in the Application Object Manager

Running a workflow process in the application object manager can be useful for enforcing business processes with mobile users or for defining business processes that involve end-user navigation.

The methods that support invoking a workflow process in the application object manager include:

- From the Process Simulator
- From a script specified to run locally in the application object manager
- From a run-time event with processing mode specified as local synchronous

Running a Workflow Process in Batch Mode

Workflow processes can be run in batch mode by running the Workflow Process Batch Manager server component. Executing a process in batch allows you to execute the actions in a workflow process for multiple records. When you are running a process in batch, you may want to specify a search specification to limit the number of records that are evaluated.

NOTE: Only 7.0 workflow processes and service workflow processes should be run in batch mode.

Workflow Process Batch Manager takes the parameter SearchSpec, then executes the search specification on the primary business component of the process business object. For each fetched record, the Workflow Process Batch Manager invokes the workflow process and sets the Object ID process property as the current active row.

Table 16. Workflow Process Batch Manager Parameters

Display Name	Description
Workflow Process Name	Required. Name of the workflow process definition to execute.
Search Specification	Search specification that identifies the work items to process.

Scheduling a Batch Workflow

You can set up a batch workflow to run a workflow process at specified intervals, for example, 7 A.M. every Monday. You do this by setting up a component request.

To set up a component request to schedule a batch workflow

1. From the application-level menu in the run-time client, choose Navigate > Site Map > Administration - Server Configuration > Enterprises > Component Definitions.
2. In the Component Request form, click New.
3. In the Component/Job field, click the selection button. The Component/Jobs dialog box appears.
4. Select Workflow Process Batch Manager.
5. In the Component Request Parameters form applet, click New.
6. In the Name field, click the selection button, then select Workflow Process Name from the dialog box.
7. In the Value field, type in the name of the workflow process to execute.
8. Click New to add another parameter.
9. In the Name field, click the selection button.
10. Select Search Specification.

In the Value field, provide a search specification.

8 For Developers: Testing Workflow Processes

This chapter explains how to test your workflow processes before deployment. You use the Process Simulator in Siebel Tools to test workflow processes, and you can also test them in real time using the run-time environment. The way you test your workflow processes depends on the type of process you are testing.

This chapter is organized as follows:

Testing Workflow Processes with the Process Simulator

Testing your workflow processes before migrating them to your production environment verifies that resulting actions are accurate and useful and the results are exactly what you want.

You need to develop a test and migration procedure for introducing changes into the production environment. Some of the considerations for this procedure are discussed in "Defining a Test and Migration Strategy for Workflow Processes" on page 40.

CAUTION: Your test environment and production environment must have identical versions of the software.

Of the various ways to invoke a workflow process, invocation from the Process Simulator is an easy way to debug a workflow process. You can debug process steps as you define them in Siebel Tools, where the Process Designer and the Process Simulator both reside.

NOTE: When the workflow process is run from the Process Simulator, it runs in the application object manager. Actual invocation of the process may be run in the application object manager or in the Workflow Process Manager server session, depending on specific parameters. Because some workflow processes that can run in the Workflow Process Manager server session may not be able to run in the application object manager, not all workflow processes can be simulated. For more information, see "About the Process Simulator and Supported Modes for Simulation" on page 138.

The other methods involve invoking a workflow process outside of the Siebel Business Process Designer. For information on these methods of invoking a workflow process, see "About Invoking a Workflow Process" on page 129. For information about invoking a workflow process from a server component, see *Overview: Siebel eBusiness Application Integration Volume I* and *Siebel eMail Response Administration Guide*.

About the Process Simulator and Supported Modes for Simulation

The Process Simulator allows you to step through a workflow process while viewing the results of each step. A workflow process does not have to be active to run it in the Process Simulator. The simulator ignores activation date, expiration date, and status.

NOTE: Before you run the Process Simulator or deploy your workflow processes, you use the Validate feature. The Validate feature has rules to make sure the workflow process works correctly. For more information, see "Using the Validate Tool to Correct Errors in Workflow Processes" on page 139.

You can use the Process Simulator to test workflows that run in the Siebel client. This includes service workflows, 7.0 workflows, and interactive workflows. You cannot use the Process Simulator to test workflows that involve a server component, such as Workflow Process Manager, Server Request Broker, Assignment Manager, and Communications Server. To test workflows that involve a server component, the workflow must be deployed to the run-time environment and tested using the application server.

When you start the Process Simulator, the Siebel Web Client is launched with a connection to the database specified in your debug options. (For information on setting debug options, see "Running the Process Simulator" on page 140.) A Watch window shows the Workflow variables (process properties) and application data (business objects and business components).

This topic is organized as follows:

- "Process Simulator Buttons" on page 138
- "Process Simulator Watch Window" on page 139

For more information on testing workflow processes, see:

- "Using the Validate Tool to Correct Errors in Workflow Processes" on page 139
- "Running the Process Simulator" on page 140
- "Testing Workflows That Involve Server Components" on page 141

Process Simulator Buttons

At the top of the Process Simulator, there is a set of buttons that you use to simulate workflow processes. Table 17 describes the buttons on the Process Simulator.

Table 17. Process Simulator Buttons

Button	Description
Start	Clicking this button activates the Start step in the process.
Next Step	Clicking this button activates the step immediately after the step that just executed.
Continue	Clicking this button resumes the process if it is stopped.
Stop	Clicking this button stops the process.

Process Simulator Watch Window

You can use the Watch window in the Process Simulator to view process properties. To view the Watch window, right-click in the Process Simulator and select Watch Window.

Table 18 describes the fields in the Watch window.

Table 18. Watch Window Fields

Field	Description
Next Step	The name of the next step in the process.
Process Status	The current status of the process.
Process Property Name	The name of the process property associated with the specific step.
Process Property Value	The value stored in the process property.
Process Property Hierarchy	Stores process property value of type hierarchy.

Using the Validate Tool to Correct Errors in Workflow Processes

The Validate tool in Siebel Business Process Designer is an error-correction mechanism that you use before simulation and deployment. You can validate a workflow process by using the context-sensitive right-click menu in the Palette Designer or in the Workflow Processes OBLE in Siebel Tools.

Validation enforces the semantic consistency of a workflow process that otherwise cannot be easily enforced by structural constraints. For example, using validation, you can make sure that an error process does not itself contain an error process. When you validate a workflow process, you are given warnings about errors the workflow may contain. You can then correct the errors before running the Process Simulator.

The Validate tool can detect the following errors:

- **Connectors not attached correctly.** Make sure all the workflow process diagram's branches are connected correctly.
- **Outgoing branches not specified for Decision steps.** Make sure to specify outgoing branches for each Decision step in the workflow process.
- **Business services and business service methods not specified for Business Service steps.** Make sure that each Business Service step in the workflow process is not missing a business service or a business service method.
- **Business components missing from Siebel Operation steps.** Make sure to specify the business component upon which each Siebel Operation step acts.
- **Subprocess names not specified for Subprocess steps.** Make sure that each Subprocess step specifies the appropriate subprocess that the workflow calls.

To validate a workflow process

1. In Siebel Tools, in the Workflow Processes OBLE, select the process you want to validate.
2. Right-click and select Validate.

   The Validate dialog box appears.

3. Click Start.

   "Starting validation..." appears in the bottom left corner of the Validate dialog box.

4. If the validation is successful, there are no errors to report and in the bottom left corner of the dialog box, a message reads, "Total tests failed: 0."

Running the Process Simulator

Before you run the Process Simulator for the first time, you must set your debug options. Then in the Process Designer, you right-click and select Simulate Workflow Process to start the Process Simulator.

CAUTION: When testing a workflow process using the Process Simulator, it is important to note that the workflow runs just as if it were called normally. For instance, if the process includes a Siebel operation such as update or add, the records in the database will be updated when you run the Process Simulator; if you are testing a workflow in which the Outbound Communications Manager is used to send email, then email will be sent to the specified recipient.

NOTE: The Workflow Process Simulator will simulate a wait period if a Wait step is specified in seconds. However, if the unit of time is specified in minutes or greater, the Simulator will simply move on to the next step.

To use the Process Simulator to test a workflow process

1. In Siebel Tools, navigate to View > Options > Debug and set your debug options. Complete the fields in the Debug tab according to the guidelines listed below, where $ represents settings specific to your setup. Enter the complete path for the Siebel executable (Siebel.exe) and the configuration file (uagent.cfg).

Executable	$SiebelClient\BIN\siebel.exe
CFG file	$SiebelClient\BIN\enu\uagent.cfg
Working directory	$SiebelClient\BIN
User name	$username
Password	$password
Data source	$datasource

2. In the Process Designer, right-click and select Simulate Workflow Process.

   A new instance of the Web Client launches according to the debug settings you entered. The My Inbox Items view appears (in the Administration - Inbox screen).

3. Click the hyperlink Debug Workflow in the Name column of the My Inbox Items list.

   The Workflow engine is invoked, and the thread is released. Control is moved to Siebel Tools.

4. Navigate back to Siebel Tools and click the Start button.

5. If the first step executes as expected, click Next Step to execute the next step in the process.

   - To use the Watch window, right-click on the canvas and select Watch Window.
   - You can use the Process Designer at any time to make changes to the step details, and then return to the Process Simulator to debug the process.

6. Continue stepping through the workflow process and verifying the results of each step in the Watch window until the process completes.

7. When you are finished, click the Stop button to terminate the simulation.

Testing Workflows That Involve Server Components

You cannot use the Process Simulator to test workflows that involve server components, such as long-running workflow processes. If a workflow process involving a server component is run in the Process Simulator, incorrect behavior will result. To test a workflow process that involves a server component, you test the workflow in the run-time environment.

For example, if you want to test a workflow process that invokes Siebel Assignment Manager, you deploy the workflow to the run-time environment. You export the workflow from Siebel Tools and import it into the Web Client. Then you test the workflow in real time with the working server components.

For interactive workflows, you must make sure that the corresponding view is assigned to the application that the user is using. For example, if the user session views are in Siebel Sales, make sure you are testing the workflow in the Siebel Sales environment, and not in the environment of another application, such as Siebel Call Center. Verify that the environment in which you are testing contains all the views you want to test.

To test a workflow process involving a server component

1. Activate the workflow process you want to test.

2. Verify that the corresponding view is assigned to the application used in the user session.

3. Reload personalization to load the run-time events in the current object manager session.

   a. From the application-level menu, choose Navigate > Site Map > Administration - Runtime Events > Events.

   b. From the applet menu, select Reload Runtime Events.

4. Test the workflow in the run-time environment.

9 For Administrators: Administering Workflow Processes

This chapter is organized as follows:

About Deploying Workflow Processes

In this release, workflow design and workflow deployment occur separately. After you have designed a workflow process in the Process Designer within Siebel Tools, you deploy the workflow process from within the run-time client application by activating it.

Figure 4 on page 31 shows the deployment architecture of Siebel Business Process Designer. Siebel Tools is connected to the server data source. The run-time client application is also connected to the server data source. The steps you and the Siebel Business Process Designer take are summarized as follows:

1. You click the Deploy button in Siebel Tools, and the workflow process is marked as Completed in the repository.
2. You log in to the run-time client and from the application-level menu, choose Navigate > Site Map > Administration - Business Process > Workflow Deployment. This is the staging environment. You activate the workflow process.
3. The workflow process definition is read from the repository and Siebel Business Process Designer writes the definition to the run-time environment as XML.
4. You deploy the workflow process with its deployment parameters (Replication, Activation Date, Expiration Date, and Monitoring Level).

You can find further information about deploying workflow processes as follows:

- See "Deploying Workflow Processes" on page 144 for general information about deploying workflow processes.
- See "Deploying Workflow Processes to Mobile Clients" on page 145 for information specific to deploying workflow processes to mobile clients.

Deploying Workflow Processes

There are two steps involved in deploying a workflow process because the workflow process definitions are stored as repository objects, while deployed workflow processes are stored in run-time tables along with their deployment parameters. You deploy workflow processes from the Siebel Tools repository to the Business Process Designer Administration client.

The first step to deploying a workflow process uses the Deploy button in the Object List Editor of Siebel Tools. The second step uses the Workflow Deployment view in the run-time client to change the status of the workflow process to Active and make it available for deployment.

NOTE: If the workflow process you are deploying includes subprocess steps or new repository objects such as business components, business services, and views, you must first make sure these subprocess steps or repository objects are available to the workflow you are deploying. In the case of subprocess steps, deploy the subprocess workflows before deploying the parent workflow, so the subprocesses are accessible to the parent workflow process. In the case of new repository objects, first compile the new repository objects so they are accessible to the workflow process you are deploying.

See "About Process Monitoring Levels" on page 152 for detailed information on deployment parameters.

To deploy a workflow process

1. After validating the workflow process by testing it, select the workflow process in the Object List Editor and click Deploy.

   The workflow's status changes from In-progress to Completed and is available as follows:

   - If you are connected to the server data source, the completed workflow process is available at run time to be activated.
   - If you are connected to the local data source, check in the workflow process. After you check in the workflow, it is available at run time to be activated.

2. In the run-time client, from the application-level menu, choose Navigate > Site Map > Administration - Business Process > Workflow Deployment and query for the workflow you just deployed.
3. With the workflow process selected, click the Activate button.

   This checks the syntax for validity, registers run-time events if used, and changes the status of the process to Active. It also changes the status of the previous active version to Outdated.

  a. If your workflow process has run-time events, you will also need to load the run-time events. From the application-level menu, choose Navigate > Site Map > Administration - Runtime Events, then click the applet menu and select Reload Runtime Events. This will load the run-time events in the current object manager session. For more information, see *Siebel Personalization Administration Guide*.

4. Set the deployment parameters for the workflow process:

   a. Set the activation date in the Activation Date/Time field.

   b. Set the expiration date in the Expiration Date/Time field.

   c. Set the replication to None, unless you are deploying the workflow process to mobile clients. If you are deploying the workflow to mobile clients, see "Deploying Workflow Processes to Mobile Clients" on page 145.

   d. Set the monitoring level in the Monitoring Level field. For more information, see "About Process Monitoring Levels" on page 152.

Now you can invoke the workflow process from any of the invocation modes: the Process Simulator, a script, or Workflow Policies.

Deploying Workflow Processes to Mobile Clients

The Replication field in the Workflow Processes view allows you to choose whether to route a workflow process definition to your mobile clients. Routing only the workflow process definitions your mobile clients need lets you reduce the amount of data in the local database. Table 19 lists the possible values for the Replication field.

Table 19. Workflow Processes View

Field	Description
Replication	Choose from these possible values: ■ **All.** The workflow process definition is routed to all mobile clients and regional nodes. ■ **Regional.** The workflow process definition is routed to the regional nodes only. ■ **None.** (Default) The workflow process definition is not routed to the mobile client or the regional nodes.

NOTE: Support of workflow processes on regional nodes is restricted. For more information, see "Deploying Workflow Processes on Regional Nodes" on page 146.

More information about using workflow processes with mobile clients is provided in the following sections:

- "Restricting Mobile Client Routing" on page 146
- "Deploying Workflow Processes on Regional Nodes" on page 146

Restricting Mobile Client Routing

Here are some things to remember when you are modifying the Replication field in order to choose whether to route a workflow process definition to your mobile clients:

- Changing the Replication field value from None to All adds the workflow process definition and all related records to the mobile client or regional node when it synchronizes with the server.
- If your workflow has a status of In Progress, you can change the Replication field value from All to None (without creating a new version of the workflow) to delete the workflow process definition and all related records from the mobile client or regional node when it synchronizes with the server.

Deploying Workflow Processes on Regional Nodes

You can execute workflow processes on regional nodes. The workflow process can be called from script or run-time events.

When executing a workflow process on a regional node, the workflow process must reside on the regional node. The settings and environment should be replicated entirely on the nodes. All the objects the workflow is referencing must be available on the regional node.

Migrating Workflow Processes from Development to Production

Once you have tested your workflow processes in your development environment, you can move them to the production environment.

NOTE: Before migrating data, make sure that all data required for the workflow process is also present in the target environment. For example, if your workflow process requires custom entries in the list of values (LOV) table, make sure that these are present and active.

Siebel applications provide two utilities for migrating workflow processes from development to production:

- **REPIMEXP.** The Repository Import/Export utility is found in the `siebel/bin` directory. Use REPIMEXP for bulk migration of repository objects, including workflow definitions. In the command-line interface, type repimexp/help to view your usage options.

  NOTE: Using REPIMEXP, you cannot pick and choose which workflows to migrate. To select a single workflow or only certain workflows for migration, use the Import/Export migration option.

- **Import/Export functionality in Siebel Business Process Designer.** Use Import/Export for incremental migration of workflow definitions. You use Siebel Tools to export workflows from one environment, and to import workflows to another environment.

  NOTE: The Workflow import/export feature is designed only to migrate individual workflow processes or small sets of workflow processes. For example, Workflow import/export will not migrate 150 workflow processes at one time. To migrate large numbers of processes, break them into smaller sets of 10 workflow processes or less.

NOTE: If you choose not to connect Siebel Tools to your production repository, you must use REPIMEXP for migration of all your workflow process definitions.

More information, see "Importing or Exporting a Process Definition" on page 147.

Importing or Exporting a Process Definition

It is a good idea to back up your process definitions regularly using the Export function. Use a meaningful naming convention when selecting a filename for an exported process to make it easy to understand the purpose of the process.

To export a process definition

1. In Siebel Tools, in the Object List Editor, select the workflow process or processes you want to export. To select more than one process, press and hold the CTRL key while selecting the processes.
2. Right-click and choose Export Workflow Process.

   The Save As dialog box appears.
3. Enter the file path, filename, and the .xml filename extension, and then click Save.

   The process or processes are exported. If you selected more than one process to export, all the processes are saved to one XML file.

NOTE: When exporting a process containing subprocesses, you must also export the subprocesses. Subprocesses are not exported automatically.

To import a process definition

1. In Siebel Tools, in the Object List Editor, select the workflow process or processes you want to import. To select more than one process, press and hold the CTRL key while selecting the processes.
2. Right-click and choose Import Workflow Process.

   The Open dialog box appears.

3. Select a path and filename of the process to import, and then click Open.

   The process is imported with a status of In Progress.

   NOTE: If a process definition of the same name exists in the target environment, the newly imported process definition's version number will be incremented by one.

Administering Workflow Processes in the Run-Time Client

You use the Administration - Business Process views in the run-time client to administer workflow processes created in Siebel Tools. The Administration - Business Process views include the following:

- **Workflow Deployment view.** For deploying and activating workflow processes. From the application-level menu, choose Navigate > Site Map > Administration - Business Process > Workflow Deployment.

- **Workflow Instance Admin view.** For viewing and controlling workflow processes that are in a running state, a waiting state (persisted workflows), and an error state. For a selected instance, you can change the value of the process properties before resuming the instance. From the application-level menu, choose Navigate > Site Map > Administration - Business Process > Workflow Instance Admin.

- **Workflow Instance Monitor view.** For monitoring all workflow process instances in all states, as well as step instances and aggregate data. An instance that is completed moves from the Workflow Instance Admin view to appear in the Workflow Instance Monitor view (when the Monitoring flag is set). From the application-level menu, choose Navigate > Site Map > Administration - Business Process > Workflow Instance Monitor.

- **Workflow Processes view.** For reviewing pre-7.7 workflow process definitions. From the application-level menu, choose Navigate > Site Map > Administration - Business Process > Workflow Processes.

  NOTE: The Workflow Processes view and its child views listed below are read-only views used only for reviewing pre-7.7 workflow processes. You view any new workflows that you create in the Workflow Processes OBLE in Siebel Tools.

  - **Process Definition view.** For reviewing the definitions of pre-7.7 workflow processes. From the application-level menu, choose Navigate > Site Map > Administration - Business Process > Workflow Processes > Process Definition.

  - **Process Designer view.** For reviewing the process flow diagrams of pre-7.7 workflow processes. From the application-level menu, choose Navigate > Site Map > Administration - Business Process > Workflow Processes > Process Designer.

  - **Process Properties view.** For reviewing the properties of pre-7.7 workflow processes. From the application-level menu, choose Navigate > Site Map > Administration - Business Process > Workflow Processes > Process Properties.

You can use the Administration - Business Process views for the following tasks:

- "Activating Workflow Processes" on page 149

- "Stopping a Workflow Process Instance" on page 149

- “Deleting a Workflow Process Instance” on page 150
- “Purging a Workflow Process Instance from the Log” on page 150
- “Recovering Workflow Processes” on page 127
- “Monitoring Workflow Process Instances” on page 151

Activating Workflow Processes

Workflow components and definitions are defined as Siebel Tools objects and are stored in the Siebel Tools repository. When you deploy a workflow process, its definition is read from the repository and written to run-time tables.

Before you can run a workflow process, you must take two steps. First you deploy the workflow process from Siebel Tools, by clicking the Deploy button in the OBLE. This marks the workflow process Completed in the repository. Then you activate the workflow process from the Workflow Deployment view in the Siebel Web Client, by clicking the Activate button. This updates the status of the workflow process record to Active.

For more details about activating workflow processes, see “To deploy a workflow process” on page 144.

Activating Fields Used by Workflow Processes

Fields that are not activated by the Object Manager must be activated in order for Workflow to be able to reference and use them. When fields are exposed on the user interface, they are activated by the Object Manager, so workflow processes (running on the Object Manager) that reference these fields are able to run properly. But when fields are not exposed on the user interface, they are not activated by the Object Manager, so the Object Manager does not reference them. In this case, a workflow process running on the Object Manager cannot use these fields, and an error results.

To activate the fields necessary for a workflow process to run, do one of the following:

- Activate the fields through scripting. For example, use a business service to activate all the fields to be used by the workflow process.
- Expose the fields on the user interface.
- At the business component level, explicitly activate the fields by setting the fields’ property Force Active = TRUE.

Stopping a Workflow Process Instance

Workflow administrators can stop a process instance if persistence is defined. After a workflow process instance is stopped, it is removed from the system. Process instances with a status of Running, Waiting, or Error can be stopped.

To stop a process instance

1. From the application-level menu, choose Navigate > Site Map > Administration - Business Process > Workflow Instance Admin.
2. In the Related Instances applet, select the process instance to stop.
3. In the applet menu, choose Stop Instance.

CAUTION: Stopped processes cannot be resumed.

Deleting a Workflow Process Instance

If you know the specific instance you want to stop, stop the workflow process. See "Stopping a Workflow Process Instance" on page 149.

NOTE: If you stop a running workflow process instance, the execution of the process instance is terminated. This is different from purging a workflow process instance from the log. See "Purging a Workflow Process Instance from the Log" on page 150.

If you want to stop all process instances of a specific workflow process definition, then you can delete the process definition from within the Workflow Deployment view.

To stop all process instances of a workflow process definition

1. From the application-level menu, choose Navigate > Site Map > Administration - Business Process > Workflow Deployment > Active Workflow Processes.
2. Select the process instance.
3. From the applet menu, select Delete Process.

Purging a Workflow Process Instance from the Log

Using the purge function, an administrator can delete all process instances with a status of Stopped or Completed before the user-specified date. If you want to delete a paused instance, stop the instance first.

NOTE: If you delete a running workflow process instance from the log, the execution of the process instance is unaffected and the workflow process will continue to run.

To purge all process instances from the log

1. From the application-level menu, choose Navigate > Site Map > Administration - Business Process > Workflow Instance Monitor > Process Instances.
2. Click Purge.
3. In the Workflow Instance Monitor Purge dialog box, select a date.
4. Click Purge.

Monitoring Workflow Process Instances

This section provides information on monitoring of workflow processes, organized as follows:

- "About Workflow Process Monitoring" on page 151
- "About Process Monitoring Levels" on page 152
- "Setting the Monitoring Level" on page 153
- "About Tracing and Event Log Levels" on page 154
- "Increasing Tracing Levels for Workflow Management Server Components" on page 155

About Workflow Process Monitoring

This release provides extended process monitoring and error notification, as well as improved execution tracing. There are two views you can use to monitor workflow processes:

- Workflow Instance Monitor view. See "Workflow Instance Monitor View" on page 151.
- Workflow Instance Admin view. See "Workflow Instance Admin View" on page 152.

Workflow Instance Monitor View

Use this view to see the history of all workflow process instances in all states, as well as step instances and aggregate data. When the Monitoring flag is set for a deployed workflow process definition, the workflow process instance remains in the Workflow Instance Monitor view after it has completed and is no longer visible in the Workflow Instance Admin view. The Workflow Instance Monitor view provides a log of workflow instances.

From the application-level menu, choose Navigate > Site Map > Administration - Business Process > Workflow Instance Monitor.

The top applet in this view lists workflow definitions.

NOTE: The top applet shows the workflow definitions for all workflow processes that have monitoring turned on, that is, the Monitoring Level is not NONE.

The other applets in this view are the following:

- **Process Instances.** This applet shows the related log instances for the selected workflow process.
- **Step Instances.** This applet shows the steps and process properties for the selected process instance.
- **Aggregate Data.** This applet shows aggregate data as a chart view for the selected workflow process.

NOTE: Depending on the monitoring level set for the selected workflow process, you may see no records in the Step Instances and Aggregate Data applets.

Workflow Instance Admin View

Use this view for monitoring running workflow processes. This view shows all processes that are in running, waiting, and error states and that have persistence set. Persistence is set if the workflow process has a Wait step or the workflow's Auto Persist flag is checked (for 7.0 workflows).

