

SIEBEL® eBUSINESS APPLICATIONS

SIEBEL WORKFLOW ADMINISTRATION GUIDE

SIEBEL 2000
VERSION 6.1 REV. A

10PA1-WF01-06100

JUNE 2001

Siebel Systems, Inc., 2207 Bridgepointe Parkway, San Mateo, CA 94404
Copyright © 2001 Siebel Systems, Inc.
All rights reserved. Published 2001
Printed in the United States of America

No part of this publication may be stored in a retrieval system, transmitted, or reproduced in any way, including but not limited to photocopy, photographic, magnetic or other record, without the prior agreement and written permission of Siebel Systems, Inc.

The full text search capabilities of Siebel eBusiness Applications include technology used under license from Fulcrum Technologies, Inc. and are the copyright of Fulcrum Technologies, Inc. and/or its licensors.

Siebel, the Siebel logo, TrickleSync, TSQ, Universal Agent, and other Siebel product names referenced herein are trademarks of Siebel Systems, Inc., and may be registered in certain jurisdictions.

Windows® is a registered trademark of Microsoft Corporation.

All other product names, marks, logos, and symbols may be trademarks or registered trademarks of their respective owners.

U.S. GOVERNMENT RESTRICTED RIGHTS. Programs, Ancillary Programs and Documentation, delivered subject to the Department of Defense Federal Acquisition Regulation Supplement, are “commercial computer software” as set forth in DFARS 227.7202, Commercial Computer Software and Commercial Computer Software Documentation, and as such, any use, duplication and disclosure of the Programs, Ancillary Programs and Documentation shall be subject to the restrictions contained in the applicable Siebel license agreement. All other use, duplication and disclosure of the Programs, Ancillary Programs and Documentation by the U.S. Government shall be subject to the applicable Siebel license agreement and the restrictions contained in subsection (c) of FAR 52.227-19, Commercial Computer Software - Restricted Rights (June 1987), or FAR 52.227-14, Rights in Data—General, including Alternate III (June 1987), as applicable. Contractor/licensor is Siebel Systems, Inc., 2207 Bridgepointe Parkway, San Mateo, CA 94404.

Proprietary Information

Siebel Systems, Inc. considers information included in this documentation and in Siebel 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

Introduction

Who Should Use This Guide	Intro-2
How This Guide Is Organized	Intro-3
What's New in This Release	Intro-5
Additional Documentation	Intro-5
Contacting Siebel Technical Support	Intro-7
Siebel Systems Welcomes Your Comments	Intro-8

PART 1. Concepts and Overview

Chapter 1. Overview of Workflow

General Principles of Workflow	1-2
Siebel Workflow Solutions	1-3
Siebel Workflow	1-3
Activity Templates	1-4
Siebel Assignment Manager	1-4
Siebel SmartScript	1-4

Chapter 2. Introduction to Siebel Workflow

Introducing Siebel Workflow	2-2
Siebel Workflow Roles	2-3
Using Siebel Workflow Designer	2-4
Understanding Workflow Processes	2-5
Workflow Processes Administration Overview	2-7

Understanding Workflow Policies 2-12
 Workflow Policies Structure 2-13
 Workflow Policies Administration Overview 2-17
Understanding State Models 2-19
 State Model Administration Overview 2-20

PART 2. Workflow Processes

Chapter 3. Planning for Workflow Processes

Gathering Information 3-2
 Researching Existing Process Information 3-2
 Researching New Processes and Areas for Improvement 3-3
Understanding Workflow Process Requirements 3-3
Planning for Workflow Processes 3-4
 Defining a Primary Component for a Business Object 3-4
 Enabling a Business Service for Workflow Processes 3-5
Defining a Test and Migration Strategy 3-6

Chapter 4. Designing Workflow Processes

Designing Workflow Processes 4-3
 Using the Workflow Processes Views 4-3
Using the All Processes View 4-5
Using the Process Designer View 4-8
Using the Process Properties View 4-11
Working with Workflow Process Definitions 4-13
 Reviewing Existing Process Definitions 4-13
 Defining a New Process 4-14
Designing a Workflow Process 4-15
Working with Process Properties 4-18
 Defining Process Properties 4-18

Working with Start Steps	4-20
Using the Start View	4-21
Defining a Start Step	4-24
Working with Decision Steps	4-28
Using the Decision Branches View	4-29
Defining a Decision Step	4-33
Working with Business Service Steps	4-37
Using the Business Service View	4-38
Defining a Business Service Step	4-42
Working with Siebel Operation Steps	4-45
Using the Siebel Operation View	4-46
Defining a Siebel Operation Step	4-48
Working with Sub Process Steps	4-52
Using the Sub Process View	4-53
Defining a Sub Process Step	4-56
Working with End Steps	4-59
Using the End Arguments View	4-60
Defining an End Step	4-61
Working with Wait Steps	4-63
Using the Wait View	4-64
Defining a Wait Step	4-65
Working with Stop Steps	4-67
Using the Stop View	4-68
Defining a Stop Step	4-71
Working with Exceptions	4-73
Using the Next Step Applet for Exceptions	4-75
Defining an Exception	4-75
Deleting a Step or Process	4-78
Copying a Step or Process	4-78
Activating a Process	4-79
Modifying a Process Definition	4-79

Chapter 5. Testing Workflow Processes

Testing Processes with the Process Simulator 5-2
 Using the Process Simulator View 5-3

Chapter 6. Running Workflow Processes

Invoking a Workflow Process 6-2
 Invoking from a Workflow Policy 6-2
 Invoking from a Script 6-4
Running a Workflow Process on the Server 6-6
Running a Workflow Process on the Object Manager 6-7
Running a Workflow Process in Batch 6-8

Chapter 7. Migrating to the Production Environment

Migrating Workflow Processes to Production 7-2
 Importing or Exporting a Process Definition 7-2

PART 3. Workflow Policies

Chapter 8. Planning for Workflow Policies

Planning Workflow Policy Groups 8-2
Planning for Workflow Policies 8-3
 Determining What to Monitor 8-4
 Planning Policies and Conditions 8-4
 Planning Workflow Policy Actions 8-5
 Planning Scenarios 8-6
Defining a Test and Migration Strategy 8-9

Chapter 9. Creating Workflow Policies

Using the Workflow Policies Views	9-2
Working with Workflow Policy Actions	9-3
Understanding the Actions Applet	9-4
Understanding the Arguments Applet	9-5
Understanding the Recipients Applet	9-13
Creating a Workflow Policy Action	9-15
Siebel Workflow Policy Action Examples	9-16
Creating a Send Page Action	9-16
Creating a Send Email Action with a Repeating Message	9-17
Creating a Send Message Broadcast Action	9-19
Creating a Database Operation Action	9-20
Creating a Run External Program Action	9-21
Working with Workflow Policy Groups	9-23
Understanding the Group Applet	9-25
Understanding the Policies Applet	9-25
Creating a Workflow Policy Group	9-26
Working with Workflow Policies	9-26
Understanding the Policies Applet	9-27
Understanding the Conditions Applet	9-29
Understanding the Actions Applet	9-34
Creating a Workflow Policy	9-35
Siebel Workflow Policy Examples	9-36
Creating a Send Page Workflow Policy	9-36
Creating a Send Email Workflow Policy	9-37

Chapter 10. Customizing Workflow Policies with Siebel Tools

Understanding Siebel Tools	10-3
Understanding Siebel Tools and Workflow Policies	10-6
Working with Workflow Policy Objects	10-9
Using the Siebel Tools Views	10-12
Understanding the Workflow Policy Column List View	10-13
Understanding the Workflow Policy Object List View	10-15
Understanding the Workflow Policy Component List View	10-16
Understanding the Workflow Policy Component Columns View	10-19
Defining a Workflow Policy Column	10-21
Defining a Workflow Policy Component	10-22
Defining a Workflow Policy Object	10-23
Modifying Policy Column Names	10-24
Adding Policy Columns to a Workflow Policy Object	10-24
Associating a Column with a Workflow Policy Component	10-26
Using the Validate Tool	10-27
Modifying an Existing Workflow Policy Object	10-29
Understanding Workflow Policy Programs	10-33
Using the Program List View	10-34
Using the Workflow Policy Program Argument List View	10-36
Defining a Workflow Policy Program	10-41
Example of Creating a Program Argument	10-43
Creating SQL Statements for Program Arguments	10-44
Using Predefined Workflow Policy Programs	10-44
Using Change SR Close Date to Today	10-45
Using Change SR Owner	10-47
Using Change SR Owner to Manager	10-49
Using Send Quote Page	10-51
Making Object Types Available in Siebel Client	10-53

Chapter 11. Workflow Policies Server Administration

Verifying Workflow Policies Installation	11-2
Verifying the Repository Setting	11-2
Verifying the License Key	11-2
Creating Database Triggers	11-4
Database Triggers and Database Administration	11-5
Running Generate Triggers	11-5
Running the SQL Script File	11-7
Database Triggers and Remote Users	11-8
Setting Up the Siebel Server for Email Manager	11-8
Setting Up the Siebel Server for Page Manager	11-14
Troubleshooting the Email and Page Managers	11-18
Starting Workflow Monitor Agent	11-19
Using the Workflow Monitor Agent	11-20
Using Workflow Action Agent	11-25
Starting Workflow Agent Processes Automatically with Siebel Server	11-26
Using Workflow Policies and Siebel Server Task Trace Files	11-27
Viewing Trace Files in Siebel Server Administration	11-28
Viewing Trace Files in the Siebel Server Log Directory	11-29
Using Workflow Policies Analysis Charts and Reports	11-31
Using the Policy Frequency Analysis Chart	11-31
Using the Policy Trend Analysis Chart	11-31
Using Workflow Policies Reports	11-31

Chapter 12. Workflow Policies and Siebel Marketing

Using Workflow Policy Programs for Campaign Execution	12-2
Using Send Campaign Email	12-2
Using Create Email Activity	12-3
Using Assign to Campaign	12-3
Understanding a Campaign Workflow Scenario	12-4

Chapter 13. Testing Workflow Policies

Overview 13-2
Testing New Policies and Monitoring the Results 13-3
Troubleshooting 13-5

Chapter 14. Migrating Policies to the Production Environment

Migrating Workflow Policies 14-1

Chapter 15. Monitoring and Performance Tuning

Monitoring Workflow Policies 15-2
 Using the Workflow Policies Log View 15-3
 Understanding the S_ESCL_REQ Table 15-5
 Understanding the S_ESCL_STATE Table 15-6
 Understanding the S_ESCL_ACTN_REQ Table 15-7
Tuning Workflow Policies for Performance 15-8
 Workflow Policy Groups to Manage Server Load 15-8
 Multiple Workflow Monitor Agents and Workflow Action Agents 15-8
 Workflow Agents on Multiple Application Servers 15-8
 Running Workflow Agent Processes through Server Manager (svrvmgr) . . 15-9
 Optimal Sleep Interval for Workflow Policy Groups 15-9
 Optimal Action Interval for Each Workflow Action Agent 15-10
 Creating Multiple Workflow Agent Processes 15-10
 Performance Tuning Tips 15-11

PART 4. State Models

Chapter 16. Using State Models

State Model Overview	16-2
Creating State Models and State Transitions	16-5
Creating State Models	16-5
Creating State Transitions	16-10
Using a State Model	16-13

PART 5. Appendixes

Appendix A. Predefined Business Services

Appendix B. Predefined Programs

Appendix C. Siebel Workflow Terminology

Index

Introduction

Who Should Use This Guide	Intro-2
How This Guide Is Organized	Intro-3
What's New in This Release	Intro-5
Additional Documentation	Intro-5
Contacting Siebel Technical Support	Intro-7
Siebel Systems Welcomes Your Comments	Intro-8

Who Should Use This Guide

This guide provides the information you need to configure, implement, test, and monitor Siebel Workflow. It also includes information to help you verify your installation.

Although job titles and duties at your company may differ from those listed in the following table, the audience for this guide consists of employees in the following categories:

Siebel Application Administrators Persons responsible for planning, setting up, and maintaining Siebel applications.

For Siebel Workflow, the Siebel Application Administrator is the department, service, or sales manager responsible for defining and changing business processes.

Siebel Application Developers Persons who plan, implement, and configure Siebel applications, possibly adding new functionality.

For Siebel Workflow, the Siebel Application Developer is responsible for configuring Workflow elements using Siebel Tools and the Siebel client.

Siebel System Administrators Persons responsible for the whole system, including installing, maintaining, and upgrading Siebel applications.

For Siebel Workflow, the Siebel System Administrator is responsible for installing Siebel Server, managing Siebel Workflow, and creating database triggers.

This guide assumes you are familiar with Siebel Tools and the concepts related to the Siebel suite of products, such as business objects, business components, and so on.

How This Guide Is Organized

This guide provides information necessary to plan, implement, and monitor Siebel Workflow effectively in a business environment. The following table shows a summary of the information contained in each chapter:

Table 1. Chapter Summary (1 of 2)

Part	Chapter / Appendix	Summary of Contents
	Introduction	Provides an overview of the Siebel Workflow Administration Guide for Siebel 2000
Part 1: Concepts and Overview	Chapter 1	Provides an overview of how you can use Siebel Workflow to automate the enforcement of your business policies
	Chapter 2	Provides an overview of how Siebel Workflow works, including the architecture, concepts, roles, and the relationship between Siebel Tools and Siebel Workflow
Part 2: Workflow Processes	Chapter 3	Provides a recommended sequence of tasks to plan for Workflow Processes
	Chapter 4	Describes how to use the Siebel Client to create Workflow Processes
	Chapter 5	Describes how to test Workflow Processes with the Process Simulator
	Chapter 6	Describes the various methods of invoking and running a Workflow Process
	Chapter 7	Describes how to migrate Workflow Processes from a test environment to a production environment
Part 3: Workflow Policies	Chapter 8	Provides a recommended sequence of tasks to properly plan Siebel Workflow Policies implementation
	Chapter 9	Describes how to use the Siebel Client to create Workflow Policies and Workflow Actions
	Chapter 10	Describes how to create and modify Workflow Objects and Programs

Table 1. Chapter Summary (2 of 2)

Part	Chapter / Appendix	Summary of Contents
	Chapter 11	Describes how to create database triggers, start email and pager processes, and Workflow Agent processes
	Chapter 12	Describes how Workflow Policies and Siebel Marketing work together to automate campaign execution
	Chapter 13	Describes how to test your Workflow Policies before implementing them in a production environment
	Chapter 14	Describes the migration process from the test to the production environment
	Chapter 15	Describes how to monitor Workflow Policies so you can tune your system for optimal performance
Part 4: State Machine	Chapter 16	Describes how to use the State Machine to manage state models and state transitions
Part 5: Appendixes	Appendix A	Provides a list of predefined Siebel Business Servers
	Appendix B	Provides a list of predefined programs for use with Siebel Workflow Policies
	Appendix C	Provides terminology related to Siebel Workflow

What's New in This Release

In this release, Siebel has extended the Siebel Workflow product to include Workflow Processes. Workflow Processes allow you to:

- Define a business process using a graphical designer interface.
- Use Business Services as steps in workflow processes.
- Support complex conditions in process decision points.
- Test workflow process definitions using a graphical process simulator.
- Base workflow processes on Business Objects.

NOTE: Your Siebel implementation may not have all the features described in this guide depending on which software modules you have purchased.

Additional Documentation

The following documents also provide information on the topics addressed in this guide.

Siebel Server Administration Guide

Siebel Applications Administration Guide

Siebel Installation Guide

Siebel Tools Guide

Siebel Release Notes

Siebel Data Model Reference

Siebel SmartScript Guide

Siebel eBusiness Application Integration Guide

Siebel Object Types Reference

Siebel Communications Server Administration Guide

This guide does not provide information about general software concepts, such as records and queries, or about using Microsoft Windows. Nor does it provide instructions for basic navigation of Siebel applications. For this kind of information about Siebel applications, see *Siebel Basics*.

Administrators and developers, such as marketing administrators, call center administrators, and application developers, should read *Siebel Applications Administration Guide* for information on how to set up and maintain Siebel applications features.

You can find information about Siebel Technical and Professional Services in *Guide to Siebel Global Services*.

For copies of these documents, please use the Siebel Books Online Web site (<http://ebusiness.siebel.com/booksonline>). Through Siebel Books Online, you can order additional Siebel documentation and copies of the *Siebel Bookshelf for Enterprise Applications* CD-ROM.

For the most current and accurate documentation, see the Documentation Updates section of the Siebel SupportWeb site (<http://ebusiness.siebel.com/supportweb/>). The Documentation Updates show changes that have been made to the documentation since it was released. For up-to-the-minute information on Siebel products, see *Siebel Release Notes*. *Siebel Release Notes* appear on the Siebel SupportWeb site.

Siebel Systems reserves the right to modify the documentation at any time by posting revised documentation on the Siebel SupportWeb site.

Contacting Siebel Technical Support

Do you know how to access Siebel Technical Support? It is crucial that you understand the requirements for getting support before you encounter technical issues that require Siebel Technical Support's assistance. This will facilitate smooth resolution of your issues. If you have questions, please don't hesitate to contact us.

To maximize your knowledge of Siebel products and your return on investment:

- You must attend Siebel training to become a *designated contact*.
- Your Siebel-trained designated contacts provide technical support to your users. Siebel Technical Support provides support directly to your designated contacts only.

To provide efficient, timely support and access to the Technical Support knowledge base:

- Siebel Technical Support is primarily Web-based; it can be accessed through Siebel SupportWeb (<http://ebusiness.siebel.com/supportweb/>). Please submit new service requests to us through SupportWeb, where you can also search the knowledge base for solutions.
- Designated contacts receive read/write access to Siebel SupportWeb. All other project team members at your company receive read-only accounts so that they can access the knowledge base.

To register for Siebel training, access <http://siebeluniversity.siebel.com/edPortal> and choose Implementation Team Training.

Please submit your technical issues and updates to Siebel SupportWeb (<http://ebusiness.siebel.com/supportweb/>). If you do not have a SupportWeb account, or if you have a question, please contact us at support@siebel.com or call your local Siebel Support Center:

- **North America:** + 800 214 0400 or + 1 650 295 5724
- **Brazil (São Paulo):** + 55 11 5110 0800
- **UK (London):** + 44 800 072 6787 or + 44 1784 494949
- **Germany (Munich):** + 49 89 957 18 400

- **France (Paris):** + 44 800 072 6787 or + 44 1784 494949
- **Ireland (Galway):** + 44 800 072 6787 or + 44 1784 494949
- **Japan (Tokyo):** 0120 606 750 (Japan domestic only),
+ 81 3 5469 3811 (outside of Japan)
- **Singapore:** + 65 212 9266

Outside of local support center hours, Gold and Rollout Support Option customers can call +1 800 214 0400 or +1 650 295 5724.

We appreciate your business and look forward to working with you.

Siebel Systems Welcomes Your Comments

To help us improve our products, we want to know about any corrections or clarifications to this guide that you would find useful. Please include in your message:

- The title and version of the guide (very important)
- The name and version number of the Siebel application you are using
- Your name, job title or functional area, company name, phone number, and email address

Contact us through regular mail or email at:

Siebel Systems, Inc.
Technical Publications Department
2207 Bridgepointe Parkway
San Mateo, CA 94404-5009

doc@siebel.com

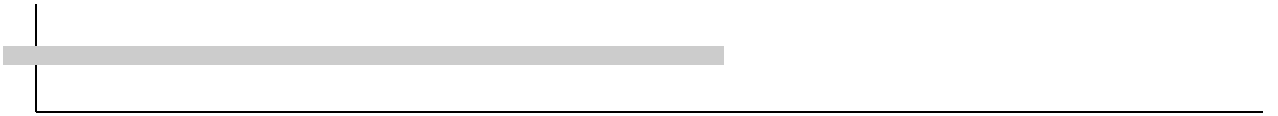
We appreciate your feedback.

Concepts and Overview

Part 1

Chapter 1. Overview of Workflow

Chapter 2. Introduction to Siebel Workflow



Overview of Workflow

1

General Principles of Workflow	1-2
Siebel Workflow Solutions	1-3
Siebel Workflow	1-3
Activity Templates	1-4
Siebel Assignment Manager	1-4
Siebel SmartScript	1-4

General Principles of Workflow

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

- Ensuring 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 to the dynamic nature of the information being monitored to enforce the processes.

The management of important events is central to the enforcement of business workflow. *Workflow* is the timely management of an event to ensure 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 ensure 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/her pipeline to make sure 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.

Siebel Workflow Solutions

Managing workflow in your organization 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

Siebel provides a comprehensive set of solutions through various software modules to ensure that Siebel applications can address all of your workflow requirements. These modules are:

- Siebel Workflow
- Activity Templates
- Siebel Assignment Manager
- Siebel SmartScript

Siebel Workflow

Siebel Workflow, the focus of this guide, is a customizable business application that allows you to define, manage, and enforce your business processes. It allows you to design complex workflow processes and automate the enforcement of business policies and procedures.

This application features the following three modules:

- **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, sub-processes, 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.
- **State Models.** Used for defining business object states and state transitions.

Activity Templates

This module allows you to automatically generate activities for processing work. You have the ability to generate activities for whatever function you are currently performing. Activities can automatically be generated for an account, contact, opportunity, or service request. Activity templates allow administrators to define standard processes for managing an opportunity or handling a service request.

You use activity templates primarily for processing work. You can create activities with dynamic due dates and assign an activity to any user on the system. See *Siebel Applications Administration Guide* for more information.

Siebel Assignment Manager

This module allows you to route and assign work. It allows sales and service organizations to assign the most qualified people to specific tasks effectively. You use Assignment Manager to match candidates to predefined and customizable assignment objects. To assign the most qualified candidate to each object, Assignment Manager applies assignment rules. For example, in a sales organization, you may create an assignment rule that scores positions based on territory definitions for an opportunity. In a service organization, you may create an assignment rule that scores employees based on product expertise for a service request. Using the sum of scores for each assignment rule, Assignment Manager assigns the best candidate for each object.

After work is assigned to an employee, it shows up in his/her personal queue. See *Siebel Assignment Manager Administration Guide* for more information.

Siebel SmartScript

This module allows business analysts, call-center managers, and Siebel developers to define the application workflow for an interactive situation in a script. These situations include both inbound communications, such as loan processing or customer service, and outbound contacts, such as telemarketing. The flow of the interaction is controlled entirely by the script, not by the agent. See *Siebel SmartScript Guide* for more information.

Introduction to Siebel Workflow

2

Introducing Siebel Workflow	2-2
Siebel Workflow Roles	2-3
Using Siebel Workflow Designer	2-4
Understanding Workflow Processes	2-5
Workflow Processes Administration Overview	2-7
Understanding Workflow Policies	2-12
Workflow Policies Structure	2-13
Workflow Policies Administration Overview	2-17
Understanding State Models	2-19
State Model Administration Overview	2-20

Introducing Siebel Workflow

Siebel Workflow is a powerful interactive software tool that lets you automate how your organization handles workflow processes. It uses as its basic model the processes 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 Workflow Designer, a graphical user interface featuring a familiar flowcharting methodology for designing workflow processes.

Siebel Workflow works with all Siebel eBusiness Applications and involves the following architectural components:

- **Siebel Tools.** Siebel Tools is a software application that allows you to reconfigure and extend the functionality of Siebel Workflow.
- **Siebel Client.** Workflow Processes, Workflow Policies, and State Model are administered through Siebel Workflow Designer views in the Siebel 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.

For a complete description of the Siebel architecture, please refer to *Siebel System Administration Guide*.

This guide describes how the members of the implementation team in your organization can plan, implement, and manage workflow for your business using the Workflow Processes, Workflow Policies, and State Models modules.

Siebel Workflow Roles

Siebel Workflow requires a number of different employee roles to plan, implement, and manage the system. You can customize these roles and have one or more persons perform them depending on your business environment. When implementing Siebel Workflow, you may want to work as part of the Siebel eBusiness Implementation Team, or you may want to create a separate implementation team for Siebel Workflow.

The roles that configure and manage Siebel Workflow are summarized as follows:

- The *Workflow Configurator* uses Siebel Tools 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 *Siebel eBusiness Application Integration Guide*.

- The *Workflow Administrator* or *Business Analyst* defines or changes workflow processes and policies in the Siebel client using Siebel Workflow Designer. This person is typically a department manager, service manager, marketing manager, or sales manager.
- The *System Administrator* activates workflow policies by generating database triggers in a script and creating them in the Siebel database. The System Administrator also 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.
- An *End User* uses the system and executes workflow processes, policies, and state models.

Using Siebel Workflow Designer

Siebel Workflow Designer is the administrative interface for Siebel Workflow. This interface brings together the three modules—Workflow Processes, Workflow Policies, and State Model—for creating a comprehensive workflow design.

Figure 2-1 shows an example of the Siebel Workflow Designer interface. The navigation bar on the left portion of the screen provides access to the views for the three modules.