From the application-level menu, choose Navigate > Site Map > Administration - Business Process > Workflow Instance Admin.

The top applet in this view lists all workflow process instances that are running, in an error state, or in a waiting state.

The other applets in this view are the following:

- **Related Instances.** This applet shows the associated processes for the selected parent workflow process—subprocesses and error processes.
- **Process Properties.** This applet shows the process properties for the process instance selected in the Related Instances applet. You can change the value of these process properties before resuming an instance that is waiting or in an error state.

Information on workflow process monitoring and administration is organized as follows:

- "About Process Monitoring Levels" on page 152
- "Setting the Monitoring Level" on page 153
- "About Tracing and Event Log Levels" on page 154
- "Increasing Tracing Levels for Workflow Management Server Components" on page 155

About Process Monitoring Levels

You set deployment parameters to determine the way you monitor your workflow processes. These deployment parameters are Monitoring Level and Log-writing Frequency.

Monitoring Level Parameters

Table 20 shows the monitoring level parameters (and their corresponding log-writing frequencies) that you can set for a workflow process. When the workflow instance is created, the monitoring level is read from the workflow process definition and stays throughout the lifetime of the instance unless the instance is paused. In the case of an instance being paused, when the instance is resumed, the monitor level is reread from the definition.

Table 20. Monitoring Level Parameters

Monitoring Level	Record Process Instance	Record Step Instances	Record Process Properties
0 None	N	None	None
1 Status	Y	None	None
2 Progress	Y	All Steps	None

Table 20. Monitoring Level Parameters

Monitoring Level	Record Process Instance	Record Step Instances	Record Process Properties
3 Detail	Y	All Steps	All Steps
4 Debug	Y	All Steps	All Steps

Log-writing Frequency Parameters

Log-writing frequency determines the frequency at which monitoring data is written to the disk. The frequency is optimized internally by the Workflow run-time environment, based on the workflow type and the monitoring level you select.

You can set the log-writing frequency parameters directly only by using the Debug monitoring level. At the Debug monitoring level, the log is written to the disk after every step.

NOTE: Enabling monitoring at any level incurs performance overhead to your workflow processes. It is best to set the monitoring level to 0 (None) or 1 (Status) on workflow processes running in production. Monitoring levels of 2(Progress) and higher should only be used for the debugging of workflow processes.

Monitoring Levels and 7.0 Compatibility

7.0 workflows with persistence frequency and persistence level set are mapped to a monitoring level based on the following logic:

- If the persistence frequency is set to NONE, then the monitoring level is set to NONE.
- If the persistence frequency is ON_PAUSE or EVERY_STEP, persistence is explicitly turned on and the monitoring level is set as follows:
  - If the persistence level is ALL_STEPS, then the monitoring level is set to PROGRESS.
  - If the persistence level is CURRENT_STATE, then the monitoring level is set to STATUS.

NOTE: In this release, persistence and monitoring are separate features that serve different functions. Persistence is a quality of service and is controlled at definition time. Monitoring is an administrative tool and is controlled at deployment time. Monitoring has *no* impact on the way a workflow process functions.

Setting the Monitoring Level

To use workflow process monitoring, set the deployment parameter called Monitoring Level.

To set the Monitoring Level parameter

1. From the application-level menu, choose Navigate > Site Map > Administration - Business Process > Workflow Deployment.

2 In the Active Workflow Processes applet, complete the Monitoring Level field. See "About Process Monitoring Levels" on page 152 for the information on the various levels you can choose.

Troubleshooting Workflow Processes

You can use tracing levels to troubleshoot workflow processes. Information in this topic is organized as follows:

- "About Tracing and Event Log Levels" on page 154
- "Increasing Tracing Levels for Workflow Management Server Components" on page 155
- "Siebel Application Response Management (Siebel ARM)" on page 155
- "Siebel Flight Data Recorder (FDR) Files" on page 156

About Tracing and Event Log Levels

Table 21 lists the events that Siebel Workflow uses for logging.

Table 21. Workflow Process Logging Events

Event	Level	Description
Workflow Engine Invoked	4	Trace methods invoked and arguments passed to the Workflow engine
Workflow Process Execution	4	Trace workflow process execution
Workflow Step Execution	4	Trace workflow step execution
Workflow Performance	4	Measure overall process execution time
Workflow Performance	5	Measure process and step execution time
Workflow Recovery	3	Trace instance recovery status and progress (applicable only to the Workflow Recovery Manager server component)
Workflow Recovery	4	Trace instance recovery details (applicable only to the Workflow Recovery Manager server component)

NOTE: Setting trace levels above default parameters will affect performance. Trace levels should be reset to default parameters after troubleshooting has been completed.

For more information on tracing levels, see "Increasing Tracing Levels for Workflow Management Server Components" on page 155.

Increasing Tracing Levels for Workflow Management Server Components

You can generate a more detailed trace file to assist in troubleshooting Workflow Process Manager, Workflow Process Batch Manager, and Workflow Recovery Manager.

NOTE: Complete these steps prior to executing the server process.

To increase tracing levels

1. From the application-level menu, choose Navigate > Site Map > Administration - Server Configuration > Server > Components > Events.
2. In the Components applet, choose the component for which you want to generate tracing: the Workflow Process Manager, the Workflow Process Batch Manager, or the Workflow Recovery Manager.
3. Click the Events tab to view all the configurable event types for the selected component.

   The log level is set to a default value of 1.

4. Change the log level value to 3, 4, or 5. Use Table 21 on page 154 to make your choice.
5. (Optional) For additional troubleshooting, you may want to repeat Step 4 to increase the tracing level for the Object Manager SQL log event.

NOTE: More tracing information is generated as the Component Event Configuration Log Level value increases.

Siebel Application Response Management (Siebel ARM)

The Workflow Process Manager server component is enabled for Siebel Application Response Measurement (Siebel ARM). Siebel ARM captures timing data useful for monitoring the performance of the Siebel application, and records this information to binary files.

Siebel ARM settings affect the Workflow Process Manager as follows:

When the Siebel ARM level is set to 1, the Workflow Process Manager business service records:

- The time it takes to execute a service method

  Examples of service methods in Workflow Process Manager are RunProcess and ResumeInstance.

When the Siebel ARM level is set to 2, the Workflow Process Manager business service records:

- The time it takes to execute a service method
- The time it takes to execute a workflow step
- The time it takes to write monitoring data to the disk

  Siebel ARM level 2 can help you determine the logging overhead when you increase the monitoring level of your workflow process.

Table 22 shows the Siebel ARM areas and subareas defined for Siebel Workflow, and their Siebel ARM levels.

Table 22. Siebel ARM Areas and Levels

Area	Sub-area	Level	Description
WORKFLOW	CORDR_RESUME	1	Resume a suspended process.
WORKFLOW	CORDR_EXECUTE	1	Execute a process.
WORKFLOW	ENGNE_INVOKE	1	Invoke a Workflow Process Manager service method.
WORKFLOW	STEPS_EXSTEP	2	Execute a step.
WORKFLOW	MONTR_WRTE	2	Write process monitoring data to disk.

For more information on enabling Siebel ARM and the Siebel ARM analyzer, see *Performance Tuning Guide*.

Siebel Flight Data Recorder (FDR) Files

Siebel flight data recorder log files (extension .fdr) are records of system and server component behavior at run time. In the event of a system or server component failure, the settings and events leading up to the failure are captured and logged. The Siebel flight data recorder log file can then be forwarded to Siebel Technical Support and used to troubleshoot and analyze the specific settings and events that occurred prior to the failure. The Siebel flight data recorder log files are stored in the Binary subdirectory of the Siebel Server root directory.

FDR instrumentation points have been embedded in the Workflow Process Manager business service and the Workflow Recovery Manager business service to provide capture-processing details in case of a system failure or server component failure.

For more information on Siebel flight data recorder files, see *System Monitoring and Diagnostics Guide for Siebel eBusiness Applications*.

10 Workflow Policies

This chapter provides information about how to use workflow policies. The information is organized as follows:

About Planning Workflow Policies

Information on planning activities for workflow policies is provided in the following topics:

Planning Workflow Policy Groups

Before you create your Workflow policies, you need to create workflow policy groups. Each Workflow Policy Agent is assigned one workflow policy group. If you are going to run only one Workflow Policy Monitor Agent and one Workflow Policy Action Agent, assign all your policies to one policy group.

The reasons to use multiple Workflow Policy Agents are:

- To shorten the time between when the policy event is triggered and when Workflow Policies notices the event
- To spread the workload across multiple application servers
- To adjust the polling interval so that polling for noncritical policies does not prevent efficient processing of critical policies

Policies with similar time intervals are generally grouped together. By creating groups of policies with similar time intervals, you can assign the workflow policy group a Workflow Policy Agent with a polling rate that matches the needs of the workflow policies—creating a more efficient use of your resources.

Creating workflow policy groups and using multiple Workflow Policy Agents are part of tuning your system to create the highest performance and can be done as you monitor your system's performance.

Planning Workflow Policies

Once you have gathered policy information, you can begin to plan the workflow policies.

Many of the workflow policy objects and programs you need to create your workflow policies have been predefined by Siebel. However, you can use Siebel Tools to augment programs, create additional workflow policy objects, or make additional workflow policy columns available for monitoring. See "About Customizing Workflow Policies with Siebel Tools" on page 189 for more information on how to perform these tasks.

The planning phase is a good time to review your company's business process tasks. You want to determine which tasks can be automated with Workflow Policies and then prioritize the implementation sequence. It is always a good idea to create and implement a small group of policies at a time. After you successfully implement the group, you can proceed to another small group of policies in a systematic manner. See "About Creating Workflow Policies" on page 163 for more information on creating workflow policies.

NOTE: After planning a new workflow policy, test the policy definition by creating a query based on the policy. Then you can execute the query on your current production environment. The query response can help you determine the frequency of the workflow conditions. You may find that a policy creates excessive notification or insufficient visibility. See "About Testing Workflow Policies" on page 244 for more information.

Determining What to Monitor When Planning Policies

The first step in planning is to identify the purpose of the policy and the specific database information that needs to be monitored. For example, if the service department wants to send an email to the service request contact whenever a service request is opened with a severity level of critical, the information to record would be that which is listed in Table 23.

Table 23. Determining What to Monitor

What to Monitor	Purpose of the Policy
Service request status Service request severity	Send an email to the service request contact when the service request is opened and its severity level is critical.

Planning Policies and Conditions

The second step in planning is to define the policy properties and conditions, identify the workflow policy object to be monitored in the Siebel database, and determine the monitoring interval period and duration.

Table 24 illustrates the type of information you need to model the general policy definition in terms of Workflow Policies. It shows the workflow policy name as Email Confirmation of SR, the workflow policy object is Service Request, monitoring interval period (Workflow Group) is Medium Frequency, and the duration is set to 0.

Table 24. Planning Policies

Name	Workflow Object	Workflow Group	Duration
Email Confirmation of SR	Service Request	Medium Frequency	0

NOTE: Duration indicates the time element that must be met before an action is performed. Each workflow policy has one duration, so if you need to cause an action to occur after one hour, two hours, and six hours, you must create a different policy for each duration.

After you determine your policy's workflow object and other properties, you need to define the workflow conditions, as shown in Table 25. Conditions are in the form of an expression.

Table 25. Workflow Policy Conditions

Field (Column Monitored in the Database)	Comparison	Value
Service Request Severity	=	1-Critical
Service Request Status	=	Open

Planning Workflow Policy Actions

The third step in planning is to define the policy actions. A policy action is executed when the conditions of the policy have been met. Table 26 illustrates the type of information you need to define a workflow policy action.

Table 26. Workflow Policy Actions

Action Name	Program	Workflow Object	Arguments
Send SR Email to Contact	Send SR Email	Service Request	Send to Contact

NOTE: Workflow Policies comes with a set of predefined actions and programs. You can use these or define your own actions or programs to suit your business needs.

Scenario for Planning Workflow Policies: Notification for 30%+ Discounts

In this scenario, the sales department manager wants to be automatically notified whenever sales representatives quote discounts over 30%. Table 27 lists a workflow policy that monitors quotes with a discount exceeding 30%, for which the purpose is to notify the sales manager to review and approve the quote.

Table 27. Determining What to Monitor

What to Monitor	Purpose of the Policy
Quotes with a discount exceeding 30% need Sales Manager approval	Notify Sales Manager to review and approve the quote.

Table 28 shows the workflow policy name as Notify Sales Manager on Sales Approval. The workflow policy object is Quote, the workflow policy group is Low Frequency, the duration is set to 0, and the quantity is set to 5. This means that the workflow policy action occurs as soon as five new quotes meet the criteria of the workflow policy conditions.

Table 28. Planning Policies

Name	Workflow Object	Workflow Group	Duration	Quantity	Comments
Notify Sales Manager on Sales Approval	Quote	Low Frequency	0	5	Notify the manager when a quote with a discount over 30% is created.

Table 29 illustrates the type of information you need for the policy conditions.

Table 29. Workflow Policy Conditions

Field (Column Monitored in the Database)	Comparison	Value
Quote Status	=	In Progress
Quote Item Discount Percent	>	30

Next, define the workflow policy actions that occur when the conditions of the policy are met. You can also define the action arguments, such as the email subject and the message template, using dynamic values. Table 30 lists definitions for the Send Email to Sales Manager action.

Table 30. Actions and Action Arguments

Action Name	Program	Workflow Object	Arguments and Substitutions
Send Email to Sales Manager	Send Quote Email	Quote	Subject: Please approve quote discount for [Account] Message Template: Please approve the quote discount for quote [Quote Number] and notify [Last User First Name] [Last User Last Name] Repeating Message: The following quotes also need approval [Quote Number]

Scenario for Planning Workflow Policies: Notification for Large Number of Open Service Requests

In this scenario, the service department wants to automate its notification policy when the number of open requests for an agent reach a critical mass of 20. The tables below show the information needed to define this type of workflow policy.

Table 31 represents the general policy definition.

Table 31. Determining What to Monitor

What to Monitor?	Purpose of the Policy
Monitor open service requests when they reach a quantity of 20	Send a Message Broadcast to the service representative to alert the representative about the situation.

Next, model the general policy definition in terms of Workflow Policies.

Table 32. Workflow Policies

Name	Workflow Object	Workflow Group	Quantity
Over 20 Open Service Requests	Service Request	High Frequency	20

Table 32 shows the policy name is Over 20 Open Service Requests, workflow policy object is Service Request, workflow policy group is High Frequency, and the quantity is 20.

After you determine the policy's workflow object and other properties, define the workflow conditions for your workflow policy. Table 33 shows the workflow condition definitions.

Table 33. Workflow Conditions

Field (Column Monitored in the Database)	Comparison	Value
Service Request Status	=	Open

Define the workflow policy actions that occur when the conditions of the policy are met. You can also define the action arguments. Table 34 on page 162 shows the action argument definitions.

Table 34. Actions and Action Arguments

Action Name	Program	Workflow Object	Arguments and Substitutions
Alert Agent of Open SR	Send SR Message Broadcast	Service Request	Abstract: You have over 20 service requests Message Template: You have over 20 service requests. Please review your service request queue.

Defining a Test and Migration Strategy for Workflow Policies

Before implementing new workflow policies, you must verify them in a test environment consisting of a sample Siebel database and workflow policies test data. Testing new policies, conditions, and actions checks that the policy you release into the production environment properly executes and does not cause conflicts with your existing workflow policies.

The following are some suggestions for setting up your test and migration policy:

- Make sure your test environment and production environment have identical versions of the software and that you are using realistic data in your database by using a partial or complete copy of the production database.
- Create a small group of workflow policies to implement as a first phase of implementation. After you have successfully implemented the first group, you can add more policies in a systematic manner.
- To verify a new workflow policy, go to your production environment, manually create a query based on the new policy, and check the response. This helps you determine if a policy creates excessive notification or misses the rows you want to monitor.

For more information on migrating your test environment to your production environment, see "Migrating Policies to the Production Environment" on page 246.

About Creating Workflow Policies

Information about creating Workflow policies is provided as follows:

- "About the Workflow Policies Views" on page 163
- "Defining Workflow Policy Actions" on page 164
- "Using the Send Page Program Type" on page 165
- "Using the Send Message Program Type" on page 166
- "Using the Message Broadcast Program Type" on page 167
- "Using the Run External Programs Type" on page 168
- "Using the Database Operation Program Type" on page 169
- "Creating a Workflow Policy Action" on page 172
- "Creating Workflow Policy Groups" on page 178
- "Creating Workflow Policies" on page 179

About the Workflow Policies Views

The views you use to create and define workflow policies are a part of the Siebel Business Process Designer. To display the Workflow Policies view, you navigate to Workflow Policies in the Business Process Administration screens of Site Map.

Several views are used when you create Workflow policies. These views include:

- **Workflow Policies Action view.** Use to create the actions you want to use with your workflow policies. See "Defining Workflow Policy Actions" on page 164.
- **Workflow Policies Groups view.** Use to create the workflow policy groups to use with workflow policies. See "Creating Workflow Policy Groups" on page 178.
- **Workflow Policies Policies view.** Use to create workflow policies. See "Creating Workflow Policies" on page 179.

- **Workflow Policies Explorer view.** Use to review the currently defined workflow policy objects.
- **Workflow Policies Log view.** Use to review the workflow policy monitor log, to check policy trends, and to check policy frequency.

Defining Workflow Policy Actions

Workflow policy actions are events that you want to occur when the conditions of your workflow policy are met. You must create the appropriate workflow policy actions before you create the policy that will use the actions.

In the Siebel Business Process Designer, you use the Workflow Policies Action view to define policy actions. This view and its associated fields are described below. For the procedure on creating a workflow policies action, go to "Creating a Workflow Policy Action" on page 172.

The Workflow Policies Action view consists of three applets. These applets are:

- **Actions applet.** This is where you create a name for your action and choose the appropriate program. See "About the Actions Applet in the Workflow Policies Action View" on page 164.
- **Arguments applet.** This is where you define the arguments for the action. The format of the Arguments applet changes depending on the program type of the action. See "About the Arguments Applet in the Workflow Policies Action View" on page 165.
- **Recipients applet.** This is the contact name, employee name, position, or relative of the workflow policy object that can receive an email, page, or message broadcast. See "About the Recipients Applet" on page 171.

NOTE: You cannot invoke DLLs or external functions using workflow policy actions. Use workflow processes for this.

About the Actions Applet in the Workflow Policies Action View

Table 35 describes the fields of the Workflow Policies Actions applet.

Table 35. Actions Applet Fields

Field	Description	Possible Value
Name	The name of the workflow policy action.	A descriptive name that is: ■ Consistent with your overall naming strategy ■ Meaningful to the policy maker
Program	The workflow policies program associated with the action.	This program is chosen from a picklist. See "Predefined Programs" on page 247

Table 35. Actions Applet Fields

Field	Description	Possible Value
Workflow Object	Workflow policy object that this action is associated with. This action is now available only to policies that are based on this workflow policy object.	Chosen from a picklist of workflow policy objects.
Comments	Comments describing the purpose or use of this action.	Any text.

About the Arguments Applet in the Workflow Policies Action View

The format of the Workflow Policies Arguments applet varies, depending on the program type you select for the workflow policy action.

NOTE: Program arguments are case sensitive. You must enter the correct case. Use the argument picklists when possible instead of entering the arguments yourself.

This section describes each workflow policy program type, the available workflow policy program arguments and valid values, and some usage scenarios.

The available workflow policy program types are:

- **Send Page.** See "Using the Send Page Program Type" on page 165.
- **Send Message.** See "Using the Send Message Program Type" on page 166.
- **Message Broadcast.** See "Using the Message Broadcast Program Type" on page 167.
- **Run External.** See "Using the Run External Programs Type" on page 168.
- **Database Operation.** See "Using the Database Operation Program Type" on page 169.

NOTE: Before using the email or paging functions, you need to perform the setup procedures described in "About Workflow Policies Server Administration" on page 217

Using the Send Page Program Type

The Send Page Arguments applet displays if you select the Send Page workflow policy program type in the Workflow Policies Actions applet.

Send Page Arguments and Values

Table 36 shows the arguments and valid values for the Send Page workflow policy program type.

NOTE: Numeric paging is inherently unreliable because of a lack of a computer protocol for sending numeric pages. If you must send a numeric page, you can use the Pager Pin field in the employee table to control the delay between dialing the paging phone number and sending the numeric message. Add commas to the Pager Pin field. Each comma is roughly equal to a half-second delay. Avoid using the numeric paging feature in mission critical applications.

Table 36. Send Page Program Type

Argument	Valid Values When Used by Action
Numeric Message Template	Numeric message when pager is numeric.
Alpha Message Template	Text message when pager is alphanumeric. "Current" is a reserved word in Siebel Workflow. Do not use this word in messages.
Available Substitutions	Dynamic fields that you can use in the Alpha Message Template. When the action executes, the substitution value is populated with the value from the record that meets all 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.

When setting the Send Page arguments, note the following:

- Siebel Workflow Policies automatically determines the correctly formatted message depending on what type of pager the person being paged has.
- If neither of the message arguments has a value, Workflow Policies logs an error message and the action is not completed.
- You can send only pages to employees. The pager information for an employee is stored in the Employee Administration view. The Siebel database currently does not store pager information for contacts.
- Messages support substitution of values that come from the Available Substitutions field.

Using the Send Message Program Type

When you select the Send Email workflow policy program in the Actions applet, the Send Message Arguments applet displays along with the Recipients applet.

The Send Message Arguments applet allows you to create an email template used to build the message sent to the recipient specified in the Recipients applet.

Send Email Arguments and Values

Table 37 shows the arguments and valid values for the Send Email workflow policy program type.

Table 37. Send Email Workflow Policy Program Type

Argument	Valid Values When Used by Action
Subject	Subject line of email message.
Message Template	Text of message. Maximum length is 2000 characters, including variable substitutions. "Current" is a reserved word in Siebel Workflow. Do not use this word in a message.
Repeating Message	Message that is repeated when the Consolidate flag is checked on the Workflow Policies Policies view. "Current" is a reserved word in Siebel Workflow. Do not use this word in messages.
Available Substitutions	Dynamic fields that you can use in Subject, Message Template, and Repeating Message. When the action executes, the substitution value is populated with the value from the record that meets all the 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.

Using the Message Broadcast Program Type

The Message Broadcast Arguments applet appears if the Send Message Broadcast workflow policy program is selected in the Actions applet.

Send Message Broadcast Arguments and Values

Table 38 shows the arguments and valid values for the Message Broadcast workflow policy program type.

Table 38. Message Broadcast Workflow Policy Program Type

Argument	Valid Values When Used by Action
Activation	Date and time for which the message broadcast is active. The variable CURRENT can be used when specifying the activation date. See "Entering Date Calculations" on page 187 for more information.
Expiration	Date and time when the message broadcast expires. The variable CURRENT can be used when specifying the activation date. See "Entering Date Calculations" on page 187 for more information.
Abstract	Short description of the message broadcast.
Message Template	Text of message to broadcast. Maximum length is 2000 characters, including variable substitutions. "Current" is a reserved word in Siebel Workflow. Do not use this word in a message.
Severity	Severity of message to broadcast.
Available Substitutions	Dynamic fields that you can use in the Abstract and Message Template. When the action executes, the substitution value is populated with the value from the record that meets all of the policy conditions.

Activating the Check New Broadcasted Message Workflow Policy

If you use any workflow policy that contains a workflow policy program type = Send Message Broadcast and you enable message broadcast caching on an object manager component, then you must activate the Check New Broadcasted Message workflow policy, which belongs to the Siebel Messaging policy group.

The Check New Broadcasted Message policy monitors the S_BRDCST_MSG table and invokes the Notify Broadcasted Message workflow process to broadcast any new message added to the table.

For information on configuring message broadcast caching, see *Applications Administration Guide*. For information on activating a workflow policy, see "Creating Database Triggers" on page 217.

Using the Run External Programs Type

The External Programs Arguments applet appears if a Run External workflow policy program type is selected in the Actions applet. An example of a Run External Program is described in "Example of a Workflow Policy Action: Creating a Run External Program Action" on page 176.

Run External Arguments and Values

Table 39 shows the arguments and valid values for the Run External workflow policy program.

Table 39. Run External Workflow Policy Program Type

Argument	Valid Values When Used by Action
Executable Name	Path and name of executable to run. For example, the executable will launch from the Siebel Server. The executable can be a batch program.
Command Line	The command line to use. The parameters that you want to pass to the executable.
Execute Type	■ **Wait.** Workflow Policies waits for the external program to complete and examines the return code of the external program. If the return code is not 0, an error condition occurs. ■ **No Wait.** Workflow Policies executes the external program in the background and then continues processing. The return code is not checked. Note that for Visual Basic programs which create files, set Execute Type to Wait to avoid possible corruption of files. When set to No Wait, Visual Basic attempts to write files twice, thus corrupting the data.
Available Substitutions	Dynamic fields that can be used as command line parameters. When the action executes, the substitution value is populated with the value from the record in violation.

If no path is supplied for the Executable Name argument, the executable is assumed to be in the current path of Workflow Policies running on the Siebel Server. For example, your Siebel Server may be installed on C:\siebsrvr. The default path for the executable name would be C:\siebsrvr\bin.

NOTE: The external program cannot be one that is interactive, requires a user interface, or accesses the Windows desktop.

Using the Database Operation Program Type

Siebel Business Process Designer has a number of database operation programs already predefined. All you need to define are the parameters.

The Arguments applet appears if you select a database operation program such as Create Opportunity Activity in the Actions applet.

Database Operation Arguments and Values

Table 40 shows the arguments and valid values for the Database Operation workflow policy program.

Table 40. Database Operation Workflow Policy Program Type

Argument	Valid Values When Used by Action
Name	Name of column to be updated.
Required	Indicates the argument is required.
Value	Updated value of the column. You can use substitutions in the value if they were defined in the program. The syntax for adding substitutions to the value is square brackets around the variable, for example, [SR Num].

About the Recipients Applet

Table 41 describes the fields in the Recipients applet.

Table 41. Recipients Applet Fields

Field	Description	Possible Value
Recipient Type	The possible values are dependent on the workflow policy program selected for the action. Recipients apply to workflow policy programs that are of the Send Email, Send Page, and Send Message Broadcast type.	■ **Send to Employee.** Picklist of employees. ■ **Send to Position.** Picklist of positions. ■ **Send to Contact.** Picklist of contacts. ■ **Send to Relative.** Send to an individual or group of individuals (such as a service request owner or opportunity team member) related to the Workflow object (such as an Opportunity or Service Request). ■ **Send to Address**. Represents a direct email address for programs that send email.
Recipient Name	Name of the recipient based on the recipient type.	Contact name, employee name, position, or relative of workflow policy object to send an email, page, or message broadcast. **NOTE:** Messages are sent based on position. When choosing an employee as the recipient, note that if multiple employees occupy the same position, the message is sent to all of them, even though you are only choosing one employee as the recipient.

The Send to Relative recipient type sends an email or page to an individual or position associated with the current record. For example, you can send an email to the primary sales representative of an opportunity (when all the conditions of the policy are met).

Send to Relative is also used when you wish to configure a custom Send to <xxxx> recipient. Because you must use one of the Recipient Type choices in the picklist (Send to Employee, Position, Contact, or Relative), you use Send to Relative to define a custom recipient type.

NOTE: Email Manager does not send email to the same recipient twice for the same action. If it detects that an email has already been sent to a specific email address, another one is not sent. If the Send to Relative type returns more than one recipient, each recipient is sent an email as long as each email address is unique.

The Send to Position recipient type allows you to send to the primary employee of this position without having to know the name of the person (the employee must be ACTIVE). The Send to Contact recipient type allows you to pick any available contact in the Siebel system.

NOTE: The only time the Action Recipients applet is available is when a Send Email, Send Page, or Send Message Broadcast program is selected in the Actions applet.

Creating a Workflow Policy Action

The procedure for creating a workflow policy action is described below. Examples of creating workflow policy actions for specific workflow policy programs follow the procedure.

To create a workflow policy action

1. From the application-level menu, choose Navigate > Site Map > Administration - Business Process > Actions.
2. In the Actions applet, click New Record from the applet-level menu and enter the name of the workflow policy action. This name is the one that appears in the Actions Applet of the Workflow Policies Policies view.
3. Pick a workflow policy program type from the picklist in the Program field.
4. Select a workflow policy object, if applicable, from the picklist in the Workflow Object field. If you specify a workflow policy object, this action appears only on the Actions Applet of the Workflow Policies Policies view for policies associated with this workflow policy object.
5. Enter a description of the purpose of the action in the Comments field.
6. In the Arguments applet, pick one or more of the arguments and enter the appropriate value. The available arguments change according to the workflow policy program type you select in Step 2.

   NOTE: See "About the Arguments Applet in the Workflow Policies Action View" on page 165 for a description of the Argument applet for the specific workflow policy program types.

7. If the workflow policy program is either Send Email, Send Page, or Send Message Broadcast, enter the recipients of the action in the Recipient applet.

NOTE: You cannot execute a business service from a Workflow Policy Action.

Examples of Workflow Policy Actions

The following are several examples of workflow policy actions for specific situations. You can use these examples as the basis for creating your own workflow policy actions.

- "Example of a Workflow Policy Action: Creating a Send Page Action" on page 173
- "Example of a Workflow Policy Action: Creating a Send Email Action with a Repeating Message" on page 174
- "Example of a Workflow Policy Action: Creating a Send Message Broadcast Action" on page 175
- "Example of a Workflow Policy Action: Creating a Database Operation Action" on page 176

- "Example of a Workflow Policy Action: Creating a Run External Program Action" on page 176

Example of a Workflow Policy Action: Creating a Send Page Action

You may want a page sent to the support manager whenever a service request priority becomes very high and the service request is not assigned to anyone. Use the following procedure to define a workflow policy action for this situation.

To send a page whenever a service request is set to the highest value

1 In the Workflow Policies Actions view, fill in the Actions applet fields as follows:

   a Create a new record in the Actions applet and enter the name of the action:

   ```
   Page Support Manager when SR request changes
   ```

   b Select a predefined workflow policy program from the Program field picklist:

   ```
   Send SR Page
   ```

   c Select a predefined workflow policy object from the Workflow Object field picklist.

   ```
   Service Request
   ```

   NOTE: The workflow object field fills in automatically only when a workflow policy object is specified in the workflow policy program being selected. You pick a workflow policy object from the picklist when it does not automatically fill in.

2 Fill in the Send Page Arguments applet.

   a Select dynamic fields from Available Substitutions.

   b Enter text and dynamic fields in the Alpha Message Template:

   ```
   The [SR Status] of [SR Number] has changed.
   ```

   You use the Numeric Message Template for numeric paging and the Alphanumeric Message Template for alphanumeric paging. The type of paging to use is indicated by the pager type in the employee table.

3 Fill in the Recipients applet.

   a Select a predefined recipient type from the Recipient Type field picklist:

   ```
   Send to Position
   ```

   b Select Recipient Name from the Recipient Name picklist:

   ```
   Support Manager
   ```

This action is now available to use in a workflow policy.

Example of a Workflow Policy Action: Creating a Send Email Action with a Repeating Message

In this example, the vice president of sales wants to be notified only after a specific number of deals fail to close. Because this action will be used with a workflow policy that uses the Batch feature, you will need to enter relevant information in the Repeating Message field of the Send Message Arguments applet. This is because the recipient receives one email with a consolidated list of the pertinent information on each of the deals. Without a Repeating Message, the email would be sent but may not contain meaningful information.

Use the following procedure to define a workflow policy action for this situation.

To send an email with a repeating message

1 In the Workflow Policies Actions view, fill in the Actions applet fields as follows:

   a Create a new record in the Actions applet view and enter the name of the action:

   ```
   Excellent Quality Opportunity
   ```

   b Select a predefined workflow policy program from the Program field picklist:

   ```
   Send Opportunity Email
   ```

   c Select a predefined workflow policy object from the Workflow Object field picklist:

   ```
   Opportunity
   ```

   d Enter text in the Comments field:

   ```
   Send an email to the VP of Sales when deals aren’t closing
   ```

2 Fill in the Send Message Arguments applet.

   a Select dynamic fields from Available Substitutions where appropriate.

   b Enter text and/or dynamic fields in Subject:

   ```
   Opportunities not Closing
   ```

   c Enter text and/or dynamic fields in Message Template:

   ```
   Meet with [Last User First Name] [Last User Last Name] about
   [Opportunity] for [Account]
   ```

   d Enter text and/or dynamic fields in Repeating Message:

   ```
   Meet with [Last User First Name] [Last User Last Name] about
   [Opportunity] for [Account]
   ```

3 Fill in the Recipients applet.

   a Select a predefined Recipient Type from the Recipient Type field picklist:

   ```
   Send To Position
   ```

   b Select the Recipient Name from the Recipient Name picklist:

```
VP Sales
```

When you create the workflow policy for this action, check the Batch field in the Policies applet of the Workflow Policies Policies view.

Example of a Workflow Policy Action: Creating a Send Message Broadcast Action

In this example, a service department wants to automate its notification policy for open service requests when open requests for one agent reach at least 20.

To create a Message Broadcast Action for open service requests

1. In the Workflow Policies Actions view, fill in the Actions applet fields.
   a. Create a new record in the Actions applet and enter the name of the action:

      ```
      Alert Agent of Open SRs
      ```

   b. Select a predefined workflow policy program from the Program field picklist:

      ```
      Send SR Message Broadcast
      ```

   c. Select a predefined workflow policy object from the Workflow Object field picklist:

      ```
      Service Request
      ```

2. Complete the Send Message Broadcast Arguments form using message arguments and typing in static text.
   a. Enter text in Abstract:

      ```
      You have over 20 Service Requests.
      ```

   b. Enter text in Message Template:

      ```
      You have over 20 service requests. Please review your
      service request queue.
      ```

3. Fill in the Recipients applet.
   a. Select a predefined Recipient Type from the Recipient Type field picklist:

      ```
      Send to Relative
      ```

   b. Select the Recipient Name from the Recipient Name picklist:

      ```
      SR Owner
      ```

Example of a Workflow Policy Action: Creating a Database Operation Action

Two kinds of database operations are possible in Workflow Policies—insert and update. Insert allows a record to be inserted into a table in the Siebel database. The update database operation allows one or more columns in an existing record to be changed.

In the following example, a database update occurs when you use Workflow Policies to update the value of the Priority field to Very High if the Severity is Critical.

To create a Database Operation to update Service Request Priority

1 In the Workflow Policies Actions view, fill in the Actions applet fields.

   a Create a new record in the Actions applet and enter the name of the action:

   ```
   Update SR Priority to Very High
   ```

   b Select a predefined workflow policies program from the Program field picklist:

   ```
   Change SR Priority
   ```

2 Fill in the Arguments applet.

   a Select from Name picklist:

   ```
   New Priority
   ```

   b Select from Value picklist:

   ```
   1-Critical
   ```

Example of a Workflow Policy Action: Creating a Run External Program Action

In Siebel Workflow you use the action type Run External Program for defining an action that runs an external program. For example, your company could write a custom executable for calculating the quality of a new lead coming into the system. You could then call this executable from Siebel Workflow whenever the parameters for calculating the lead change.

In the first of the following examples, a program named “leadcalc.exe” is in the C:\bin directory and the action is being defined to call and execute this program. The second example provides the procedure for running external programs on UNIX.

To run an external lead calculation program

1 In the Workflow Policies Actions view, fill in the Actions applet fields as follows.

   a Create a new record in the Actions applet and enter the name of the action:

   ```
   Run Lead Calculation Program
   ```

   b Select a predefined workflow policy program from the Program field picklist:

```
Run External Program
```

c. Select a predefined workflow policy object from the Workflow Object field picklist.

NOTE: The workflow object field fills in automatically only when a workflow policy object is specified in the workflow policy program being selected. You pick a workflow policy object from the picklist when it does not automatically fill in.

2. Fill in the Run External Program Arguments applet.

   a. Enter the name of the executable:

   ```
   leadcalc.exe
   ```

   b. Enter any command line parameters.

      These are the parameters you want to pass to the executable.

   c. Select an execute type.

   d. Select dynamic fields from Available Substitutions.

3. Fill in the Recipients applet.

   a. Select a predefined recipient type from the Recipient Type field picklist:

   ```
   Send to Position
   ```

   b. Select Recipient Name from the Recipient Name picklist:

   ```
   Support Manager
   ```

This action is now available to use in a workflow policy.

To run an external program on a UNIX platform

The Run External Program workflow policy program is not supported on UNIX. However, you can use the following procedure as a workaround.

1. Define a business service that executes an external program.

   a. From the application-level menu, choose View > Site Map > Business Service Administration > Business Service Methods.

   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 Proc: Service_PreInvokeMethod.

   f. Call Clib.system in the function body, for example:

   ```
   var program = Inputs.GetProperty (“Program”)
   if (program)
   {
   Clib.system(program);
   }
   return (CancelOperation);
   ```

2. Create a workflow process calling the business service created in step 1.
   a. Add and connect a Start step, a Business Service step, and an End step.
   b. For the Business Service step, specify Run Program and Run.
   c. For the input argument for Program, specify the external program you want to run. For example, /bin/mail hkim@pcs.com </home/users/hkim/letter.txt.
3. Run your workflow process.

Creating Workflow Policy Groups

Workflow policy groups provide a means of identifying policies having similar system requirements. By grouping policies you can optimize your system, balance system loads, and provide scalability. All workflow policies must be assigned to a workflow policy group.

You create groups for workflow policies in the Workflow Policies Groups view. The Workflow Policies Groups view has two applets:

- **Workflow Groups applet.** Allows you to create new policy groups and to view and select previously existing policy groups. Specifying a workflow policy group determines the monitoring cycle for a workflow policy. Each policy group should contain policies that need to be monitored within similar time intervals. See "About the Workflow Groups Applet" on page 179.
- **Policies applet.** Lists the workflow policies assigned to the selected group. See "About the Workflow Policies Applet" on page 179.

To create a policy group

1. From the application-level menu, choose Navigate > Site Map > Administration - Business Process > Policy Groups.
2. From the applet drop-down menu in the Policy Groups applet, select New Record and enter the name for the group.
3. (Optional) Enter comments in the Comments field.

About the Workflow Groups Applet

Table 42 describes the fields in the Workflow Groups applet.

Table 42. Workflow Groups Applet Fields

Field	Description
Name	The name of the workflow policy group. All workflow policies belong to one and only one policy group. This name can be no more than 30 characters long.
Comments	Any comments describing the purpose or use of the policy group.

About the Workflow Policies Applet

Table 43 describes the fields in the Policies applet of the Workflow Policies Groups view.

Table 43. Policies Applet Fields

Field	Description
Name	The name of the policy included in the workflow policy group.
Workflow object	The workflow policy object for which the policy was created, for example, Service Request, Opportunity, or Quote.
Activation Date/Time	The date and time that the policy was or will be activated. If this field is null, the policy is activated immediately.
Expiration Date/Time	The date and time that the policy expires. If this field is null, the policy is active indefinitely.
Comments	Any comments describing the purpose or use of the policy.

Creating Workflow Policies

After creating your workflow policy actions and workflow policy groups, you are ready to go to the Workflow Policies view to complete your workflow policy creation.

You create a new workflow policy after you create the policy action and the policy group.

The Workflow Policies view is made up of four applets:

- **Policies List applet.** Where you enter and view information about the workflow policy. The entry applet toggles with a list applet so that you can quickly move between working on an individual policy and viewing information about several policies or groups of policies. See "About the Policies List Applet" on page 182.

- **Conditions applet.** Where you define or change the conditions for the workflow policy. You can define as many conditions as necessary. All the conditions for the policy must be met to trigger the workflow policy action. If you want the policy to be triggered when one or another condition is true, you must create a separate workflow policy for each condition. See "About the Conditions Applet" on page 183.
- **Actions applet.** Where you enter the name of the previously defined workflow policy action you want to take place when the conditions of the workflow policy are met. See "About the Actions Applet" on page 187.
- **Arguments applet.** Where you can review the workflow policy action arguments.

To create a Workflow policy

1. From the application-level menu, choose Navigate > Site Map > Administration - Business Process > Policies.
2. In the Policies List applet, select New Record from the applet-level menu, create a policy name and fill in the other applet fields.
3. Fill in the fields in the Conditions applet.
4. Fill in the name of the action you created in the Workflow Policies Actions view and, if necessary, check the Consolidate field.

NOTE: Workflow policies can not be based on the table S_DOCK_TXN_LOG. The unique index for this table is TXN_ID, rather than ROW_ID for other standard Siebel tables. Because Workflow uses ROW_ID to do the comparison and execute actions, it will error out if used against S_DOCK_TXN_LOG.

NOTE: You cannot execute a Business Service from a Workflow Policy.

Examples of Workflow Policies

The following are examples of workflow policies for specific situations. You can use these examples as the basis for creating your own workflow policy actions.

- "Example of a Workflow Policy: Creating a Send Page Workflow Policy" on page 188
- "Example of a Workflow Policy: Creating a Send Email Workflow Policy" on page 188

NOTE: Workflow policies update the database fields directly through the Data Layer, and do not go through the Business Object Layer; therefore, any Workflow Processes that include Business Component events related to the updated fields are *not* executed.

Using Batch Mode with Workflow Policies

You can create Workflow policies as batch policies by checking the Batch check box in the Policies applet. When you start Workflow Monitor in batch mode, it checks for policies with the Batch check box marked. Each policy causes an SQL statement to be issued to identify the specific records that meet the policy conditions. The records identified are then processed in turn and the appropriate actions are carried out.

You can use the batch function to consolidate email messages for a designated recipient by checking the Batch field in the Actions applet in the Workflow Policies Policies view.

If you consolidate email messages, the recipient would receive one email with the information of multiple actions rather than multiple emails. For example, you can create a workflow policy that sends an email to the director of sales each time a quote is submitted with a discount over 30%. If 20 sales representatives submit quotes with the 30% discount and the Batch field is checked, the director of sales receives one email listing the 20 quotes. If the Batch field is not checked, the director of sales will receive 20 email messages.

NOTE: When creating a batch type workflow policy, the comparison operators IS ADDED, IS UPDATED, or IS DELETED must be used in conjunction with regular conditions. These comparison operators are considered special conditions intended for Dynamic mode when triggering rows to look up regular conditions.

About the Policies List Applet

Table 44 defines the Policies List applet fields.

Table 44. Policies Applet Fields

Field	Description
Policy Name	The name of the workflow policy.
Workflow Object	The workflow policy object for which the policy was created, for example, Service Request, Activity, or Accounts. This field is required.
Group	The workflow policy group to which the policy belongs. Each policy must be assigned to a workflow policy group. See the note following this table for information about changing workflow policy groups.
Comments	Any comments about the purpose or use of the policy.
Activation Date/Time	The date and time that the policy was or will be activated. If this field is null, the policy is activated immediately.
Expiration Date/Time	The date and time that the policy expires. If this field is null, the policy is active indefinitely.
Duration	The duration fields specify how long in days, hours, or minutes all conditions must be true for the workflow policy to be executed. This field is ignored if the policy is run in batch mode.
Created By	The login name of the person who created the workflow policy. The information in this field is automatically filled. Read-Only.
Created On	The date and time the workflow policy was created. The information in this field is supplied for you. Read-Only.
Quantity	The number of records that meet the policy conditions before the policy action executes. If you do not specify a quantity, Siebel Workflow assumes a quantity of 1. Quantity allows policy administrators to create conditions that are based on a number of records that meet the policy conditions. For example, an administrator may create a workflow policy that sends a message broadcast when 20 or more critical service requests are open.
Batch	When Batch is checked, this indicates that this policy should evaluate all records that potentially meet the conditions of the policy. The Workflow Monitor Agent scans all records using the conditions of the policy to find the matches. When this field is checked, run Workflow Monitor Agent with the Batch Mode flag set to TRUE. The default is unchecked.

NOTE: Before you can move a Workflow Policy from one group to another group, all requests associated with that Workflow Policy must complete. If a Workflow Policy's group is changed while associated requests are pending, the Workflow Monitor Agent will fail with the error Rule Not Found. If this occurs, restore the Workflow Policy to its original group, wait for the requests to complete, and then proceed with the change.

About the Conditions Applet

Table 45 defines the Conditions applet fields in the Workflow Policies Policies view.

Table 45. Workflow Policy Conditions Applet Fields

Field	Description
Field	The workflow policy component column in the workflow policy object on which the workflow policy condition is based, for example, service request priority or service request open date. Select the workflow policy column instance from the picklist for the field. This field is required.
Comparison	The comparison to make between a workflow policy agent's column value and the value you specify, for example, equals (=) or greater than (>). Select the comparison from the picklist for the field. This field is required. For more information, refer to "Using Comparison Values in the Conditions Applet."
Value	The value to compare to the workflow policy column value instance, for example, not started or very high. This field is required except when the Comparison field has a value of Is Null, Is Not Null, Is Updated, Is Deleted, or Is Added. For more information, refer to "Using Comparison Values in the Conditions Applet" and "Entering Date Calculations" on page 187.

Using Comparison Values in the Conditions Applet

You use comparison values in the Operation field. The field exposes the Workflow policy component column for monitoring.

Standard Comparisons

The Comparison field supports <, >, <>, >=, <=, =, LIKE, IN, NOT IN, BETWEEN, IS NULL, and IS NOT NULL operators. An 'AND' is implied between multiple conditions defined using these comparison values. 'AND' means that all conditions must be met before the action occurs.

When you specify values for the comparison operands LIKE, IN, NOT IN, and BETWEEN in the Value field of the Conditions applet of the Workflow Policies Policies view, it must be in a form that the underlying database expects. IN, NOT IN, and BETWEEN require you to enter the database specific format for the field being examined, for example, IN ('a', 'b', 'c') or IN (1, 2, 3, 4) and BETWEEN 'A' and 'M' or BETWEEN 1 and 10.

NOTE: It is up to the policy creator to make sure the syntax is correct. Siebel Business Process Designer only passes the BETWEEN clause to the database. It does not verify syntax, except for date and time. For date and time fields, Siebel Business Process Designer converts the date and time columns to the format of month/day/year, hour:minute:second.

LIKE and NOT LIKE allow you to use wildcards, for example, LIKE Smith% or NOT LIKE Sm%th%.

Table 46 shows comparison values for a typical database (your specific database syntax requirements may vary). Note that when using LIKE, IN, NOT IN, or BETWEEN with character fields, you use single quotes around the value. In addition, when using IN or NOT IN, you must place the value within parentheses.

Table 46. Comparisons for a Typical Database

Comparison	Value
<	5
>	5
<>	5
>=	5
<=	5
=	A
LIKE	Abc%
IN	(1, 2, 3)
NOT IN	('A', 'B', 'C')
BETWEEN	1 and 2
BETWEEN	'A' and 'B'

NOTE: On an MS SQL Server database, when you create a workflow policy condition on a LONG column, the available comparisons are IS NULL, IS NOT NULL, LIKE, and NOT LIKE.

Specialized Comparisons

The Comparison field also supports the specialized operators IS ADDED, IS UPDATED, and IS DELETED.

The following comparisons work at the workflow policy component level. They do not operate at the field level.

- **IS ADDED.** If a new row is added for this workflow policy component, then trigger this workflow policy to be examined.

  NOTE: If used in conjunction with standard comparisons, IS ADDED can be triggered when a record is updated.

- **IS DELETED.** If a row is deleted from this workflow policy component, then trigger this workflow policy to be examined.

The following comparison operates at the field level. To monitor if a field within the workflow policy component was modified, use the field that is named after the workflow policy component.

- **IS UPDATED.** If the field's value has changed, either by adding a new record with the specific field or by modifying the field in an existing record, then trigger this policy to be examined. To monitor if any field for a particular table was updated, use the workflow policy component column that represents the LAST_UPD column for that table.

The IS operators serve as a starting point for the examination of the workflow policy.

NOTE: When creating a batch type workflow policy, the comparison operators IS ADDED, IS UPDATED, or IS DELETED *must* be used in conjunction with regular conditions. These comparison operators are considered special conditions intended for Dynamic mode when triggering rows to look up regular conditions.

Table 47 describes the specialized comparisons for all database platforms that can be used in creating workflow conditions.

Table 47. Specialized Comparisons

Comparison	Value
IS ADDED	Use IS ADDED with a workflow policy component column specified in the Condition field and nothing specified in the Condition value. The condition is met when an instance of the workflow policy component is added. For example, if the Service Request policy component column is selected in the Condition field and IS ADDED is selected in the comparison, the condition will be met when you create a new service request.
IS UPDATED	Use IS UPDATED with any field specified in the Condition field and nothing specified in the Condition value. The condition is met when the field changes. For example, if service request status is specified in the Condition field and IS UPDATED is selected in the comparison, the condition is met when the Service Request status changes.
IS DELETED	Use IS DELETED when you specify a child workflow policy component in the Condition field, and nothing is specified in the Condition value. A child workflow policy component is a workflow policy component that is associated with a major entity in Siebel (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. If IS DELETED is used in conjunction with other conditions, the other conditions must be based on the parent workflow policy component. For example, you may want to notify a service request owner if an activity is deleted from a service request that has a sub-status of In Process. The policy would be based on the Service Request Workflow policy object. The first condition would be field = Activity Component, comparison = IS DELETED, value = blank. The second condition would be field = Service Request sub-status, Comparison = '=', value = In Process. The action is to send an email to the SR owner.

NOTE: 'OR' is implied between conditions defined using these specialized comparison values, where 'OR' means that one or more of the conditions must be met before the action occurs. An 'AND' is implied between conditions using standard comparisons and conditions using specialized comparisons.

For example, you may want a service representative to receive an email when an open service request has an activity added to it. You would then create a policy that has conditions Service Request Status = 'Open', Service Request Activity Component IS ADDED.

Entering Date Calculations

Workflow Monitor considers both date and time when evaluating Workflow Policy conditions that perform a date comparison. CURRENT can be used when entering a value for a date comparison. The format for using CURRENT is CURRENT +/- d:h:m where "d" is day, "h" is hours, and "m" is minutes. You can use CURRENT in the comparison value for date fields. You can also use CURRENT when you specify the activation and expiration dates for a message broadcast action.

About the Actions Applet

Table 48 defines the Actions applet fields on the Workflow Policies Policies view.