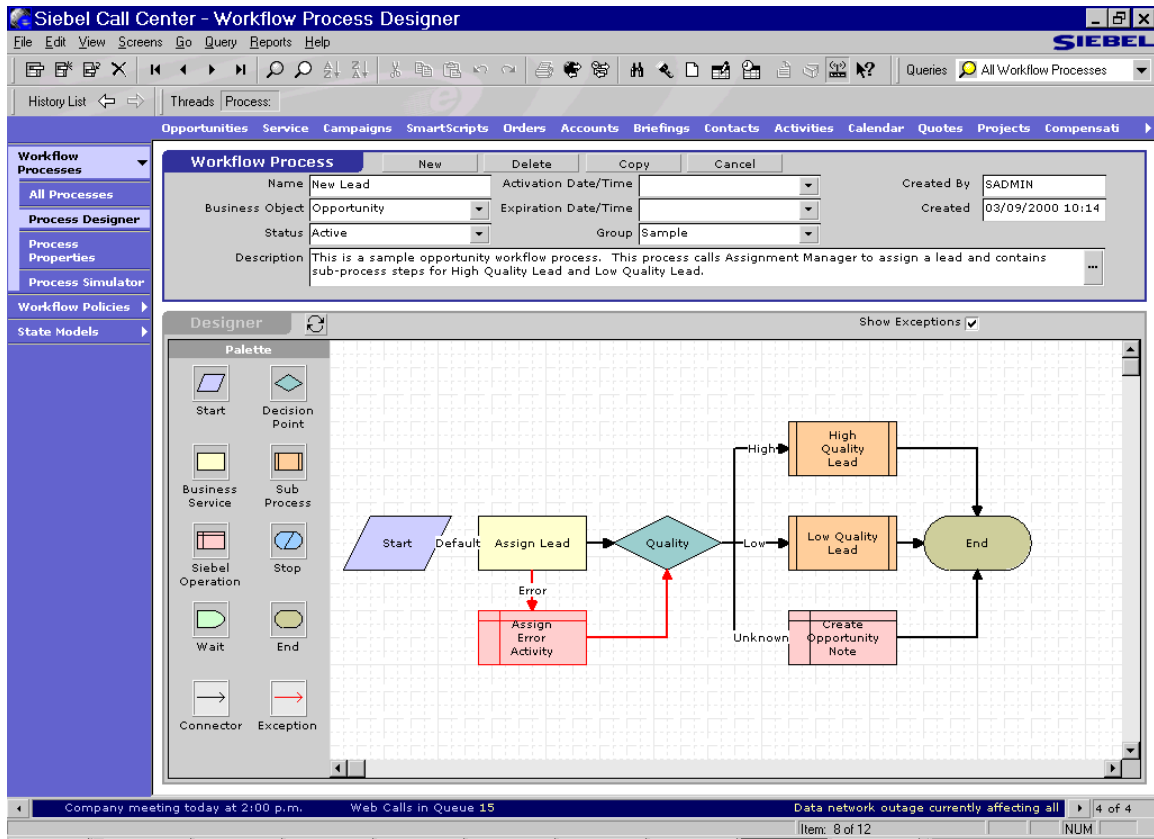


Figure 2-1. Siebel Workflow Designer View

Understanding Workflow Processes

Workflow Processes is the module you use to create workflow processes. It allows you to define your company's business processes using the Siebel Workflow Designer interface on the Siebel client. Using Workflow Processes, you can define a process that consists of one or more process steps such as start steps, decision points, business services, sub-processes, or tasks.

A task can be a predefined or 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.

The Workflow Processes module consists of a series of views for designing the flow of a process and then defining each step in the process.

[Figure 2-2 on page 2-6](#) shows an example of one of the Workflow Processes views.

Introduction to Siebel Workflow

Understanding Workflow Processes

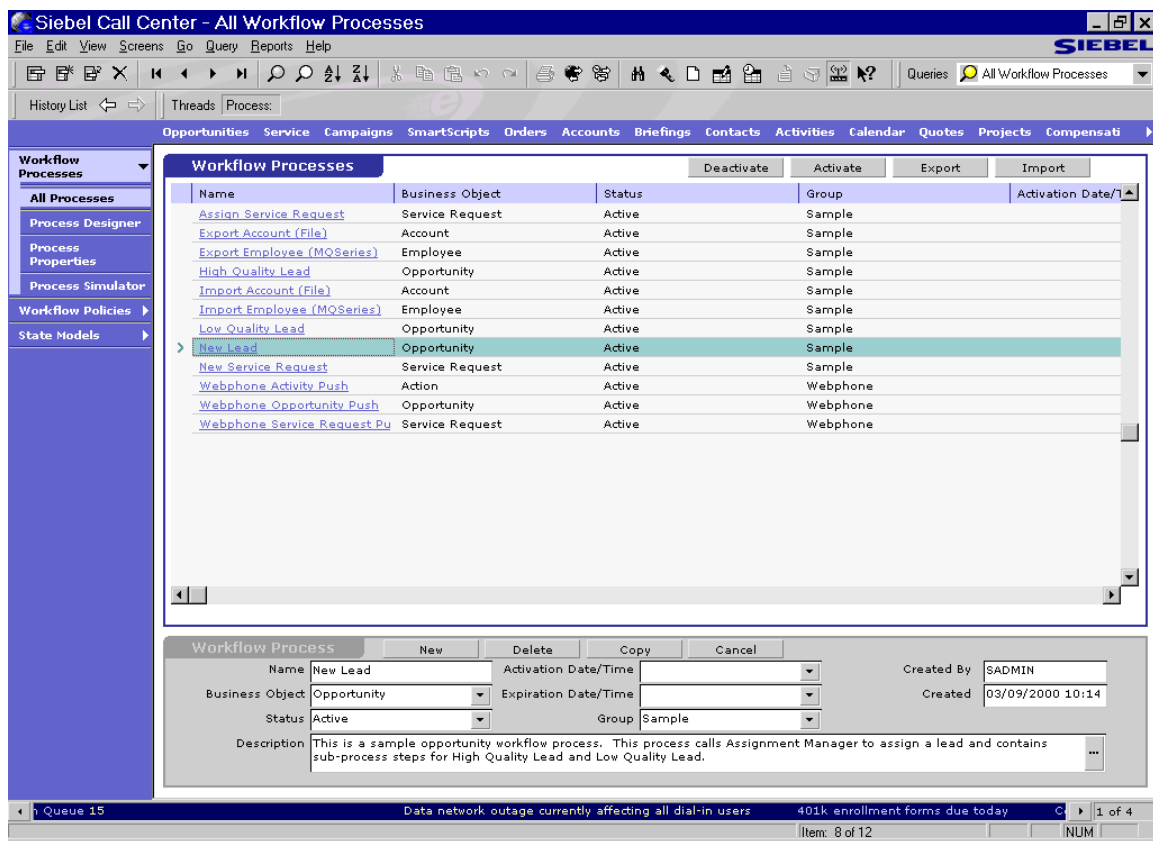


Figure 2-2. Workflow Processes View

Workflow Processes Administration Overview

Usage of workflow 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.

To help you understand how Workflow Processes works, review the usage scenarios “[Scenario 1: New Service Request](#)” on page 2-8 and “[Scenario 2: New Lead](#)” on page 2-10. You can view the example scenarios in detail by connecting to the sample database installed during your Siebel Client installation.

To access the sample database

- 1 Launch Siebel Call Center.
- 2 Enter sadmin/sadmin as the username and password.
- 3 Select Sample in the Connect to dialog box.
- 4 Navigate to Screens → Siebel Workflow Administration → Workflow Processes → All Processes.

Workflow Processes Views

Workflow Processes are administered through the Siebel Workflow Designer on Siebel client. Instructions for accessing and using the Workflow Processes views are in [Chapter 4, “Designing 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 insert or update on a business component, a user interface event (such as a button click), or a server component.

From an external system, processes can be invoked using COM or CORBA. Procedures for invoking a workflow process are in [Chapter 6, “Running Workflow Processes.”](#)

Scenario 1: New Service Request

ABC Computing defines its business process for a new service request with Workflow Processes. Figure 2-3 illustrates a diagram of the process as drawn in the Process Designer view.

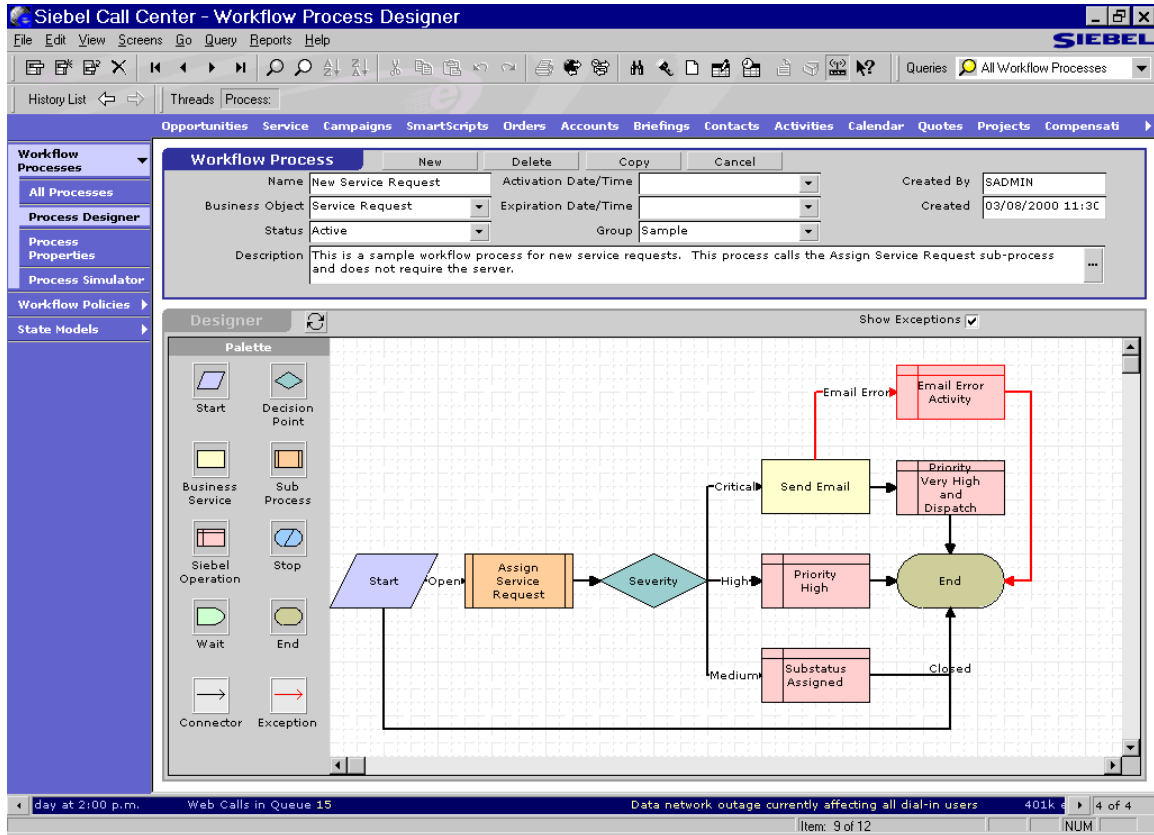


Figure 2-3. Service Request Diagram

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. 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 sub process task. The service request is assigned to the appropriate agent based on the assignment rules defined in the Assign Service Request sub process.
- **Severity.** This is a decision step. The service request priority 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.

Scenario 2: New Lead

WYZ Computing defines its business process for creating a new lead with Workflow Processes. [Figure 2-4](#) illustrates a diagram of the process as drawn in the Process Designer view. The process calls Assignment Manager to assign a lead and contains sub-process steps for High Quality Leads and Low Quality Leads.

Introduction to Siebel Workflow

Understanding Workflow Processes

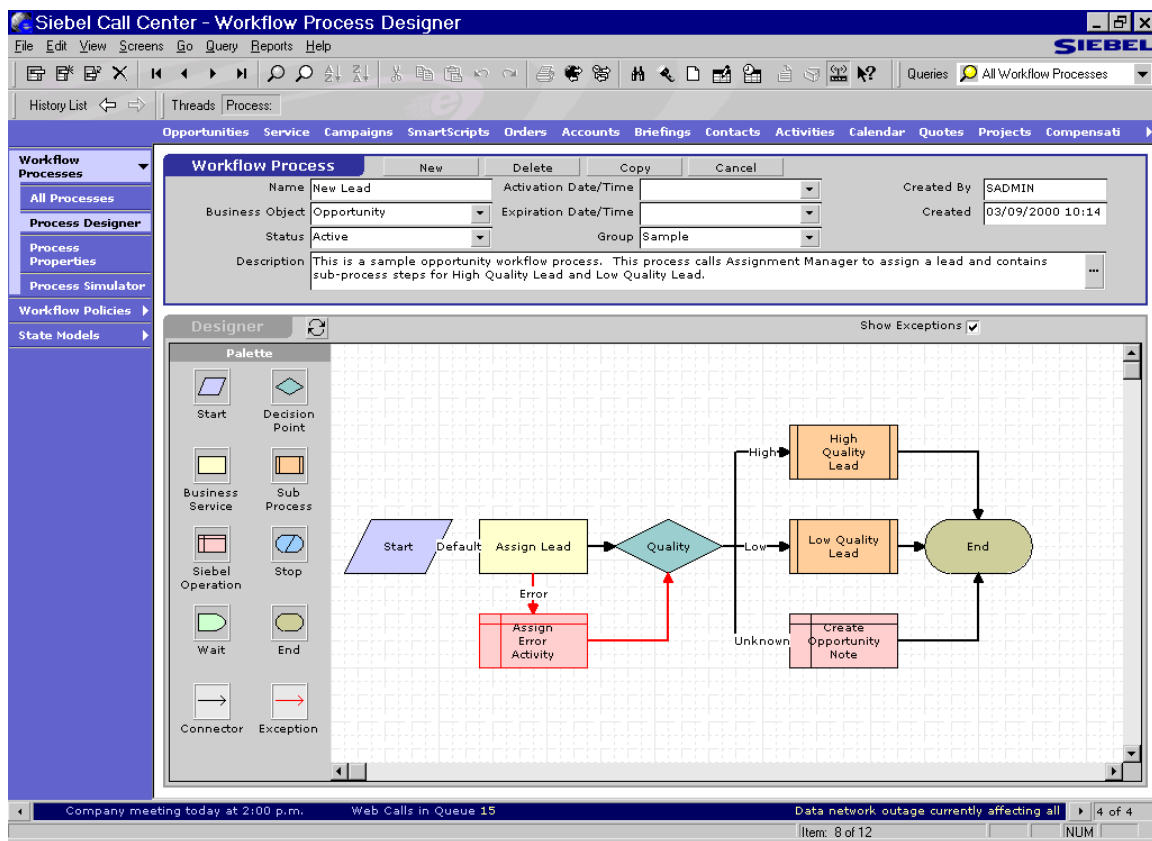


Figure 2-4. New Lead Diagram

The steps in this diagram are interpreted as follows:

- **Start.** This is the start step that initiates the process instance.
- **Assign Lead.** This is a business service task. It calls the Assignment Manager server component to assign the leads using Assignment Manager rules.
- **Assign Error Activity.** This is a Siebel Operation insert task that is triggered if an error is returned from the Assign Lead step.
- **Quality.** This is a decision point. The quality of the lead determines the path that the work item takes.
- **High Quality Lead.** This is a sub-process step that calls the High Quality Lead workflow process.
- **Low Quality Lead.** This is a sub-process step that calls the Low Quality Lead workflow process.
- **Create Opportunity Note.** This is a Siebel Operation insert task that inserts a note in the Opportunity business object.
- **End.** This step defines the completion of the process.

Understanding Workflow Policies

Workflow Policies is the module within the Siebel Workflow Designer that you use to create workflow policies. A policy consists of one or more policy conditions actions. When the policy conditions are met, the policy action is executed.

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

NOTE: A number of the functions available with Workflow Policies can be supported using Workflow Processes in Siebel 2000. It is recommended that workflow policies be used to define conditions for invoking workflow processes. Workflow processes should be used for defining the actions.

NOTE: In Siebel 2000 the name Workflow Policies replaces the name Policy Manager, which was used to refer to the Siebel workflow automation tool prior to the release of Siebel 2000.

Workflow Policies Structure

The basic underlying construct of workflow policies is the “rule.” The structure of this rule is, if all conditions are true, then an event 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 as shown in [Figure 2-5](#).

Rule

If	Condition	Is true
Then	Action	

A rule contains a workflow condition and a workflow action. If the condition is true, the action occurs.

Condition

If	Condition Clause	Is true
And if	Condition Clause	Is true

A workflow condition may contain one or more condition clauses. The clauses are joined by “and.” You must create multiple policies if you are constructing a join. When all the conditions are true, then the actions occur.

Condition Clause

Object	Attribute	Compares with	Value
--------	-----------	---------------	-------

The value of the object attribute is compared with the value specified in the condition clause. The result of the comparison is true or false. When the result is true, the conditions of the policy are met.

Action

Action Clause	Invoke Workflow Process Send Email Send Page
Action Clause	Send Message Broadcast Database Operation Run External Program

An action may contain one or more action clauses. When all policy conditions are met, all the actions occur.

Figure 2-5. Workflow Policies Rule Structure

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 circumstance or situation that causes something to happen, a triggering event. A *workflow policy action* is an event 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 (True) 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
```

SiebelWorkflow Policies supports only AND linkages between policy conditions. 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 9-30.

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 also reuse actions in multiple workflow policies. See [Chapter 10, “Customizing Workflow Policies with Siebel Tools,”](#) for a discussion of actions and their parameters.

[Figure 2-6](#) illustrates the key parts of a workflow policy.

Workflow Policy		Duration: 2 Hours	
Workflow Condition			
Service Request Severity	=	1-Critical	
Service Request Status	=	Open	
Workflow Action			
	Action		Recipient
	Send Urgent Page	=	Owner's Mgr
	Send Alert E-mail	=	Director

Figure 2-6. Key Parts of a Workflow Policy

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 are generic events and can be one of the following types:

- **Run Workflow Process.** This program invokes a workflow process.
- **Send Message.** This program sends an email to one or more recipients.
- **Send Page.** This program sends a page to one or more recipients.
- **Send Message Broadcast.** This program inserts a message broadcast for one or more recipients.
- **Database Operation.** This program either inserts or updates the data records of a Siebel database table for selected workflow policy components.
- **Run External Program.** This program allows you to run an executable.

You can use programs in multiple action definitions and you can use action definitions in multiple workflow policies. [Appendix B, “Predefined Programs,”](#) contains a list of the predefined programs.

Table 2-1 illustrates how you can use a pre-configured program such as Send SR Email to create an action. In the example below, a workflow policy action named Send High Severity SR Email is being designed to use the Send SR Email program. This program is one of the Send Message type programs. The Send SR Email program allows you to use drag-and-drop substitutions to build the text of the email message. These substitutions are SR Num, Account, and Severity. Once you create an action, you can use it in any number of workflow policies.

Table 2-1. Using a Preconfigured Program to Create a Workflow Action

Program Type	Program Name	Workflow Action Name	Substitutions	Workflow Action Email Template
Send Message	Send SR Email	Send High Severity SR Email	SR Num Account Severity	The [SR Num] from [Account] has a severity level [Severity] and has been open for two hours.

Workflow Policy Groups

Workflow policies are organized into workflow groups. A *workflow group* is a collection of workflow policies to facilitate load balancing on the servers. Workflow groups allow you to manage and optimize Siebel 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 Workflow Designer, and policy execution by the Siebel Server Workflow Components.

Workflow Policies Views

Workflow Policies is administered through the Siebel Workflow Designer in the Siebel Client. Instructions for accessing and using the Workflow Policies views are in [Chapter 10, “Customizing Workflow Policies with Siebel Tools.”](#)

Figure 2-7 shows an example of one of the Workflow Policies views.

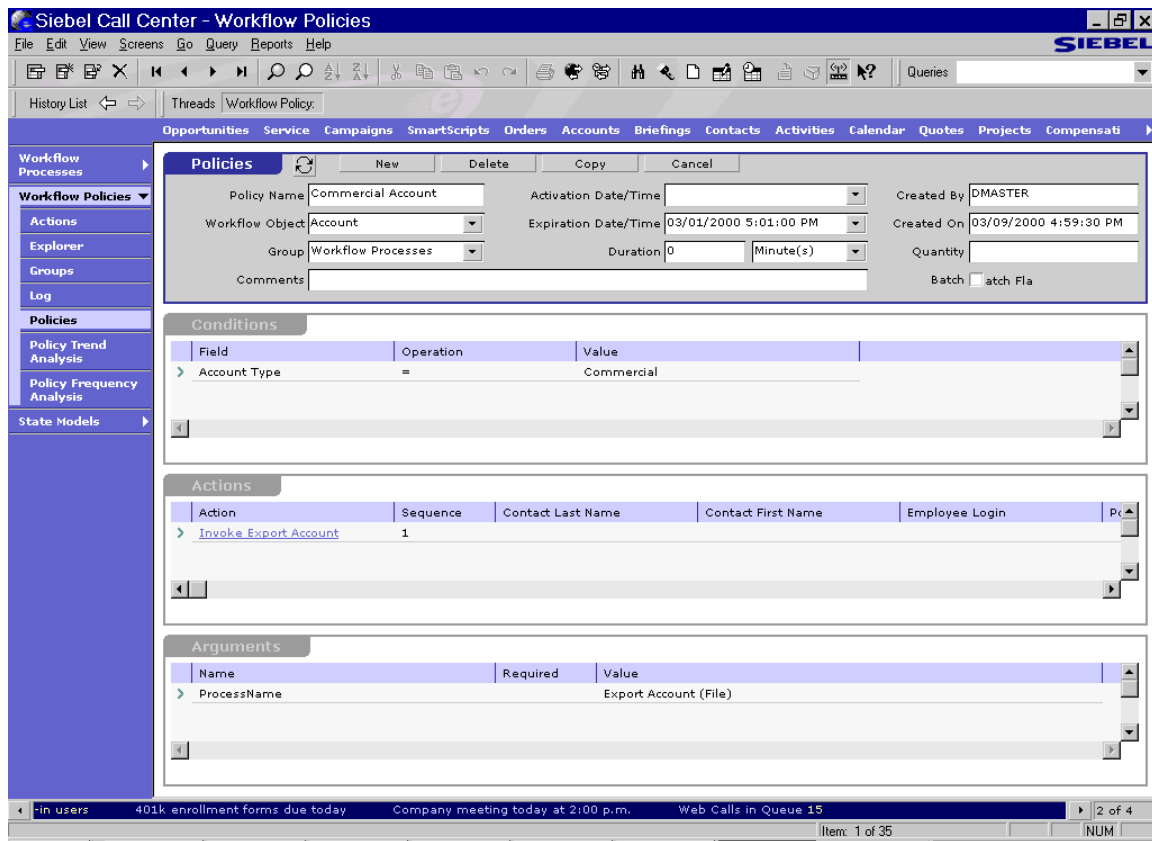


Figure 2-7. Workflow Policies View

Understanding State Models

The State Models module provides a data-driven method for extending workflow control based on the status of an object such as a service request or a product defect. A state represents the status of an object, such as Open, Closed, or Pending.

You use State Models primarily to enforce authorization and transition rules. A state model restricts who can change a given attribute on an object, such as the status on a service request and the conditions that must be met to make the change.

State Model is the blueprint of acceptable states and state transitions that the state machine enforces. The state machine then makes sure that these objects go through the desired process defined in the state model.

The *state machine* is the engine that enforces the transitions between states for an object during its lifetime. The state represents where the object is in its lifetime. The state can also control whether or not the data of that object can be modified. As an example, a service request that is in a Closed state may be considered “frozen,” such that its attributes cannot be modified.

A *state transition* defines the allowable migration of an object from one state to the next. For instance, a service request that has been closed but must be re-opened may go from the Closed state to an Open state, and may go from Open to Pending, but may not transition directly from Closed to Pending. The allowable migration of a service request from Closed to Open, or Open to Pending, represents defined state transitions.

State Model Administration Overview

State Model is administered through the Siebel Workflow Designer on the Siebel client. Instructions for accessing and using the State Model views are in [Chapter 16, “Using State Models.”](#)

Figure 2-8 shows an example of one of the State Model views.

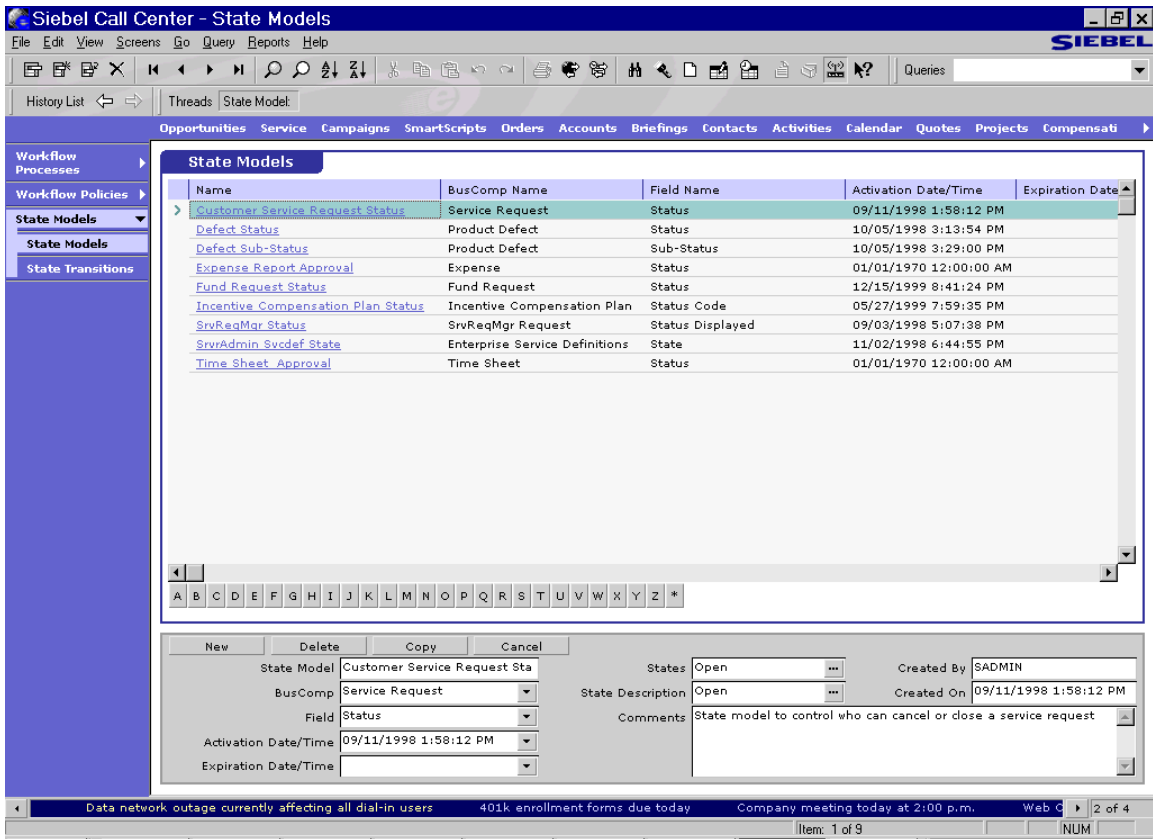


Figure 2-8. State Model View

Workflow Processes

Part 2

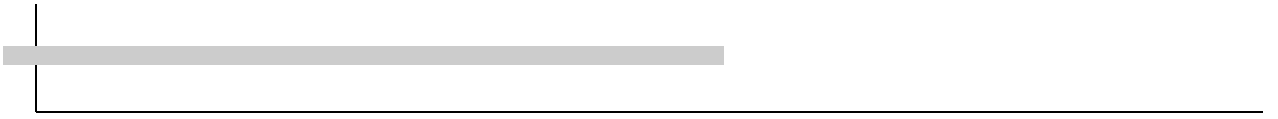
Chapter 3. Planning for Workflow Processes

Chapter 4. Designing Workflow Processes

Chapter 5. Testing Workflow Processes

Chapter 6. Running Workflow Processes

Chapter 7. Migrating to the Production Environment



Planning for Workflow Processes

3

- Gathering Information 3-2
 - Researching Existing Process Information 3-2
 - Researching New Processes and Areas for Improvement 3-3
- Understanding Workflow Process Requirements 3-3
- Planning for Workflow Processes 3-4
 - Defining a Primary Component for a Business Object 3-4
 - Enabling a Business Service for Workflow Processes 3-5
- Defining a Test and Migration Strategy 3-6

Gathering Information

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 in Siebel Workflow Processes.

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. Each workflow process should be 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 Manager 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 *Siebel Tools Guide*.

Planning for Workflow Processes

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

- Each workflow process should be associated with a business object that has a primary component defined in Siebel Tools. For more information, see [“Defining a Primary 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 *Siebel Tools Guide* for information about defining a business service with Siebel Tools.

Defining a Primary 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 business object picklist on the Workflow Processes views.

Enabling a Business Service for Workflow Processes

Siebel provides a number of predefined business services. (See [Appendix A, “Predefined Business Services,”](#) for a list of these services.) You can also define your own custom business services using Siebel Tools or Siebel Administration views.

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 on 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.

NOTE: When using the HTTP Transport business service, it should be invoked from Business Integration Manager, not Workflow Process Manager. For more information, see *Siebel eBusiness Application Integration Guide*.

Defining a Test and Migration Strategy

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 [Chapter 5, “Testing Workflow Processes,”](#) and [Chapter 7, “Migrating to the Production Environment.”](#)

Designing Workflow Processes

4

Designing Workflow Processes	4-3
Using the Workflow Processes Views	4-3
Using the All Processes View	4-5
Using the Process Designer View	4-8
Using the Process Properties View	4-11
Working with Workflow Process Definitions	4-13
Reviewing Existing Process Definitions	4-13
Defining a New Process	4-14
Designing a Workflow Process	4-15
Working with Process Properties	4-18
Defining Process Properties	4-18
Working with Start Steps	4-20
Using the Start View	4-21
Defining a Start Step	4-24
Working with Decision Steps	4-28
Using the Decision Branches View	4-29
Defining a Decision Step	4-33
Working with Business Service Steps	4-37
Using the Business Service View	4-38
Defining a Business Service Step	4-42
Working with Siebel Operation Steps	4-45
Using the Siebel Operation View	4-46

Defining a Siebel Operation Step	4-48
Working with Sub Process Steps	4-52
Using the Sub Process View	4-53
Defining a Sub Process Step	4-56
Working with End Steps	4-59
Using the End Arguments View	4-60
Defining an End Step	4-61
Working with Wait Steps	4-63
Using the Wait View	4-64
Defining a Wait Step	4-65
Working with Stop Steps	4-67
Using the Stop View	4-68
Defining a Stop Step	4-71
Working with Exceptions	4-73
Using the Next Step Applet for Exceptions	4-75
Defining an Exception	4-75
Deleting a Step or Process	4-78
Copying a Step or Process	4-78
Activating a Process	4-79
Modifying a Process Definition	4-79

Designing Workflow Processes

This chapter provides details and procedures for designing workflow processes in the Siebel Workflow Designer with the Workflow Processes views. The basic order of design is:

- Review existing process definitions.
- Define the process.
- Diagram the process.
- Enter details for each step in the process.
- Test the process.
- Implement the method for invoking the process.
- Migrate the process to the production environment.

Using the Workflow Processes Views

Workflow Processes views are part of the Siebel Workflow Designer interface on the Siebel Client.

The views available from the navigation bar on the left of the screen are the top layer of views. These are the main process definition views. To access these views, navigate to Screens → Siebel Workflow Administration → Workflow Processes. A second layer of views is available for defining the process step details. The top layer of views includes:

- **All Processes view.** For reviewing existing business processes. See [“Using the All Processes View” on page 4-5](#) for a description of this view.
- **Process Designer view.** For designing the process flow diagram. See [“Using the Process Designer View” on page 4-8](#) for a description of this view.
- **Process Properties view.** For defining the properties for a process. See [“Using the Process Properties View” on page 4-11](#) for a description of this view.
- **Process Simulator view.** For testing the functionality of the processes you create. It is similar to a debugging application. See [“Using the Process Simulator View” on page 5-3](#) for a description of this view.

The second layer of views is accessed by double-clicking the appropriate step icon in the flowcharted process on the Process Designer view. The second layer of views includes:

- **Business Service view.** This is where you define process steps that call a business service. See [“Using the Business Service View” on page 4-38](#) for a description of this view.
- **Decision view.** This is where you define a decision point step, including decision branches, conditions, and values. See [“Using the Decision Branches View” on page 4-29](#) for a description of this view.
- **End view.** This is where you define the ending process step. See [“Using the End Arguments View” on page 4-60](#) for a description of this view.
- **Siebel Operation view.** This is where you define an insert or update to a Siebel database record. See [“Using the Siebel Operation View” on page 4-46](#) for a description of this view.
- **Start view.** This is where you identify the conditions that must be met for the process to be initiated. See [“Using the Start View” on page 4-21](#) for a description of this view.
- **Stop view.** This is where you define the input and output arguments that result in a process termination. See [“Using the Stop View” on page 4-68](#) for a description of this view.
- **Sub-Process view.** This is where you define sub-process steps, including input and output arguments. See [“Using the Sub Process View” on page 4-53](#) for a description of this view.
- **Wait view.** This is where you define a pause in the workflow process execution. See [“Working with Wait Steps” on page 4-63](#) for a description of this view.

Using the All Processes View

This view displays a list of the currently defined workflow processes.

Figure 4-1 illustrates the All Processes view.

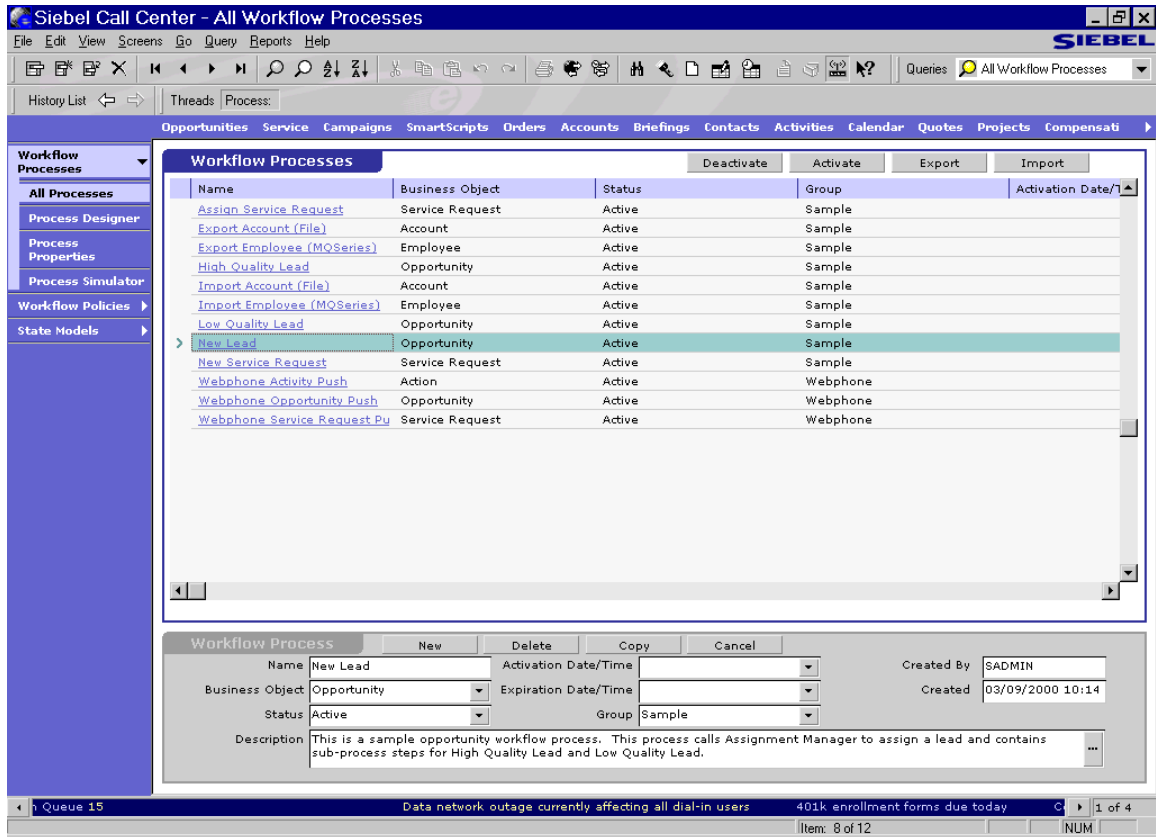


Figure 4-1. All Processes View

The top portion of the view displays a list of all workflow processes. The lower portion is the Workflow Process applet.

View Tasks

The tasks you can perform in this view are:

- **Define a new process.** See [“Defining a New Process”](#) on page 4-14.
- **Delete an existing process.** See [“Deleting a Step or Process”](#) on page 4-78.
- **Copy a process.** See [“Copying a Step or Process”](#) on page 4-78.
- **Activate or deactivate a process.** See [“Activating a Process”](#) on page 4-79.
- **Export or import a process.** See [“Importing or Exporting a Process Definition”](#) on page 7-2.

Workflow Process Applet Field Descriptions

[Table 4-1](#) describes the fields where you enter data in the Workflow Process applet.

Table 4-1. Workflow Process Applet Fields (1 of 2)

Field	Description	Possible Value
Name	The name of the process.	Required A descriptive name that is: <ul style="list-style-type: none">■ Consistent with your overall naming strategy.■ Meaningful to the process designer.
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 “Planning for Workflow Processes” on page 3-4 for more information on defining the primary component for a business object.
Status	The current status of the process.	Read-only. The default status is Inactive. To change the status to Active, click the Active button. To change the status to Inactive, click the Deactivate button.

Table 4-1. Workflow Process Applet Fields (2 of 2)

Field	Description	Possible Value
Group	The name of the associated group for the business process.	Optional Selected from a picklist, this value allows you to segregate business processes by group. Note that you cannot type a value for the group. The value must be chosen from the picklist or left null. Groups for business processes are defined in the Application Administration/List of Values view under WF_GROUP_CD.
Description	A text narrative describing the purpose of the process.	Optional
Activation Date	The activation start date.	Optional This value is selected from a calendar widget.
Expiration Date	The activation end date.	Optional This value is selected from a calendar widget.
Created By	The login ID of the person who created the process.	Read-only This value is automatically populated based on the logon name of the user.
Created	The date that the process is created.	Read-only This value is automatically populated by the system.

Using the Process Designer View

This view features a graphical flowchart tool for designing business processes. The Process Designer view operates by dragging and dropping step icons from the Palette area to the workspace to diagram the process flow.

Figure 4-2 illustrates the Process Designer view.

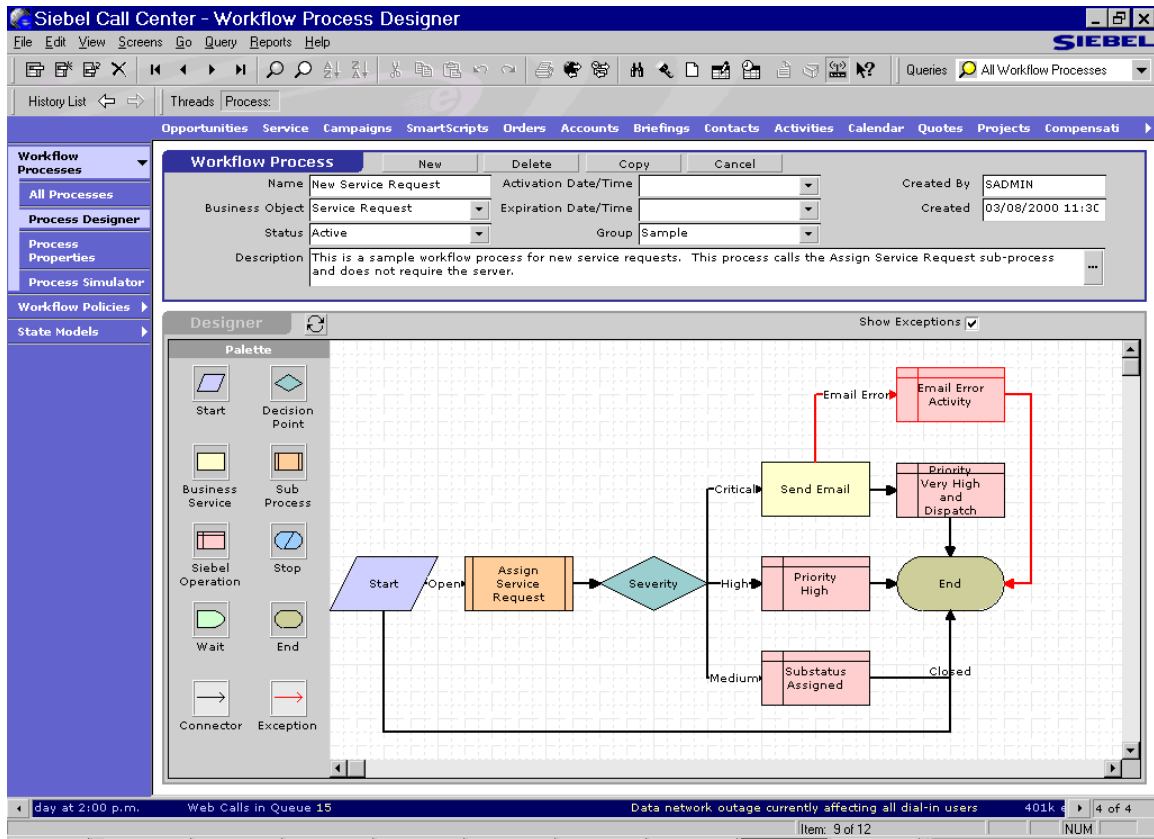


Figure 4-2. Process Designer view

The top portion of the view, the Workflow Process applet, is where you can perform various tasks by clicking the appropriate button and filling in or modifying the necessary fields. The lower portion is the Designer applet.

View Tasks

The tasks you can perform in this view are:

- **Define a new process.** See [“Defining a New Process”](#) on page 4-14.
- **Delete an existing process.** See [“Deleting a Step or Process”](#) on page 4-78.
- **Copy a process.** See [“Copying a Step or Process”](#) on page 4-78.
- **Design a process flow.** See [“Designing a Workflow Process”](#) on page 4-15.

Workflow Process Applet Field Descriptions

This applet also appears in the All Processes view. See [“Workflow Process Applet Field Descriptions”](#) on page 4-6 for a description of these fields.

Designer Applet Palette Item Descriptions

[Table 4-2](#) describes the items available on the Designer applet palette.

Table 4-2. Designer Applet Palette Items (1 of 2)

Item	Description	Possible Value
Start	Represents the conditions that must be met to execute an instance of a business process.	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.	A process can have one or more business service steps.
Siebel Operation	Represents a type of action. It is an insert or update to a business component record or field.	Business object logic applies to all Siebel operations. A process can have one or more Siebel operations.
Connector	Represents the direction of flow between steps in a business process.	A process can have one or more connectors.
Decision	Represents a type of step. A decision is a step in the process definition where the work item branches off to different steps depending on a set of conditions.	A process can have one or more decision steps.

Table 4-2. Designer Applet Palette Items (2 of 2)

Item	Description	Possible Value
Sub Process	Represents a process embedded into another workflow process. A sub-process has its own process definition.	A process can have one or more sub-process steps.
End	Indicates when process execution is complete. An End step allows you to save output arguments into process properties.	A process can have one or more end steps.
Wait	Represents a pause in execution for a specific duration.	A process can have one or more wait steps.
Stop	Represents an end to a process and the presentation of an error to the user.	A process can have one or more stop steps.
Exceptions	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.

Each palette item can be double-clicked from the workspace on the Process Designer view to drill down to the detail view for that specific step.

From some detail views, by clicking the Next Step button, you can view the Next Step applet. The Next Step applet lists steps that are accessible from the current step through connectors or exceptions. To return to the step detail, click the Back button.

Using the Process Properties View

This view lists the currently defined properties for business processes. Process properties store values that the 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 business process completes, the final results of the process properties are available as output arguments. You can also use process property values in expressions.

Figure 4-3 illustrates the Process Properties view.

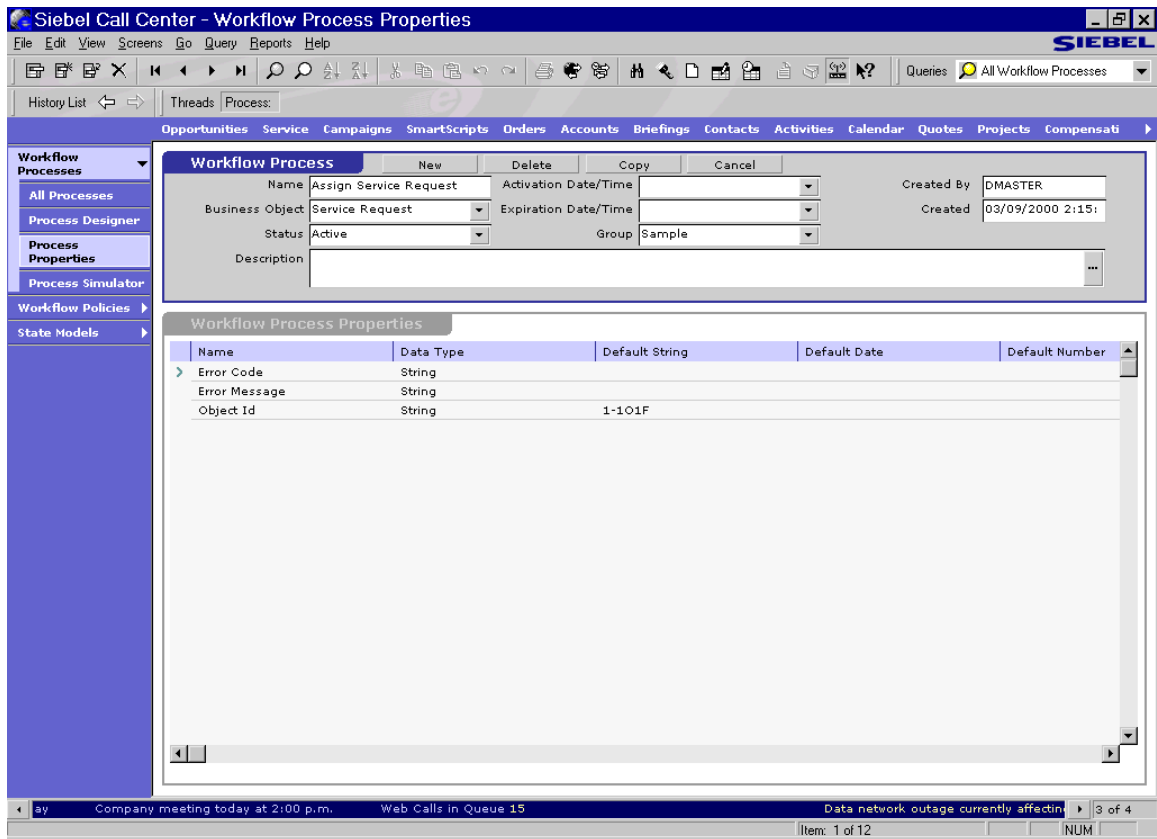


Figure 4-3. Process Properties view

The top portion of the view, the Workflow Process applet, is where you can perform various tasks by clicking the appropriate button and filling in or modifying the necessary fields. The lower portion is the Workflow Process Properties applet.

View Tasks

The tasks you can perform in this view are:

- **Define a new process.** See [“Defining a New Process”](#) on page 4-14.
- **Delete an existing process.** See [“Deleting a Step or Process”](#) on page 4-78.
- **Copy a process.** See [“Copying a Step or Process”](#) on page 4-78.
- **Set process properties.** See [“Working with Process Properties”](#) on page 4-18.

Workflow Process Applet Field Descriptions

This applet also appears in the All Processes view. See [“Workflow Process Applet Field Descriptions”](#) on page 4-6 for a description of these fields.

Workflow Process Properties Applet Field Descriptions

[Table 4-3](#) describes the fields in the Workflow Process Properties applet.

Table 4-3. Workflow Process Properties View Fields

Field	Description	Possible Value
Name	The name of the process property.	Free form
Data Type	The type of data that can be stored in the process property.	<ul style="list-style-type: none">■ 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 <i>Siebel eBusiness Application Integration Guide</i>.
Default String	Initial value if the process property is a string type.	Free form
Default Date	Initial value if the process property is a date type.	Calendar widget
Default Number	Initial value if the process property is a numeric type.	Numeric widget

Working with Workflow Process Definitions

The first part of designing a workflow process is defining the general process parameters and laying out the flow of the process steps. This involves the following tasks:

- **Review existing processes.** See [“Reviewing Existing Process Definitions”](#) next.
- **Define the new process.** See [“Defining a New Process”](#) on page 4-14.

Reviewing Existing Process Definitions

You should review your existing process definitions 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 All Processes view provides a list of all the current process definitions. See [“Using the All Processes View”](#) on page 4-5 for information about this view. Use the following procedure to review existing process definitions.

To review process definitions

- 1** From the Workflow Processes screens, choose the All Processes view.
- 2** Use the scroll bar in the Workflow Processes applet to review the list of available process definitions.
- 3** To view the basic details of a specific process, click on the process name hyperlink.

The details for the process appears in the Workflow Process applet on the lower portion of the view.

- 4** If you find a process you want to copy as the basis for a new process definition, click the Copy button.

See [“Copying a Step or Process”](#) on page 4-78 for more details on copying an existing process definition. Alternatively, you can create a new process by clicking the New button. See [“Defining a New Process”](#) on page 4-14 for more information on creating a new process.

Defining a New Process

To begin defining a new process, you must enter some basic information about the process, such as the process name, the name of the associated business object, and the activation dates. You can enter this information in the Workflow Process applet on the All Processes view, the Process Designer view, or the Process Properties view.

The procedure below describes using the Process Designer view because this is the view that you use to diagram the process steps, which is the next part of creating a new process. See [“Using the Process Designer View” on page 4-8](#) for more information about this view.

To define a new process

- 1** From the Siebel Workflow Designer screens, choose the Process Designer view.
- 2** Click New in the Workflow Process applet.
- 3** Enter a name for the new process.

The process name should be short but meaningful.

- 4** Select a business object from the picklist.

The picklist contains only those business objects having a defined primary business component. See *Siebel Tools Guide* for information about defining business objects. See [“Defining a Primary Component for a Business Object” on page 3-4](#) for information about defining a primary business component for a business object.

NOTE: A business object must be selected to support Siebel operation steps or steps that compare values in a business component. Once a business object is selected, it should not be changed.

- 5** Enter a description.

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

- 6** Select a Start date and an End date from the pop-up calendars.

These dates represent the time span during which the process is valid.

7 Select a group from the picklist.

Groups can be used to categorize your workflow processes. To define a new group, use the following steps:

- a** Navigate to Screens → Application Administration → List of Values → List of Values.
- b** Enter a new record.
- c** Select WF_GROUP_CD in the type field.
- d** In the Display Value column, enter the new group name.

Designing a Workflow Process

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

NOTE: See *Siebel SmartScript Guide* for more details on using the drag and drop functions of the Designer applet.

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 double-clicking on a step icon or connector arrows in the flowchart workspace. Each specific icon drills down to a detail view. [Table 4-4](#) lists the palette icons and the corresponding detail view.

Table 4-4. Step Icons and Detail Views (1 of 2)

Step icons	Detail view
Start	Start
Business Service	Business Service Arguments
Siebel Operation	Siebel Operation Details

Table 4-4. Step Icons and Detail Views (2 of 2)

Step icons	Detail view
Decision	Decision Branches
Sub Process	Sub Process Arguments
Wait	Wait
End	End Arguments
Stop	Stop
Connector	Detail view of originating step
Exception	Detail view of originating step

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

To diagram the process steps

- 1 From the Siebel Workflow Designer screens, choose the Process Designer view.

The name and details of the process definition you are working on should display in the Workflow Process applet on the top portion of the view.

- 2 Add a Start step to the diagram area.

All processes must have one and only one Start step. Details on defining a Start step are in [“Working with Start Steps” on page 4-20](#).

- 3 Add one or more middle steps to the diagram area. Processes can have one or more of any of the action step types, such as Business Service, Decision, Sub Processes, 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 4-33](#)
- [“Defining a Business Service Step” on page 4-42](#)
- [“Defining a Siebel Operation Step” on page 4-48](#)
- [“Defining a Sub Process Step” on page 4-56](#)

- [“Defining a Wait Step” on page 4-65](#)
 - [“Defining a Stop Step” on page 4-71](#)
 - [“Defining an Exception” on page 4-75](#)
- 4** 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 4-61](#).
- 5** 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 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 4-34](#) and [“Defining Conditions and Values” on page 4-35](#) for information about defining Decision Branches.

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

- a** Select the connector, branch, or exception.
- b** Click the right mouse button.
- c** Navigate to Edit → Add Point or Edit → Remove Point.

Alternatively, you could select Ctrl + A to add a point or Ctrl + R to remove a point.

Working with Process Properties

Process Properties are fields for storing values that you can use in steps as input and output arguments.

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

- **Object Id.** The Object Id of the work item being processed.
- **Error Code.** An error symbol of the process instance if a step returns an error.
- **Error Message.** An error message text of the process instance if a step returns an error.
- **Siebel Operation Object Id.** The object id of an object that is updated or created during a Siebel Operation step.

NOTE: Error Code and Error Message are automatically populated when an error occurs. Siebel Operation Object Id is automatically populated when a Siebel Operation step is executed.

Defining Process Properties

To define a process property

- 1 From the Siebel Workflow Designer screens, choose the Process Properties view.
- 2 Verify that the process definition for which you are defining the process property appears in the Workflow Process applet.

NOTE: If the correct process does not appear in the Workflow Process applet, go to the All Processes view and select the correct process definition.

- 3 Click the Workflow Process Properties applet to make it active and choose Edit → Add New Record to begin the definition of a process property.
- 4 Enter a name for the process property.

NOTE: Do not use the period (“.”) character in a process property name.

- 5** Select a data type code from the picklist.

The choices are:

- **String.** If property holds a character value
- **Number.** If property holds a numeric value
- **Date.** If property holds a date value
- **Hierarchy.** If property holds hierarchical data

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.

NOTE: If you wish to use Process Property values in your expressions, you need to add the symbol “&” before the Process Property name. For example, in the following expression `&Object Id` represents the Process Property named Object Id.