Table 48. Actions Applet Fields

Field	Description
Action	The name of the action.
Sequence	The sequence of the action relative to other actions. This field is required.
Contact Last Name	The last name of a contact when the recipient of the action is a contact in the database.
Contact First Name	The first name of a contact when the recipient of the action is a contact in the database.
Employee Login	The login name of an employee when the recipient of the action is an employee.
Position	The position of an employee when the recipient of the action is a position.
Relative	The relative type when the recipient of the action is determined by the workflow object, for example, service request owner.
Consolidate Flag	Consolidates the action to one instance if more than one record meets all the conditions of the workflow policy during the same action interval. Default is FALSE. The consolidate flag is unavailable with actions that send pages.

Many of the choices in the fields in the Workflow Policies Policies view are predefined in other Siebel Business Process Designer views, either in Siebel Client or in Siebel Tools. You can modify the predefined choices or create new choices for these fields. These choices show up as a picklist in the Workflow Policies Policies view applets.

Example of a Workflow Policy: Creating a Send Page Workflow Policy

In this situation, the support manager wants a page sent whenever a service request priority becomes Very High and no one has been assigned to the service request. The Send Page action has already been created; now you must create the workflow policy to implement the workflow policy action.

To create a Send Page Workflow policy

1. From the application-level menu, choose Navigate > Site Map > Administration - Business Process > Policies.
2. In the Policies List applet, select New Record from the applet-level menu.
3. Fill in the Policies applet as follows:
   a. Name: `Page support manager`
   b. Workflow Object: `Service Request`
   c. Policy Group: `High Frequency`
   d. Duration: `2 hours`
4. Fill in the Conditions applet:
   a. `Service Request Priority = Very High`
   b. `Service Request Owner IS NULL`
5. Fill in the Actions applet with the name of the appropriate Send Page action.

Example of a Workflow Policy: Creating a Send Email Workflow Policy

In this situation, the vice president of sales wants to be notified when the number of deals that are not closed reaches a designated level. In this case, you have already created an workflow policy action that batches information on the deals and sends an email message containing information on the number of deals you designated.

To create a Send Email Workflow policy

1. From the application-level menu, choose Navigate > Site Map > Administration - Business Process > Policies.
2. In the Policies List applet, select New Record from the applet-level menu.
3. Fill in the Policies applet as follows:
   a. Name: `Very High Value Opportunity`
   b. Workflow Object: `Opportunity`

c. Policy Group: `Medium Frequency`

d. Quantity: `5`

e. Comments: `Reminder to talk to the sales rep on opportunities nearing closure > $400,000, but low probability`

f. Batch: checked

NOTE: You do not need to fill in the Quantity field to have a repeating message.

4. Fill in the Conditions applet with the following:

Field	Comparison	Value
Forecast Probability	<=	50
Forecast Revenue	>	400000

5. Fill in the Actions applet.

   a. Enter the Action name with the appropriate workflow policy action:

   ```
   Excellent Quality Opportunity
   ```

   b. Check the Consolidate field.

The Send Message Arguments fills in automatically with the information from the defined workflow policy action.

NOTE: To make sure that the email action works properly, you can restart the Workflow Policy Monitor and Workflow Policy Action agents and set the Workflow Policy Action parameter Sleep to at least 5 minutes. This makes sure that your email lists all the opportunities meeting the conditions of the workflow policy. If the Workflow Policy Action agent runs too fast, then there will be an individual email each time the conditions of the workflow policy are met. Starting the server processes is discussed in "About Workflow Policies Server Administration" on page 217

About Customizing Workflow Policies with Siebel Tools

This section describes how to customize workflow policies with Siebel Tools.

"Siebel Tools and Workflow Policies" on page 190

"Siebel Tools Definitions in the Workflow Policies Views" on page 191

Siebel Tools and Workflow Policies

Using Siebel Tools, you can define new workflow policy objects and modify existing workflow policy objects to meet your business needs. A brief discussion of basic Siebel Tools concepts is provided here as they relate to Workflow Policies. For a complete discussion of Siebel Tools, see *Using Siebel Tools* and *Configuring Siebel eBusiness Applications*.

NOTE: When you use Siebel Tools to modify or create workflow policy objects on your local system, the changes are not available on the server until they are applied to the server.

Siebel Tools consists of an Object Explorer window and one or more Object List Editor windows. Figure 7 on page 45 shows the Object Explorer and Object List Editor window. The Object List Editor window lists object definitions for each object type and allows you to edit object type properties. The Object Explorer provides navigation between each group of object definitions of a particular object type.

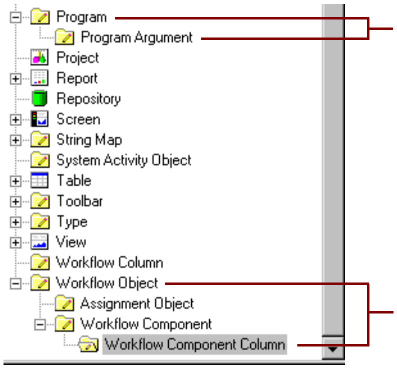
Object type is an entity that is displayed as a node on the Object Explorer. An object type is the template from which object definitions are created and have a predefined set of properties. Workflow policy programs, workflow policy columns, and workflow policy objects are all object types.

An *object definition* implements one piece of the software such as Service Request or Contact. This object definition consists of *properties*. Properties are characteristics of the software that the object definition implements. For example, the properties of workflow policy column (object type) Service Request Severity (object definition) include Name (Service Request Severity), Table Name, Picklist, and so on.

NOTE: Workflow policy objects are not included in the Object Explorer by default. Click View > Options > Object Explorer to add the workflow policy objects to the Object Explorer view.

Properties correspond to the columns in Object List Editor windows. The information entered under the columns is *values*. You can also use the Properties window to edit the properties of the currently selected object definition in an Object List Editor window by changing the values in the columns. You may change the property values in an object definition but not the set of properties to which values are assigned.

Object type definitions have a specific set of properties, as discussed above. They also have hierarchical relationships with other definitions called parent-child. The arrangement of folder icons in the Object Explorer is hierarchical (in the Types view). An object type (folder) beneath and slightly to the right of another is the child object type of the one it is below. The one above the *child object type* is the child's *parent object type*. Figure 11 shows parent-child relationships in the Object Explorer. A parent object type can have multiple child object types.



Parent-child relationship between Program and Program Argument

Parent-child relationship between Workflow Object, Workflow Component, and Workflow Component Column

Figure 11. Parent-Child Object Type Relationships in the Object Explorer

Siebel Workflow Policies accesses the following object types to create workflow policies, workflow actions, and workflow conditions:

- Workflow policy program
- Workflow policy program arguments
- Workflow policy column
- Workflow policy object
- Workflow policy component
- Workflow component column

Workflow policy programs and program arguments must be created and defined in Siebel Tools for use by workflow policies in the Workflow Policies Action view. Workflow policy objects, workflow policy components, workflow policy component columns, and workflow policy columns must be created and defined in Siebel Tools for use by workflow polices in the Workflow Policies Policies view.

Siebel Tools Definitions in the Workflow Policies Views

The Workflow objects that were defined in Siebel Tools are displayed in Workflow Policies in the Workflow Object picklist in the Workflow Policies view.

The workflow policy component columns defined in Siebel Tools are available to Workflow Policies views. The workflow policy component columns in Siebel Tools are exposed in Workflow Policies in the Condition Field picklist in the Workflow Policies view.

The policy programs defined in Siebel Tools are available in Workflow Policies views. The policy programs in Siebel Tools are exposed in Workflow Policies in the Program Field picklist in the Workflow Policy Actions view.

You use Siebel Tools to configure or create custom workflow policy objects and custom policy programs. The use of Siebel Tools is described in *Using Siebel Tools*. Only information specific to Workflow Policies is described in this document.

About Workflow Policy Objects

Although Siebel Tools includes many of the workflow policy components you need for workflow policy creation, you can reconfigure entities in Siebel Tools to meet the full range of your business needs.

Workflow policy objects provide the context in which Workflow Policies operate. The workflow policy object, through its workflow policy components, defines the set of tables and columns that can be monitored by a policy and how each table in the workflow policy object relates to the other tables. This collection of columns and the relationships between the tables of the workflow policy object represent the entity within Siebel Tools that you would like to monitor.

Workflow policy objects comprise:

- **Workflow policy components.** Defines the Siebel database tables that you can monitor. Workflow policy components define the relationships between the primary workflow policy component and all other policy components of a workflow policy object.
- **Workflow policy component columns.** Defines the columns in the Siebel database table that you can monitor. You expose these columns for monitoring when you define workflow policy conditions for a workflow policy.

Siebel Tools includes many of the workflow policy objects for common business needs such as Opportunity, Service Request, and Contact. You may find that you need to reconfigure existing workflow policy objects or create custom workflow policy objects to meet your specific business needs.

CAUTION: Do not try to monitor Enterprise Integration Manager (EIM) table columns. To recognize EIM tables, look for table names that begin with EIM_ or end with _IF.

Creating a Workflow Policy Object

Creating a workflow policy object consists of four main steps:

- Defining the workflow policy columns. See "Defining a Workflow Policy Column" on page 199.
- Defining the workflow policy components. See "Defining a Workflow Policy Component" on page 200.
- Defining the workflow policy object. See "Defining a Workflow Policy Object" on page 200.
- Associating the workflow policy column with the workflow policy component. See "Associating a Column with a Workflow Policy Component" on page 202.

About the Relationship Between Workflow Policy Components

Figure 12 shows the entity relationship diagram for four Service Request Workflow Policy components. The diagram shows each of the components, their relationship to one another, and which columns are of interest. Service Request is the primary workflow policy component, and the other three components are joined directly or indirectly to it.

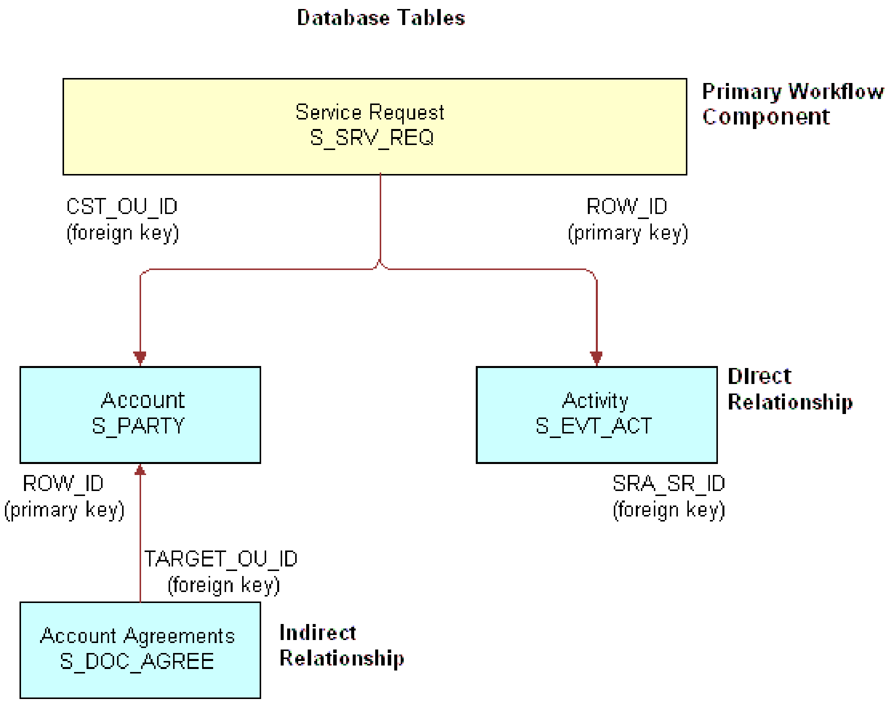


Figure 12. Relationships of Workflow Policy Objects and Workflow Policy Component

Relationship Between Workflow Policy Object and Workflow Policy Component

If the Service Request has the primary field checked, then it is the primary component. All the other components in the list are the nonprimary components of the Service Request workflow policy object.

Workflow Policies and the Siebel Tools Views

This section discusses the following Siebel Tools views:

- **Workflow Policy Column List view.** Displays a list of the available workflow policy columns. You must activate extension columns in Siebel Tools in order to make them available for use in workflow database operations. See "About the Workflow Policy Column List View" on page 194.
- **Workflow Policy Object List view.** Displays a list of the available workflow policy objects. See "About the Workflow Policy Object List View" on page 196.

- **Workflow Policy Component List view.** Displays a list of all workflow policy components for the selected workflow policy object. This view shows both the primary policy component and any nonprimary policy components and how each of the policy components is related. See "About the Workflow Policy Component List View" on page 196.
- **Workflow Policy Component Columns List view.** Displays a list of all the policy columns that can be monitored from the selected workflow policy component. See "About the Workflow Policy Component Columns View" on page 198.

About the Workflow Policy Column List View

Table 49 describes the fields in the Workflow Policy Columns List view.

Table 49. Workflow Policy Columns Applet Fields

Field	Description	Comments
Name	The name of the workflow policy column. This is the default name that appears in the Conditions applet on the Workflow Policies Policies view.	A descriptive name that is ■ Consistent with your overall naming strategy. ■ Meaningful to the policy maker. ■ Descriptive of how the column is used.
Changed	The identifier for whether the record was added or edited.	A check mark or a blank value.
Project	The project the workflow policy column belongs to. The project must be locked by you before you can modify the column.	A project from the picklist of projects you currently have checked out.
Table Name	The name of the Siebel database table that contains the column.	A table name from the picklist of all Siebel database tables.
Column Name	The name of the column in the Siebel table.	A database column on the database table specified in Table Name.
Picklist	This is the picklist that is used when selecting a comparison value for the column in the Workflow Policies Policies view.	A picklist defined in the repository. The column selected would have a corresponding Business Component field. If the corresponding Business Component field has a picklist defined, the picklist should be entered here. For more information on picklists, see *Configuring Siebel eBusiness Applications*.
Source Field	The field in the business component of the picklist that is the source of the comparison value.	A Business Component field name from the picklist specified in the Picklist field.

Table 49. Workflow Policy Columns Applet Fields

Field	Description	Comments
Applet	Pick applet used to display the picklist in the Workflow Policies view.	An applet chosen from the picklist. Only Pick applets should be selected.
Inactive	Determines if this column is active or inactive. If column is inactive, the column is not compiled when you compile your .srf and is not accessible by any object.	A check mark indicates this is inactive and is not compiled or accessible.
Comments	Comments describing the purpose or use of column.	Any text.

Configuring a Workflow Condition Based on a Foreign Key

You can configure a workflow condition that is based on a foreign key that exists in the primary table of the workflow object. For example, SOPTY.CURR_STG_ID, where S_OPTY is the primary table of the Opportunity workflow object, and CURR_STG_ID is a foreign key from S_STG.NAME.

To configure a workflow condition based on a foreign key existing on the primary table

1. Create a new Workflow column, S_STG.NAME.
2. Make sure CURR_STG_ID is added under the Opportunity workflow component.
3. Create a new workflow component in the Opportunity workflow object based on S_STG table:
   - Name = your choice
   - Source Table Name = S_STG
   - Source Column Name = ROW_ID
   - Target Component Name = Opportunity
   - Target Column Name = CURR_STG_ID
4. Add the new workflow column S_STG.NAME (from Step 1) to the new workflow component.

You can now create a workflow condition that is based on the new workflow column.

About the Workflow Policy Object List View

Table 50 describes the properties for a workflow policy object.

Table 50. Workflow Policy Object Properties Fields

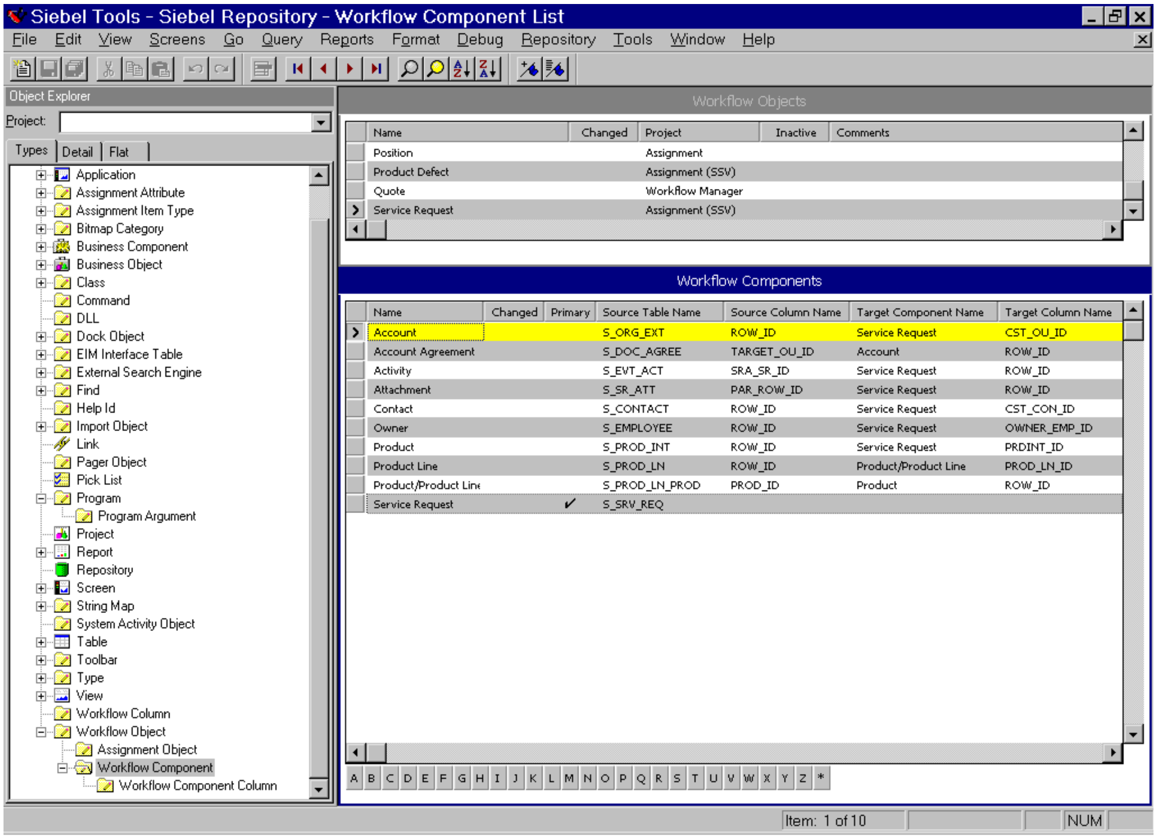
Field	Description	Comments
Name	The name of the workflow policy object.	A descriptive name that is: - Consistent with your overall naming strategy. - Meaningful to the policy maker.
Changed	The identifier for whether the record has been added or edited.	A check mark indicates the record has been added or edited.
Inactive	Indicates if the object is active or inactive.	A check mark indicates this field is inactive and will not be compiled or accessible. If object is inactive, the object is not compiled when you compile your .srf and is not accessible by any other object or policy.
Comments	Comments relating to the workflow policy object.	Descriptive text.
Project	The project name.	Defined in the project picklist.

About the Workflow Policy Component List View

A *workflow policy component* is a logical mapping of a database table. Figure 13 on page 197 shows the Workflow Policy Component list view. Except for the primary workflow policy component, each workflow policy component defines a relationship to another workflow policy component. This relationship is defined by specifying a source policy column and a target policy column. The source and target columns on a workflow policy component identify foreign key relationships between the tables.

A *primary workflow policy component* is a workflow policy component that all other workflow policy components are directly or indirectly related to. From these workflow policy components, the workflow policy columns that are available for monitoring in the workflow policy can be defined.

To define a workflow policy object and its components, you should be familiar with the Siebel Data Model and *Siebel Data Model Reference*. *Siebel Data Model Reference* describes the tables and how the tables are related.



Workflow Objects

Name	Changed	Project	Inactive	Comments
Position		Assignment		
Product Defect		Assignment (SSV)		
Quote		Workflow Manager		
Service Request		Assignment (SSV)		

Workflow Components

Name	Changed	Primary	Source Table Name	Source Column Name	Target Component Name	Target Column Name
Account			S_ORG_EXT	ROW_ID	Service Request	CST_OU_ID
Account Agreement			S_DOC_AGREE	TARGET_OU_ID	Account	ROW_ID
Activity			S_EVT_ACT	SRA_SR_ID	Service Request	ROW_ID
Attachment			S_SR_ATT	PAR_ROW_ID	Service Request	ROW_ID
Contact			S_CONTACT	ROW_ID	Service Request	CST_CON_ID
Owner			S_EMPLOYEE	ROW_ID	Service Request	OWNER_EMP_ID
Product			S_PROD_INT	ROW_ID	Service Request	PRDINT_ID
Product Line			S_PROD_LN	ROW_ID	Product/Product Line	PROD_LN_ID
Product/Product Line			S_PROD_LN_PROD	PROD_ID	Product	ROW_ID
Service Request		✓	S_SRV_REQ			

Figure 13. Workflow Policy Component List View

Table 51 describes all of the properties of a workflow policy component.

Table 51. Workflow Policy Component Properties Fields

Field	Description	Comments
Name	Name of the workflow policy component.	A descriptive name that is: ■ Consistent with your overall naming strategy. ■ Meaningful to the policy maker.
Changed	Indicates whether the record has been added or edited.	A check mark indicates the record has been added or edited.

Table 51. Workflow Policy Component Properties Fields

Field	Description	Comments
Primary	Indicates whether this workflow policy component is primary for the workflow policy object selected in the workflow Object applet.	A check mark indicates this is the primary workflow component. Note: Each workflow policy object must have only one primary workflow policy component.
Source Table Name	The table that the workflow policy component is based on.	A table name from the picklist.
Source Column Name	The column in the source table that relates to another workflow policy component.	A picklist of columns from the table specified in the Source Table Name field. (Not required for the primary workflow policy component.)
Target Component Name	The target workflow policy component that this workflow policy component is related to.	A table name from the picklist. (Not required for the primary workflow policy component.)
Target Column Name	The column in the target workflow policy component that the source column in this workflow policy component is joined to.	A picklist of columns from the workflow policy component specified in the Target Component Name field. (Not required for the primary workflow policy component.)
Inactive	Indicates if the component is active or inactive.	A check mark indicates this field is inactive and is not compiled or accessible. If the component is inactive, it is not compiled when you compile your .srf and is not accessible by any policy.
Comments	Any comments for the workflow policy component.	Descriptive text.

About the Workflow Policy Component Columns View

This view displays a list of columns that can be monitored from the selected workflow policy component. To navigate to workflow policy component columns, choose Workflow Policy Object > Workflow Policy Component > Workflow Policy component column. The Workflow Policy Component Column view lists all the columns available for monitoring.

Table 52 describes the workflow policy component columns values.

Table 52. Workflow Policy Component Columns Properties Fields

Field	Description	Comment
Workflow Column Name	The name of the column defined in the workflow Policy Component Column view.	A picklist of all columns that were defined in the Workflow Policy Column view for the table that the workflow policy component is based on.
Alias	The name of the column as it appears in the Conditions Field picklist in the Workflow Policies Policies view.	The default is the workflow policy column name. ■ A descriptive name that is: ■ Consistent with your overall naming strategy. ■ Meaningful to the policy maker. ■ Descriptive of how the column is used.
Changed	Indicates whether the record was added or edited.	A check mark indicates that the record has changed.

Defining a Workflow Policy Column

Before you can add a workflow policy column to a workflow policy component, you must define the workflow policy column in the Workflow Policy Column List view.

The procedure has two basic parts. First, you need to identify the business object, business component, and applet that will use the new workflow policy column. Second, you create the new workflow policy column.

To add a new policy column

1. Start Siebel with /x.
2. In Siebel Client, navigate to the view that will use the new policy column, for example, the Account > Activities view.
3. Pull down Help > About View.

   A dialog box appears identifying the business object, business components, and applets the view uses. Make note of this information.
4. In Siebel Tools, select the Business Component identified in step 3 from the Object Explorer and scroll to the Table field.

   This Table field identifies the Siebel database table that this business component represents. Make note of this information.
5. In Siebel Tools, select Workflow Policy Column from the Object Explorer.

6 Navigate to Edit >New Record.

7 Fill in the rest of the fields in the Workflow Policy Columns view with the values you found in previous steps.

   NOTE: The table name/column name combination must be unique. You are not allowed to add a record if your table name/column name combination has already been defined in another record.

Defining a Workflow Policy Component

To define a workflow policy component

1 In Siebel Tools, select Workflow Policy Object, then select Account. Then, expand the Object Explorer field Workflow Policy Object to Workflow Policy Component.

2 Create a new record and enter a name for the new policy component.

3 Enter the source table name for the policy component.

4 Set the source policy column name.

   This determines the relationship between this policy component and the primary policy component.

5 Determine the relationship between this policy component and the primary policy component.

Next, you need to identify the set of columns from this workflow policy component that you would like to monitor. To do this in Siebel Tools, navigate to Workflow Policy Object > Workflow Policy Component > Workflow Policy Component Column and Workflow Policy Column view. Here you need to identify the column in the predefined workflow policy columns that has an activity assigned to it but is not currently exposed in the Workflow Policy Component Column view.

Defining a Workflow Policy Object

A workflow policy object is defined by its parent-child relationship to workflow policy components and workflow policy component columns. A *workflow policy object* is a collection of workflow policy components. Each workflow policy object has one and only one primary workflow policy component. All the other workflow policy components of a workflow policy object are related to the primary workflow policy component, either directly or indirectly. A workflow policy component defines a database table that includes those columns you would like to monitor. Workflow policy component relationships are based on their corresponding table relationships. A workflow policy component column is the specific column that is available for monitoring.