```
[Account Id] = [&Object Id]
```

Working with Start Steps

Start steps identify the input conditions that must be met for a process to execute. A Start step is similar to a Decision step in that you can define branches.

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

- Define the Start step
- Define the Start step branches, conditions, and values

A Start step defines the process start condition. For example, to handle open service requests, you could define a start condition of “Status = Open.”

Figure 4-4 illustrates the Start view.

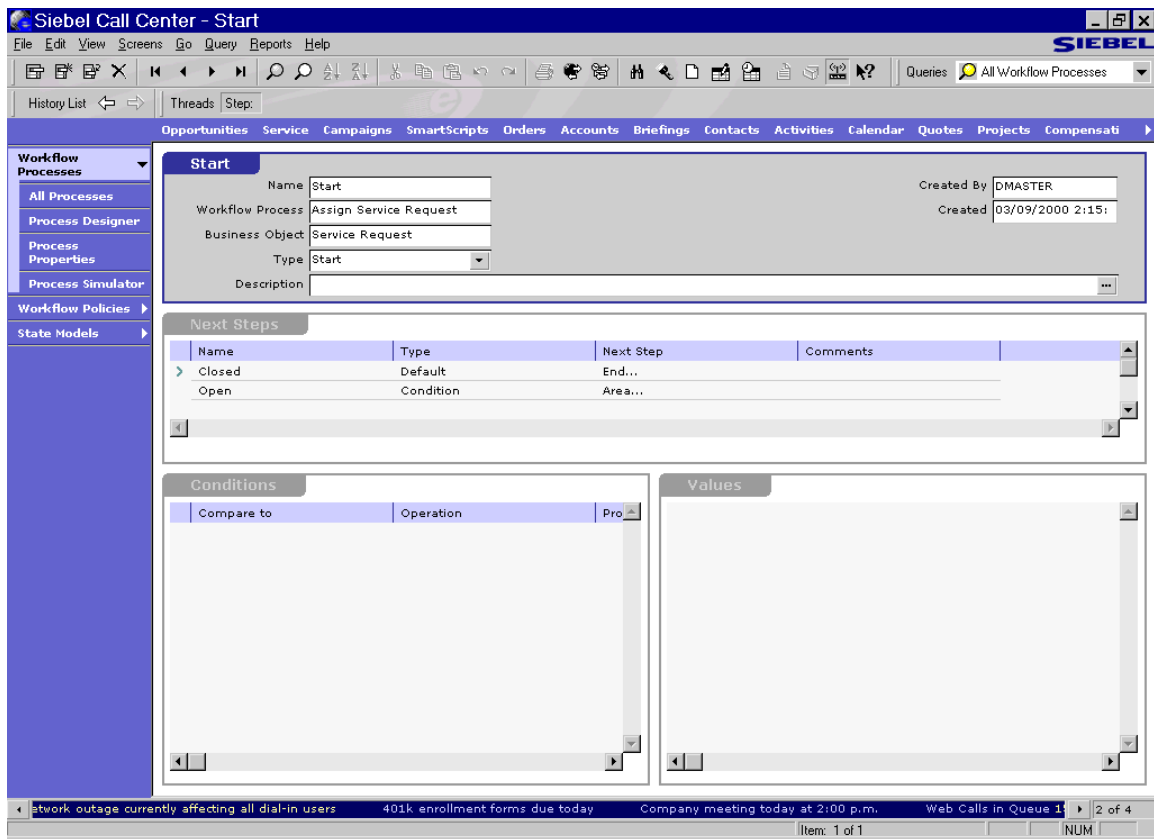


Figure 4-4. Start View

Using the Start View

This view is for entering and viewing the details for a Start step. A Start step is similar to a decision point, and it can contain many decision branches. Each branch is based on multiple conditions.

The top portion of the view, the Start applet, is where you can perform various tasks by clicking the appropriate button and filling in or modifying the necessary fields. The lower portion has three applets: Branches, Conditions, and Values.

View Tasks

The tasks you can perform in this view are:

- **Define a start step.** See [“Working with Start Steps” on page 4-20.](#)
- **Delete an existing start step.** See [“Deleting a Step or Process” on page 4-78.](#)
- **Copy a start step.** See [“Copying a Step or Process” on page 4-78.](#)
- **Define next step branches.** See [“Defining Decision Branches” on page 4-34.](#)
- **Define branch conditions and values.** See [“Defining Conditions and Values” on page 4-35.](#)

Start Applet Field Descriptions

[Table 4-5](#) describes the fields where you enter data in the Start applet.

Table 4-5. Start Applet Fields (1 of 2)

Field	Description	Possible Value
Name	The name of the start step.	A descriptive name that is: <ul style="list-style-type: none"> ■ Consistent with your overall naming strategy ■ Meaningful to the process designer
Business Object	The name of the associated business object.	Business object of the parent process. The business object determines the business components that are available for conditions. This is a read-only field.
Type	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.

Table 4-5. Start Applet Fields (Continued) (2 of 2)

Field	Description	Possible Value
Description	A text narrative describing the purpose of the start step.	Free-form text.
Created By	The name of the person who creates the process.	This value is automatically entered based on the log on name of the user.
Created	The date that the process is created.	This value is automatically entered.

Next Step Branches Applet Field Descriptions

[Table 4-6](#) describes the fields in the Next Step Branches applet.

Table 4-6. Next Step Branches Applet Fields

Field	Description	Possible Value
Name	The name of the next step branch.	The name of the branch must be unique or you will not be able to import or export the workflow process.
Type	The type of branch.	The value can be one of the following choices: <ul style="list-style-type: none">■ Condition. This value indicates that a condition is defined for the branch.■ 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.
Next Step	The name of the step that follows when conditions are met.	Picklist of existing process steps.
Comments	Additional statements relative to the decision branch.	Free-form text.

Conditions Applet Field Descriptions

Table 4-7 describes the fields in the Conditions applet.

Table 4-7. Conditions Applet Fields (1 of 2)

Field	Description	Possible Value
Compare to	Indicates where the comparison value is coming from.	<p>This is a required field, with the following choices:</p> <ul style="list-style-type: none"> ■ Business Component ■ Process Property ■ Expression ■ Applet
Operation	Identifies the comparison operation.	<ul style="list-style-type: none"> ■ 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. ■ 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.
Property Name	Identifies the specific Business Process property on which to base the condition.	This is a required field when Process Property is the Compare to value. This displays a picklist of process properties.

Table 4-7. Conditions Applet Fields (2 of 2)

Field	Description	Possible Value
Business Component	Identifies the specific business component on which to base the condition.	This is a required field when Business Component or expression is the Compare to value. The picklist includes business components defined for the business object selected for the process.
Business Component Field	Identifies the specific field within the business component on which to base the condition.	This is a required field when Business Component is the Compare to value. The picklist displays all fields defined for the selected business component.
Applet Name	The name of the applet to compare.	This is a required field when Applet is the Compare to value.
Applet Field	The name of the field within the named applet.	This is a required field when Applet is the Compare to value. The picklist displays all fields defined for the selected applet.

Values Applet

The Values applet is dynamic based on the Compare to field in the Conditions applet. The Values applet is for storing data to be used in the condition evaluation.

Defining a Start Step

To define a start step

- 1 Select the appropriate process in the All Processes view.
- 2 Navigate to the Process Designer view and double-click the Start icon to drill down to the Start view.

NOTE: To add a Start step to the workspace, drag and drop the Start icon from the palette to the workspace.

- 3 Enter or modify the step name.
- 4 Enter a description of the purpose of the Start step.
- 5 Go to [“Defining Next Step Branches”](#) below, for the next steps.

Defining Next Step Branches

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, 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 view, 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, double-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 Double-click the connector arrow to drill down to the Start view and activate the Next Step applet.
- 4 Enter or modify the branch name.

NOTE: The name of the branch must be unique or you will not be able to import or export the workflow process.

- 5 Select a branch type. The choices for this field are:
 - **Condition.** Define zero or more conditions. If no conditions are defined, it means the branch always passes.
 - **Default.** When all next step branches fail (in other words, nothing matches the conditions), process execution follows the default branch.

NOTE: You must always define a Default branch step.

- 6 Enter the name of the next step.
- 7 Enter comments, if applicable.
- 8 Go to the next procedure, “[Defining Conditions and Values for Next Step Branches](#),” to define the conditions that apply to each branch.

Defining Conditions and Values for 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.

To define conditions and values

- 1 Select the appropriate branch in the Next Step applet in the Start view.
- 2 Click the Conditions applet to make it active.
- 3 Select a Compare to value from the picklist. The available choices are:
 - **Process Property.** Select this when you want to compare a value in a process property in the process instance with a specified value.
 - **Expression.** Select this when you want to use an expression to evaluate a specific value. You must also select a business component if your condition is based on an expression.
 - **Business Component.** Select this when you want to use the value in a business component field for the condition comparison or when you are defining an expression.
 - **Applet.** Select this when you want to use the value in an applet field for the condition comparison.
- 4 If you are using a process property value, select the name of the property from the picklist.
- 5 If you are using a business component field or an expression, select the name of the component and the name of the field from the Business Component Name and the Business Component Field picklists.

- 6** Select the operation to use for evaluating the values. The available choices are:
- **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.
 - **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.

- 7** Enter an applet name and applet field, if applicable.

The applet works similar to a business component, except that it has translatable names and picklists. Only list applets can be used in conditions.

- 8** Enter any appropriate values in the Values applet.

You can enter multiple records in the values applet. Workflow Processes assumes an “or” condition between values.

If you selected Expression in the Compare to field, enter your expression in the Values applet. The syntax is the same as the syntax used in Siebel Tools. See *Siebel Tools Guide* for more information.

NOTE: Multiple conditions can be defined for each branch. Workflow Processes treats multiple conditions with an “and.” Use expressions to define multiple “or” conditions.

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')
```

Working with 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:

- Define a decision step
- Define the branches
- Define the conditions and values

Figure 4-5 illustrates the Decision Branches view.

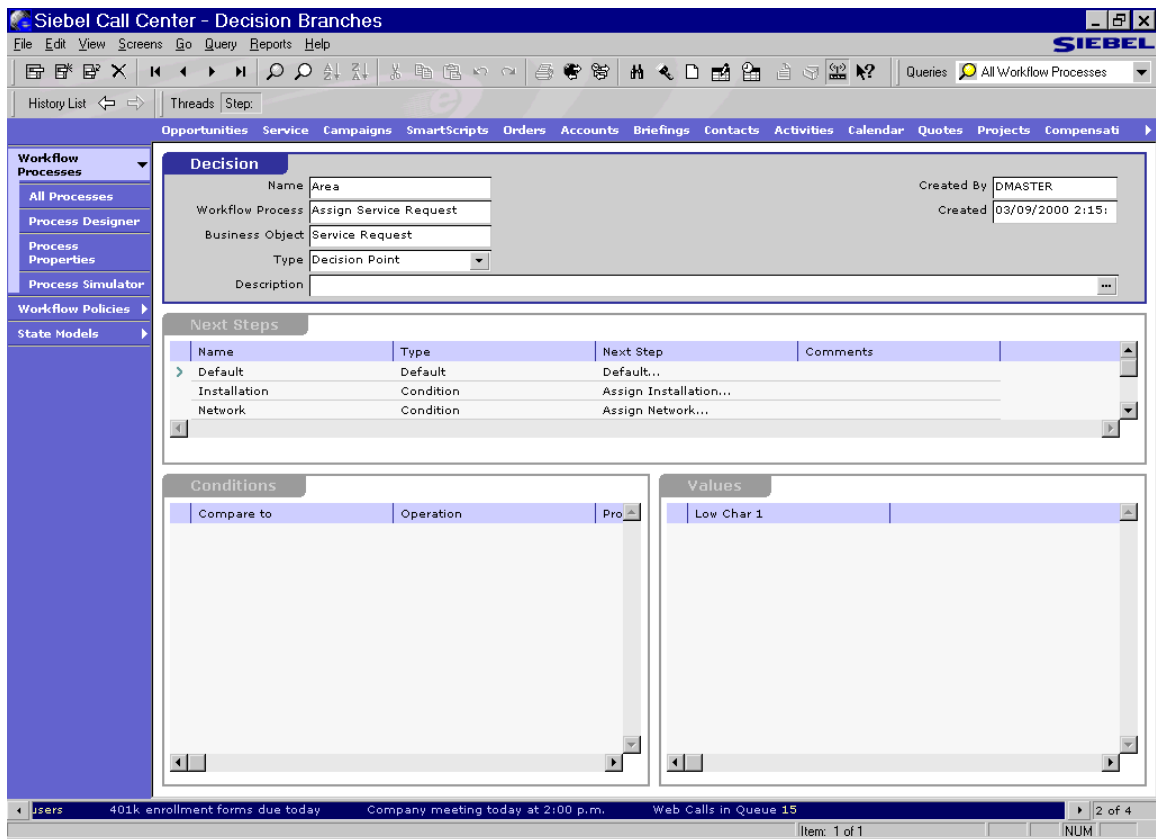


Figure 4-5. Decision Branches View

Using the Decision Branches View

This view is for entering and viewing the details for a Decision step. A decision can contain many branches. Each branch is based on multiple conditions.

The top portion of the view, the Decision applet, is where you can perform various tasks by clicking the appropriate button and filling in or modifying the necessary fields. The lower portion has three applets: Branches, Conditions, and Values.

View Tasks

The tasks you can perform in this view are:

- **Define a decision step.** See [“Defining a Decision Step”](#) on page 4-33.
- **Delete an existing decision step.** See [“Defining Decision Branches”](#) on page 4-34.
- **Copy a decision step.** See [“Copying a Step or Process”](#) on page 4-78.
- **Define branches.** See [“Defining Decision Branches”](#) on page 4-34.
- **Define conditions and values.** See [“Defining Conditions and Values”](#) on page 4-35.

Decision Applet Field Descriptions

[Table 4-8](#) describes the fields where you enter data in the Decision applet.

Table 4-8. Decision Applet Fields

Field	Description	Possible Value
Name	The name of the decision step.	A descriptive name that is: <ul style="list-style-type: none">■ Consistent with your overall naming strategy■ Meaningful to the process designer
Business Object	The name of the associated business object.	Read-only.
Type	The type of the step.	This value is automatically entered when you create the step on the Process Designer view.
Description	A text narrative describing the purpose of the decision step.	Free form text.
Created By	The name of the person who creates the process.	Read-only. This value is automatically entered based on the logon name of the user.
Created	The date that the process is created.	Read-only. This value is automatically entered.

Branches Applet Field Descriptions

[Table 4-9](#) describes the fields in the Branches applet.

Table 4-9. Branches 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 import or export the workflow process.
Type	The type of branch.	The value can be one of four choices: <ul style="list-style-type: none">■ Default. This value indicates that if all conditional branches fail, this branch will be followed.■ Condition. Define zero or more conditions. If no conditions are defined, it means the branch always passes.
Next Step	The name of the step that follows when conditions are met.	
Comments	Additional statements relative to the branch.	

Conditions Applet Field Descriptions

Table 4-10 describes the fields in the Conditions applet.

Table 4-10. Conditions Applet Fields (1 of 2)

Field	Description	Possible Value
Compare to	Indicates where the comparison value is coming from.	<p>This is a required field, with the following choices:</p> <ul style="list-style-type: none"> ■ Business Component ■ Process Property ■ Expression ■ Applet
Operation	Identifies the comparison operation	<ul style="list-style-type: none"> ■ 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. ■ 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.
Property Name	Identifies the specific business process property on which to base the condition.	This is a required field when Process Property is the Compare to value.

Table 4-10. Conditions Applet Fields (2 of 2)

Field	Description	Possible Value
Business Component	Identifies the specific business component on which to base the condition.	This is a required field when Business Component or expression is the Compare to value.
Business Component Field	Identifies the specific field within the business component on which to base the condition.	This is a required field when Business Component is the Compare to value.
Operation	The type of operation to perform on the comparison value.	A picklist contains the possible values for this required field.
Applet Name	The name of the applet to compare.	This is a required field when Applet is the Compare to value.
Applet Field	The name of the field within the named applet.	This is a required field when Applet is the Compare to value.

Values Applet

The Values applet is dynamic based on the Compare to field in the Conditions applet. The Values applet is for storing data to be used in the condition evaluation.

Defining a Decision Step

To define a decision step

- 1 Make the appropriate process active by selecting it in the All Processes view.
- 2 Navigate to the Process Designer view and double-click the Decision icon to drill down to the details view.

NOTE: To add a Decision step to the workspace, drag and drop the Decision icon from the palette to the workspace.

- 3 Enter a step name.
- 4 Enter a description of the purpose for the decision step.

NOTE: Go to the next procedure, “[Defining Decision Branches](#),” to create the branches for the decision step. From the Decision view, you can flip through all of the decision steps for the current process by holding down the CTRL key and using the up and down keyboard arrows.

Defining Decision Branches

If you connected the decision steps to the next steps in the process with connector arrows, as described in [“Designing a Workflow Process” on page 4-15](#), 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 [“Defining Conditions and Values” on page 4-35](#) for the procedure on defining conditions and values for each branch.

To define a branch

- 1 With the appropriate Decision step name displayed in the Decision applet of the Decision view and the Next Step applet active, choose Edit → Add New Record.
- 2 Enter a branch name.

NOTE: The name of the branch must be unique or you will not be able to import or export the workflow process.

- 3 Select a branch type. The choices for this field are:
 - **Default.** When all decision branches fail (in other words, nothing matches the conditions), process execution follows the default branch.
 - **Condition.** Define zero or more conditions. If no conditions are defined, it means the branch always passes.
- 4 Enter the name of the next step.
- 5 Enter comments, if applicable.
- 6 Verify on the Process Designer view 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 through step 6 for additional branches.

- 7 Go to the next procedure, [“Defining Conditions and Values,”](#) to define the conditions that apply to each branch.

Defining Conditions and Values

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.

To define conditions and values

- 1 Select the appropriate branch in the Branches applet on the Decision view.
- 2 Click the Conditions applet to make it active.
- 3 Select a Compare to value from the picklist. The available choices are:
 - **Process Property.** Select this when you want to compare a value in a process property in the process instance with a specified value.
 - **Expression.** Select this when you want to use an expression to evaluate a specific value.
 - **Business Component.** Select this when you want to use the value in a business component field for the condition comparison.
 - **Applet.** Select this when you want to use the value in an applet field for the condition comparison.
- 4 Select the operation to use for evaluating the values. The available choices are:
 - **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.
 - **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.
- 5 If you are using a process property value, select the name of the property from the picklist.
 - 6 If you are using a business component field or an expression, select the name of the component and the name of the field from the Business Component Name and the Business Component Field picklists.
 - 7 Enter an applet name and applet field, if applicable.
 - 8 The applet works similar to a business component, except that it has translatable names and picklists. Only list applets can be used in conditions.
 - 9 Enter any appropriate values and expressions in the Values applet.

NOTE: Expressions should be entered using the same format as found in Siebel Tools. For more information, see *Siebel Tools Guide*.

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

```
([Close Date] <= Today()) AND ([Revenue] > 1000)
```

Working with 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.** You can have Assignment Manager assign an object in a workflow process by calling the Synchronous Assignment Manager Request business service.
- **Navigation.** You can navigate a user to a view using the Workflow UI Utilities business service.
- **Server tasks.** You can run a server component task using the Asynchronous or Synchronous Server Requests business service.

For a list of predefined business services, see [Appendix A, “Predefined Business Services.”](#)

Another option available within Siebel Workflow is to define your own custom business services and invoke them from a workflow process. Custom business services can be defined using Siebel VB or Siebel eScript. You can define business services by navigating to Screens → Business Service Administration in the Siebel client 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 Business Service Arguments view.

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

- Define a business service step
- Define input arguments for the step
- Define output arguments for the step

Figure 4-6 illustrates the Business Service view.

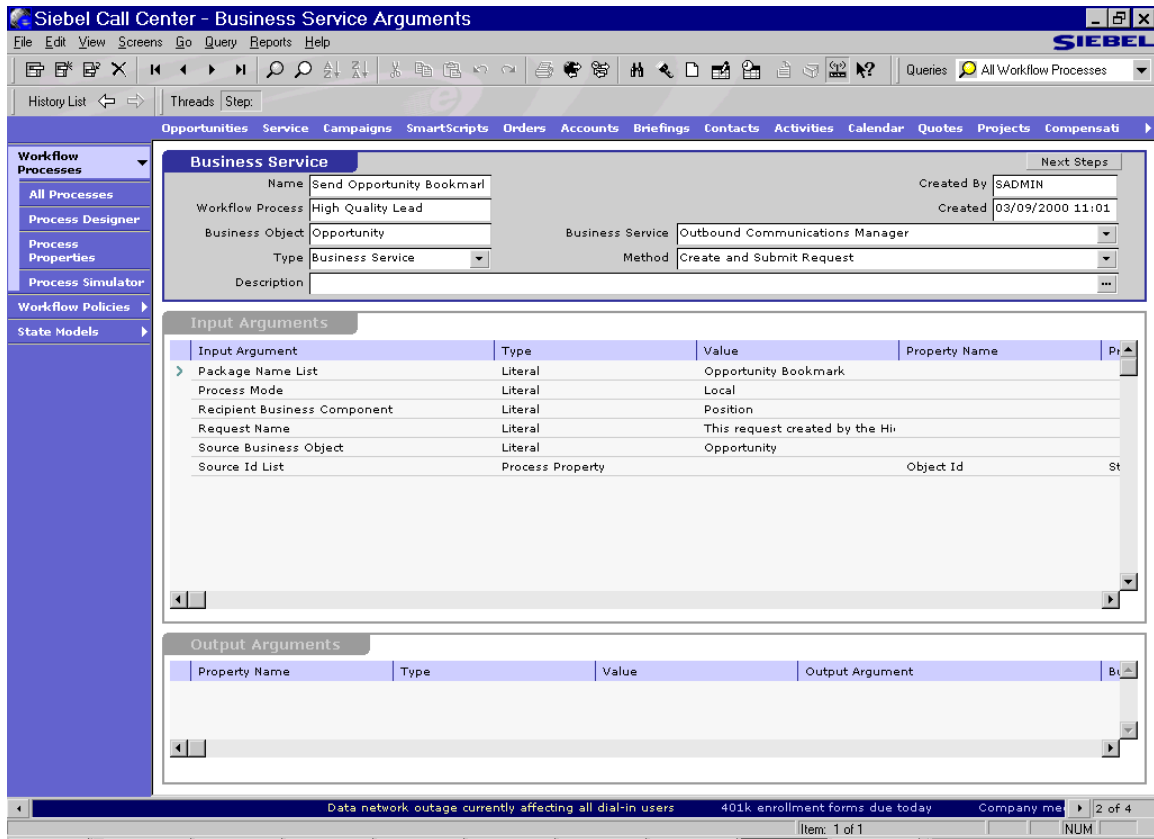


Figure 4-6. Business Service View

Using the Business Service View

You use the Business Service view to define how to invoke a Business Service, including the input and output arguments.

The top portion of the view, the Business Service applet, is where you can perform various tasks by clicking the appropriate button and filling in or modifying the necessary fields. The lower portion has two applets: Input Arguments and Output Arguments.

View Tasks

The tasks you can perform in this view are:

- **Define a business service step.** See [“Defining a Business Service Step”](#) on page 4-42.
- **Delete an existing business service step.** See [“Deleting a Step or Process”](#) on page 4-78.
- **Copy a business service step.** See [“Copying a Step or Process”](#) on page 4-78.

Business Service Applet Field Descriptions

[Table 4-11](#) describes the fields where you enter data in the Business Service applet.

Table 4-11. Business Service Applet Fields

Field	Description	Possible Value
Name	The name of the business service step.	A descriptive name that is: <ul style="list-style-type: none"> ■ Consistent with your overall naming strategy ■ Meaningful to the process designer
Type	The type of task.	This value is automatically entered when you create the step on the Process Designer view.
Description	A text narrative describing the purpose of the task.	Free-form text.
Business Service Name	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 3-5 for more information.
Method	The name of the method to invoke on the service.	The picklist displays methods defined for the selected business service.
Created By	The name of the person who creates the step.	This value is automatically entered based on the logon name of the user.
Created	The date that the step is created.	This value is automatically entered.

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. The name that is displayed in all picklists is the value of the Display name property for the object. The Display Name and the object name may be different. For more information, please refer to [“Enabling a Business Service for Workflow Processes” on page 3-5](#).

Input Arguments Applet Field Descriptions

Input arguments allow you to define values that you want to pass to a service method. Many methods require input arguments. [Table 4-12](#) describes the fields in the Input Argument applet.