Each of these workflow policy components can expose any number of workflow policy component columns. In the Siebel Tools Object Explorer, a workflow policy component column is a child object of a workflow policy component, which is a child object of a workflow policy object.

Follow these steps when you need to create a new workflow policy object.

To define a new workflow policy object

1. Open Siebel Tools and go to the Workflow Policy Object view.
2. Select the Workflow Policy Object applet.
3. Navigate to Edit >New Record and a row for a new record appears. Add the information for the new record in the appropriate fields.
4. Select the Workflow Policy Components applet. Add your primary workflow policy component and designate it as primary in the Primary field.

   CAUTION: You can have one, and only one, primary workflow policy component.

5. Add more workflow policy components and correctly define relationships to your primary workflow policy component.
6. Select the Workflow Component Columns applet. Add your workflow policy component columns for each of your workflow policy components.

Modifying Policy Column Names

Each business uses specialized terminology that clearly defines conditions within that organization. You can easily change the names of columns using the Alias column in the Workflow Component Column applet.

NOTE: The fields that appear in the Conditions applet in the Workflow Policies view are called workflow policy component columns in Siebel Tools. The Column Instances available in the picklist in the Workflow Policies view are names from the Alias field in the Workflow Component Column applet.

To change a policy column name

1. Open Siebel Tools and go to the Workflow Component Column view.
2. Select the Alias field in the Workflow Component Column applet for the condition you want to change.
3. Type in the new name.

Adding Policy Columns to a Workflow Policy Object

If you are creating a new workflow policy object or if you want to add new columns to an existing workflow policy object, you must first verify that the column is available in the Workflow Policy Columns applet in the Workflow Policy Column view. If the column is not listed, you need to add the column in the Workflow Policy Column view before you perform the following steps.

To add a column to a workflow policy object

1. Open Siebel Tools and select the Types tab in Object Explorer.

2 From Object Explorer, select the workflow policy object for the new column and then the workflow policy component under that object.

3 In the Workflow Policy Component Column applet, navigate to Edit > Add Record and add the information for the new record in the appropriate fields.

   You should see a list of workflow policy columns that were defined for the database table that the workflow policy component is based on.

Associating a Column with a Workflow Policy Component

To associate a column with a workflow policy component

1 In Siebel Tools, from Object Explorer navigate to Workflow Policy Object and select Account. Then navigate to Workflow Policy Component > Workflow Policy Component Column, with Activities selected in Workflow Policy Component.

2 Create a new record in the Workflow Policy Component Column applet.

3 In the Workflow Policy Column Name field, click on the picklist to see the current set of columns available from this workflow policy component's database table.

4 Select each workflow policy column you would like to monitor.

5 Change Display Name to match your business needs as appropriate.

   Note that if the workflow policy component column that you would like to monitor is not in the list, you must first define it under the Workflow Policy Column Explorer view.

About the Validate Tool in Siebel Tools

Siebel Tools provides a Validate Tool that allows you to check for high-level errors in new workflow policy objects or columns. To bring up the menu with Validate, right-click your mouse.

Selecting Validate brings up the Validate screen. Clicking the Start button runs the validation process and returns information either as a caution or as error messages that appear in the Details box.

Modifying an Existing Workflow Policy Object

When defining the types of workflow policies you need to operate your business, you may find that the predefined workflow policy objects do not contain the policy components you need. Use the procedural steps in this section as a general guideline for modifying a workflow policy object.

Before modifying a workflow policy object:

- Find out the name of its database table and column names. If you are going to add or modify a component, you need to know the relationship between the component and the primary workflow policy component.
- Make sure you do not have other records referencing this object that may be affected by your change. For example, before inactivating a component column, verify that no policy conditions are referencing the component column.

To determine a database table

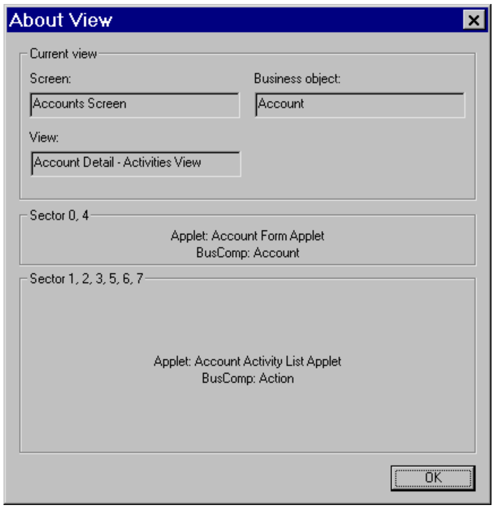
1. Start Siebel by entering the following from the command line:

   ```
   C:\Siebel\bin\siebel.exe /x
   ```

2. In Siebel Client, navigate to the appropriate workflow policy object view. This is the view that contains the business data you want to monitor.

   For example, if you need to modify the workflow for an account activity, you would navigate to the Account > Activities view.

3. Choose Help > About View.



Screen: Accounts Screen

Business object: Account

View: Account Detail - Activities View

Sector 0, 4

Applet: Account Form Applet
BusComp: Account

Sector 1, 2, 3, 5, 6, 7

Applet: Account Activity List Applet
BusComp: Action

About View identifies the Business object, Business components, and applets this view uses.

In the case of the Account Activities view, the dialog box identifies Action as the business component used by the Activities applet.

4. In Siebel Tools, select Business Component in the Object Explorer and find the appropriate component name.
5. Select the component (for example, Action) and find the table name. In the illustration above, the table name is S_EVT_ACT.

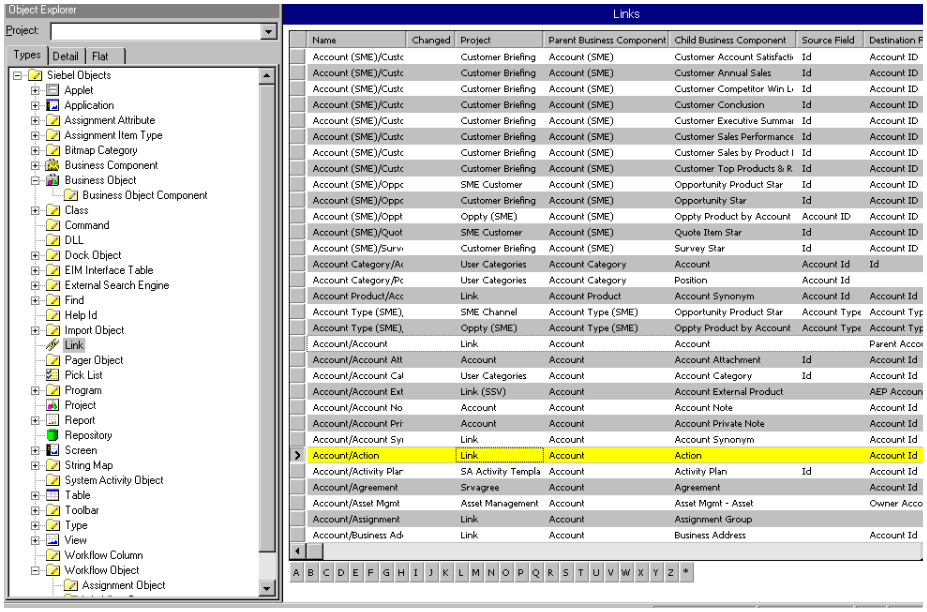
   You use this table name when you create a workflow policy component.

To determine the relationship between a component and the primary workflow policy component

1. In Siebel Client, navigate to the appropriate workflow policy object view.
2. Choose Help > About View.
3. Find the business component for the appropriate applet and the business object this view uses.
4. In Siebel Tools, select Business Object in the Object Explorer and search for the business component object name you noted in Step 3 on page 204.
5. In Object Explorer, expand Business Object to Business Object Component and select the appropriate Business Object Component.

   The attribute in the Link field identifies the link defining the relationship between the account and action business components.

6. In Object Explorer, select Link and find the applet/object link. The illustration below shows the Account/Action Link selected.



Links

Name	Changed	Project	Parent Business Component	Child Business Component	Source Field	Destination F
Account (SME)/Custc		Customer Briefing	Account (SME)	Customer Account Satisfacti	Id	Account ID
Account (SME)/Custc		Customer Briefing	Account (SME)	Customer Annual Sales	Id	Account ID
Account (SME)/Custc		Customer Briefing	Account (SME)	Customer Competitor Win L	Id	Account ID
Account (SME)/Custc		Customer Briefing	Account (SME)	Customer Conclusion	Id	Account ID
Account (SME)/Custc		Customer Briefing	Account (SME)	Customer Executive Summa	Id	Account ID
Account (SME)/Custc		Customer Briefing	Account (SME)	Customer Sales Performance	Id	Account ID
Account (SME)/Custc		Customer Briefing	Account (SME)	Customer Sales by Product I	Id	Account ID
Account (SME)/Custc		Customer Briefing	Account (SME)	Customer Top Products & R	Id	Account ID
Account (SME)/Oppc		SME Customer	Account (SME)	Opportunity Product Star	Id	Account ID
Account (SME)/Oppc		Customer Briefing	Account (SME)	Opportunity Star	Id	Account ID
Account (SME)/Oppt		Oppty (SME)	Account (SME)	Oppty Product by Account	Account ID	Account ID
Account (SME)/Quot		SME Customer	Account (SME)	Quote Item Star	Id	Account ID
Account (SME)/Surv		Customer Briefing	Account (SME)	Survey Star	Id	Account ID
Account Category/Ac		User Categories	Account Category	Account	Account Id	Id
Account Category/Pc		User Categories	Account Category	Position	Account Id	
Account Product/Acc		Link	Account Product	Account Synonym	Account Id	Account Id
Account Type (SME),		SME Channel	Account Type (SME)	Opportunity Product Star	Account Type	Account Typ
Account Type (SME),		Oppty (SME)	Account Type (SME)	Oppty Product by Account	Account Type	Account Typ
Account/Account		Link	Account	Account		Parent Accou
Account/Account Att		Account	Account	Account Attachment	Id	Account Id
Account/Account Cal		User Categories	Account	Account Category	Id	Account Id
Account/Account Ext		Link (SSV)	Account	Account External Product		AEP Accoun
Account/Account No		Account	Account	Account Note		Account Id
Account/Account Pri		Account	Account	Account Private Note		Account Id
Account/Account Syı		Link	Account	Account Synonym		Account Id
Account/Action		Link	Account	Action		Account Id
Account/Activity Plar		SA Activity Templa	Account	Activity Plan	Id	Account Id
Account/Agreement		Srvagree	Account	Agreement		Account Id
Account/Asset Mgmt		Asset Management	Account	Asset Mgmt - Asset		Owner Acco
Account/Assignment		Link	Account	Assignment Group		
Account/Business Ad		Link	Account	Business Address		Account Id

   This Link defines the relationship between the parent Business Component and the child Business Component through the Source field and Destination field.

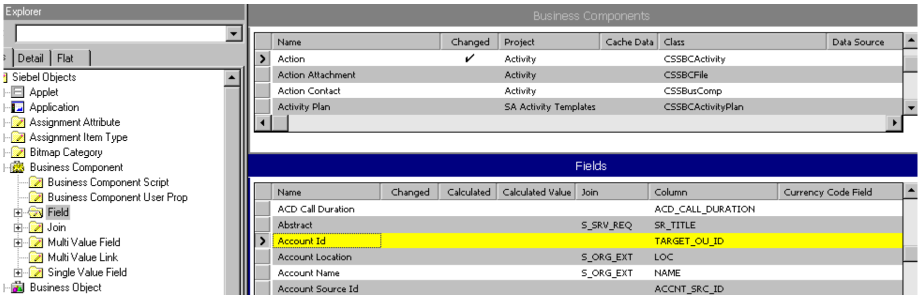
   A blank Source field indicates that the join is using the ROW_ID column of the parent business component.

   The Destination field is the field within the child business component that is a foreign key to the Business Component.

7. In Object Explorer, select Business Component, then select the appropriate component name.

8 Expand the Business Component to Field. Select the appropriate field and find the Column attribute.

In the illustration below, Account ID is the field and TARGET_OU_ID is the Column attribute.



Business Components

Name	Changed	Project	Cache Data	Class	Data Source
Action	✓	Activity		CSSBCActivity	
Action Attachment		Activity		CSSBCFile	
Action Contact		Activity		CSSBusComp	
Activity Plan		SA Activity Templates		CSSBCActivityPlan	

Fields

Name	Changed	Calculated	Calculated Value	Join	Column	Currency Code Field
ACD Call Duration					ACD_CALL_DURATION	
Abstract				S_SRV_REQ	SR_TITLE	
Account Id					TARGET_OU_ID	
Account Location				S_ORG_EXT	LOC	
Account Name				S_ORG_EXT	NAME	
Account Source Id					ACCNT_SRC_ID	

The column indicates which column within the table this field represents. You use this information when you define the workflow policy component.

About Workflow Policy Programs

Workflow policies use workflow policy actions based on the workflow policy programs that are predefined in Siebel Tools. (See "Predefined Programs" on page 247 for a complete list.) To meet your business needs, you can also reconfigure workflow policy programs to create new types of workflow policy actions.

A workflow policy program is a generic event that actions are based on. A workflow policy program defines the particular action that takes place when the conditions of a workflow policy are met.

There are five types of programs in the Siebel application:

- **Send Message.** For more information, see "Send Message Program Arguments" on page 209.
- **Send Page.** For more information, see "Send Page Program Arguments" on page 209.
- **External Program.** For more information, see "Run External Program Arguments" on page 210.
- **Send Message Broadcast.**
- **Database Operation.**

About the Program List View

Table 53 describes the Program properties fields.

Table 53. Program Properties

Property	Required	Description
Name	Yes	The name of the action to perform. This name is exposed in the Actions view in the Siebel client.
Changed	No	Indicates recent modifications.
Project	No	Name of the project as defined in the project picklist.
Type	Yes	Select one of the following types from the picklist: ■ **DB Operation.** Insert or update a database table based on arguments. ■ **External Program.** Execute an external program in Windows. ■ **Send Message.** Compose and send an automatic email message. ■ **Send Page.** Send a page to a pager. ■ **Send Message Broadcast.** Send a message broadcast to a group of users.
Workflow object	No	Limits use of this program to policies associated with this workflow policy object.
Inactive	No	Checked if program is not active.
Comments	No	Text to describe the program.

About the Workflow Policy Program Argument List View

Workflow policy program arguments define recipients, database actions, and available substitutions. Each workflow policy program typically has several program arguments. The argument fields that display in this view depend on the type of workflow policy program you select. A workflow policy program argument is a child process of a workflow policy program.

Table 54 shows the Workflow Policy Program Argument properties values.

Table 54. Workflow Policy Program Argument Common Properties

Property	Required	Description
Applet	Optional	Picklist applet.
Default Value	Optional	Text value of a type that depends on the Name of the program argument—an SQL statement, the text of a message, the email address of a recipient, and so on. Maximum length is 2000 characters.
Name	Required	Identifies the parameter from a predefined list.
Picklist	Optional	Picklist object.
Required	Boolean	Value is TRUE or FALSE. Indicates whether or not data entry is required.
Source Field	Optional	Picklist Source field.
Visible	Boolean	Value is TRUE or FALSE. Indicates whether the data supplied by this argument is displayed.
Inactive		Checked if program is not active.

When setting a Default Value for time/date fields, use the following formats:

- Date Column format: 2001-03-16
- Time Column format: 19:26:26
- Date Time Column format: 2001-04-05 21:25:00

Common Workflow Policy Program Argument Values

You can add functionality to workflow policy programs by creating a new workflow policy program argument record. Workflow policy program arguments determine how the workflow policy program behaves, including what substitutions are available for a workflow policy program and how the recipients are defined. Valid values for workflow policy program arguments common to all workflow policy programs are shown in Table 55.

Table 55. Valid Database Operations Workflow Policy Program Argument Values

Name	Description	Allowable Default Value
Primary ID	The row ID of the violating row that the Workflow policy program is acting on.	Should be empty.
Primary Table	The base table to which the action is applied. The base table can be unrelated to the record of the primary ID. Examples include: The violating row is in a child table and you now want to insert or update a record in the parent table. Tables can also be updated that are not related to the primary ID table. For example, create a Message Broadcast record when a certain monitored condition in the Opportunity record is true.	Any of the tables defined within the Siebel business object repository (as compared to the workflow business object). Workflow business objects are used for monitoring conditions but are not used in the coding of action programs.
Update Row ID	The row ID of a table other than the primary table of the workflow policy object. You can associate a workflow policy action with a workflow policy that updates any table. This value is used only when the Operation Type is set to update.	The row ID you want to update.
Operation Type	What operation to perform—update or insert.	Two possible values for DB Operation: Update or Insert.
Field Name	Name of the column in the base table to which the operation is performed. This is one of two field column pairs.	Allowable values: Text, Variable, Function.
New Row ID	For insert operations, this argument is automatically populated with the row ID of the row about to be inserted.	Should be empty.
Field Name (Column)	Name must be identical to the Field Name of the first column pair and (Column) appended to the name. This is the second of two Column Pairs.	Actual field name in the base table. The value can not contain any leading spaces.

Table 55. Valid Database Operations Workflow Policy Program Argument Values

Name	Description	Allowable Default Value
Sql Statement	Selects additional data from the database to be used as substitutions when the action is performed.	Valid SQL query statement for the RDBMS used (that is, Oracle, MS SQL, Informix, or Sybase).
Sql Statement Inputs	Name of the column in the base table on which the operation will be performed.	
Sql Statement Outputs	A placeholder for the values selected in the SQL Statement argument.	Variable Name.

When you run a database operation with Insert as the Operation Type, you can select a Default Value—New Row ID—as described previously, which provides the value for the ROW_Id field for the row being inserted.

Send Message Program Arguments

Table 56 describes program arguments specific to the Send Message program.

Table 56. Send Message Program Argument Properties

Name	Description	Value
Email Message	Body of the email message.	Any text with available substitutions.
Email Message Repeated	Text that is repeated when the Consolidate feature is used.	Any text with available substitutions.
Email Subject	Text in subject line of the email message.	Any text.
Send to Contact	All contacts available in Siebel.	
Send to Position	List of the positions available in Siebel.	
Send to Employee	List of all employees available in Siebel.	

Send Page Program Arguments

Table 57 describes the Program Arguments particular to the Send Page program.

Table 57. Send Page Program Argument Properties

Name	Description	Value
Send to Contact	All contacts available in Siebel.	Picklist of contacts.
Send to Employee	List of all employees available in Siebel.	Picklist of employees.
Send to Position	List of the positions available in Siebel.	Picklist of positions.

Table 57. Send Page Program Argument Properties

Name	Description	Value
Send to Relative	Send to an individual or group of individuals related to the Workflow object.	
Alpha Numeric Page Message	Body of the text message.	Any text with available substitutions.
Numeric Page Message	Body of the numeric message.	Any text with available substitutions.

Run External Program Arguments

Table 58 describes the Program Arguments particular to the Run External Program program.

Table 58. Run External Program Argument Properties

Name	Description	Value
Command Line	What parameters to pass to the executable.	
Executable Name	Full path to the executable to execute.	
Executable Type	The mode in which the Workflow Action Agent will execute the external program.	Wait. No wait.

Creating a Workflow Policy Program

A workflow policy program is a generic event that actions are based on. You define a program by defining the workflow policy event.

CAUTION: Do not rename or change the name of an existing workflow policy program. If you do so, you will lose all the actions created for the program.

When creating a workflow policy program that inserts new records, you must determine and provide the minimum field values that constitute a valid record as defined in the repository for the table:

- Provide values for all required columns. If a default value is defined for a column, that default value is used on the insert if the program specifies none. For example, S_EVT_ACT has two required columns: NAME and ROW_STATUS. ROW_STATUS defaults to Y so you do not have to set a value in the program (although you can).
- You do not need to provide a value for system-generated columns such as CREATED, CREATED BY, LAST_UPD, LAST_UPD_BY, ROW_Id, MODIFICTION_NUM, CONFLICT_Id.

For more information, see *Siebel Data Model Reference*.

Siebel Systems recommends that when you want to define a new workflow policy program that you copy an existing program that is similar to what you need and then modify the copy to suit your specific business needs. The advantage to using this method is that if something goes wrong with your customized program, you can always start over with the original existing program. Additionally, modifying a copy of an existing program is less error–prone that creating an entirely new program.

CAUTION: Thoroughly test any SQL queries that you plan to use with customized policy programs. Be aware that if the SQL statement fails to find rows, the workflow policies action is unable to process any tokens.

To create a workflow policy program

1. In Siebel Tools, choose Program.
2. Select an existing program that is similar to what you need for the new workflow policy program.
3. Click the right mouse button and choose Copy Record. This copies the entire program including the program arguments.
4. Modify the appropriate fields, such as Workflow Object, to meet the needs of the new program.
5. Define the program arguments.

   Enter the arguments carefully to make sure capitalization, punctuation, spelling, and so on are correct:

   - Type the entries in the Name column *exactly* as indicated in Table 55 on page 208. Primary ID, Primary Table, Operation Type, SQL Statement, and SQL Statement Outputs must have one space between each word and each word must be properly capitalized. For example, Primary ID must have one space between the two words, capital P, and lowercase d.

     NOTE: In program arguments, the carriage return character that exists in SQL Statement and SQL Statement Outputs can cause unexpected behavior for a workflow policy program. In most cases, the substitution value is not substituted with the intended value but is instead substituted with the [Label] literally. Avoid using the carriage return character.

   - When using SQL statements in program arguments, make sure that the statements are specific to the particular RDBMS you are using.
   - Type the names of the column pairs exactly: One space between each word, identically capitalized, one space in front of the left parenthesis and no spaces in the (Column).

   The order of the rows is not important.

   NOTE: Before using a program and its related program arguments in a workflow policy, you must delete any inactive or incomplete program argument definitions. These could cause Workflow Monitor Agent errors.

Example of Creating a Workflow Policy Program Argument: Send Opportunity Email

The following is an example of adding a new a workflow policy program argument, in this case, Send Opportunity Email. The current recipients of type relative are limited to the Primary Sales representative. You want to add a relative for Primary Contact. This allows policy makers to create an action that sends an email to the Primary Contact for an opportunity.

To add an alternative Send to Relative to the Send Opportunity Email program

1. In Siebel Tools, choose Workflow Policy Program > Send Opportunity Email > Workflow Policy Program applet.

   NOTE: To create a new workflow policy program argument for Send Opportunity Email, check the existing program arguments and make sure that the Send to Relative program argument exists.

2. Create a new record, `Primary Contact`.

   NOTE: When creating new program arguments, they cannot have the same name as a SQL Statement Output or the Workflow Monitor Agent server task will hang with the message "Examining request for policy..." when inserting a record.

3. Bring down the box under Default Value and create your SQL statement. For example:

   ```
   select O.PR_CON_ID, 'Send to Contact'
   from &Table_Owner.S_OPTY O
   where O.ROW_ID=?
   ```

   Workflow passes the ROW_ID of the violating row, so make sure to write all your SQL queries to use the same ROW_ID. In this example, the WHERE clause is written to use the ROW_ID of the opportunity row that violates the policy.

   NOTE: SQL statements are database vendor–specific. Use an external SQL tool to build and test your statements. When the test works, copy the statement into the field.

4. Select Workflow Relative Type Picklist in the PickList field.

   This picklist describes this argument as a relative.

   The Visible field is checked. The changed field becomes checked when you create a new program argument.

Creating SQL Statements for Workflow Policies Program Arguments

Before you begin to create the Recipient Type, Send to Relative, you must create a SQL statement in the Workflow Policy Program applet in Siebel Tools.

SQL statements written for Workflow Policies Program Arguments must have the following characteristics:

- The table name and column name you reference must be uppercase.
- The case-sensitive table name should be prefixed with:

  `&Table_Owner.`

- The SQL statement must be valid for the specific database vendor being used.

About Predefined Workflow Policy Programs

You can use the following examples as explanation for how to interpret the program arguments. Review these examples to understand the format of a program.

The examples use the predefined workflow policy programs included with Workflow Policies (see "Predefined Programs" on page 247). These programs can be viewed in the Siebel Tools Object Explorer by clicking Program and then Program Argument.

- "Example of Using a Predefined Workflow Policy Program: Change SR Close Date to Today" on page 213
- "Example of Using a Predefined Workflow Policy Program: Change SR Owner" on page 214
- "Example of Using a Predefined Workflow Policy Program: Change SR Owner to Manager" on page 215
- "Example of Using a Predefined Workflow Policy Program: Send Quote Page" on page 216

Example of Using a Predefined Workflow Policy Program: Change SR Close Date to Today

Using this program, you can define a policy such that if a Service Request has an activity of type Resolution, and the SR is open for more than five days, the SR close date is changed to today's date.

When the policy triggers the workflow policy program, the program enters the current system date into the Close Date field of the Service Request record. Table 59 shows the arguments for the Change SR Close Date to Today program.

Table 59. Change SR Close Date to Today Program Arguments

Argument Name	Comment
Primary ID	Contains the row ID of the Service Request record meeting the policy condition.
Primary Table Operation Type	Specifies the table (S_SRV_REQ) and what action is to take place (Update).

Table 59. Change SR Close Date to Today Program Arguments

Argument Name	Comment
Sql Statement	`select {fn now()} from &Table_Owner.S_DUAL` This statement calls the Siebel function now() to obtain the current date and uses the table &Table_Owner.S_DUAL to hold the value temporarily. The S_DUAL table is used to hold temporary values. Math functions can also be performed. For example, SQL Statement = select {fn now()}+7 from &Table_Owner.S_DUAL returns the current date plus seven days. Different RDBMS have different formats for the same function (for example, in MS SQL, the function GetDate() is used to return the current date).
Sql Statement Outputs	The "Today" variable obtains its value from the SQL statement.
New Close Date (Column)	Specifies the column in the record to be updated (ACT_CLOSE_DT).
New Close Date	Specifies the field to be updated to the value of Today.
Update Row ID	The row ID of the record you want to update. (The same as the value of the Primary ID.)

Example of Using a Predefined Workflow Policy Program: Change SR Owner

If an open service request is not assigned for a certain length of time, this workflow policy program could be used to assign (change owner) a service request to the expert in the area of the specific service request. This would allow the right people to see the incoming service request and assign it appropriately.

This workflow policy program allows you to select a new owner from a picklist and put it into the field of the Service Request record matching the policy condition. Table 60 shows the arguments for the Change SR Owner program.

Table 60. Change SR Owner Program Arguments

Argument Name	Comment
Primary ID	Contains the row ID of the Service Request record meeting the policy condition.
Primary Table Operation Type	Specifies the table (S_SRV_REQ) and what action is to take place (Update).
New Owner (Column)	Specifies the field in the record to be updated (Owner_EMP_ID).

Table 60. Change SR Owner Program Arguments

Argument Name	Comment
New Owner	Indicates that a picklist is to be displayed for assigning a new owner. The picklist is defined by columns picklist = Picklist SR Owner, Source = ID, and Applet = SR Owner Pick Applet.
Visible	When checked, indicates the picklist is visible to the user.

Example of Using a Predefined Workflow Policy Program: Change SR Owner to Manager

If the service request is not closed within a specific duration of time, assign the service request to the owner's manager. This would allow a proper response time to service calls.

This workflow policy program does the following:

- Uses the Primary ID as input into a SQL statement
- Uses a query SQL statement to retrieve the current value of the field Manager
- Sets the New Owner field to default to the current value of Manager
- Allows the end user to update the New Owner field optionally through a picklist

Table 61 shows the arguments for the Change SR Owner to Manager program.

Table 61. Change SR Owner to Manager Program Arguments

Argument Name	Comment
Primary ID	Contains the row ID of the Service Request record meeting the policy condition.
Primary Table Operation Type	Specifies the table (S_SRV_REQ) and what action is to take place (Update).
New Owner (Column)	Specifies the field in the record to be updated (Owner_EM_ID).
New Owner	Indicates that a picklist is to be displayed for assigning a new owner.

Table 61. Change SR Owner to Manager Program Arguments

Argument Name	Comment
Sql Statement	SELECT MGRPOS.PR_EMP_ID FROM &TABLE_OWNER.S_POSTN POS, &TABLE_OWNER.S_EMPLOYEE EMP, &TABLE_OWNER.S_POSTN MGRPOS, &TABLE_OWNER.S_SRV_REQ SR WHERE SR.ROW_ID = ? AND SR.OWNER_EMP_ID = EMP.ROW_ID AND EMP.PR_POSTN_ID = POS.ROW_ID AND POS.PAR_POSTN_ID = MGRPOS.ROW_ID SR.ROW_ID = ? uses a question mark as a placeholder for inputting the value of the Primary ID. The system knows to substitute the Primary ID for the question mark. This SQL statement joins four tables, giving access to data from all of them. In this example, only one field is retrieved. Policy Monitor requires the definitions contained in the workflow policy object, workflow policy components, and workflow policy columns. In working and coding workflow policy action programs using the Siebel tables, explicit joining of the base table through SQL code is required.
Sql Statement Inputs	Set to the value of Primary ID.
Sql Statement Outputs	Set to the value of Manager.

Example of Using a Predefined Workflow Policy Program: Send Quote Page

If a created quote has a value less than some percentage of the opportunity’s revenue (very highly discounted), send a page to a designated employee.

This workflow policy program sends out a pager message. The SQL statement is configured for the different RDBMS syntax.

There are four SQL statements, one default and three specific to an RDBMS (Informix, Oracle, and SQL Anywhere).

The default SQL Statement query retrieves five values from four tables using an outer join specified by *=:

```
select
q.QUOTE_NUM, q.REV_NUM, o.NAME, a.NAME, a.LOC
from
&Table_Owner.S_DOC_QUOTE q, &Table_Owner.S_ORG_EXT a, &Table_Owner.S_OPTY o
where
q.ROW_ID = ? and q.OPTY_ID *= o.ROW_ID and q.TARGET_OU_ID *= a.ROW_ID
```

The SQL statement (Oracle) query retrieves five values from four tables using an outer join specified by the (+):

```
select
q.QUOTE_NUM, q.REV_NUM, o.NAME, a.NAME, a.LOC
from
&Table_Owner.S_DOC_QUOTE q, &Table_Owner.S_ORG_EXT a, &Table_Owner.S_OPTY o
where
q.ROW_ID = ? and q.OPTY_ID = o.ROW_ID (+) and q.TARGET_OU_ID = a.ROW_ID (+)
```

The SQL Statement is required. However, if an SQL Statement (<SQL style>) is present, this takes precedent over SQL Statement.

The SQL statement outputs define five variables (Quote Number, Revision, Opportunity, Account, Site) to hold the result of the query statement.

In an outer join, there may not be an associated table, in which case the variable will be set to null.

Making Object Types Available in the Siebel Client

The workflow policy objects, columns, and programs that are created in Siebel Tools are available to the policy maker to create policies and actions in the Siebel client.

For these Siebel Tools objects to be accessible in the Siebel client, the Siebel Repository must be updated in the Siebel database. Workflow policy objects, columns, and programs are read from the Repository, not from the compiled Siebel repository file (.srf). The client must also have the correct repository name specified in the configuration file (.cfg) in the parameter “DockRepositoryName.”

About Workflow Policies Server Administration

This section describes the server administration tasks relevant to Workflow Policies.

Creating Database Triggers

The Generate Trigger (GenTrig) component on the Siebel Server allows you to create database triggers. Workflow Policies uses database triggers to identify which records match policy conditions. Run Generate Triggers when you:

- Create or delete new policies (including Assignment Policies), with the exception of Workflow Policies that have Batch Flag set to TRUE.
- Amend policy conditions or policy criteria.
- Change activation or expiration dates of policies, including Assignment Policies.

To run Generate Triggers, you must have installed Siebel Server, and the client you are using must be enabled to access the Siebel Server Administration screens. See the installation guide for the operating system you are using for more information on installing Server Manager.

CAUTION: If you have incorrectly defined a policy condition, running Generate Triggers may result in invalid triggers. An invalid trigger can prevent execution of normal user transactions. For this reason, thoroughly test your policies in your test environment before you deploy them in your production system.

Generating triggers is a one- or two-step process, depending on how the EXEC parameter is set; the default setting is FALSE.

- If the EXEC parameter is set to TRUE, the Generate Trigger component automatically creates the SQL script and applies it to the server database.
- If the EXEC parameter is set to FALSE, generating triggers is a two-step process:
  - Use the Generate Triggers component from a Siebel Server to create the SQL script file, which is placed in the root directory of the Siebel Server installation.
  - Use your database vendor's SQL tool to execute the SQL script file against the server database.

You can run the Generate Triggers component from either the Server Manager graphical user interface (GUI) or command line mode. Both the GUI and the command line use the same parameters.

So the triggers are only there to create indicators for the Workflow engine check the policies conditions.

About Database Triggers and Database Administration

It is important to keep your database administrators informed of any active Workflow database triggers, as any database Update or Insert event will cause the database trigger to react, regardless of how the event is executed.

For example, if you have Workflow triggers on Inserts to the S_SRV_REQ table, and the database administrator does a Table export and import of these records, the triggers will treat every record in the database as if it were a newly inserted record, which may result in inappropriate actions being taken on old records that were simply re-imported.

NOTE: In this release, the Generate Triggers task now requires the Privileged User Name and Password instead of Table Owner ID and Password.

Running Generate Triggers

When running Generate Triggers, remember the following tips, especially if you are deleting a policy:

- Deleting a policy and then running Generate Triggers does not remove the database trigger. When you delete a policy, you must run Generate Triggers with the remove parameter set to TRUE. This removes *all* triggers. You must then rerun Generate Triggers to reset the triggers for existing policies.
- You need to stop and restart the Workflow Monitor Agents when running Generate Triggers.
- Generate Triggers needs to be rerun whenever you change policy conditions. Generate Triggers does not need to be rerun when changing policy actions.
- For SQL Server, have your default database set correctly. To determine your default database, launch the SQL Server Enterprise Manager and navigate to the SQL Server Machine name. Then, click Security and then click LOGIN. The default database will be listed to the right.

To generate triggers using the GUI

1. In the Siebel client, from the application-level menu, choose Navigate > Site Map > Administration - Server Management > Jobs.
2. In the Jobs list, click New.
3. From the Component/Job drop-down list, select Generate Triggers. This creates a new line entry but does not start the task.
4. In the Job Parameters list, click New to modify parameter settings. The component-specific parameters for Generate Triggers are in Table 62.
5. Enter your Privileged User name and password.
6. In the Job Detail form applet, from the applet-level menu, select Start Job.
7. To view changes to the state, refresh the screen by clicking Run Query from the applet menu.
8. Upon completion, the Status field contains either Success or Error. It is recommended that you view the log details.

Table 62. Component-Specific Parameters for Generate Triggers

Name	Value	Description
Remove	TRUE or FALSE (default)	Set to TRUE to generate “DROP TRIGGER” statements to clean up the triggers. Remove does not generate “CREATE TRIGGER” statements.
Trigger File Name	Valid filename on the Siebel Server	Name and output location for the SQL script file. The default is TRIGGER.SQL. The file is created in the root directory of the Siebel Server during installation.

Table 62. Component-Specific Parameters for Generate Triggers

Name	Value	Description
EXEC	TRUE or FALSE (default)	Determines if the SQL script file runs automatically or manually. If TRUE, the SQL script file runs automatically. EXEC should be set to FALSE if you are running a Sybase server (all Siebel versions) or MS_SQL server (Siebel versions 4.x). This is to prevent connected users from getting an error message when Siebel generates database triggers. Make sure that no one is logged in to the database before you generate triggers. Also, if you are creating a large number of triggers because there are too many workflow policies, the triggers should be applied by the user and not by the Generate Triggers server process. The Exec parameter should be set to FALSE in this case.
Mode	ALL or WORK or ASGN	Set to ALL to create both Workflow Policy triggers and Assignment Manager triggers. Set to WORK to create only Workflow Policy triggers. Set to ASGN to create only Assignment Manager triggers.
Privileged User Name/ Privileged User Password	Assigned Privileged User name and password	All users must enter a Privileged User name and password. The Table Owner is considered a Privileged User, so you may enter the Table Owner name and password in the Privileged User name and password fields.

Running the SQL Script File

Once Generate Triggers has completed, run the SQL script file if the EXEC parameter is FALSE.

To run the SQL script file

1 Connect to the database server as the Siebel tableowner using your RDBMS vendor’s SQL tool (for example, ISQL for Microsoft or SQL*Plus for Oracle).

2 Run the SQL script file specified by the Trigger File Name parameter. The default filename is TRIGGER.SQL. The default location of this file is the root of the directory that the Siebel Server was installed in. For example, this might be:

```
C:\siebsrvr\trigger.sql
```

3 Verify that no errors are reported.

For example, the policy administrator, Bill Stevens, has finished creating policies in the test Siebel Client and wants the database triggers set in the Siebel database for the new policies. Using the Generate Triggers component, he sets the file output name.

This creates a file, TRIGGER.SQL, for the database administrator containing all the triggers that need to be modified or created in the test database for these policies.

The database administrator then runs the following command in SQL*Plus to create the triggers in the Oracle database:

```
SQL>@<path>\mytrig.sql
```

The successful creation of each database trigger in the Oracle database is indicated on the screen. For information on the syntax required for other databases, refer to your database documentation.

NOTE: On an MS SQL server database, execute the script trigger.sql as the database owner (dbo) login for the Siebel database.

About Database Triggers and Remote Users

When a remote user synchronizes, the changes get incorporated into the database (for example, account information in the S_ORG_EXT table is updated on synchronization). If you run a workflow which creates database triggers that compare changes in the database against specific conditions, then the triggers will fire and rows get written to S_ESCL_REQ if the changes are of interest to the workflow conditions during synchronization.

Setting Up the Siebel Server for Email Manager

Some workflow policy actions allow you to send email messages to specific individuals. To send email using Workflow Policies, an SMTP/POP3-compliant mail system must be working properly on your network and the Email Manager component of the Siebel Server must be running. You must also set the Mail Profile parameter to the name of the messaging profile you want to use for sending the email. You can use any SMTP/POP3-compliant client programs to verify your settings, including Microsoft Exchange or Microsoft Outlook.

When you create your messaging profile, you must be logged into the same account that is used to start Siebel Server. This account is specified when Siebel Server is installed.

To find the account used to start Siebel Server

1. Under Settings, bring up the Control Panel.
2. Click Services.
3. Scroll down and select Siebel Server.
4. Click the “Startup...” button.
5. When the Service dialog box appears, locate the account used to run the Siebel Server Service in the This Account field.

More information on using Email Manager is provided in the following sections:

- "Setting Up the Communications Profile to Send Email through Workflow" on page 222
- "Starting Email Manager" on page 223

Setting Up the Communications Profile to Send Email through Workflow

Sending email through Workflow involves creating an SMTP/POP3 communications profile.

NOTE: In order to create a new communications profile, CompGrp "CommMgmt" must be enabled. Verify that it is enabled before beginning the following task. For more information, see *Siebel Communications Server Administration Guide*.

To create a communications profile

1. From the application-level menu, choose Navigate > Site Map > Administration - Communications > Communications Drivers and Profiles.
2. In the Communications Drivers list applet, select the "Internet SMTP/POP3 Server" communications driver.
3. Click the Profiles tab.
4. Create a new profile called <Profile Name>.
5. Click the Driver Parameters tab, and complete the fields for the parameters described in Table 63 in the Profile Parameters Overrides applet:

Table 63. Driver Parameters for Creating a Communications Profile

Parameter	Entry
From Address	<the sender's email address for outbound communications>
POP3 Account Name	<the account name for the POP3 mailbox from which to retrieve inbound communications>
POP3 Account Password	<the password for the POP3 mailbox account>
POP3 Server	<the host name or IP address of the machine on which the Internet POP3 server is running, as appropriate for your network configuration>
SMTP Server	<the host name or IP address of the machine on which the Internet SMTP server is running, as appropriate for your network configuration>

NOTE: For details on the parameters listed in Table 63, see *Siebel Communications Server Administration Guide*.

After creating your communications profile, you need to create the component definition of Email Manager. The Email Manager component executes email actions once the conditions of a Workflow policy are met.

Starting Email Manager

You can start Email Manager from the command line, or from the Email Manager Component view.

To start Email Manager from the command line

- To start the Email Manager task with the <Profile Name> profile, in the server manager command line, use the following command to start a task for Email Manager:

  ```
  start task for comp MailMgr with MailProfile=<Profile Name>
  ```

To start Email Manager from the Server Administration view

1. From the application-level menu, choose Navigate > Site Map > Administration - Server Configuration > Servers > Components.
2. In the Components list, find the Email Manager component.
3. In the Component Parameters list, set Mail Profile = Test.
4. If you want to automatically start up a task for Email Manager whenever the component is restarted or the Siebel server services are restarted, set the component parameter Default Tasks = 1.

   When the workflow policy is violated, Workflow Monitor Agent will insert a record into the S_APSRVR_REQ table for workflow actions that invoke any of the Send Email workflow policy programs. Email Manager will then pick up records from the S_APSRVR_REQ table, setting their status from QUEUED to ACTIVE and then to SUCCEEDED during the course of the execution. Subsequently, Outbound Communications Manager will be invoked to log onto the <Profile Name> profile and send the emails out to the recipients using the Outbound Communications Manager business service, Send Message method.

Mail Profile Parameter

The Mail Profile parameter specifies the mail profile to use and is defined in Control Panel. The parameter establishes the connection between the application and the email system. If you do not specify a profile here, the default profile is used. The names must match exactly.

Setting Up the Siebel Server for Page Manager

Some Workflow policy actions allow you to send page messages to specific individuals. The Page Manager component of the Siebel Server must be running for you to send a page. Some actions can page specific individuals with alphanumeric or numeric pagers. To send a page using Workflow Policies, make sure these prerequisites are met:

- The server running the Page Manager component has access to a local or network modem.
- The Page Manager component of the Siebel Server is running. Several parameters, similar to dial-up networking set up in Windows, must be set prior to running the Page Manager component.
- Enter the appropriate telephone numbers for paging in the Employee view. These are the numbers used by Workflow.
- Change the regional configuration to avoid inputting the country code prior to the telephone number. This could cause errors.
- Change the list of values (PAGE_TYPE) parameters to make the page manager accept an alphanumeric send. This means the language and the value shown.

NOTE: Alphanumeric paging is more reliable than numeric paging because the pager messages are transmitted by the computer at the pager companies. This is not true for numeric paging, where pager messages are sent by emulating key presses on a phone. Failures in sending numeric pager messages are very hard to detect.

The Page Manager component uses the industry standard protocol Telocator Alphanumeric Protocol (TAP) for alphanumeric paging. Check with your pager company for the phone number to send your alphanumeric paging.

Several parameters affect how the Page Manager component interacts with the modem. You can change these parameters in the Server Administration screen. The available parameters are listed. The modem parameters are the defaults for Hayes-compatible modems. Verify that the settings are compatible with your modem.

To run the Page Manager component

1. From the application-level menu, choose Navigate > Site Map > Administration - Server Management > Jobs.
2. In the Jobs list, click New.
3. From the Component/Job drop-down list, select the Page Manager component.
4. In the Job Parameters list, click New.

5 Click on Parameters in the Server Tasks applet and enter your parameters.

See Table 64 for a list of parameters.

The most important parameters are Modem Port, Dial Prefix, Long Distance Prefix, and Local Area Code. Change the values for these parameters to match your system. If you do not specify a parameter, the default values described in Table 64 are used.

Table 64. Parameters for the Page Manager Component

Parameter	Value
Modem Port	The component object model (COM) port where the modem is attached. Valid values are COM1, COM2, and so on. Default = COM1
Dial Prefix	A number or sequence of numbers that needs to be dialed to get an outside line. If no dial-out prefix is use, insert a “,” (comma). Default = 9 Note that when dialing 9 for an outside line is not required and you are using srvrmgr.exe from the command line, specifying a comma for this parameter returns an error. But, if you set it to a hyphen (or any other non-dialable character) it will work correctly. Example command: `SRVRMGR.EXE /g NT01022 /e SBLPRD_ENT502 /s SBLPRD_APP502 /u ***** /p ***** /c "START TASK FOR COMPONENT PageMgr WITH DialPrefix = '-'"`
Long Distance Prefix	The long-distance prefix. This value is added in front of all long-distance phone numbers. Set this parameter to equal an empty string if your location does not require a long-distance prefix to be dialed. Default = 1
Local Area Code	The area code of your location. If the beginning digits of a phone number are equal to this code, they are removed before the phone number is dialed, and the long-distance prefix will not be added. Default = <empty>
Delay1	The number of seconds to wait between dialing a phone number and simulating key presses for the first set of numbers. This applies only to numeric paging. It is ignored for alphanumeric paging. Default = 12
Delay2	The number of seconds to wait between simulating key presses for the first and second set of numbers. This applies only to numeric paging. It is ignored for alphanumeric paging. This is also ignored if the numeric pager does not have a personal identification number (PIN). Default = 4

Table 64. Parameters for the Page Manager Component

Parameter	Value
Modem Reset String	A modem command used to reset the modem. Default = ATZ Refer to your modem documentation for the correct command.
Modem Init String	A modem command used to initialize the modem. Default = AT&FQ0V1 Refer to your modem documentation for the correct command. For example, some modems require a numeric value after &F.
Modem Dial String	A modem command used to dial the modem. You should rarely need to change this parameter. Default = ATDT
Modem Hangup String	A modem command used to hang up the modem. You should rarely need to change this parameter. Default = ATH
Modem Restore String	A modem command used to restore the power-up settings of the modem. You should rarely need to change this parameter. Default = AT&F
Request Key	When you have more than one Page Manager, the request key serves as the ID for each Page Manager. You can then specify this key for the Workflow action in the Workflow Action Argument view. The request key can be any string.

Troubleshooting the Email and Page Managers

Email Manager stops processing when it is unable to log on to the mail server (SMTP/POP3-compliant server) and logs an error message in the trace files.

Page Manager stops processing if the modem is not available. The requests continue to accumulate in the Requests table. After you fix your processing problems, you must restart the servers. The servers will continue processing from where they left off.

If Email Manager is able to log on but has a problem sending a particular email, it logs an error message and continues on to the next request. If Page Manager is able to interface with the modem but has a problem with a given page send, it will log an error and move on to the next request.

When Workflow Policies executes email and paging actions, it actually inserts email requests and paging requests into the database. These requests are inserted as records in the S_APSRVR_REQ table, which are then processed by Email Manager and Page Manager.

New requests have a status of "QUEUED." After a request is picked up by Email Manager or Page Manager, but before it is processed, it has a status of "ACTIVE." After a request is processed, its status becomes SUCCEEDED if the processing is successful, or FAILED if an error occurs.

To generate the sending emails, Siebel Server uses the UNIX “Mail” command. To verify that your server platform can run the command to a valid recipient and verify the email was successfully sent, do the following:

1. From the UNIX command prompt type:

   ```
   >mail recipient email address
   ```

   where *recipient email address* is a valid address.

2. Then, type a message ending with a period on the last line to indicate the end of the message. Then, press enter.

If email written to S_APSRVR_REQ is not sent, though the Email Manager trace file shows status SUCCEEDED, check that the following Outlook settings on the server are set:

- Send messages immediately
- Check for new messages every <*x*> minutes

Both of these options must be enabled in Outlook for email messages to be sent successfully.

Executing Workflow Policies with Workflow Monitor Agent

To execute your Workflow policies, you need to start Workflow Monitor Agent. Workflow Monitor Agent checks when the conditions of policies are met, and executes actions once those conditions are met.

You start and stop the Workflow Monitor Agent task in the Administration - Server views.

Table 65 describes Workflow policies database tables.

Table 65. Workflow Policies Database Tables

Table	Description
S_ESCL_REQ	This table holds the potential matching requests caused by applications.
S_ESCL_STATE	This table holds the time-based policy matches.
S_ESCL_ACTN_REQ	(Optional) This table holds the requests to execute actions. This is only used if you use Action Agent=TRUE.
S_ESCL_LOG	This table holds a history of base table rows that have matched policies.

What Workflow Monitor Agent Does

Workflow Monitor Agent performs several server processes that monitor the Siebel database. Workflow Monitor Agent:

- Checks the Escalation Requests table to see when the conditions of a policy are met

- Monitors all policies within a single group

  NOTE: You can only run one Workflow Monitor Agent process against a specific group at one time. You can run multiple Workflow Monitor Agent processes at the same time, but they must be against different groups; if you run two Workflow Monitor Agent processes against the same group, deadlocks will occur.

- If you use Action Agent = True, this generates requests for Workflow Action Agent in the Action Request table (S_ESCL_ACTN_REQ)
- Purges requests from the S_ESCL_REQ table after processing. When a database trigger is activated because a workflow policy condition is met, a record is inserted into the Escalation Request table, S_ESCL_REQ. Workflow Monitor Agent (Workmon) evaluates the request against the rules set up by the policies in the workflow policy group

If you use Action Agent = True, and if Workmon determines that the request in the S_ESCL_REQ table has no duration defined in the policy, Workmon either takes direct action and logs an entry into the S_ESCL_LOG table or sends it to the S_ESCL_ACTN_REQ table.

If Workmon determines that the request has a time element that must be met, the request is sent to the S_ESCL_STATE table along with the expiration time. The request stays in the S_ESCL_STATE table until the expiration time is met, or the request is removed because the conditions of the policy are no longer met. Workmon evaluates each of the requests that remains in the S_ESCL_STATE table for a time duration match or to determine if the condition still matches in the S_ESCL_STATE table. If you use Action Agent = True, then as each match occurs WorkMon either takes direct action and logs an entry into the S_ESCL_LOG table or sends it to the S_ESCL_ACTN_REQ table.

NOTE: If a workflow policy has a specified duration, the duration time is calculated from the time WorkMon detects that the row is in violation of the policy, not from the time the row was inserted into S_ESCL_REQ. For example, if you create a policy and set the duration as one week, but then WorkMon is not started until several days after Generate Triggers is run, the policy action will fire one week from when WorkMon is started, not one week from when the policy is created or Generate Triggers is run.

When the request for an action is made to the S_ESCL_ACTN_REQ table, Workflow Action Agent executes the action and logs an entry into the S_ESCL_LOG table.More information about Workflow Monitor Agent is provided in the following sections:

- "Using Workflow Monitor Agent" on page 228
- "Using Workflow Action Agent" on page 235
- "Starting Workflow Agent Processes Automatically with Siebel Server" on page 236

Using Workflow Monitor Agent

Before you start Workflow Monitor Agent, you must create a separate server component definition for each Workflow Monitor Agent task. You can start Workflow Monitor Agent from the Server Manager command-line interface.