Table 4-12. Arguments Applet Fields (1 of 2)

Field	Description	Possible Value
Input Argument	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. Note that the picklist will show the Display Name for the Business Service as defined in Siebel Tools.
Type	The type of argument.	This is a required field. The picklist contains the following choices: <ul style="list-style-type: none"> ■ 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. Note that 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.

Table 4-12. Arguments Applet Fields (2 of 2)

Field	Description	Possible Value
Business Component	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–type input arguments.

Output Arguments Applet Field Descriptions

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

[Table 4-13](#) describes the fields in the Output Arguments applet.

Table 4-13. Output Arguments Applet Fields (1 of 2)

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 4-18 .
Type	The type or argument.	This is a required field. The picklist contains the following choices: <ul style="list-style-type: none"> ■ Literal ■ Output Argument ■ Business Component ■ Expression
Output Argument	The name of the output argument from the business service.	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.

Table 4-13. Output Arguments Applet Fields (Continued) (2 of 2)

Field	Description	Possible Value
Value	A string value	For Literal or Expression arguments. Note that string values can only be a maximum of 32,767 characters.
Business Component	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 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 *Siebel Tools Guide* for more information.

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

Defining a Business Service Step

To define a new business service step

- 1 Make the appropriate process active by selecting it in the All Processes view.
- 2 Navigate to the Process Designer view and double-click the Business Service icon to drill down to the details view.

NOTE: To add a Business Service step to the workspace, drag and drop the Business Service icon from the palette to the workspace.

- 3 Enter a step name.
- 4 Enter a description of the step.

- 5 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 *Siebel Tools Guide* for information on creating customer-defined services.

- 6 Enter 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 Business Service Input Arguments.”](#)
- 8 If you need to define output arguments for this task, continue to [“Defining Business Service Output Arguments” on page 4-44.](#)

NOTE: From the Business Service detail view, you can flip through all of the Business Service steps for the current process by holding down the CTRL key and using the up and down keyboard arrows.

Defining Business Service Input Arguments

To define input arguments for a business service step

- 1 With the appropriate business service name displayed in the Business Service applet of the Business Service view and the Input Argument applet active, choose Edit → Add New Record.
- 2 Select the name of the input argument. The picklist contains input arguments for the selected business service method.
- 3 Choose an input argument type. This indicates the type of data that you are passing to the input argument. The choices available are:
 - Literal
 - Process Property
 - Business Component
 - Expression

- 4 If the input argument type selected is Literal, enter a value.
This may be a picklist, depending on the argument selected.
- 5 If the input argument type is a Process Property, select the property name.
- 6 If the input argument type is a Business Component, select the applicable business component name and business component field.
- 7 If the input argument type is an Expression, enter an expression in the Value field.
- 8 Enter any appropriate comments.

Defining Business Service Output Arguments

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 name displayed in the Business Service applet of the Business Service view and the Output Argument applet active, choose Edit → Add New Record.
- 2 Select the property name from the picklist.
- 3 Choose an output argument type. The choices available are:
 - Literal
 - Output Argument
 - Business Component
 - Expression
- 4 If the output argument type selected is Literal, enter a value.
- 5 If the output argument type is an Output Argument, enter the argument.
- 6 If the output argument type is a Business Component, select the applicable business component name and business component field.
- 7 If the output argument type is an Expression, enter the expression in the value field.
- 8 Enter any appropriate comments.

Working with Siebel Operation Steps

Siebel Operation steps handle Siebel database operations, such as insert or update.

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

- Define a Siebel Operation step
- Define any fields for the Siebel operation
- Define search specifications for the Siebel operation

Figure 4-7 illustrates the Siebel Operation view.

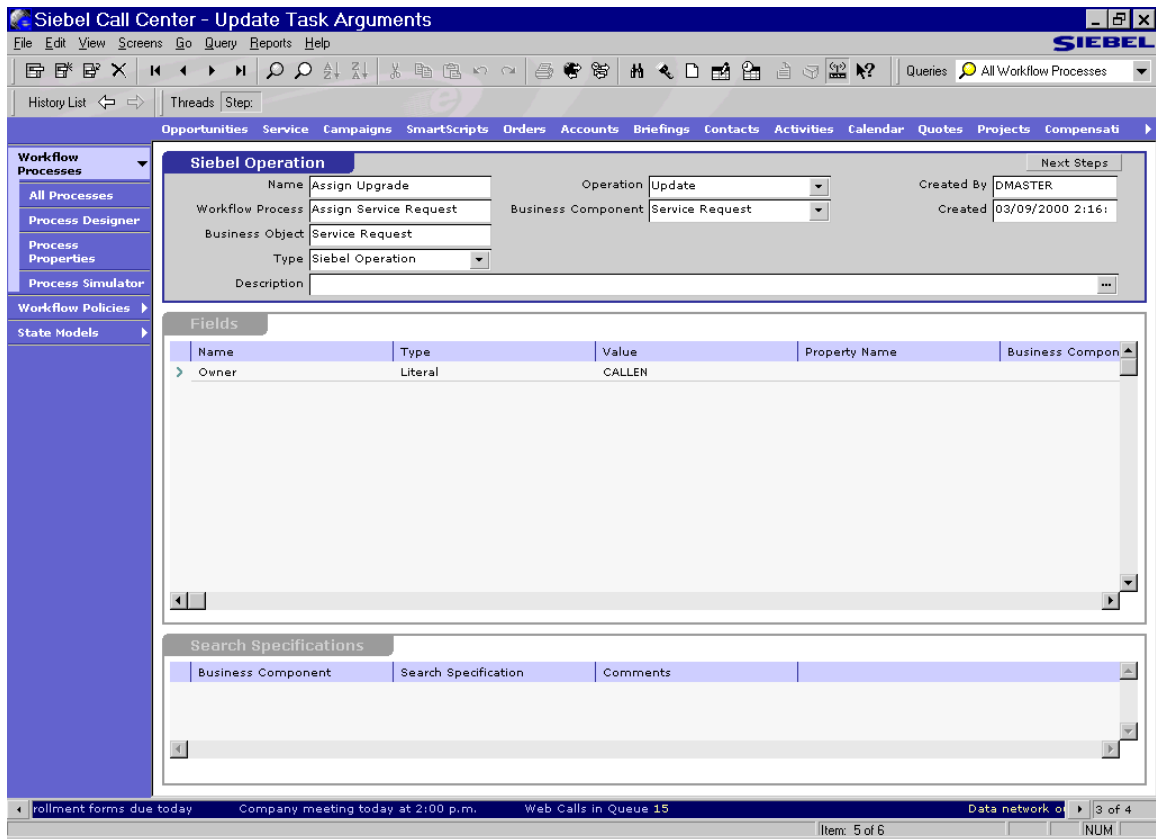


Figure 4-7. Siebel Operation View

Using the Siebel Operation View

You use the Siebel Operation view to define an operation type step, such as inserting or updating a business component.

The top portion of the view, the Siebel Operation applet, is where you can perform various tasks by clicking the appropriate button and filling in or modifying the necessary fields. The lower portion has two applets: Fields and Search Specifications.

The Fields applet allows you to define the field values for the operation. The Search Specifications applet is only for update operations. It is used to define which records you want to update when updating a business component with multiple records.

View Tasks

The tasks you can perform in this view are:

- **Define a Siebel operation step.** See [“Defining a Siebel Operation Step” on page 4-48.](#)
- **Delete an existing Siebel operation step.** See [“Deleting a Step or Process” on page 4-78.](#)
- **Copy a Siebel operation step.** See [“Copying a Step or Process” on page 4-78.](#)

Siebel Operation Applet Field Descriptions

[Table 4-14](#) describes the fields where you enter data in the Siebel Operation applet.

Table 4-14. Siebel Operation Applet Fields (1 of 2)

Field	Description	Possible Value
Name	The name of the Siebel operation step.	A descriptive name that is: <ul style="list-style-type: none">■ Consistent with your overall naming strategy■ Meaningful to the process designer
Description	A text narrative describing the purpose of the Siebel operation.	Free-form text.
Operation	Specifies the type of operation.	This is a required field. Possible operations are Insert or Update.

Table 4-14. Siebel Operation Applet Fields (2 of 2)

Field	Description	Possible Value
Business Component	The name of the business component within the workflow process business object.	This is a required field. This is a picklist of business components that have been defined for the selected business object.
Created By	The name of the person who creates the step.	This value is automatically entered based on the logon name of the user.
Created	The date that the step is created.	This value is automatically entered.

Fields Applet Field Descriptions

[Table 4-15](#) describes the fields in the Fields applet.

Table 4-15. Fields Applet Fields

Field	Description	Possible Value
Name	The name of the business component field.	This is a picklist of fields on the business component. The list displays all fields defined on the business component except multi-value groups and calculated fields.
Type	The type of value to be entered in the field.	This is a required field. The picklist contains the following choices: <ul style="list-style-type: none"> ■ Literal ■ Process Property ■ Business Component ■ Expression
Value	A string value.	For literal and expression arguments.
Property Name	The name of the business process property.	For Process Property–type arguments.
Business Component	The name of the associated business component.	For Business Component–type arguments.
Business Component Field	The name of the field associated with the business component.	For Business Component–type arguments.

Search Specifications Applet Field Descriptions

Table 4-16 describes the fields in the Search Specifications applet.

Table 4-16. Search Specifications Applet Fields

Field	Description	Possible Value
Business Component	The name of the business component to search.	
Search Specification	The search parameters.	Written in the form of an expression, for example, [Status] LIKE '*Open*'.
Comments	Text narrative of the search purpose.	Free-form text.

Defining a Siebel Operation Step

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 sub process 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 would be 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.

Calculated fields cannot be updated using Siebel Operations steps because typically they require values from other business component fields. Calculations should be performed using expressions.

The object id for the process is automatically passed to Siebel Operation steps, with the exception of workflows called from a script (in those cases, the row id of the object needs to be passed as part of the property set that calls the workflow). 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.

NOTE: The Workflow programs and Siebel Operation steps use different object layers to update data. For example, you may have a Workflow Policy that calls a Workflow 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.

To define a Siebel operation

- 1** Make the appropriate process active by selecting it in the All Processes view.
- 2** Navigate to the Process Designer view and double-click the Siebel Operation icon to drill down to the details view.

NOTE: To add a Siebel Operation step to the workspace, drag and drop the Siebel Operation icon from the palette to the workspace.

- 3** Enter a name for the step.
- 4** Enter a description of the purpose of the step.

- 5 Select the type of operation. The available choices are:
 - Insert
 - Update

NOTE: You will need to verify that updates/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 verify that the values selected for the sub-area field are valid for that associated area.

- 6 Select the name of the business component.
- 7 Select the business component method if the operation type is Method.
- 8 If you need to define fields for this Siebel operation, continue to [“Defining Fields for the Siebel Operation.”](#)
- 9 If you need to define search specifications for this Siebel operation, continue to [“Defining Siebel Operation Search Specifications” on page 4-51.](#)

NOTE: From the Siebel Operation view, you can navigate through all of the Siebel operation steps for the current process by holding down the CTRL key and using the up and down keyboard arrows.

Defining Fields for the Siebel Operation

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 name displayed in the Siebel Operation applet of the Siebel Operation view and the Fields applet active, choose Edit → Add New Record.
- 2 Select the name of the field to be updated.

- 3** Choose an input argument type. The choices available are:
 - Literal
 - Process Property
 - Business Component
 - Expression
- 4** If the field type selected is Literal, enter a value.
- 5** If the field type is a Process Property, select a property name.
- 6** If the field type is a Business Component, select the applicable business component name and business component field.
- 7** If the field type is an Expression, enter the expression in the value field.
- 8** 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 name displayed in the Siebel Operation applet of the Siebel Operation view and the Search Specifications applet active, choose Edit → Add New Record.
- 2** Select the appropriate business component name.
- 3** Enter search specifications.
- 4** Enter any comments, if necessary.

NOTE: After executing an insert Siebel Operation step, the row id of the record that was created is automatically stored in the Siebel Operation object id process property.

Working with Sub Process Steps

A Sub Process step allows you to invoke an entirely separate process within a process. A process definition can have one or more Sub Process steps.

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

- Define a sub process step
- Define input arguments for the sub process
- Define output arguments for the sub process

Figure 4-8 illustrates the Sub Process view.

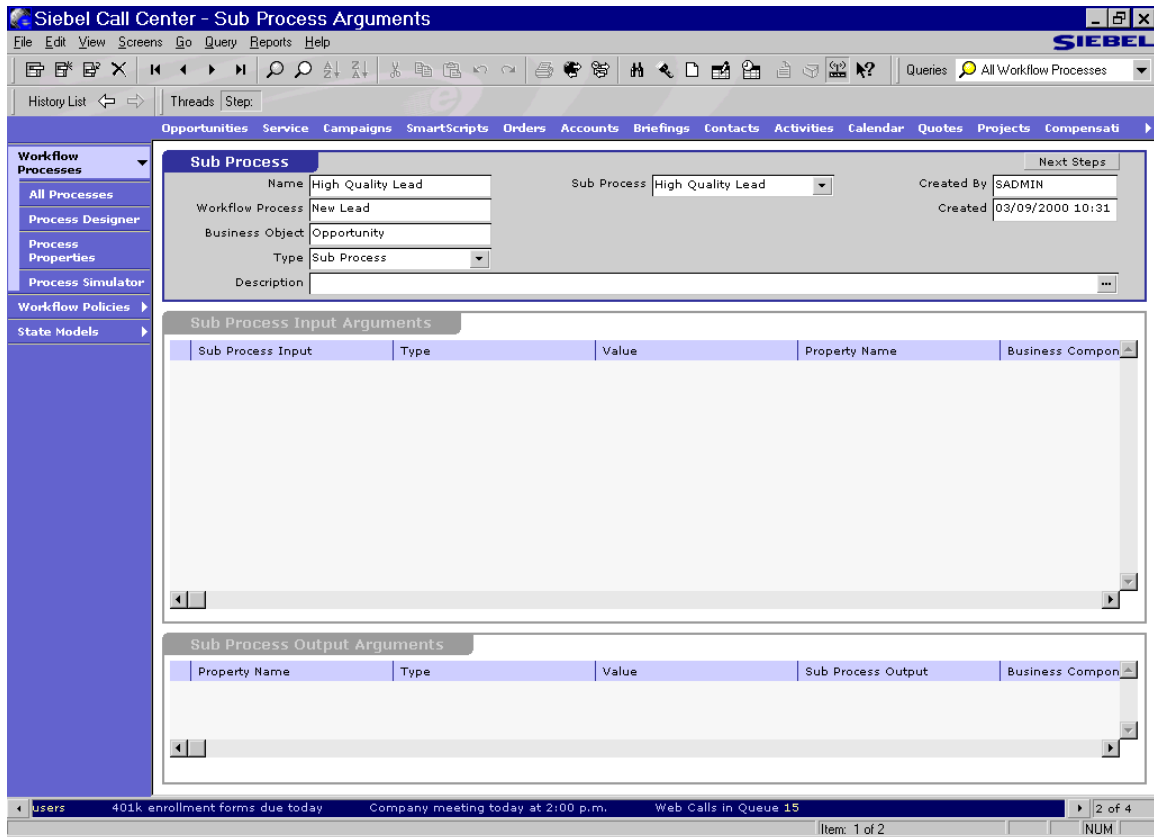


Figure 4-8. Sub Process View

Using the Sub Process View

You use the Sub Process view to define a step for invoking a sub process, including the input and output arguments.

The top portion of the view, the Sub Process applet, is where you can perform various tasks by clicking the appropriate button and filling in or modifying the necessary fields. The lower portion has two applets: Input Arguments and Output Arguments. Input and output arguments allow you to pass values to the sub process properties.

View Tasks

The tasks you can perform in this view are:

- **Define a sub process step.** See “[Defining a Sub Process Step](#)” on page 4-56.
- **Delete an existing sub process step.** See “[Working with End Steps](#)” on page 4-59.
- **Copy a sub process step.** See “[Copying a Step or Process](#)” on page 4-78.

Sub Process Applet Field Descriptions

[Table 4-17](#) describes the fields where you enter data in the Sub Process applet.

Table 4-17. Sub Process Applet Fields

Field	Description	Possible Value
Step	The name of the sub process step.	A descriptive name that is: <ul style="list-style-type: none"> ■ 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 sub process.	Free-form text.
Sub Process	The name of the process to run.	This value is selected from a picklist of defined workflow processes.
Created By	The name of the person who creates the step.	This value is automatically entered based on the log on name of the user.
Created	The date that the step is created.	This value is automatically entered.

Input Arguments Applet Field Descriptions

Table 4-18 describes the fields in the Input Argument applet.

Table 4-18. Input Arguments Applet Fields

Field	Description	Possible Value
Sub Process Input	The name of the sub process property.	This is a required field. The picklist contains process properties for the selected sub process.
Type	The type of argument.	This is a required field. The picklist contains the following choices: <ul style="list-style-type: none">■ 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.
Property Name	The name of the business process property.	For Process Property type input arguments.
Business Component	The name of a business component within the business object of the business process.	For Business Component type input arguments.

Output Arguments Applet Field Descriptions

Table 4-19 describes the fields in the Output Arguments applet.

Table 4-19. 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 4-18.
Type	The type or argument.	This is a required field. The picklist contains the following choices: <ul style="list-style-type: none"> ■ Literal ■ Business Component ■ Expression
Value	A string value	For Literal or Expression arguments.
Sub Process Output	The name of the output argument from the sub process.	For Output Argument type.
Business Component	The name of the business component within the business object of the business process.	For Business Component type.

Defining a Sub Process Step

NOTE: Before you define a Sub Process step, the workflow process you will call with the step must already be defined.

To define a sub process step

- 1** Make the appropriate process active by selecting it in the All Processes view.
- 2** Navigate to the Process Designer view and double-click the Sub Process icon to drill down to the details view.

NOTE: To add a Sub Process step to the workspace, drag and drop the Sub Process icon from the palette to the workspace.

- 3** Enter a name for the step.
- 4** Enter a description of the purpose of the step.
- 5** Select the process to be called by the Sub Process step.

NOTE: You can display the process design for the selected Sub Process by clicking the right mouse button, and selecting Go to Sub Process.

- 6** If you need to define input arguments for this sub process, continue to [“Defining Sub Process Input Arguments.”](#)
- 7** If you need to define output arguments for this sub process, continue to [“Defining Sub Process Output Arguments” on page 4-58.](#)

NOTE: From the Sub Process view, you can flip through all of the sub process steps for the current process by holding down the CTRL key and using the up and down keyboard arrows.

Defining Sub Process Input Arguments

Input arguments allow you to populate process properties in the sub process. For example, if you want to pass the object id from the current process to the sub process, you would do this through input arguments.

To define input arguments for a sub process step

- 1** With the appropriate sub process name displayed in the Sub Process applet of the Sub Process view and the Input Arguments applet active, choose Edit → Add New Record.
- 2** Enter the name of the sub process input argument.
- 3** Choose an input argument type. The choices available are:
 - Literal
 - Process Property
 - Business Component
 - Expression
- 4** If the input argument type selected is Literal, enter a value.
- 5** If the input argument type is Process Property, select a property name.
- 6** If the input argument type is Business Component, select the applicable business component name and business component field.
- 7** If the input argument type is Expression, enter the expression in the value field.
- 8** Enter any appropriate comments.

Defining Sub Process Output Arguments

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

To define output arguments for a sub process step

- 1** With the appropriate sub process name displayed in the Sub Process applet of the Sub Process view and the Output Arguments applet active, choose Edit → Add New Record.
- 2** Select the property name from the picklist.
- 3** Choose an output argument type. The choices available are:
 - Literal
 - Output Argument
 - Business Component
 - Expression
- 4** If the output argument type selected is Literal, enter a value.
- 5** If the output argument type is Output Argument, enter the argument.
- 6** If the output argument type is Business Component, select the applicable business component name and business component field.
- 7** If the output argument type is Expression, enter the expression in the value field.
- 8** Enter any appropriate comments.

Working with 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.

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

- Define an end step
- Define output arguments for the end step

Figure 4-9 illustrates the End Step view.

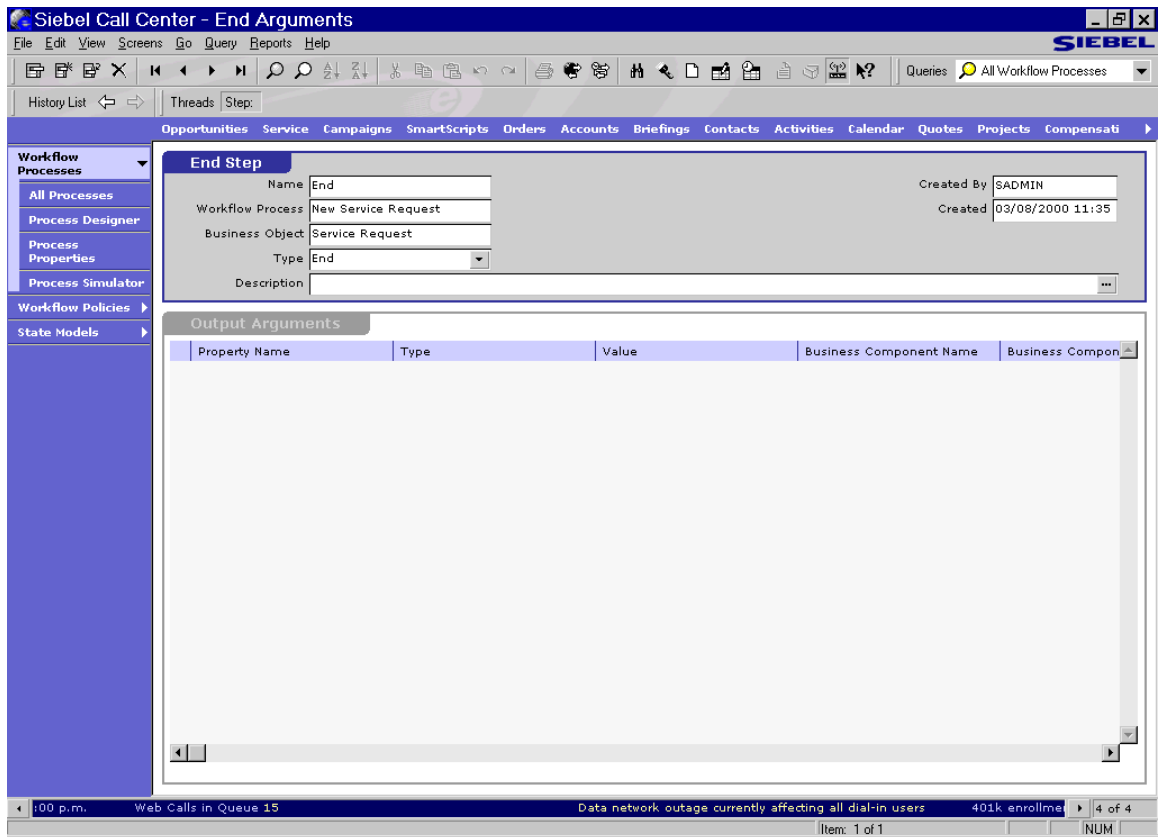


Figure 4-9. End Step View

Using the End Arguments View

You use the End Arguments view to store any results from the workflow process in a process property.

The top portion of the view, the End Step applet, is where you can perform various tasks by clicking the appropriate button and filling in or modifying the necessary fields. The lower portion has an Output Arguments applet.

View Tasks

The tasks you can perform in this view are:

- **Define a new end step.** See [“Defining an End Step” on page 4-61](#).
- **Delete an existing end step.** See [“Deleting a Step or Process” on page 4-78](#).
- **Copy an end step.** See [“Copying a Step or Process” on page 4-78](#).

End Step Applet Field Descriptions

[Table 4-20](#) describes the fields where you enter data in the End Step applet.

Table 4-20. End Step Applet Fields

Field	Description	Possible Value
Name	The name of the end step.	A descriptive name that is: <ul style="list-style-type: none">■ Consistent with your overall naming strategy■ Meaningful to the process designer
Type	The type of end step.	This value is automatically set based on the icon you drag to the workspace. Read-only.
Description	A text narrative describing the purpose of the end step.	Free-form text.
Created By	The name of the person who creates the step.	This value is automatically entered based on the logon name of the user.
Created	The date that the step is created.	This value is automatically entered.

Output Arguments Applet Field Descriptions

[Table 4-21](#) describes the fields in the Output Arguments applet.

Table 4-21. 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 4-18.
Type	The type of argument.	This is a required field. The picklist contains the following choices: <ul style="list-style-type: none"> ■ Literal ■ Output Argument ■ Business Component ■ Expression
Value	A string value.	For Literal or Expression arguments.
Name	The name of the output argument from the end step.	For Output Argument type.
Business Component	The name of the business component within the business object of the business process.	For Business Component type.

Defining an End Step

To define an end step

- 1 Make the appropriate process active by selecting it in the All Processes view.
- 2 Navigate to the Process Designer view and double-click the End icon to drill down to the details view.

NOTE: To add an End step to the workspace, drag and drop the End icon from the palette to the workspace.

- 3 Enter a name for the step.

- 4 Enter a description of the purpose of the step.
- 5 Enter output arguments in the Output Arguments applet.

Defining End 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 end step

- 1 With the appropriate End step name displayed in the End Step applet of the End view and the Output Arguments applet active, choose Edit → Add New Record.
- 2 Select the property name from the picklist.
- 3 Choose an output argument type. The choices available are:
 - Literal
 - Output Argument
 - Expression
 - Business Component
- 4 If the output argument type selected is Literal, enter a value.
- 5 If the output argument type is Output Argument, enter the argument.
- 6 If the output argument type is Business Component, select the applicable business component name and business component field.
- 7 Enter any appropriate comments.

Working with Wait Steps

Wait steps allow you to suspend process execution for a specific period of time. The main parts of creating a wait step for a workflow process are:

- Define a wait step
- Define input arguments for the wait step

Figure 4-10 illustrates the Wait view.