Replication and Workflow Monitor Agent

Within the entire enterprise architecture of a Siebel deployment, there can be only one Workflow Monitor Agent monitoring a particular workflow group.

For example, a regional node can be running a Workflow Monitor Agent that monitors a group called Group 1. Meanwhile, in the headquarters, there is another WorkMon running, which monitors a group called Group 2. In this way, the organization is able to run Workflow Policies where needed, while working with the restriction of one WorkMon for one group.

NOTE: You cannot run more than one instance of Workflow Monitor Agent and Workflow Action Agent for a particular workflow group. However, you are allowed to have multiple Workflow Monitor Agent and Workflow Action Agent processes for different groups running at the same time.

Starting Workflow Monitor Agent

The following tasks are involved in starting Workflow Monitor Agent:

- "To create a Workflow Monitor Agent component definition" on page 229
- "To set parameters and activate a Workflow Monitor Agent component definition" on page 230
- "To stop or restart a Workflow Monitor Agent component" on page 230
- "To start Workflow Monitor Agent using the Server Manager command-line interface" on page 230
- "To run the Workflow Monitor Agent task" on page 230

Table 67 on page 232 shows Workflow Monitor Agent parameters.

To create a Workflow Monitor Agent component definition

1. In the Siebel client, from the application-level menu, choose Navigate > Site Map > Administration - Server Configuration > Enterprises > Component Definitions.
2. In the Component Definitions list, click New.
3. Complete the fields described in Table 66:

Table 66. Component Definitions Fields

Field	Description
Name	Name of the component
Component Type	WorkMon
Component Group	Select an existing component group
Description	Description of the component
Alias	Alias for the component. The alias can not contain blank spaces.

4. From the applet-level menu, choose Save Record.

   The component definition is saved. To view the definition, you must perform a query.

To set parameters and activate a Workflow Monitor Agent component definition

1. In the Component Definitions list, perform a query for the component definition.
2. (Optional) You may make additional changes to the component parameters. For a description of Workflow Monitor Agent parameters, see Table 67 on page 232.
3. From the Component Definitions list applet-level menu, choose Enable Component Definition.

   The definition state changes from “Creating” to “Active.”
4. Restart the Siebel server.

   Your changes take effect.

To stop or restart a Workflow Monitor Agent component

1. From the application-level menu, choose Navigate > Site Map > Administration - Server Management > Jobs.
2. In the Jobs list, click New.
3. From the link bar, click Servers.
4. Click the Components tab.
5. Select the component you would like to stop or restart, then click Shutdown or Startup.

To start Workflow Monitor Agent using the Server Manager command-line interface

1. Start the server manager by entering:

   ```
   srvrmgr /g <Siebel Gateway Name Server address> /s <Siebel server name> /e
   <enterprise server name> /u <server administrator username> /p <server administrator
   password>
   ```
2. Start a new Workflow Monitor Agent task in background mode by entering:

   ```
   start task for component WorkMon with SleepTime=<time>,GroupName=<group name>
   ```

You can start the Workflow Monitor Agent from the command line.

NOTE: You will need to create a separate server component definition for each Workflow Monitor Agent task.

To run the Workflow Monitor Agent task

1. From the application-level menu, choose Navigate > Site Map > Administration - Server Management > Jobs.
2. In the Jobs list, click New.
3. From the Component/Job drop-down list, select the name of the server component defined for this Workflow Monitor Agent task.

4. In the Job Parameters list, click New.

   Specify the parameters for the Workflow Monitor Agent. See Table 67 on page 232 for a list of parameters.

5. In the Job Detail form applet, from the applet-level menu, select Start Job to begin Workflow Action Agent task.

   NOTE: Run only one instance of Workflow Monitor and Workflow Action Agent for a given workflow group. For example, you can start only one instance for the "Sales" group at a specific time. However, you are allowed to have multiple Workflow Monitor and Workflow Action Agent processes for different groups running at the same time.

Table 67 shows Workflow Monitor Agent parameters.

Table 67. WorkMon Command-Line Interface Parameters

Parameter Name	Display Name	Description	Default Value
ActionAgent	Use Action Agent	Determines if Action Agent is automatically run with Monitor Agent. If set to FALSE (the default setting), the Workflow Action Agent server component starts within Workflow Monitor Agent, and actions are then executed by Workflow Monitor Agent. You must start Workflow Action Agent separately when using email consolidation and when the parameter Use Action Agent is set to TRUE.	FALSE
ActionInterval	Action Interval	Action execution interval in seconds. This argument determines when actions for a given policy are executed again on a given base table row. The purpose of this argument is to limit the number of times actions are executed if a row keeps going in and out of a matching condition. In other words, if the same record repeatedly violates the same policy before the action interval expires, the record will be removed from the S_ESCL_REQ table and the action will not be performed again. Note: The default is 3600 seconds. If this parameter is used, the value *must* be greater than 0 (zero) or unexpected behavior may result.	3600
BatchMode	Processes the batch policies	Determines if Monitor Agent is running in batch mode. When the value is set to TRUE, only the policies that have the Batch flag set to TRUE will be evaluated. When FALSE, only the policies that have the Batch flag set to FALSE will be evaluated. Note that when starting with Batch Mode set to TRUE, Workflow Monitor Agent will run once; that is, it will go through all records in the table and then exit out.	FALSE

Table 67. WorkMon Command-Line Interface Parameters

Parameter Name	Display Name	Description	Default Value
CheckLogCacheSz	Cache size of Policy violations	Number of policy violations to store in cache.	100
DeleteSize	Request delete size	This indicates the number of records to commit at a time. The minimum is 1. If Workflow Monitor encounters deadlocks, you may reduce the default to 125 with minimal performance degradation. Note: to avoid call stack errors, do not set the Request Delete Size value to zero.	500
GenReqRetry	Number of seconds to retry	Number of seconds to retry sending a Generic Request message.	120
GroupName	Group Name	Required. Workflow policy group that Monitor Agent works on.	
IgnoreError	Ignore errors	Ignore errors while processing requests. By default, the Workflow Monitor and Action agents will not ignore errors that occur while processing the requests. If Ignore Errors is set to TRUE and an error is encountered, the agent processes will log the error condition, delete the request, and continue working. By setting this argument to FALSE, the agent processes will exit on an error and send an email message to the mail ID specified by the Mailing Address argument. When running Workflow with Ignore Errors=TRUE, note that valid errors will be ignored. Whereas, if Ignore Errors is set to FALSE, the agent stops and exits with the error. It is recommended that you set Ignore Errors to FALSE so that valid errors are not ignored.	FALSE
KeepLogDays	Number of days to keep violation information	Number of days worth of violation information that should be retained. Sets the number of days log information is stored. Log information older than the number of days set is automatically removed from the system. This value can be set to 0 to prevent the purging of this log information.	30

Table 67. WorkMon Command-Line Interface Parameters

Parameter Name	Display Name	Description	Default Value
LastUsrCacheSz	Cache size of last user information	Number of last user information items to cache. When executing actions, the information about the last user to modify the base table row is available as a token substitution in the program arguments. By caching this information in the server, the throughput performance of executing actions can potentially increase.	100
MailServer	Mail Server	Name of email server to send notification of abnormal termination.	
MailTo	Mailing Address	Mail address to review notification of abnormal termination. Mail to <mail ID> if a Workflow Agent process exits with an error condition. An error can be caused by the failure of an action to execute, invalid object definitions, and so on.	
NumRetries	Number of Retries	Number of retries for recovery. This parameter works with the Retry Interval and Retry Up Time parameters to reconnect MTS or Siebel Server mode components to the database if database connectivity has been lost.	10000
ReloadPolicy	Reload Policy	Policy reload interval in seconds. This argument defines the frequency that policies are reloaded into the engine. This allows changes to be made on the screens and with the Generate Triggers component; the engine acts on the changes within some time frame. The default is 600 seconds.	600

Table 67. WorkMon Command-Line Interface Parameters

Parameter Name	Display Name	Description	Default Value
Requests	Requests per iteration	Maximum number of requests read per iteration. This controls the maximum number of requests WorkMon reads from the requests queue within one iteration. Between iterations WorkMon deletes processed requests from the requests queue and commits, optionally reloads policies from the database, checks for shutdown request, and optionally sleeps. In other words, you can think of the Requests Per Iteration parameter as a way to control the maximum amount of work WorkMon performs before taking these between-iteration steps.	5000
Sleep Time	Sleep Time	The time in seconds that the Workflow Agent process "sleeps" after it has polled for events and fulfilled all obligations to notify. Once it has completed its obligations, the Workflow Agent process stops polling for the time period set by the sleep interval. This parameter affects both the performance of the Workflow Agent process and the responsiveness of the application server.	60

NOTE: You can separate the processes for load balancing or run one process for ease of testing.

Using Workflow Action Agent

The Workflow Action Agent process submits a request to Email Manager and Page Manager when actions are to be taken.

NOTE: You will need to create a component definition for each Workflow Action Agent task.

Workflow Action Agent:

- Processes requests logged in the action request table (S_ESCL_ACTN_REQ) for a single group.
- Invokes all actions linked with the Workflow policy being processed.
- Logs email and page actions in the S_APPSRVR_REQ table for execution by Email Manager and Page Manager.
- Purges requests from S_ESCL_ACTN_REQ after processing.

If the Use Action Agent parameter is set to TRUE in the Monitor Agent process, you need to perform the following steps to start the Action Agent process.

To run the Workflow Action Agent process

- You start the Workflow Action Agent in the same way that you start the Workflow Monitor Agent. See "Starting Workflow Monitor Agent" on page 229.

To shut down the Workflow Agent process

- You shut down the Workflow Action Agent in the same way that you shut down the Workflow Monitor Agent. See "To stop or restart a Workflow Monitor Agent component" on page 230.

When restarting a workflow policy process, a Workflow Agent process immediately begins tracking all relevant activities that have occurred since it was shut down.

Starting Workflow Agent Processes Automatically with Siebel Server

You can specify that the Workflow Agent Process for a Workflow Group automatically starts when the Siebel Server is started.

To start a Workflow Monitor Agent Process automatically

1. In the Siebel client, from the application-level menu, choose Navigate > Site Map > Administration - Server Configuration > Enterprises > Component Definitions.
2. In the Component Definitions list, select the Workflow Monitor Agent in Server Components and set the parameters in Component Parameters to the following values:

   - **Group Name.** Enter the name of Workflow Group you want to start under Current Value. It will be copied to Value on Restart.
   - **Default Tasks.** Enter 1 under Value on Restart for starting one Workflow Agent.
   - **Use Action Agent.** Default is False, which means Workflow Action Agent is run automatically in Workflow Monitor Agent.

   See Table 67 on page 232 for a detailed description of these parameters.

   NOTE: If you want to Workflow Action Agent to run as a separate process for the above Workflow Monitor Agent, follow the above steps plus the following revised Step 2: Enter True under Current Value for Use Action Agent.

To start multiple Workflow Monitor Agent Processes for multiple Workflow Groups

1. Create a defined component as a Server mode component with WorkMon Component Type, and then assign the component to Siebel Server.
2. Create a defined component for each additional Workflow Group.

3 Follow the steps listed above to configure each component to start automatically.

About Workflow Policies and Siebel Server Task Trace Files

Whenever you start a Workflow Policies server process, a Siebel Server task trace file is created so that you can check for error messages and other information about the process. Trace files are created for the following Siebel Server processes:

- Generate Triggers
- Page Manager
- Email Manager
- Workflow Monitor Agent
- Workflow Action Agent

You can view trace file information in one of two places:

- The Siebel Server Tasks > Tasks Info Log in Administration - Server Management. See "Viewing Trace Files in Siebel Server Administration" on page 237.
- The log directory of your Siebel Server. See "Viewing Trace Files in the Siebel Server Log Directory" on page 237.

Viewing Trace Files in Siebel Server Administration

You can view trace files from the Administration - Server Management view.

To view trace files

- In the Siebel client, from the application-level menu, choose Navigate > Site Map > Administration - Server Management > Tasks > Log.

  The Tasks applet lists the status of all the Siebel Server tasks either running or started.

Viewing Trace Files in the Siebel Server Log Directory

You can also use Windows Explorer to navigate to your Siebel Server log directory. Under \log, select the server's name to see a file that lists all the trace files for each server process.

You can double-click the Trace File icon to access the trace file. You can view the trace file for any application server task.

About Tracing and Event Log Levels

Workflow Policies uses the events described in Table 68 for logging.

Table 68. Workflow Policies Logging Events

Event	Level	Description
SqlParseAndExecute	4	Traces all SQL statements and execution times.
Object Assignment	3	Traces Workflow Monitor Agent while it is doing Dynamic Assignment. Use in conjunction with Rules Evaluation.
Rules Evaluation	4	Traces Workflow Monitor Agent while it is doing Dynamic Assignment. Use in conjunction with Object Assignment.

NOTE: Setting trace levels above default parameters will affect performance. Reset trace levels to default parameters after troubleshooting has been completed.

About Workflow Policies Analysis Charts and Reports

Siebel Workflow Policies provides several charts for analyzing how frequently a workflow policy condition is met and the total number of policy instances occurring in a specified period of time. Workflow Policies also provides reports that summarize Workflow Policy and Workflow Log information.

More information about analysis charts and reports is provided in the following sections:

- "Using the Policy Frequency or Trend Analysis Chart" on page 238
- "Using Workflow Policies Reports" on page 239

Using the Policy Frequency or Trend Analysis Chart

Policy Frequency Analysis provides you with information about the number of times a Workflow policy executes. Policy Trend Analysis provides you with information about policy execution trends.

To view the Policy Frequency or Trend Analysis chart

- From the application-level menu, choose Navigate > Site Map > Administration - Business Process > Policy Frequency Analysis. This view has two applets:
  - **Monitor Log.** This lists the workflow policies.
  - **Workflow Policy Frequency/Trend Analysis.** The Workflow Policy Frequency Analysis applet displays a chart illustrating the execution frequency of a selected policy. The Workflow Policy Trend Analysis applet shows the total number of workflow policy conditions met over a specified period of time. To toggle between the two analysis applets, use the toggle list on the chart applet.

Using Workflow Policies Reports

In the Workflow Policies Policies and Log views, you can bring up a Reports page that you can print out. To bring up the report, select Reports from View.

The Reports page that appears provides summary information of the Workflow Policy.

If you need to review all the business rules for your organization, you can print out the Reports page for each of your workflow policies.

About Workflow Policies and Siebel Marketing

This section describes workflow policy programs for campaign execution.

Using Workflow Policy Programs for Campaign Execution

The Workflow Policy programs in Siebel Marketing were designed to allow a marketer to create complex campaign policies to automate the different stages of the campaign. Actions are based on the type of workflow policy program used and are used by Workflow Policies to create campaign policies.

Three workflow policy programs are designed for creating actions to execute campaigns:

- **Send Campaign Email.** Sends email to all contacts and prospects associated with a campaign. See "Using the Send Campaign Email Workflow Policy Program" on page 239.
- **Create Email Activity.** Creates an activity record on all the contacts or prospects that were sent an email. See "Using the Create Email Activity Workflow Policy Program" on page 240.
- **Assign to Campaign.** Takes a contact or a prospect and assigns it to a chosen campaign. See "Using the Assign to Campaign Workflow Policy Program" on page 240.

Using the Send Campaign Email Workflow Policy Program

The Send Campaign Email program provides marketers with the ability to send emails to campaign contacts and prospects.

Send Campaign Email has new Available Substitutions in the Send Message Arguments applet, such as [Prospect First Name], to allow for personalization of campaign emails.

To add a new substitution, you need to edit the SQL Statement corresponding to your Database Platform in Siebel Tools, Program | Programs Argument. Modify the Default Value for SQL Statement Outputs. These variables are for holding the result of the query statement. These variables also correspond to the Available Substitution in the Send Message Argument applet.

The Recipients applet is where you select the Recipient Type. The campaign contacts and prospects to whom the email will be sent are seen in the Contacts/Prospects applet in the Campaign Administration view.

Using the Create Email Activity Workflow Policy Program

This workflow policy program in a campaign creates an activity record on all the contacts or prospects that were sent an email. In the Arguments applet, you specify the data that fills in the columns on the record you are creating on the Contact Activity table. Table 69 describes valid values for the Arguments applet.

Table 69. Create Email Activity Program Arguments

Argument	Value
Name	Description: Text of activity. Status: Choose activity status such as planned or active from the picklist. Type: Choose Activity type from the picklist.
Required	This value indicates whether the argument is required.
Value	Text or picklist.

Using the Assign to Campaign Workflow Policy Program

This workflow policy program adds the selected contact or prospect to the list of campaign contacts or prospects for the designated campaign. Table 70 describes the value for the New Campaign argument.

Table 70. Assign to Campaign Program Argument

Argument	Value
New Campaign	Picklist that allows you to choose a campaign to which you will assign the contact or prospect.

Scenario for Creating a Marketing Campaign with Workflow Policies

In this scenario, a marketer wants to run a two-tier campaign with different actions taken depending on how the campaign recipient responds. The marketer is calling the campaign the “CD-ROM Promotion.” This is how the marketer wants the campaign to work:

- An email is sent telling recipients they can receive a discount by ordering a new product over the phone. The marketer wants to keep track of the recipients and to give them two weeks to respond.
- At the end of the two-week period, any recipients who did not respond to the offer are assigned to a new campaign.

To set up this campaign, the marketer must perform the following tasks:

- Define the actions to be used by the policies. See "Defining the Workflow Policy Actions" on page 241.
- Create a workflow policy group for the campaign. See "Creating the Workflow Policy Group" on page 242.
- Create the policies for the two tiers of the campaign. See "Creating the Policies" on page 243.

Defining the Workflow Policy Actions

Three Workflow policy actions are required for this scenario:

- **Send Campaign Email.** To send the offer email to the campaign recipients.
- **Create Email Activity.** To record the email activity in a table.
- **Assign to Campaign.** To assign nonrespondents to a new campaign.

The steps for creating the three actions are described below.

To create a Send Campaign Email action

1. Create a new record in the Workflow Policies Actions view.

   a. Enter the name of the action:

      `Send First Campaign Contact`

   b. Select a predefined program from the Program field:

      `Send Campaign Email`

   c. Select the following predefined Workflow Policy Object from the Workflow Object field picklist:

      `Campaign Contact`

   d. Enter any appropriate text in the Comments field if needed.

2. Fill in the Send Message Arguments applet.

   Select dynamic fields from Available Substitutions where appropriate.

   a. Enter text and dynamic fields in Subject.

   b. Enter text and dynamic fields in Message Template for sending email to Contacts.

3. Fill in the Recipients applet.

   a. Select a predefined Recipient Type from the Recipient Type field picklist.

  b Select the Recipient Name from the Recipient Name picklist.

To create a Create Email Activity action

1. Create a new record in the Workflow Policies Actions view Actions applet.
   - **a** Enter the name of the action:

     ```
     First CD-ROM Campaign
     ```

   - **b** Select a predefined Workflow policy program from the Program field:

     ```
     Create Email Activity
     ```

   - **c** Select a predefined workflow policy object from the Workflow Object field picklist:

     ```
     Campaign Contact
     ```

   - **d** Enter text in the Comments field if needed.
2. Fill in the Arguments applet with the activity table field name and the appropriate text.

To create an Assign to Campaign Email action

1. Create a new record in the Workflow Policies Actions view Actions applet.
   - **a** Enter the name of the action:

     ```
     Assign to Campaign
     ```

   - **b** Select a predefined workflow policy program from the Program field:

     ```
     Assign to Campaign
     ```

   - **c** Select a predefined workflow policy object from the Workflow Object field picklist:

     ```
     Campaign Contact
     ```

   - **d** Enter text in the Comments field if needed.
2. Fill in the Arguments applet to indicate the name of the new campaign.

Creating the Workflow Policy Group

All policies must be assigned to a workflow policy group, so in this scenario a group is created just for campaigns.

To define a Workflow Policy Group

1. Create a new record in the Workflow Policies Groups view.
2. Enter the name of the workflow policy group for this policy:

   ```
   Campaign Group
   ```

   This is the name entered into the Group field in the Workflow Policies Policies view.

Creating the Policies

Once the workflow policy actions and the workflow policy group are ready, the policies can be created. Two policies are required in this scenario:

- Email for CD-ROM campaign—to trigger the sending of the offer email and the email activity record.
- Assign Non-Respondents—to trigger the reassignment of nonrespondents to a new campaign.

In the procedures below for creating the policies, it is important to note how the fields in the Conditions applet are set.

To create the Email for CD-ROM Campaign policy

1. Fill out the Policies Applet in the Workflow Policies Policies view.
   1. Enter the policy name:

      ```
      Email for CD-ROM campaign
      ```

   2. Choose a workflow policy object from the picklist:

      ```
      Campaign Contact
      ```

   3. Choose a workflow policy group from the picklist:

      ```
      Campaign Group
      ```

   4. Enter a zero in the Duration field.
2. Fill out the Conditions Applet in the Workflow Policies Policies view.
   1. Enter a campaign name:

      ```
      1st CD-ROM Promotion
      ```

   2. Enter a start date.
   3. Enter a campaign status of Active. This is the trigger that sets off the campaign.

To create the Assign Non-Respondents policy

1. Fill out the Policies Applet in the Workflow Policies Policies view.
   1. Enter the policy name:

      ```
      Non-Respondents of CD-ROM campaign
      ```

   2. Choose a workflow policy object from the picklist:

      ```
      Campaign Contact
      ```

   3. Choose a workflow policy group from the picklist:

      ```
      Campaign Group
      ```

   4. Enter 14 days in the Duration field.
2. Fill out the Conditions Applet in the Workflow Policies Policies view.

a. Enter a campaign name:

   `1st CD-ROM Promotion`

b. Enter a campaign status of Active. This is the trigger that sets off the campaign.

c. Enter N in the Done field.

If the Policy duration is set to 14 days and Done is equal to N in the Conditions applet (meaning that no flag exists in the activity record for this recipient), then the policy executes in 14 days. That is, everyone who did not respond to the first campaign is assigned to a new campaign after 14 days.

About Testing Workflow Policies

This section describes testing and troubleshooting procedures.

Testing your workflow policies before implementing them into your production environment allows action recipients to receive accurate and useful information and the results are exactly what you want.

You need to develop a testing and migration procedure for introducing changes into the production environment. Some of the considerations for creating a test and migration environment are discussed in "Defining a Test and Migration Strategy for Workflow Policies" on page 162.

Before you can test your new workflow policies, you must install the Siebel Server workflow policy components on the Siebel Server.

CAUTION: Your test environment and production environment must have identical versions of the software.

Testing New Policies and Monitoring the Results

You need to test your new workflow policies by entering data that meets all the workflow policy conditions you defined in the policy. Test each of the newly defined workflow policies, workflow policy conditions, and workflow policy actions to verify that:

- The policies, conditions, and actions are correctly defined
- The policies, conditions, and actions accurately define the transactions (the correct columns) you want to monitor
- The actions are what you want and occur when you want them
- The action interval and sleep times are correctly defined

Correctly testing your workflow policies and eliminating any problems are critical before implementing the policies in your production environment.

Make sure your database triggers are created, the email and pager server processes are running, and your Workflow Agent processes are running before you test and monitor the new policies.

You verify your action by checking to see if the proper action occurs. That is, you can check that the email arrives or the pager goes off. You can monitor Workflow Agent progress using the Workflow Policies Log view.

The Workflow Policy Log view displays a log of all the records that meet a policy condition tracked by the Workflow Monitor Agent process. You access the Workflow Policy Log view from the Siebel client.

The view contains the following fields:

- **Policy.** The name of the policy.
- **Workflow Object.** The name of the workflow policy object.
- **Object Identifier.** The ID of the workflow policy object for which the policy condition was met.
- **Object Values.** Identifying information for the row that met the policy condition.
- **Event Date/Time.** The date and time the policy condition was met.

Once you have verified that the workflow policies work as expected, you can migrate the workflow policies to your production environment.

Troubleshooting Workflow Policies

Because workflow policies are based on database triggers, a workflow policy can take effect on a database record only after the record is committed. If you have a policy that is based on multiple database tables, the policy takes effect only if the records on all tables are committed. For example, Opportunity Revenue is stored in the S_OPTY_POSTN table, and lead quality is stored in the S_OPTY table. A policy with conditions Opportunity Revenue > 10M and Lead Quality = high takes effect only when the records are committed on both tables.

Also keep in mind that multiple business components can be created for the same database table using search specifications. If you are creating a workflow policy component to monitor a business component, be sure to include all the fields that are being used in search specifications as workflow policy columns. The workflow policy column can then be used in the policy conditions to allow appropriate behavior to be enforced.

If your workflow policy action does not occur, check the following:

- Verify that your test record meets *ALL* your workflow policy conditions.
- Verify that the client configuration file is pointing to the correct enterprise server (one error that can occur if the server is incorrect is ESC-00053, “Error loading rule definitions”).
- Check the workflow policy activation date/time.
- Check the monitor task:
  - Is the monitor awake and running against the correct group?
  - Search the Task Information log for the Row_Id of your test record.
    - ❑ If Row_Id does not exist, run GENERATE TRIGGERS.
    - ❑ Update your test record.