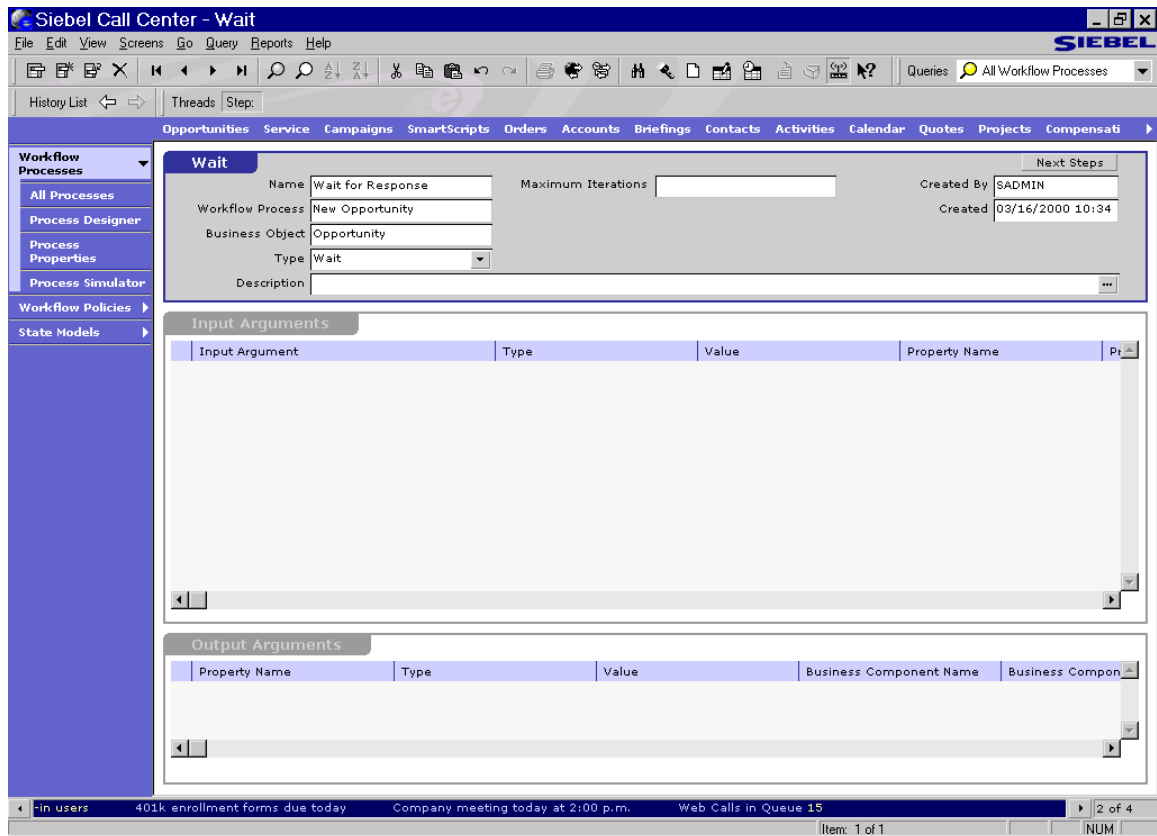


Figure 4-10. Wait View

Using the Wait View

The top portion of the view, the Wait applet, is where you can perform various tasks by clicking the appropriate button and filling in or modifying the necessary fields. The lower portion is for Input Arguments.

View Tasks

The tasks you can perform in this view are:

- **Define a wait step.** See [“Defining a Wait Step” on page 4-65.](#)
- **Delete an existing wait step.** See [“Deleting a Step or Process” on page 4-78.](#)
- **Copy a wait step.** See [“Copying a Step or Process” on page 4-78.](#)

Wait Applet Field Descriptions

[Table 4-22](#) describes the fields where you enter data in the Wait applet.

Table 4-22. Wait Applet Fields

Field	Description	Possible Value
Name	The name of the wait step.	A descriptive name that is: <ul style="list-style-type: none">■ 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 in the Process Designer view.
Description	A text narrative describing the purpose of the step.	Free-form text.
Maximum Iterations	The maximum number of times you can execute this step within a process instance.	
Created By	The name of the person who creates the step.	This value is automatically entered based on the logon name of the user.
Created	The date that the step is created.	This value is automatically entered.

Input Arguments Applet Field Descriptions

Table 4-23 describes the fields in the Input Argument applet.

Table 4-23. Input Arguments Applet Fields

Field	Description	Possible Value
Name	The name of the input argument.	
Type	The type of argument.	This is a required field. The picklist contains the following choices: <ul style="list-style-type: none"> ■ Literal ■ Process Property ■ Expression ■ Business Component
Value	A string value.	For Literal and Expression–type input arguments. This could be a picklist depending on the argument selected.
Property Name	The name of the business process property.	For Process Property–type input arguments.
Business Component	The name of a business component within the business object of the business process.	For Business Component–type input arguments.

Defining a Wait Step

To define a wait step

- 1 Make the appropriate process active by selecting it in the All Processes view.
- 2 Navigate to the Process Designer view and double-click the Wait icon to drill down to the details view.

NOTE: To add a Wait step to the workspace, drag and drop the Wait icon from the palette to the workspace.

- 3 Enter a name for the step.

- 4 Enter a description of the purpose of the step.
- 5 Enter input arguments in the Input Arguments applet.

NOTE: The wait duration is defined by entering a value for the Seconds to Sleep input argument.

NOTE: For durations greater than 60 seconds, you should specify minutes or a greater unit of measure so that business component data is refreshed.

Defining Wait Step Input Arguments

To define input arguments for a wait step

- 1 With the appropriate Wait step name displayed in the Wait Step applet of the Wait view and the Input Argument applet active, choose Edit → Add 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 selected is Literal, enter a value.
- 5 If the input argument type is Input Argument, enter the argument.
- 6 If the input argument type is Business Component, select the applicable business component name and business component field.
- 7 Enter any appropriate comments.

Working with 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
- Define input arguments for the stop step
- Define output arguments for the stop step

Figure 4-11 illustrates the Stop view.

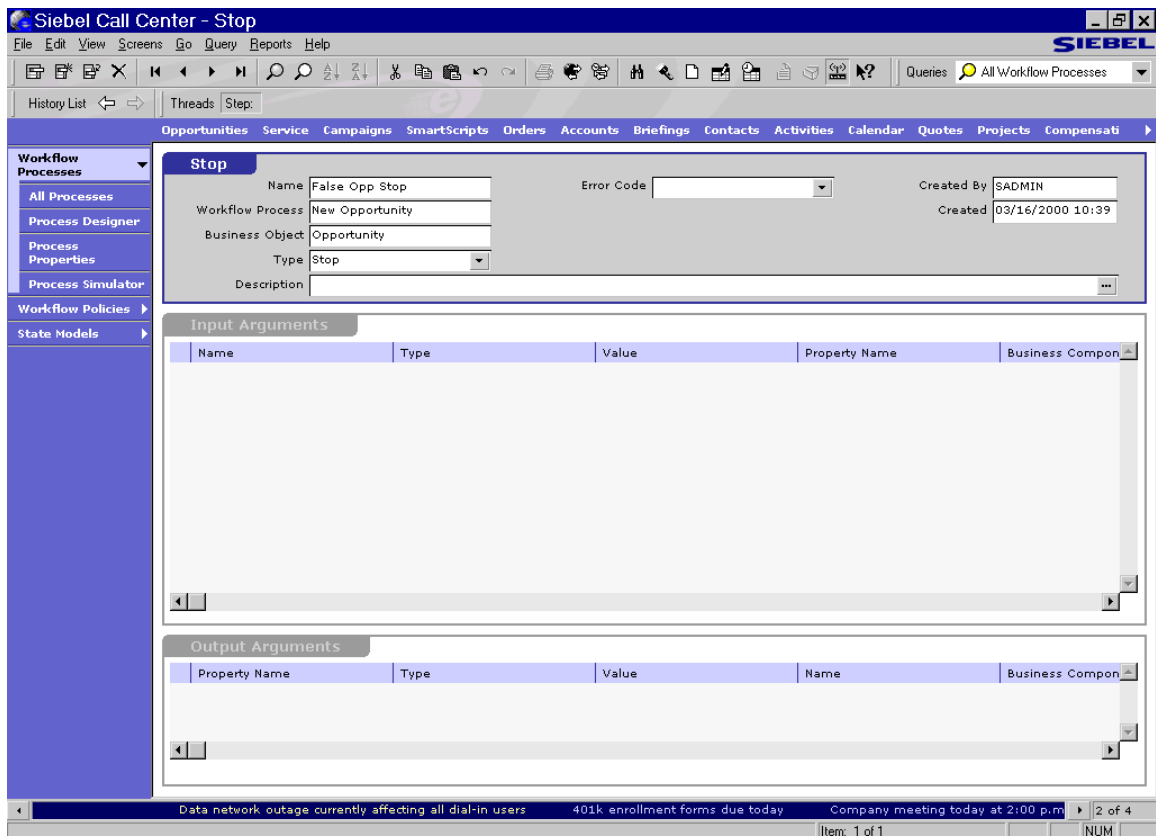


Figure 4-11. Stop View

Using the Stop View

The top portion of the view, the Stop applet, is where you can perform various tasks by clicking the appropriate button and filling in or modifying the necessary fields. The lower portion has two applets: Input Arguments and Output Arguments.

View Tasks

The tasks you can perform in this view are:

- **Define a new stop step.** See [“Defining a Stop Step” on page 4-71.](#)
- **Delete an existing stop step.** See [“Deleting a Step or Process” on page 4-78.](#)
- **Copy a stop step.** See [“Copying a Step or Process” on page 4-78.](#)

Stop Applet Field Descriptions

[Table 4-24](#) describes the fields where you enter data in the Stop applet.

Table 4-24. Stop Applet Fields

Field	Description	Possible Value
Name	The name of the business service step.	A descriptive name that is: <ul style="list-style-type: none">■ Consistent with your overall naming strategy■ Meaningful to the process designer
Type	The type of task.	This value is automatically entered when you create the step in the Process Designer view. Read-only.
Description	A text narrative describing the purpose of the task.	Free-form text.
Error Code	The name of the error code entered in the log.	This value is selected from a picklist.
Error Message	Text of the error message.	Automatically populated when an error code is selected.
Created By	The name of the person who creates the step.	This value is automatically entered based on the logon name of the user.
Created	The date that the step is created.	This value is automatically entered.

Input Arguments Applet Field Descriptions

Table 4-25 describes the fields in the Input Argument applet.

Table 4-25. Input Arguments Applet Fields

Field	Description	Possible Value
Name	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: <ul style="list-style-type: none"> ■ Literal ■ Process Property ■ Expression ■ Business Component
Value	A string value.	For Literal and Expression-type input arguments.
Property Name	The name of the business process property.	For Process Property type-input arguments.
Business Component	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-type input arguments.

Output Arguments Applet Field Descriptions

Table 4-26 describes the fields in the Output Arguments applet.

Table 4-26. 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.
Type	The type or argument.	This is a required field. The picklist contains the following choices: <ul style="list-style-type: none">■ Literal■ Process Property■ Expression■ Business Component■ Output Argument
Value	A string value.	For Literal or Expression arguments.
Name	The name of the output argument from the business procedure.	For Output Argument types.
Business Component	The name of the business component within the business object of the business process.	For Business Component types.
Business Component Field	The name of a field within the business component.	For Business Component types.

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 Make the appropriate process active by selecting it in the All Processes view.
- 2 Navigate to the Process Designer view and double-click the Stop icon to drill down to the details view.

NOTE: To add a Stop step to the workspace, drag and drop the Stop icon from the palette to the workspace.

- 3 Enter a name for the step.
- 4 Enter 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_X. 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 If you need to define input arguments for this step, continue to [“Defining Stop Step Input Arguments.”](#)
- 7 If you need to define output arguments for this step, continue to [“Defining Stop Step Output Arguments”](#) on page 4-72.

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** With the appropriate name displayed in the Stop applet of the Stop view and the Input Arguments applet active, choose Edit → Add New Record.
- 2** Enter a name for the input argument.
This should be the substitution variable appearing in the error message.
- 3** Choose an input argument type. The choices available are:
 - Literal
 - Process Property
 - Expression
 - Business Component
- 4** If the input argument type selected is Literal, enter a value.
- 5** If the input argument type is Process Property, select a property name and a property data type.
- 6** If the input argument type is Business Component, select the applicable business component name and business component field.
- 7** If the input argument type is Expression, enter the expression in the value field.
- 8** Enter any appropriate comments.

Defining Stop Step Output Arguments

To define output arguments for a stop step

- 1** With the appropriate name displayed in the Stop applet of the Stop view and the Output Arguments applet active, choose Edit → Add New Record.
- 2** Select the property name from the picklist.
- 3** Choose an output argument type. The choices available are:
 - Literal
 - Output Argument

- Expression
 - Business Component
- 4 If the output argument type selected is Literal, enter a value.
 - 5 If the output argument type is Output Argument, enter the argument.
 - 6 If the output argument type is Business Component, select the applicable business component name and business component field.
 - 7 If the output argument type is Expression, enter the expression in the value field.
 - 8 Enter any appropriate comments.

Working with Exceptions

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.

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

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

[Figure 4-12 on page 4-74](#) illustrates an exception in the Next Step applet.

Designing Workflow Processes

Working with Exceptions

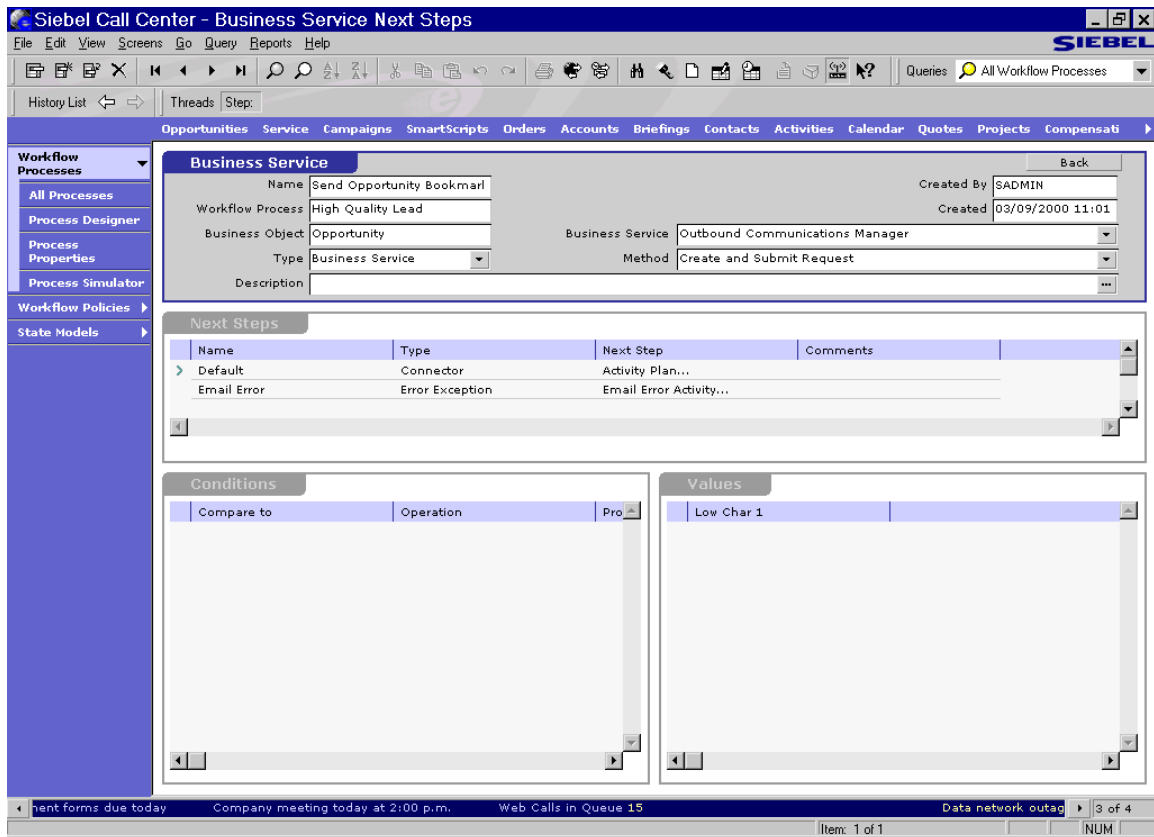


Figure 4-12. Next Step Applet for Exceptions

Using the Next Step Applet for Exceptions

An exception is a type of connector between two steps. When you double-click an exception connector, the view for the first step appears with a Next Step applet instead of the input or output argument applets.

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.

Exceptions are illustrated in the Process Designer view as red connectors. You can choose to view or not view exceptions by clicking the Show Exceptions box in the upper right portion of the Process Designer view.

Defining an Exception

Exceptions are defined in the same view as Start and Decision steps.

To define an exception

- 1** Make the appropriate process active by selecting it in the All Processes view.
- 2** 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.
- 3** Double-click the exception arrow in the workflow diagram.
- 4** In the Next Step applet, select the exception and enter a name for the exception.
- 5** Under Type, select Error Exception or User Defined Exception.
- 6** Go to the next procedure, [“Defining Exception Conditions and Values,”](#) to define the conditions that apply to the exception.

Defining Exception Conditions and Values

Conditions and values define the nature of the exception. In general, you define conditions that compare to the process properties' Error Code and Error Message when defining a system exception. Whenever the system encounters an error, the error code and error message are automatically populated in the process properties. To branch on a specific error, create a condition comparing to a process property, then select Error Code or Error Message and enter the code in the Values applet.

NOTE: It is recommended that you use Error Messages in your conditions rather than error codes to determine the next step in a Workflow Process.

To define exception conditions and values

- 1 With the exception selected in the Next Step applet and the Compare applet active, choose Edit → Add New Record.
- 2 Select a Compare to value from the picklist. The available choices are:
 - **Process Property.** Select this when you want to compare a value in a process property with a specified value in the process instance.
 - **Expression.** Select this when you want to use an expression, such as greater than or between, to evaluate a specific value.
 - **Business Component.** Select this when you want to use the value in a business component field for the condition comparison.
 - **Applet.** Select this when you want to use the value in an applet field for the condition comparison.
- 3 Select the operation to use for evaluating the values. The available choices are:
 - **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.
 - **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 If you are using a process property value, select the name of the property from the picklist.
 - 5 If you are using a business component field, select the name of the component and the name of the field from the Business Component Name and the Business Component Field picklists.
 - 6 Enter an applet name and applet field, if applicable.
 - 7 Enter any appropriate values in the Values applet.

Deleting a Step or Process

Deleting a process definition will delete all associated process instances, whether they are stopped, waiting, or running.

To back up or save the records of the process, you can query the process instances and then use the Export button on the All Processes view. This saves a copy of the process in an XML format file that you can later import back into Siebel Workflow.

NOTE: Before you delete a process, be sure that it is not a sub process in another process.

To delete a step or a process

- 1 From the Siebel Workflow Designer screens, choose the appropriate view for the step or process.

For example, to delete a process, choose the All Processes view. To delete a step, navigate to the appropriate step view by double-clicking the icon on the workflow diagram.

- 2 Locate and select the process or step you want to delete.
- 3 Click Delete.

Copying a Step or Process

To copy a step or process

- 1 Locate and select the step or process you want to copy.
- 2 Click Copy.
- 3 Enter a new name for the step or process.
- 4 Modify the other definition fields as necessary for the new step or process.

Activating a Process

When you have defined and tested a workflow process, you can make it available for use by changing the status to Active.

To activate a process

- 1 Select the process in the All Processes view.
- 2 Click the Activate button.

This compiles the process, checks the syntax for validity, and then changes the status of the process to Active.

Modifying a Process Definition

For changes to take effect after you modify a process definition, you must deactivate and then activate the workflow process.

To modify a process definition

- 1 Change the status of the workflow process to Inactive by selecting the process in the All Processes view and clicking the Deactivate button.
- 2 Make the necessary modifications to your process definition.
- 3 Make the process available for use again by changing the status to Active. To do this, select the process in the All Processes view and click the Activate button.

Testing Workflow Processes

5

Testing Processes with the Process Simulator	5-2
Using the Process Simulator View	5-3

Testing Processes with the Process Simulator

Testing your workflow processes before migrating them to your production environment ensures 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 [Chapter 3, “Planning for Workflow Processes.”](#)



Caution: Your test environment and production environment must have identical versions of the software.

Of the various ways to invoke a workflow process, using the Process Simulator provides an easy way to debug a workflow process from the Siebel Workflow Designer screens. You can debug process steps as you define them by switching from the Process Designer view to the Process Simulator view and back again.

NOTE: When the Workflow Process is run from the Workflow Process Simulator, it runs in the Siebel Client's Object Manager

The other methods involve invoking a workflow process outside of the Siebel Workflow Designer screens. For information on these methods of invoking a workflow process, see [“Invoking a Workflow Process” on page 6-2](#). Invoking a workflow process from a server component is covered in *Siebel eBusiness Application Integration Guide* and *Siebel eMail Response Guide*.

Using the Process Simulator View

The Process Simulator view is similar in appearance to the Process Designer view. It allows you to step through a process while viewing the results of each step.

Figure 5-1 illustrates the Process Simulator view.

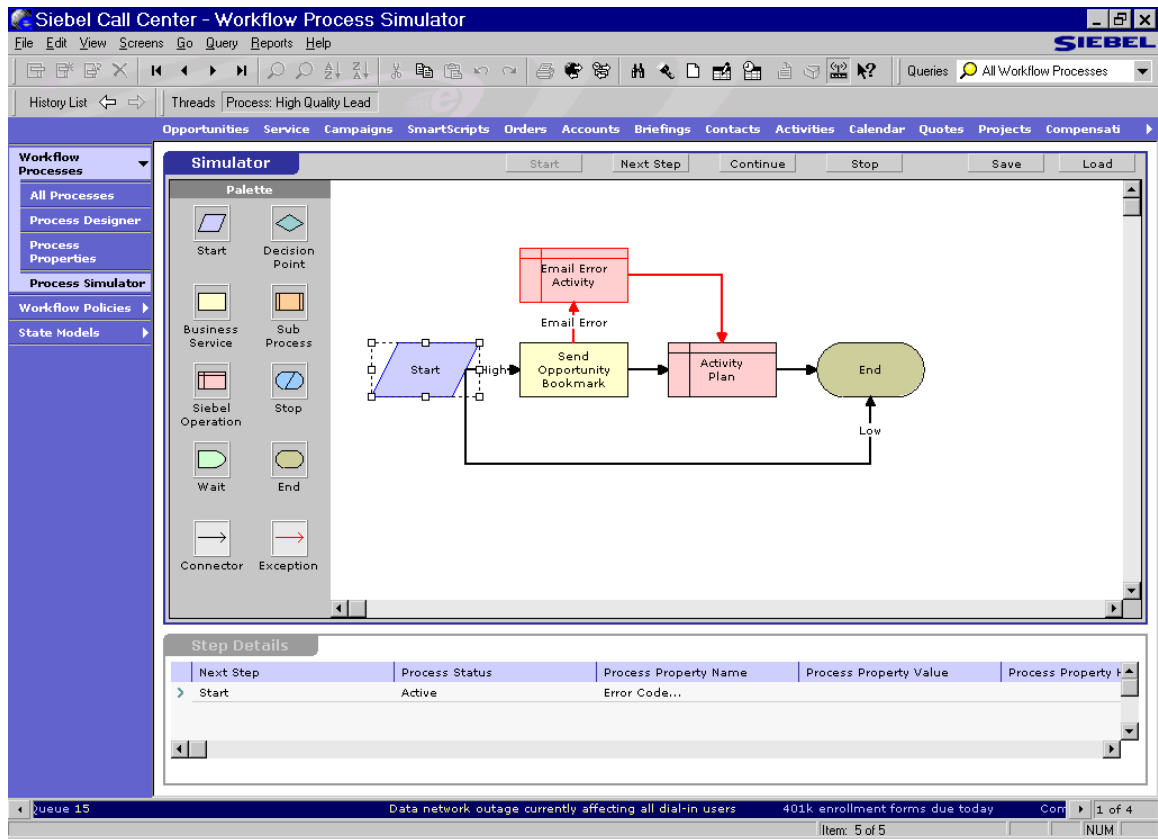


Figure 5-1. Process Simulator View

The top portion of the view, the Simulator applet, is where you can incrementally move through the steps of a process to verify that they are working correctly. The lower portion of the view features the Step Detail applet. This applet displays the results of each step.

A process does not have to be active to run it in the simulator. The simulator ignores activation date, expiration date, and status.

Simulator Applet

The top applet on the Process Simulator view displays the process design for the selected workflow process. The design is shown in a read-only mode in this view, but you can drill down to the detail steps by double-clicking the step icons.

This applet also contains the buttons for running the process. [Table 5-1](#) describes the buttons on the Simulator applet.

Table 5-1. Simulator Applet

Item	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.

Step Details Applet Field Descriptions

As a workflow process is being executed in the simulator, the step details are populated in the Step Details applet on the lower portion of the view. These details are shown in read-only mode.

Table 5-2 describes the fields in the Step Details applet.

Table 5-2. Step Details Applet 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.

To debug a process with the Process Simulator

NOTE: The Process Simulator will not work with processes that have run time events associated with them, for example, an event defined in the Start Step, Wait Step, or User Interact Step.



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.

- 1 Select the process to debug in the All Processes view.
- 2 Navigate to the Process Properties view and enter a valid row id in the default string field for the object id property.

To locate a valid row id for the object, you can select the object from any view in the Siebel application and navigate to Help → About Record. Click the Details button and the row id is displayed in the Row # field. You should select a row id for a record in the primary business component for the workflow process business object. The simulator tests your process on this record.

- 3** Click Process Simulator in the Siebel Workflow navigation bar.

The Process Simulator view appears with the selected process flow displayed in the workspace.

- 4** Click Start.

The Start step is executed, and the results are shown in the Step Details applet.

- 5** If the first step executes as expected, click Next Step to execute the next step in the process.

- 6** Continue stepping through the process and verifying the results of each step in the Step Details applet until the process completes.

You can at any time go to the Process Designer view, drill down to the step details to make modifications as necessary, and then return to the Process Simulator to debug the process.

Running Workflow Processes

6

Invoking a Workflow Process	6-2
Invoking from a Workflow Policy	6-2
Invoking from a Script	6-4
Running a Workflow Process on the Server	6-6
Running a Workflow Process on the Object Manager	6-7
Running a Workflow Process in Batch	6-8

Invoking a Workflow Process

A workflow process can be invoked in the following ways:

- From the Process Simulator view
- From a workflow policy
- From a script
- From a server component

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 5, “Testing Workflow Processes,”](#) invoking a process from the Process Simulator view is described. *Siebel eBusiness Application Integration Guide* and *Siebel eMail Response Guide* discuss invoking a process from a server component. The other invocation methods are explained in this chapter.

Invoking from a Workflow Policy

To invoke a workflow process from a workflow policy, you define a policy 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 [Chapter 9, “Creating Workflow Policies.”](#)

To invoke a process from a workflow policy

- 1** From the Siebel Workflow Designer, navigate to the Workflow Policies Action view and define a new action.
 - a** In the Program field, use the picklist to select the Run Workflow Process program.
-
- NOTE:** If you are invoking a workflow process containing eBusiness Application Integration steps, select the Run Integration Process program.
-
- b** In the Arguments applet, select ProcessName and enter the name of the workflow process you want to invoke as the argument value.
- 2** From the Siebel Workflow Designer, navigate to the Workflow Policies Group view and define a new group.
 - 3** From the Siebel Workflow Designer, navigate to the Workflow Policies Policies view and define a new policy.
 - a** In the Conditions applet, define a condition for the policy that must be met to invoke the workflow process.
 - b** In the Actions applet, enter the name of the action you defined in step 1.
 - 4** Run Gentrig.

See [“Creating Database Triggers” on page 11-4](#) for more information about running the trigger generator.
 - 5** Run Workmon.

See [“Using the Workflow Monitor Agent” on page 11-20](#) for more information about monitoring workflow policies.
 - 6** Initiate the policy. The action should invoke the workflow process.

Invoking from a Script

Workflow processes can be invoked programatically 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.

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 object manager, call the service “Workflow Process Manager.”

Example: Invoking Process 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.

```
function Invoke_Process ()
{
    var svc;
    var input;
    var output;
    var rowid;
    var buscomp;

    svc = TheApplication().GetService("Workflow Process Manager");
    input = TheApplication().NewPropertySet();
    output = TheApplication().NewPropertySet();
    buscomp = TheApplication().ActiveBusComp();
    rowid = buscomp.GetFieldValue("Id");
    input.SetProperty("ProcessName", "My Account Process");
    input.SetProperty("RowId", rowid);
    svc.InvokeMethod("RunProcess", input, output);
}
```


Example: Passing 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.

```
function Invoke_Process ()
{
    var svc;
    var input;
    var output;
    var rowid;
    var accountid;
    var buscomp;

    svc = TheApplication().GetService("Workflow Process Manager");
    input = TheApplication().NewPropertySet();
    output = TheApplication().NewPropertySet();
    buscomp = TheApplication().ActiveBusComp();
    rowid = buscomp.GetFieldValue("Id");
    accountid = buscomp.GetFieldValue("Account Id");
    input.SetProperty("ProcessName", "My Opportunity Process");
    input.SetProperty("RowId", rowid);

    // Pass value of Account Id field to the Account Id process
    property
    input.SetProperty("Account Id", accountid);
    svc.InvokeMethod("RunProcess", input, output);
}
```

Running a Workflow Process on the Server

To run a workflow process on a server, you must have a Siebel server available. The methods of invoking a workflow process that support running a process on the server include:

- From a workflow policy
- From a script

If a user invokes a process to be run on the server, the process executes only if the user is connected to the server. If the user is not connected to the server or the Siebel server is not running, the request is queued and executes when the user synchronizes or the server becomes available.

For the workflow process to be executed successfully on the server, the Workflow Process Manager must be available and the Server Request Manager must be running.

To verify these processes, navigate to Server Administration → Servers → Server Components.

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 server components).

NOTE: Navigation steps are not supported when processes run on the server. For example, the Go to View method on the Workflow UI Utilities business service results in an error if it is run on the server.

Running a Workflow Process on the Object Manager

Running a workflow process on the Object Manager does not require a Siebel server or connectivity to a server. For a mobile user, the Object Manager is on the laptop. For a connected user, the Object Manager is also on the user's machine. For thin clients and HTML clients, the Object Manager is on a Siebel server.

The methods that support invoking a workflow process on the Object Manager include:

- From the Process Simulator
- From a script

Running a workflow process on the Object Manager can be very useful for enforcing business processes with mobile users or for defining business processes that involve end-user navigation.

Running a Workflow Process in Batch

Workflow processes can be run in batch mode by invoking the Method Execute Batch Process on the Workflow Process Manager business service or by running the Workflow Process Manager Batch 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.

For more information on running a workflow process in batch, see *Siebel Server Administration Guide*.

Migrating to the Production Environment

7

Migrating Workflow Processes to Production	7-2
Importing or Exporting a Process Definition	7-2

Migrating Workflow Processes to Production

Once you have tested your workflow process in your development environment, you can move them to the production environment. To move workflow processes from one environment to another, you can use the Import and Export function buttons on the All Processes view.

Import and export is also useful for managing different versions of workflow processes and backing up old processes.

Importing or Exporting a Process Definition

When importing a process definition, you should make sure that you are not inadvertently importing over an existing process definition. It is a good idea to back up your process definitions regularly using the Export function. You should use a meaningful naming convention when selecting a file name for an exported process to make it easy to understand the purpose of the process.

To export a process definition

- 1 Navigate to Siebel Workflow Administration → Workflow Processes → All Processes.
- 2 Select the process you want to export.
- 3 Click the Export button.
- 4 Select a path and file name for the process.
- 5 Click Save.

NOTE: After exporting a process containing sub processes, you must individually export the sub processes. Sub processes are *not* exported automatically.

To import a process definition

- 1 Navigate to Siebel Workflow Administration → Workflow Processes → All Processes.
- 2 Click the Import button.
- 3 Select a path and file name of the process to import.
- 4 Click Open.

Chapter 8. Planning for Workflow Policies

Chapter 9. Creating Workflow Policies

Chapter 10. Customizing Workflow Policies with Siebel Tools

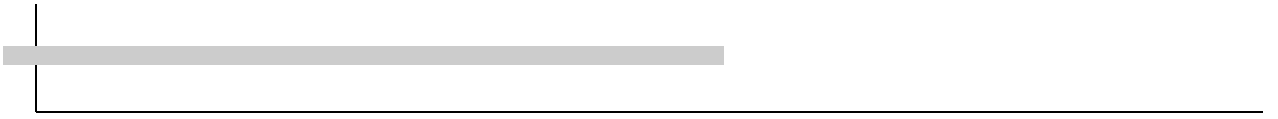
Chapter 11. Workflow Policies Server Administration

Chapter 12. Workflow Policies and Siebel Marketing

Chapter 13. Testing Workflow Policies

Chapter 14. Migrating Policies to the Production Environment

Chapter 15. Monitoring and Performance Tuning



Planning for Workflow Policies

8

Planning Workflow Policy Groups	8-2
Planning for Workflow Policies	8-3
Determining What to Monitor	8-4
Planning Policies and Conditions	8-4
Planning Workflow Policy Actions	8-5
Planning Scenarios	8-6
Defining a Test and Migration Strategy	8-9

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, all your policies should be assigned 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. For a complete discussion of workflow policy groups and how to use them to improve your system's performance, see [Chapter 15, "Monitoring and Performance Tuning."](#)

Planning for 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 [Chapter 10, “Customizing Workflow Policies with Siebel Tools”](#) 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 [Chapter 9, “Creating Workflow Policies,”](#) for more information on creating workflow policies.

NOTE: After planning a new workflow policy, it is always a good idea to test the policy definition by creating a query based on the policy, which you then run against 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 [Chapter 13, “Testing Workflow Policies,”](#) for more information.

Determining What to Monitor

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:

Table 8-1. 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 8-2](#) 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 8-2. Planning Policies

Name	Workflow Object	Workflow Group	Activation/Expiration Date/Time	Duration	Quantity	Comments
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 8-3](#). Conditions are in the form of an expression.

Table 8-3. 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 8-4](#) illustrates the type of information you need to define a workflow policy action.

Table 8-4. Workflow Policy Actions

Action Name	Program	Workflow Object	Arguments	Comments
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.

Planning Scenarios

The scenarios below provide you with examples of the type of information needed for various types of policies.

Scenario 1: Notification for 30%+ Discounts

In this scenario, the sales department manager wants to be automatically notified whenever sales representatives quote discounts over 30%.

Table 8-5. 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 8-6. Planning Policies

Name	Workflow Object	Workflow Group	Activation/Expiration Date/Time	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 8-6 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 8-7 illustrates the type of information you need for the policy conditions.

Table 8-7. 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 8-8. Actions and Action Arguments

Action Name	Program	Workflow Object	Arguments and Substitutions	Comments
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 2: 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 8-9 represents the general policy definition.

Table 8-9. 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 8-10. Workflow Policies

Name	Workflow Object	Workflow Group	Activation/ Expiration Date/Time	Duration	Quantity	Comments
Over 20 Open Service Requests	Service Request	High Frequency			20	

Table 8-10 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 8-11. 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 8-12. Actions and Action Arguments

Action Name	Program	Workflow Object	Arguments and Substitutions	Comments
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

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 ensures 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 [Chapter 13, “Testing Workflow Policies,”](#) and [Chapter 14, “Migrating Policies to the Production Environment.”](#)

Planning for Workflow Policies

Defining a Test and Migration Strategy

Creating Workflow Policies

9

Using the Workflow Policies Views	9-2
Working with Workflow Policy Actions	9-3
Understanding the Actions Applet	9-4
Understanding the Arguments Applet	9-5
Understanding the Recipients Applet	9-13
Creating a Workflow Policy Action	9-15
Siebel Workflow Policy Action Examples	9-16
Creating a Send Page Action	9-16
Creating a Send Email Action with a Repeating Message	9-17
Creating a Send Message Broadcast Action	9-19
Creating a Database Operation Action	9-20
Creating a Run External Program Action	9-21
Working with Workflow Policy Groups	9-23
Understanding the Group Applet	9-25
Understanding the Policies Applet	9-25
Creating a Workflow Policy Group	9-26
Working with Workflow Policies	9-26
Understanding the Policies Applet	9-27
Understanding the Conditions Applet	9-29
Understanding the Actions Applet	9-34
Creating a Workflow Policy	9-35
Siebel Workflow Policy Examples	9-36
Creating a Send Page Workflow Policy	9-36
Creating a Send Email Workflow Policy	9-37

Using the Workflow Policies Views

The views you use to create and define workflow policies are a part of the Siebel Workflow Designer. To display the Workflow Policies view menu, you click Workflow Policies on the Siebel Workflow Designer navigation bar.

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.
- **Workflow Policies Groups view.** Use to create the workflow policy groups to use with workflow policies.
- **Workflow Policies Policies view.** Use to create workflow policies.
- **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.
- **Workflow Policies Policy Trend Analysis.** Use to review the workflow policy monitor log and to check policy trends.
- **Workflow Policies Policy Frequency Analysis.** Use to review the workflow policy monitor log and to check policy frequency.

Working with 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 Workflow 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 9-15](#).

The Workflow Policies Actions 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.
- **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.
- **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.

Creating Workflow Policies

Working with Workflow Policy Actions

Understanding the Actions Applet

Figure 9-1 shows the Workflow Policies Actions view.

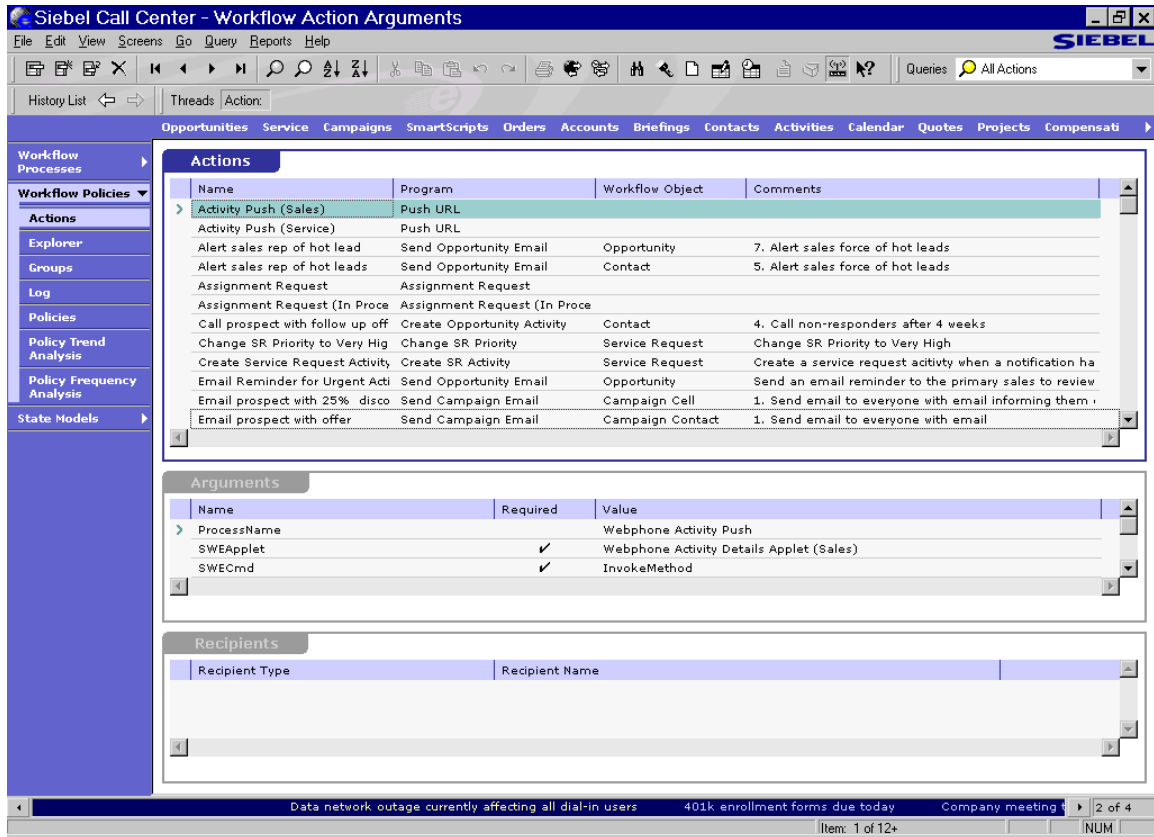


Figure 9-1. Workflow Policies Action View

Table 9-1 describes the fields of the Workflow Policies Actions applet.

Table 9-1. Actions Applet Fields

Field	Description	Possible Value
Name	The name of the workflow policy action.	A descriptive name that is: <ul style="list-style-type: none"> ■ 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 Appendix B, "Predefined Programs."
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.

Understanding the Arguments Applet

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 9-6.
- **Send Message.** See ["Using the Send Message Program Type"](#) on page 9-8.
- **Message Broadcast.** See ["Using the Message Broadcast Program Type"](#) on page 9-9.

- **Run External.** See “Using the Run External Programs Type” on page 9-10.
- **Database Operation.** See “Using the Database Operation Program Type” on page 9-12.

NOTE: Before using the email or paging functions, you need to perform the setup procedures described in [Chapter 11, “Workflow Policies Server Administration.”](#)

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, as shown in [Figure 9-2](#).

The screenshot shows the 'Send Page Arguments' applet. It is divided into two main sections: 'Send Page Arguments' and 'Recipients'.

Send Page Arguments:

- Numeric Message Template:** A dropdown menu.
- Alpha Message Template:** A text area containing the text: "An opportunity [Opportunity] for account [Account] has been added in Siebel."
- Request Key:** A text input field.
- Available Substitutions:** A list box containing: Opportunity, Account, Site, Created, Last User Login Name, Last User First Name, Last User Last Name.

Recipients:

Recipient Type	Recipient Name
> Send to Relative	Primary Sales Rep

Figure 9-2. Send Page Arguments Applet

Send Page Arguments and Values

[Table 9-2 on page 9-7](#) 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 9-2. 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, as shown in [Figure 9-3](#).

The screenshot displays two applets. The top applet, 'Send Message Arguments', has a 'Subject' field with the text 'Class Confirmation: [Course Name]', a 'Message Template' field with a preview of an email message, and 'Repeating Message' and 'Request Key' fields. The bottom applet, 'Recipients', is a table with two columns: 'Recipient Type' and 'Recipient Name'. It contains one row with 'Send to Relative' and 'Employee Email'.

Figure 9-3. Send Message Arguments 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 9-3](#) shows the arguments and valid values for the Send Email workflow policy program type.

Table 9-3. Send Email Workflow Policy Program Type (1 of 2)

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.

Table 9-3. Send Email Workflow Policy Program Type (Continued) (2 of 2)

Argument	Valid Values When Used by Action
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. [Figure 9-4](#) shows the Send Message Broadcast Arguments applet.

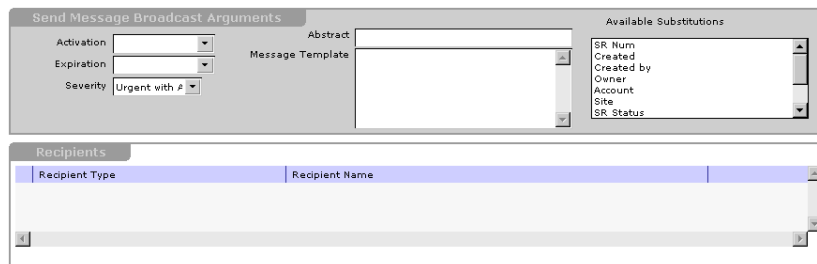


Figure 9-4. Send Message Broadcast Arguments Applet

Send Message Broadcast Arguments and Values

Table 9-4 shows the arguments and valid values for the Message Broadcast workflow policy program type.

Table 9-4. 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 9-34 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 9-34 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.

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. [Figure 9-5 on page 9-11](#) shows the External Programs Arguments applet.

An example of a Run External Program is described in “Creating a Run External Program Action” on page 9-21.

Figure 9-5. External Program Arguments Applet

Run External Arguments and Values

Table 9-5 shows the arguments and valid values for the Run External workflow policy program.

Table 9-5. 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	<ul style="list-style-type: none"> ■ Wait. Siebel 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.¹ ■ No Wait. Siebel Workflow Policies executes the external program in the background and then continues processing. The return code is not checked.
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.

1. For Visual Basic programs which create files, Execute Type should be set to Wait to avoid possible corruption of files. When set to No Wait, Visual Basic attempts to write files twice, thus corrupting the data.

Creating Workflow Policies

Working with Workflow Policy Actions

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 Workflow 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. [Figure 9-6](#) shows the Arguments applet.

Name	Required	Value
> New Priority	<input checked="" type="checkbox"/>	Very High

Recipient Type	Recipient Name
----------------	----------------

Figure 9-6. Database Arguments Applet

Database Operation Arguments and Values

Table 9-6 shows the arguments and valid values for the Database Operation workflow policy program.

Table 9-6. 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].

Understanding the Recipients Applet

Table 9-7 describes the fields in the Recipients applet.

Table 9-7. 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.	<ul style="list-style-type: none"> ■ 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 to.

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). You use Send to Relative when you want to infer whom to send your action to based on the workflow policy object.

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** Navigate to Siebel Workflow Administration → Workflow Policies → Actions.
- 2** In the Actions applet, select Edit → Add New Record 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 [“Understanding the Arguments Applet” on page 9-5](#) 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.

Siebel Workflow Policy Action Examples

This section provides you with several examples of workflow policy actions for specific situations. You can use these examples as the basis for creating your own workflow policy actions.

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.

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.

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 a critical mass of 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** Fill in the Send Message Broadcast Arguments applet 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`

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`

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 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** Navigate to Screens → 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);
}
else
{
TheApplication().MsgBox("No 'Program' specified");
}
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.

Working with 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.
- **Policies applet.** Lists the workflow policies assigned to the selected group.

How you use workflow policy groups to enhance performance is more fully described in [Chapter 15, “Monitoring and Performance Tuning.”](#)

Creating Workflow Policies

Working with Workflow Policy Groups

Figure 9-7 shows the Workflow Policies Groups view.

The screenshot displays the Siebel Call Center - Workflow Groups interface. The left sidebar contains navigation options: Workflow Processes, Workflow Policies (selected), Actions, Explorer, Groups, Log, Policies, Policy Trend Analysis, Policy Frequency Analysis, and State Models. The main content area is divided into two sections: 'Workflow Groups' and 'Policies'.

Workflow Groups Table:

Name	Comments
High Frequency	
Medium Frequency	
Low Frequency	

Policies Table:

Name	Object	Activation Date/Time	Expiration Date/Time	Comments
Expired Service Agreement	Service Request			Send an Alert to t
Notification of Critical SR	Service Request			
Page SR Owner (Gold)	Service Request			Page SR Owner if
Page SR Owner (Silver)	Service Request			Page SR owner if

The status bar at the bottom shows: enrollment forms due today, Company meeting today at 2:00 p.m., Web Calls in Queue 15, Data network, 3 of 4, Item: 1 of 3+, and NUM.

Figure 9-7. Workflow Policies Groups View

Understanding the Group Applet

Table 9-8 describes the fields in the Workflow Groups applet.

Table 9-8. 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.

Understanding the Policies Applet

Table 9-9 describes the fields in the Policies applet of the Workflow Policies Groups view.

Table 9-9. 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 a Workflow Policy Group

To create a policy group

- 1 Navigate to Workflow Administration → Workflow Policies → Groups.
- 2 Select Edit → New Record and enter the name for the group in the Workflow Groups applet.
- 3 Enter comments in the Comments field (optional).

Working with Workflow Policies

After creating your workflow policy actions and workflow policy groups, you are ready to go to the Workflow Policies 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 Policies view is made up of four applets:

- **Policies 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.
- **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.
- **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.
- **Arguments applet.** Where you can review the workflow policy action arguments.

Understanding the Policies Applet

Figure 9-8 shows the Policies applet in the Workflow Policies Policies view.

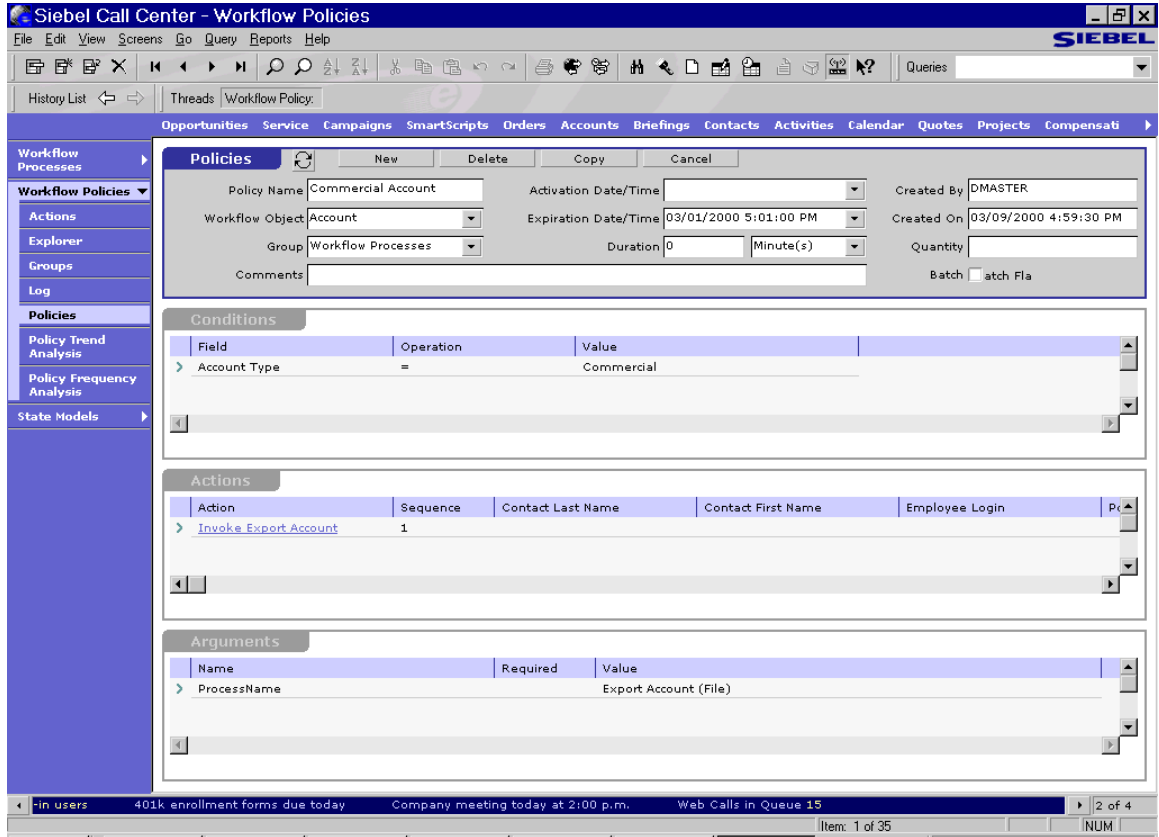


Figure 9-8. Workflow Policies Policies View

Table 9-10 defines the Policies applet fields.

Table 9-10. 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: If a Workflow Policy needs to be moved from one group to another group, all requests associated with that Workflow Policy must complete before this change can occur. 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.

Understanding the Conditions Applet

Table 9-11 defines the Conditions applet fields in the Workflow Policies Policies view.

Table 9-11. 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,” next.
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,” next, and “Entering Date Calculations” on page 9-34.

Using Comparison Values in the Conditions Applet

You use comparison values in the Operation field, as shown in [Figure 9-9](#). The field exposes the Workflow policy component column for monitoring.