- Check the Action Agent task:
  - Is Action Agent awake and running against the correct workflow policy group?
  - Search the Task Information log for the Row_Id of the test record.
- Make sure your triggers are generated.

Workflow Policies and Tracing

Workflow Policies uses the General Events event for logging. To view informational messages, set the log level to 3. To view debugging information, set the log level to 4.

Migrating Policies to the Production Environment

To migrate fully tested policies to your production environment, you need to follow a process similar to the one used for implementing the policies in your test environment.

To migrate to your production environment

1. Back up your production environment database.
2. Migrate your test repository environment into your production repository environment. The process is described in the upgrade guide for the operating system you are using.
3. Re-enter your workflow policy action types, workflow policies, and workflow policy groups exactly as they are in the test environment into the production environment.

   NOTE: Information that you have entered using Siebel Tools does not need to be re-entered.
4. In the Siebel client, from the application-level menu, choose Navigate > Site Map > Administration - Server Management > Jobs.
5. In the Jobs list, click New.
6. From the Component/Job drop-down list, select Generate Triggers. This creates a new line entry but does not start the task.
7. In the Job Parameters list, click New to modify parameter settings.

   For a description of the component-specific parameters for Generate Triggers, see "About Workflow Policies Server Administration" on page 217.
8. Select Submit Query.

   See "About Workflow Policies Server Administration" on page 217 for more information on trace files.

NOTE: To help prevent invalid triggers from being applied to your production environment, apply your database triggers to your test environment before you apply them to your production environment.

Predefined Programs

The following is a list of all predefined programs. These programs have been created from the five program types:

- Send Page
- Send Email
- Run External Program
- Send Message Broadcast
- Database Operation

Table 71 contains common actions that you can use by inserting your own message text.

Table 71. Predefined Programs

Program	Description
Send Page	
Send Page	Send a generic page message.
Send Opportunity Page	Send a page regarding an opportunity.
Send Quote Page	Send a page regarding a quote.
Send SR Page	Send a page regarding a service request.
Send Email	
Send Email	Send a generic email message.
Send Opportunity Email	Send an email regarding an opportunity.
Send Quote Email	Send an email regarding an opportunity quote.
Send SR Email	Send an email regarding a service request.
Message Broadcast	
Send Message Broadcast	Send a generic message broadcast.
Send SR Message Broadcast	Send a message broadcast regarding a service request.
Send Opportunity Message Broadcast	Send a message broadcast regarding an opportunity.
Run External Program	
Run External Program	Run an external program.
Database Operation	
Change SR Close Date to Today	Update the service request’s close date to today’s date.
Change SR Owner	Change the service request’s owner.
Change SR Group	Change the service request’s group.

Table 71. Predefined Programs

Program	Description
Change SR Owner to Manager	Change the service request’s owner to the current owner’s manager.
Change SR Priority	Change the service request’s priority to a new value.
Change SR Severity	Change the service request’s severity to a new value.
Change SR Status	Change the service request’s status to a new value.
Change SR Sub-status	Change the service request’s sub-status to a new value.
Create SR Activity	Create a service request activity.
Create Opportunity Activity	Create an opportunity activity.

11 Reference Materials for Siebel Workflow

This chapter provides reference information organized as follows:

- "Siebel Workflow Terminology" on page 250
- "Predefined Business Services" on page 253
- "Passing Parameters to and from Workflow and Data Manipulation within Workflows" on page 258
- "Using Expressions with Workflow Processes" on page 264

Siebel Workflow Terminology

Table 72 and Table 73 describe the common terms for Workflow Processes and Workflow Policies.

Table 72. Workflow Processes Terms

Term	Definition
Arguments	Data passed to or received from a process or step.
Branch	A possible outcome of a workflow process step. A branch can have one or more conditions. A branch is followed by a step in the workflow process definition. If all of the conditions for the branch are met, the work item proceeds to the step following the branch.
Business object	A group of one or more business components. A business object represents an entity in the Siebel application that you would like to monitor. A workflow process is based on one and only one business object. Business objects are defined in Siebel Tools.
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 service	A type of step in a process in which an automated call is made to a service, such as the Outbound Communications service that handles inbound and outbound messaging. A workflow process definition can have one or more business service steps.
Connector	A definition of the relationship between two workflow process steps.
Decision point	A type of step in a workflow process definition in which the work item branches off to different steps depending on a set of conditions. A decision point consists of all possible branches for that point in the business process. Each branch consists of one or more conditions that must be met for a work item to follow that branch. A workflow process definition can have one or more decision steps.
End	A type of workflow process step that specifies when a process instance is finished.
Exception	A type of workflow process step that specifies when a process instance should follow an alternative branch instead of the normal branch path.
Process property	A storage field that contains values for use in steps as input and output arguments or for performing evaluations.
Process Simulator	A graphical flowchart interface used for debugging workflow processes.
Siebel Operation	A type of workflow process step that handles database operations such as insert, query, or update of a business component record or field.

Table 72. Workflow Processes Terms

Term	Definition
Start	A type of step that defines the conditions for initiating an instance of a workflow process. When the conditions have been met, the process instance is initiated. A workflow process definition has one and only one start step.
Step	An activity within a workflow process. Steps are logically linked together to create a process definition.
Step instance	The instance of a process definition step that has been initiated. A start step is initiated when all conditions defined for the start step have been met. A decision step is initiated when all conditions for a decision branch have been met. All other steps are initiated when the previous step has completed.
Stop	A type of workflow process step that specifies the conditions that cause a process instance to terminate prior to completion.
Subprocess	A workflow process embedded into another workflow process as part of the workflow process definition. A subprocess has its own workflow process definition. A subprocess is a type of step. There can be one or more subprocess steps in a workflow process.
Wait	A type of workflow process step that specifies when a process instance should pause in execution and the duration of the pause.
Workflow process	The representation of a business process. A workflow process comprises one or more steps that indicate when a business process starts and ends and includes information about individual activities within the business process.
Workflow process instance	An instance of a workflow process that has been initiated. A process instance is initiated when the input conditions for a process definition have been met. A process instance consists of one or more step instances and contains one or more work items.
Work item	The representation of the work being processed in the context of a step within a process instance. A work item is an instance of a business object.

Table 73. Workflow Policies Terms

Term	Definition
Business object	A group of one or more business components. A business object represents an entity in Siebel that you would like to monitor. A workflow policy object is based on one and only one business object. Business objects are defined in Siebel Tools.
Business rule	The definition of how an organization wants to carry out a process in its operations.
Object type	An entity in Siebel Tools displayed as a node on the Object Explorer. For example, workflow policy objects, workflow policy components, workflow policy columns, and policy programs are all object types.

Table 73. Workflow Policies Terms

Term	Definition
Policy action	An event that Siebel executes when all policy conditions are true and all the workflow policy properties are satisfied. Policy actions are based on programs. Policy actions are defined in the Workflow Policies Actions view. Once you define a policy action, it can be used in a workflow policy.
Policy condition	A policy condition is an expression that is compared against the data in the Siebel database. The result of the comparison is either true or false. Workflow policy conditions are defined in the Workflow Policies Policies view. A policy condition is defined by selecting a workflow policy column, selecting a comparison operator, and entering or selecting a value, if appropriate.
Program	The definition of an event. Types of events are Send Email, Send Page, Database Operation, Send Message Broadcast, and Run External Program. Different properties are associated with a program based on the event type. Some of the properties that can be defined for a program include the fields that can be substituted into a message, the possible recipients of a message, and the database columns that you would like to update. Programs are defined in Siebel Tools.
Workflow policy	A systematic expression of a business rule. A workflow policy contains one or more policy conditions and one or more policy actions. If all the policy conditions for a workflow policy are true, then the policy action occurs. (That is, when all policy conditions are met.) A workflow policy is contained by one workflow policy group and is related to one workflow policy object. A workflow policy contains additional properties that govern its behavior. Workflow policies are defined in the Workflow Policies Policies view.
Workflow policy column	A column that defines the column on the Siebel database table that you would like to monitor. You use workflow policy columns when defining workflow policy conditions for a workflow policy. A workflow policy column must be associated with a workflow policy component for it to be used in a workflow policy. A workflow policy column that is associated with a workflow policy component is called a workflow policy component column. Workflow policy columns are defined in Siebel Tools.
Workflow policy component	Components that define the Siebel database tables you would like to monitor. Workflow policy components also define the relationships between tables. Workflow policy components contain workflow policy columns. Workflow policy components are defined in Siebel Tools.
Workflow policy component column	A workflow policy column that is associated with a workflow policy component. Workflow policy component columns define the database columns that can be used in workflow policy conditions for a workflow policy. Workflow policy component columns are defined in Siebel Tools.

Table 73. Workflow Policies Terms

Term	Definition
Workflow policy group	A group of one or more workflow policies. Workflow policy groups allow you to group workflow policies sharing common desired behavior. Siebel Server processes monitor workflow policy groups. For example, workflow policies that need to be monitored hourly would be in a different workflow policy group than those that need to be monitored weekly. Workflow policy groups are defined in the Workflow Policies Groups view.
Workflow policy object	A group of one or more workflow policy components. A workflow policy object represents an entity in the Siebel application that you would like to monitor. A workflow policy is based on one and only one workflow policy object. Workflow policy objects are defined in Siebel Tools.

Predefined Business Services

This section describes the following predefined business services:

- **Outbound Communications Manager.** See "Outbound Communications Manager Business Service" on page 253.
- **Synchronous Assignment Manager Requests.** See "Synchronous Assignment Manager Requests Business Service" on page 253.
- **Server Requests.** See "Server Requests Business Service" on page 254.
- **Workflow Utilities.** See "Workflow Utilities Business Service" on page 257.

For more predefined business services, see *Business Processes and Rules: Siebel eBusiness Application Integration Volume IV*.

Outbound Communications Manager Business Service

The Outbound Communications Manager business service is for sending notifications, such as notifications to contacts or employees. For information on methods and arguments, see *Siebel Communications Server Administration Guide*.

Synchronous Assignment Manager Requests Business Service

The Synchronous Assignment Manager Requests business service is for assigning an object using Assignment Manager rules. For more information on Assignment Manager rules, see *Siebel Assignment Manager Administration Guide*.

This service has one method available, Assign. This method sends a request to the assignment manager server component.

Assign Arguments

Table 74 describes the Assign method arguments.

Table 74. Assign Method Arguments

Argument	Description
Assignment Object Name	Required. This is the object that you want to assign.
Object Row ID	Required. This is the row ID for the object you want to assign. To assign the work item for the workflow process, set this to the Object ID process property.

TIP: The Synchronous Assignment Manager Requests business service attempts to assign all records that meet the appropriate criteria, even if they are locked. To prevent errors in your process due to locked records, set up a condition in your workflow process or workflow policy to skip records that do not meet the condition ASGN_USR_EXCLD_FLG = N.

Server Requests Business Service

The Server Requests business service is for sending generic requests to the server request broker. The Server Requests business service can send requests in three different modes: asynchronous, synchronous or schedule mode. While in synchronous mode, it will send the request to the server request broker and wait for a response. Otherwise, it will just send the request but does not wait for a response.

When invoking the Server Requests business service to submit a component request, you need to specify SRBroker parameters in the input property set and all component specific parameters in a child property set. There are two points to note: There is no validation for component parameters passed in the child property set, and these arguments do not appear in the picklist in the workflow administration views.

NOTE: If you want to pass parameters to the server component that are not listed as available arguments, you can create a custom business service that contains the necessary parameters. Alternatively, you can create a component job that has the parameters defined as part of the job definition.

This service has two methods available:

- **Submit Request.** Use this method to submit a request to the server request broker.
- **Cancel Request.** Use this method to cancel any server request that is currently awaiting to be run.

Submit Request Arguments

Table 75 describes the Submit Request method arguments.

Table 75. Submit Request Method Arguments

Argument	Description
Component	Required (if Component Job is not entered). Enter the name of the server component to run.
Component Job	Required (if Component is not entered). Enter the name of the component job to run.
Delete After	Optional. Number of iterations before deleting the request. Works with Delete After Units. The default value is 0 (zero).
Delete After Units	Optional. The units to measure the iterations for the Delete After argument. The default value is "NoReq" for synchronous (request is not saved to the database) and "Eon" for asynchronous (request is never deleted). Other possible values are: ■ ASAP ■ SECONDS ■ MINUTES ■ HOURS ■ DAYS ■ WEEKS ■ MONTHS ■ YEARS
Description	Optional. A description of the server request.
Hold Flag	Optional. For asynchronous requests only. Flag to indicate whether or not to hold the request.
Maximum Execution Time	For future use.
Method	Optional. Only applicable for service-based server components (for example, Workflow Process Manager, Communications Manager). Specify the business service method to invoke.

Table 75. Submit Request Method Arguments

Argument	Description
Mode of Server Request	Required. This tells the server request broker how to handle the server request. While in auto mode, the server request broker will set the mode to either synchronous or schedule, depending if the client is connected or mobile. ■ Sync: Synchronous ■ Async: Asynchronous ■ Schedule: Schedule ■ Auto: Automatic configuration
Request ID Needed	Optional. This is only applicable to asynchronous and schedule mode. If this is set to false, these two server requests will return even faster.
Request Key	For future use.
Repeat Interval	Optional. The interval for repeating requests.
Number of Repetitions	Optional. The number of repetitions for repeating requests.
Repeat From	Optional. Possible values are Scheduled Start, Actual Start, and End.
Repeat Interval Units	Optional. Unit of intervals for repeating requests.
Server Name	Optional. Enter the specific server that this request is to be run from.
Start Date	Optional. Start date and time.
Storage Amount	Optional. Enter the amount of time that the server request will be stored in the database in the event that the server is down.
Storage Units	Optional. Enter the units to measure the iterations for the Storage Amount argument. The units are the same as Delete After Units.

Cancel Request Method Arguments

Table 76 describes the Cancel Request method arguments.

Table 76. Cancel Request Method Arguments

Argument	Description
Request ID	Required. This is the ID of the server request to be cancelled.
Repeat Number	Optional. This is the number of repetitions of the repeating server requests that are to be cancelled.

Workflow Utilities Business Service

The Workflow Utilities business service contains generic utilities that can be used in process definitions.

This business service has one available method, Return Property Values. This method returns a mirror image of the input arguments. The Return Property Values method is also referred to as the Echo method.

Return Property Values Arguments

Table 77 describes the Return Property Values method arguments.

Table 77. Return Property Values Method Arguments

Argument	Description
Input Arguments	This method accepts any input arguments.
Output Arguments	An exact copy of the input arguments.

Passing Parameters to and from Workflow and Data Manipulation within Workflows

This section explains how to manipulate data within workflow processes and how to pass parameters to and from Workflow.

Manipulating Data Within Workflows

You can manipulate data within workflow processes. This section explains how to access run-time event parameters from workflow processes.

To access run-time event parameters from workflow processes

1 Set up a run-time event for each applet that is required to trigger a workflow process.

Fill in the fields of the WF Step Branch applet. Example field values are shown in the following table.

Field	Value
Branch Name	Enter a name for the branch.
Type	Condition
Next Step	Enter a name for the next step.
Event Object Type	Applet
Event	InvokeMethod
Event Object	Choose the name of the event object that will trigger the workflow.
Sub Event	NewRecord
Comments	Optional
Event Cancel Flag	The default is blank.

2 Set up your process properties in Process Designer as usual. For information on using process properties, see "About Process Properties" on page 69.

3 Drill down into the first step in your workflow process and add a new output argument for each process property you want to populate using run-time events. Add new output arguments by filling in the fields of the Output Arguments applet. Example field values are shown in the following table.

Field	Value
Property Name	Enter a name for the property.
Type	Expression
Value	GetProfileAttr(“RestrucOut”) Note: As you are filling in the Profile Attribute field, make note of the name you give to each of the Profile Attributes property values, for use in Step 8 on page 260.
Output Argument	Leave as default.
Business Component Name	Leave as default.
Business Component Field	Leave as default.
Comments	Optional

4 Activate the workflow process. For more information, see “Deploying Workflow Processes” on page 144.

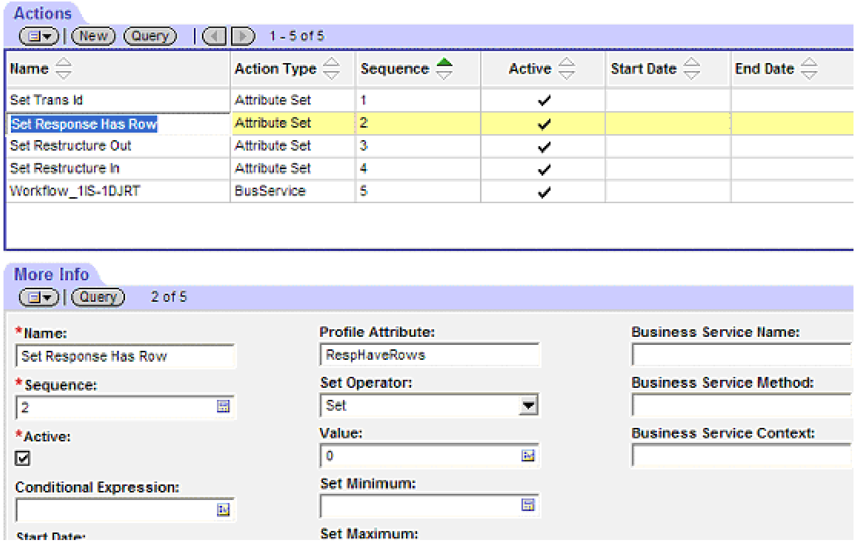
5 Since the workflow process includes run-time events, you must load the run-time events.

   a From the application-level menu, choose Navigate > Site Map > Administration - Runtime Events > Events.

   b From the applet menu, select Reload Runtime Events.

6 From the Events view tab, find the run-time event that is associated with the workflow run-time event that you created. As an example, you can query for the InvokeMethod event on the Opportunity List Applet object.

7 Drill down on the Action Set Name. The Actions view is shown in the following figure.



Name	Action Type	Sequence	Active	Start Date	End Date
Set Trans Id	Attribute Set	1	✓		
Set Response Has Row	Attribute Set	2	✓		
Set Restructure Out	Attribute Set	3	✓		
Set Restructure In	Attribute Set	4	✓		
Workflow_1IS-1DJRT	BusService	5	✓		

*Name: Set Response Has Row

*Sequence: 2

*Active: ☑

Conditional Expression:

Start Date:

Profile Attribute: RespHaveRows

Set Operator: Set

Value: 0

Set Minimum:

Set Maximum:

Business Service Name:

Business Service Method:

Business Service Context:

8 Add a new Action for each process property that will be populated using run-time events. For example, you can create a new action to set the ACU Transaction ID and call it Set ACU Trans ID.

a For each new action:

- Set the Action type to Attribute Set.
- Set the Profile Attribute to match the value you used in the GetProfileAttr call in the Process Designer in Step 3 on page 259, such as TransType.
- Set the Set Operator value to Set.
- Assign the Value field the appropriate value, such as the literal string FA-0001. You assign the value using the standard Siebel expression builder.
- Set the Sequence for the action to be less than the default sequence (for the Workflow_XXXXXXX Action).

b Make sure the Sequence setting of the Workflow_XXXXXXX Action is the highest number so that this action happens after all the other actions.

NOTE: Whenever you modify the workflow process, you must repeat this step because modifying a workflow process resets the workflow action's sequence to 1.

c From the applet menu, select Reload Runtime Events.

Passing Parameters to and from Workflow with the Workflow Process Manager Business Service

The Workflow engine can be invoked programmatically, that is, by calling a business service. The Workflow Process Manager business service is a standard Siebel business service used for this purpose. When you run a workflow process by invoking the Workflow Process Manager business service, you can pass inputs to Workflow, and in some cases, obtain outputs from Workflow.

Passing Inputs to Workflow

The input property set is required to contain a property named ProcessName, which specifies the name of the workflow process to be run. In addition to the ProcessName property, you can put other values, such as strings, numbers, and property sets, into the property set. These values will be passed to the workflow process by the Workflow Process Manager business service.

Simple data type process properties (such as String, Number, and DateTime) that are marked In or In/Out will be initialized if the input property set has a property (in the top-level property set) with a name matching the name of the workflow process property. The value of such a property in the input property set will initialize the value of the matching workflow process property.

Hierarchical data type process properties that are marked In or In/Out will be initialized if the input property set has a child whose property set Type field contains a string matching the name of the hierarchical workflow process property. If such a match is found, the matching child (and everything below the child) in the input property set is copied into the process property.

Passing Outputs from Workflow

Not all workflow processes that are started programmatically return outputs. For example, an interactive workflow process may be programmatically started, but since it can pause, the output from the call to start the workflow process may reflect the state at an intermediate point. For this reason, only workflow processes that are guaranteed to run to completion in one call, that is, service flows, should be expected to provide output in the output arguments of the call into the Workflow Process Manager business service.

Output arguments follow the same convention as input arguments. Simple workflow process properties (such as String, Number, and DateTime) that are marked Out or In/Out will appear as properties on the top-level property set. Hierarchical process properties will appear as children of the output property set. Hierarchical process properties can be located by examining the Type field of the child, which will match the workflow process property name.

Example Scripts

You can use scripts for invoking Workflow programmatically and for passing parameters. Example scripts are provided in the following sections:

- "Example Script: Invoking Workflow Programmatically and Constructing an Input Property Set" on page 262
- "Example Script: Defining Property Sets for the Input Property Set" on page 262

- "Example Script: Constructing Property Sets" on page 262
- "Example Script: Assembling Properties and Child Property Sets into the Input Property Set" on page 263
- "Example Script: Invoking the Workflow Process Manager Business Service and Passing It the Input Property Set" on page 263

Example Script: Invoking Workflow Programmatically and Constructing an Input Property Set

The following is a sample script which programmatically invokes the Workflow Process Manager business service and which constructs an input property set, psInputs, for the business service. This script defines strings that will be put into the input property set as properties.

```
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:\\sea752\\XMLMessages\\AgentAuthorizationVBCReq-final.xml"

var resFileName = "C:\\sea752\\XMLMessages\\AgentAuthorizationVBCResponse-final.xml"
```

Example Script: Defining Property Sets for the Input Property Set

The following is a sample script that defines property sets, which will be put into the input property set as child property sets.

```
//Request PS

var psRequest = app.NewPropertySet();

var psAgentNumTag = app.NewPropertySet();

var psType = app.NewPropertySet();

var sAgentID;
```

Example Script: Constructing Property Sets

The following is a sample script that constructs property sets.

```
//Build property set hierarchy
```

```
sAgentID = app.LoginName();

psRequest.SetType("XMLHierarchy");

psAgentNumTag.SetType("DataAgentNumber");

psAgentNumTag.SetValue(sAgentID);

psRequest.AddChild(psAgentNumTag);
```

Example Script: Assembling Properties and Child Property Sets into the Input Property Set

The following is a sample script that assembles properties and child property sets into the input property set.

```
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);
```

Example Script: Invoking the Workflow Process Manager Business Service and Passing It the Input Property Set

The following is a sample script that invokes the Workflow Process Manager business service and passes it the input property set.

```
var svc = TheApplication(). GetService(“Workflow Process Manager”);

svc.InvokeMethod("RunProcess", psInputs, psOutputs);//Call the Workflow

var sErr = psOutputs.GetProperty("sErr");//Check the Workflow status
```

Passing Parameters from Workflow to Global Variables (Profile Attributes)

You can use a business service to access and pass parameters from Workflow to global variables.

To access workflow parameters for a running workflow process

1. Define a business service with relevant 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 "About Business Service Steps" on page 86.
4. Use the following script to take the business service argument values and assign them to Profile Attributes.

```
function Service_PreInvokeMethod (MethodName, Inputs, Outputs)

{

if( MethodName == "XXX" ) {

var isWorkflowRunning, viewValidCurrent, viewValidNext;

// read the input arguments into profile attributes

isWorkflowRunning = Inputs.GetProperty("Workflow Running");

viewValidCurrent = Inputs.GetProperty("Valid View Current");

viewValidNext = Inputs.GetProperty("Valid View Next");

TheApplication().SetProfileAttr("WFRunning", isWorkflowRunning);

TheApplication().SetProfileAttr("WFViewCurrent", viewValidCurrent);

TheApplication().SetProfileAttr("WFViewNext", viewValidNext);

}
```

5. Use the profile attributes for further processing. All the necessary information has been put into the profile attributes of the application. Users can use the standard procedures for accessing the profile attributes to extract this information. For more information, see *Siebel Personalization Administration Guide*.

Using Expressions with Workflow Processes

Expressions can be used in workflow processes. The timestamp argument is of particular note, as its arithmetic varies from the time information used in workflow policy programs.

For more information, see "Using the Timestamp Argument" on page 265.

Using the Timestamp Argument

You can use the timestamp argument to get the current system time and to do time arithmetic based on the current time.

The arithmetic involving time information for workflow processes is different from that for workflow policy programs. The second operand to the 'Timestamp () ' function must be provided in a scale of minutes, that is, as a fraction of the whole day.

For example, if the intended result of an arithmetic operation is 30 minutes, then the argument should appear as follows:

```
Timestamp()+0.021
```

The operation is explained as follows:

- 0.021 = 30/(24*60)
- (24*60) represents a whole day in minutes
- 30 represents the required minutes

Index