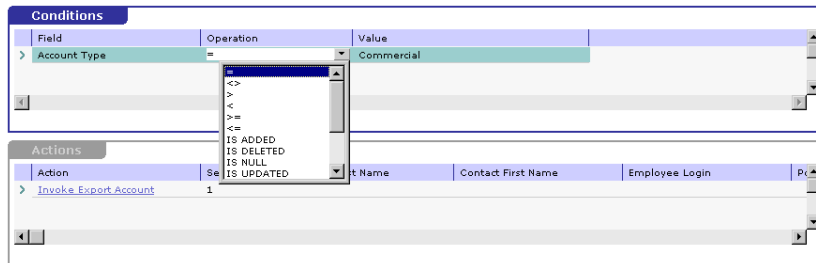


Figure 9-9. Comparison Values in the Conditions Applet

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.

The syntax for an MS SQL server is similar but requires the use of the convert function.

NOTE: It is up to the policy creator to make sure the syntax is correct. Siebel Workflow only passes the BETWEEN clause to the database. It does not verify syntax.

LIKE and NOT LIKE allow you to use wildcards, for example, LIKE Smith% or NOT LIKE Sm%th%.

Table 9-12 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 9-12. 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 9-13 describes the specialized comparisons for all database platforms that can be used in creating workflow conditions.

Table 9-13. Specialized Comparisons

Comparison	Value
IS ADDED	<p>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.</p> <p>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.</p>
IS UPDATED	<p>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.</p> <p>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.</p>
IS DELETED	<p>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.</p> <p>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.</p>

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

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.

NOTE: To have CURRENT compare the date only and not the time, set the Physical Type to Date for the workflow column under Table → Column in Siebel Tools.

Understanding the Actions Applet

Table 9-14 defines the Actions applet fields on the Workflow Policies Policies view.

Table 9-14. 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 Workflow 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.

Creating a Workflow Policy

To create a Workflow policy

- 1** Navigate to Siebel Workflow Administration → Workflow Policies → Policies.
- 2** In the Policies applet, 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.

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.

Siebel Workflow Policy Examples

This section provides several examples of creating workflow policies for specific situations. You can use these examples as the basis for creating your own policies.

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.

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** Navigate to the Workflow Policies Policies view.
- 2** Click New and fill in the Policies applet as follows:
 - a** The Policy Name: Page support manager
 - b** Workflow Object: Service Request
 - c** Group: High Frequency
 - d** Duration: 2 hours

- 3 Fill in the Conditions applet:
 - a Service Request Priority = Very High
 - b Service Request Owner IS NULL
- 4 Fill in the Actions applet with the name of the appropriate Send Page action.

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 Navigate to the Workflow Policies Policies view.
- 2 Click New and fill in the Policies applet as follows:
 - a Policy Name: Very High Value Opportunity
 - b Workflow Object: Opportunity
 - c 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.

- 3 Fill in the Conditions applet with the following:

Field	Comparison	Value
Forecast Probability	< =	50
Forecast Revenue	>	400000

- 4** 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 [Chapter 11, “Workflow Policies Server Administration.”](#)

Customizing Workflow Policies with Siebel Tools **10**

Understanding Siebel Tools	10-3
Understanding Siebel Tools and Workflow Policies	10-6
Working with Workflow Policy Objects	10-9
Using the Siebel Tools Views	10-12
Understanding the Workflow Policy Column List View	10-13
Understanding the Workflow Policy Object List View	10-15
Understanding the Workflow Policy Component List View	10-16
Understanding the Workflow Policy Component Columns View	10-19
Defining a Workflow Policy Column	10-21
Defining a Workflow Policy Component	10-22
Defining a Workflow Policy Object	10-23
Modifying Policy Column Names	10-24
Adding Policy Columns to a Workflow Policy Object	10-24
Associating a Column with a Workflow Policy Component	10-26
Using the Validate Tool	10-27
Modifying an Existing Workflow Policy Object	10-29
Understanding Workflow Policy Programs	10-33
Using the Program List View	10-34
Using the Workflow Policy Program Argument List View	10-36
Defining a Workflow Policy Program	10-41
Example of Creating a Program Argument	10-43
Creating SQL Statements for Program Arguments	10-44

Using Predefined Workflow Policy Programs	10-44
Using Change SR Close Date to Today	10-45
Using Change SR Owner	10-47
Using Change SR Owner to Manager	10-49
Using Send Quote Page	10-51
Making Object Types Available in Siebel Client	10-53

Understanding Siebel Tools

Siebel Tools is the Siebel solution where 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 *Siebel Tools Guide*.

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, as shown in [Figure 10-1 on page 10-4](#). 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.

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.

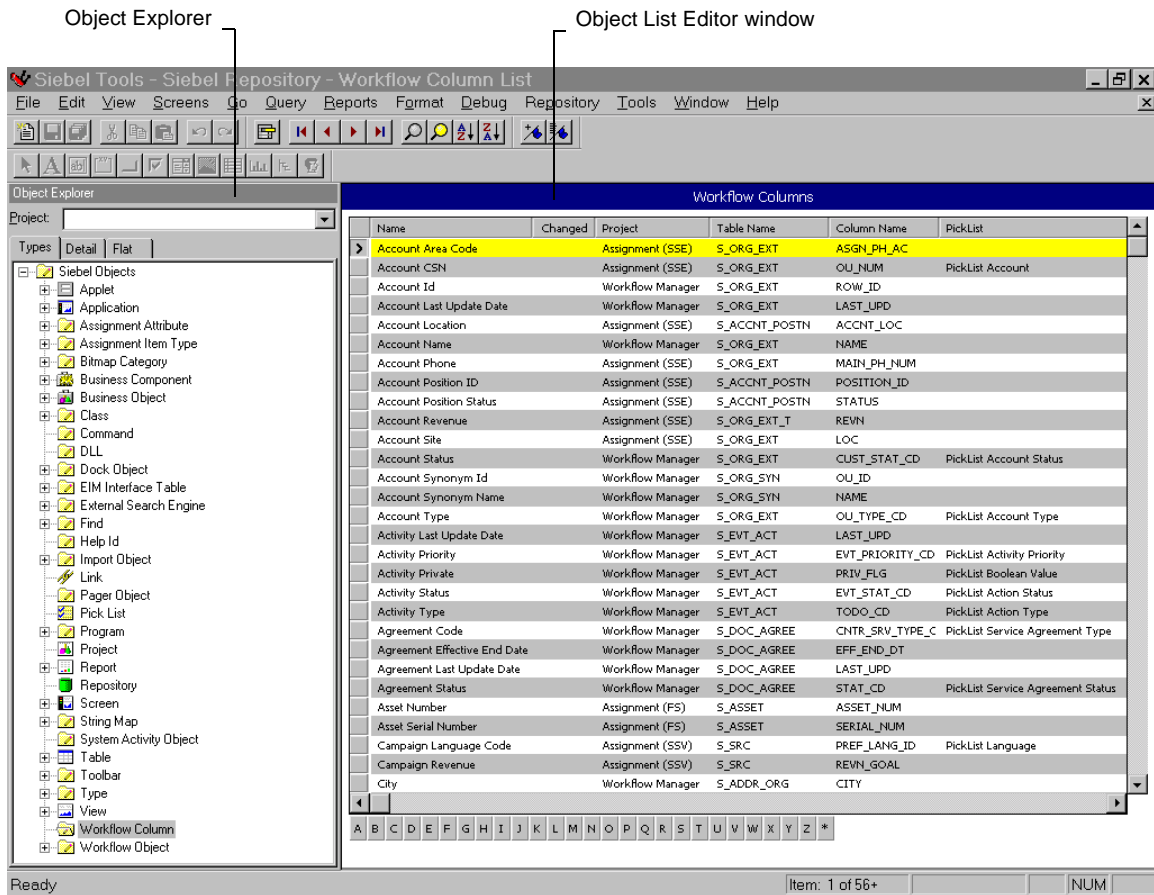


Figure 10-1. Object Explorer and Object List Editor Window

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 10-2 shows parent-child relationships in the Object Explorer. A parent object type can have multiple child object types.

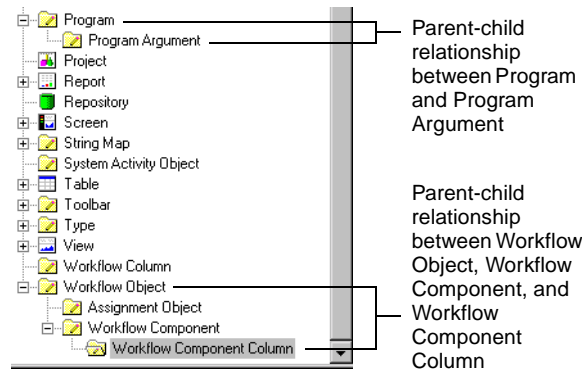


Figure 10-2. Parent-Child Object Type Relationships in the Object Explorer

Siebel Workflow Policies accesses the following object types to create workflow policies, workflow actions, and workflow conditions:

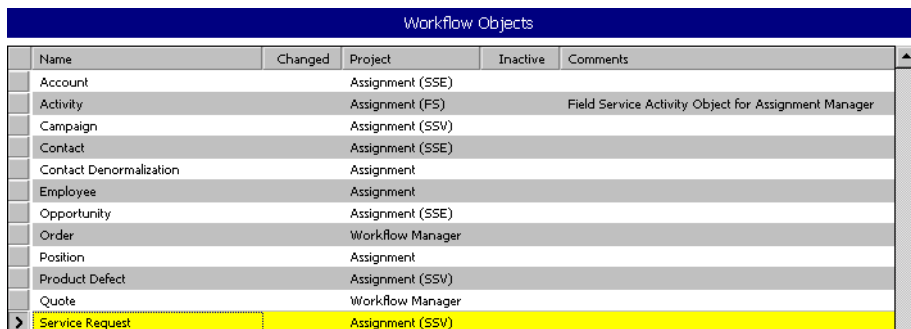
- Workflow policy program
- Workflow policy program arguments
- Workflow policy column
- Workflow policy object
- Workflow policy component
- Workflow component column

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.

Understanding Siebel Tools and Workflow Policies

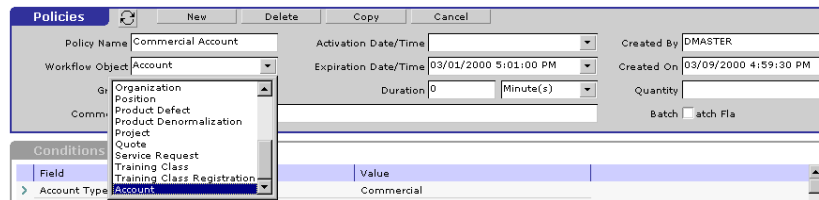
The Workflow objects that were defined in Siebel Tools are displayed in Workflow Policies in the Workflow Object picklist in the Workflow Policies Policies view. As [Figure 10-3](#) shows, the Workflow policy objects in Siebel Tools are exposed in the Workflow Object picklist in Workflow Policies.

Siebel Tools Workflow policy objects



Name	Changed	Project	Inactive	Comments
Account		Assignment (SSE)		
Activity		Assignment (FS)		Field Service Activity Object for Assignment Manager
Campaign		Assignment (SSV)		
Contact		Assignment (SSE)		
Contact Denormalization		Assignment		
Employee		Assignment		
Opportunity		Assignment (SSE)		
Order		Workflow Manager		
Position		Assignment		
Product Defect		Assignment (SSV)		
Quote		Workflow Manager		
Service Request		Assignment (SSV)		

Workflow Object picklist in Workflow Policies



Policies [New] [Delete] [Copy] [Cancel]

Policy Name: Commercial Account Activation Date/Time: [Dropdown] Created By: DMASTER

Workflow Object: Account Expiration Date/Time: 03/01/2000 5:01:00 PM Created On: 03/09/2000 4:59:30 PM

Organization: [Dropdown] Duration: 0 Minute(s) Quantity: [Input]

Comments: [Text Area]

Batch: [Input] Attach Fla: [Input]

Conditions

Field	Value
Account Type	Account
	Commercial

Figure 10-3. Workflow Policy Objects in Siebel Tools and Workflow Object Picklist in Workflow Policies

The workflow policy component columns defined in Siebel Tools are available to Workflow Policies views. As shown in Figure 10-4, the workflow policy component columns in Siebel Tools are exposed in Workflow Policies in the Condition Field picklist in the Workflow Policies Policies view.

Siebel Tools Workflow policy component columns

Workflow Component Columns				
Workflow Column Name	Alias	Changed	Inactive	Comments
Service Request Area	Service Request Area			
Service Request Last Update Date	Service Request Component			
Service Request Owner	Service Request Owner			
Service Request Owner Group	Service Request Owner Group			
Service Request Priority	Service Request Priority			
> Service Request Severity	Service Request Severity			
Service Request Source	Service Request Source			
Service Request Status	Service Request Status			
Service Request Sub-area	Service Request Sub-area			
Service Request Sub-status	Service Request Sub-status			

Workflow Policies Workflow Policy Component Columns picklist

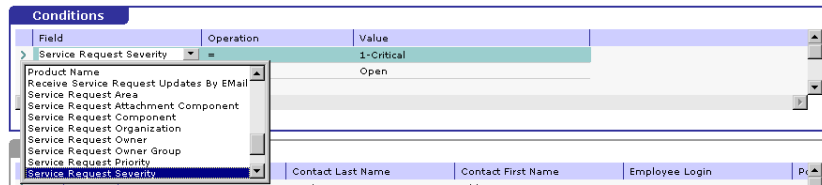


Figure 10-4. Workflow Policy Component Columns in Siebel Tools and Workflow Policy Component Columns Picklist in Workflow Policies

Customizing Workflow Policies with Siebel Tools

Understanding Siebel Tools

The policy programs defined in Siebel Tools are available in Workflow Policies views. As shown in [Figure 10-5](#), the policy programs in Siebel Tools are exposed in Workflow Policies in the Program Field picklist in the Workflow Policies Actions view.

Siebel Tools programs

Programs				
Name	Changed	Project	Type	Workflow Object
Assignment Request		Workflow Manager	Generic Request Server	
Change SR Close Date to Today		Workflow Manager	DB Operation	Service Request
Change SR Group		Workflow Manager	DB Operation	Service Request
Change SR Owner		Workflow Manager	DB Operation	Service Request
Change SR Owner to Manager		Workflow Manager	DB Operation	Service Request
Change SR Priority		Workflow Manager	DB Operation	Service Request
Change SR Severity		Workflow Manager	DB Operation	Service Request
Change SR Status		Workflow Manager	DB Operation	Service Request
Change SR Sub-status		Workflow Manager	DB Operation	Service Request
Create Opportunity Activity		Workflow Manager	DB Operation	Opportunity
Create SR Activity		Workflow Manager	DB Operation	Service Request
Generic Request		Workflow Manager	Generic Request Server	
Run External Program		Workflow Manager	External Program	
Send Broadcast Message		Workflow Manager	Send Broadcast Message	
Send Email		Workflow Manager	Send Message	

Workflow Policies Program picklist

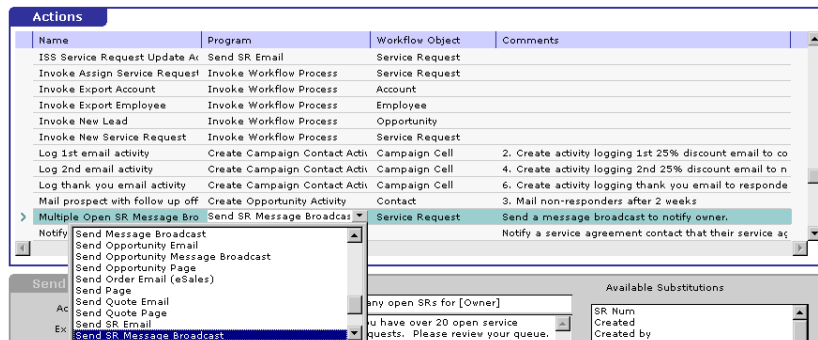


Figure 10-5. Programs in Siebel Tools and Workflow Policies Program Picklist

You use Siebel Tools to configure or create custom workflow policy objects and custom policy programs. The use of Siebel Tools is described in *Siebel Tools Guide*. Only information specific to Workflow Policies is described in this document.

Working with 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.

Creating a workflow policy object consists of four main steps:

- Defining the workflow policy columns.
- Defining the workflow policy components.
- Defining the workflow policy object.
- Associating the workflow policy column with the workflow policy component.

Understanding the Relationship Between Workflow Policy Components

Figure 10-6 shows the entity relationship diagram for four Service Request Workflow Policy components. The diagram shows each component, their relationship to each other, 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.

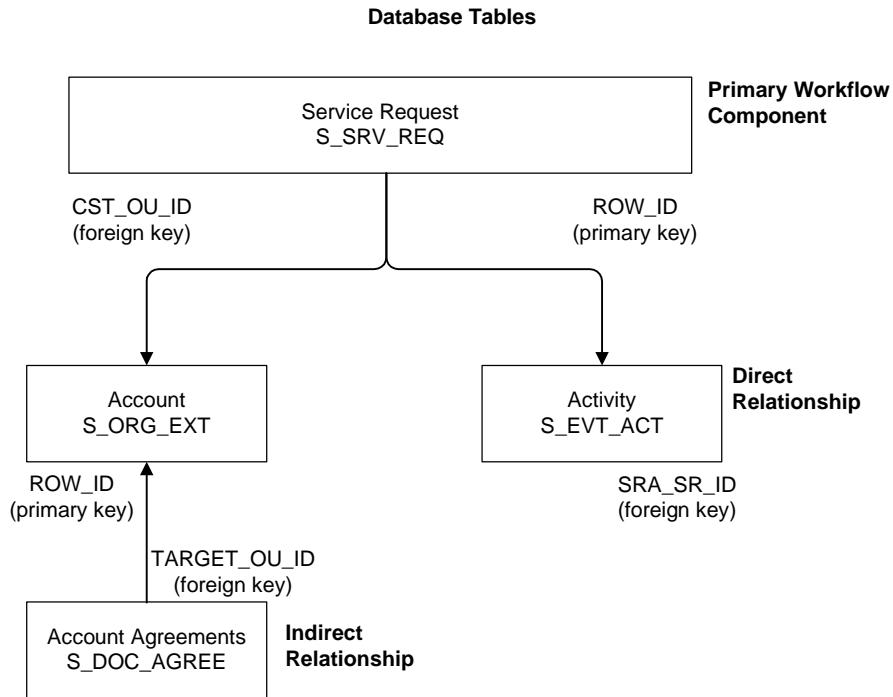


Figure 10-6. Relationships of Workflow Policy Objects and Workflow Policy Component

Figure 10-7 shows these component relationships in the Workflow Policy Component List view.

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 10-7. Service Request Policy Component Relationships

Relationship Between Workflow Policy Object and Workflow Policy Component

Figure 10-8 shows the relationship between the Service Request workflow policy object and the workflow policy components of the workflow policy object. The highlighted object, Service Request, has the primary field checked, making it the primary component. All the other components in the list are the nonprimary components of the Service Request workflow policy object.

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 10-8. Relationship Between Workflow Policy Object and Workflow Policy Component

Using 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.
- **Workflow Policy Objects List view.** Displays a list of the available workflow policy objects.
- **Workflow Policy Components 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 non primary policy components and how each of the policy components is related.
- **Workflow Policy Component Columns List view.** Displays a list of all the policy columns that can be monitored from the selected workflow policy component.

Understanding the Workflow Policy Column List View

Figure 10-9 shows the Workflow Policy Column List view.

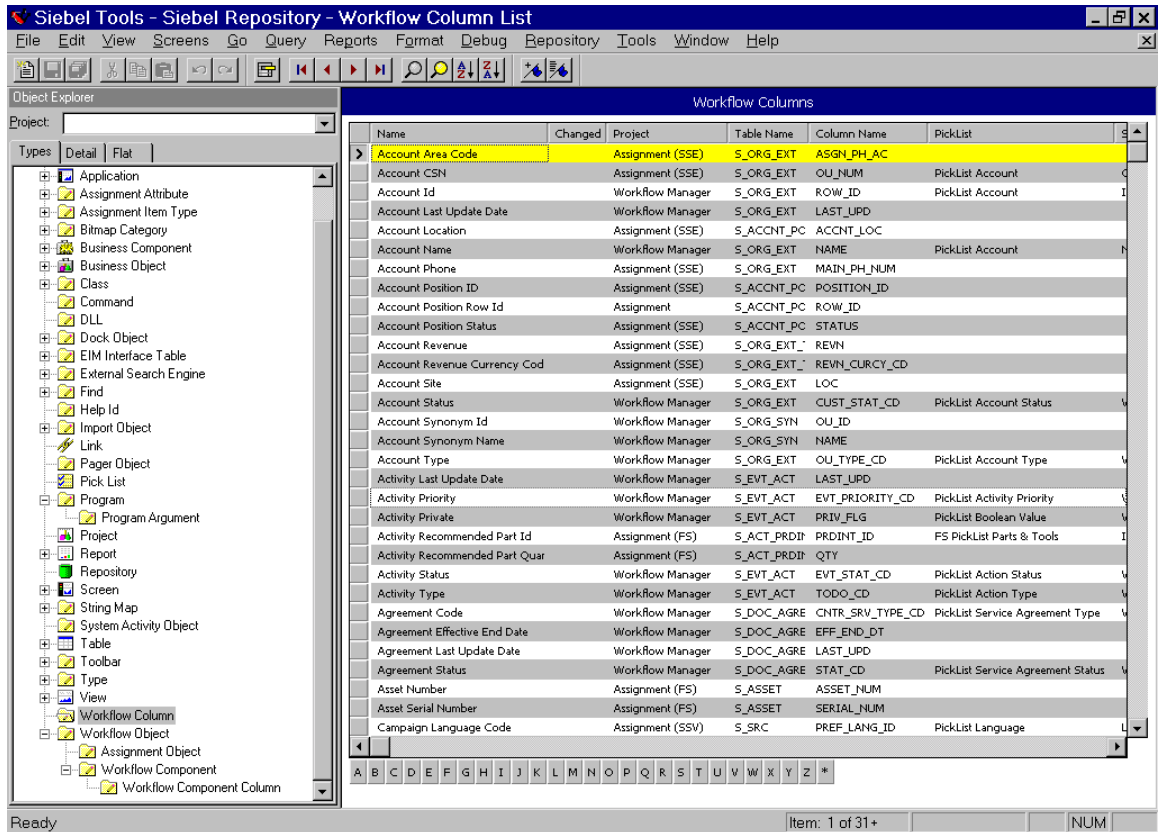


Figure 10-9. Workflow Policy Column List View

Table 10-1 describes the fields in the Workflow Policy Columns List view.

Table 10-1. 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 <ul style="list-style-type: none">■ 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 <i>Siebel Tools Guide</i> .
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.
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.

Understanding the Workflow Policy Object List View

Figure 10-10 shows the Workflow Policy Object List view that displays existing workflow policy objects.

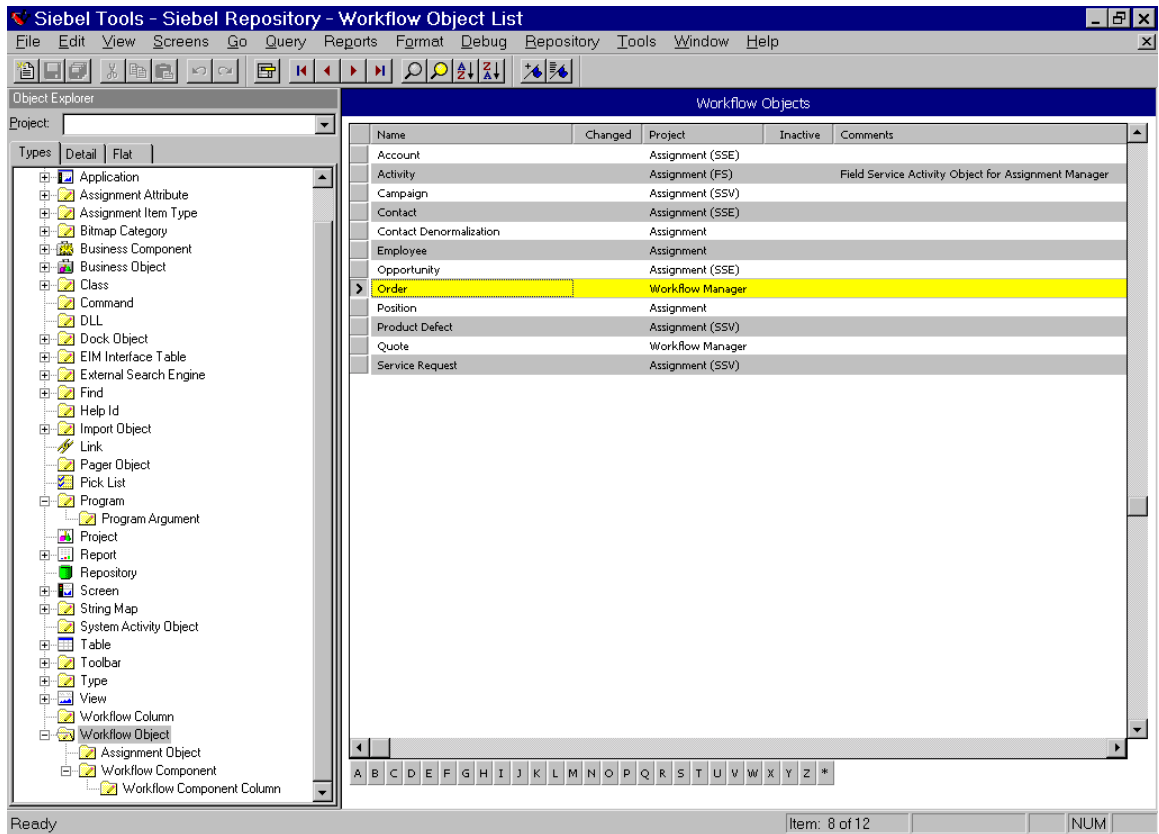


Figure 10-10. Workflow Policy Object List View

Table 10-2 describes the properties for a workflow policy object.

Table 10-2. Workflow Policy Object Properties Fields

Field	Description	Comments
Name	The name of the workflow policy object.	A descriptive name that is: <ul style="list-style-type: none">■ 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.

Understanding the Workflow Policy Component List View

A *workflow policy component* is a logical mapping of a database table. 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.

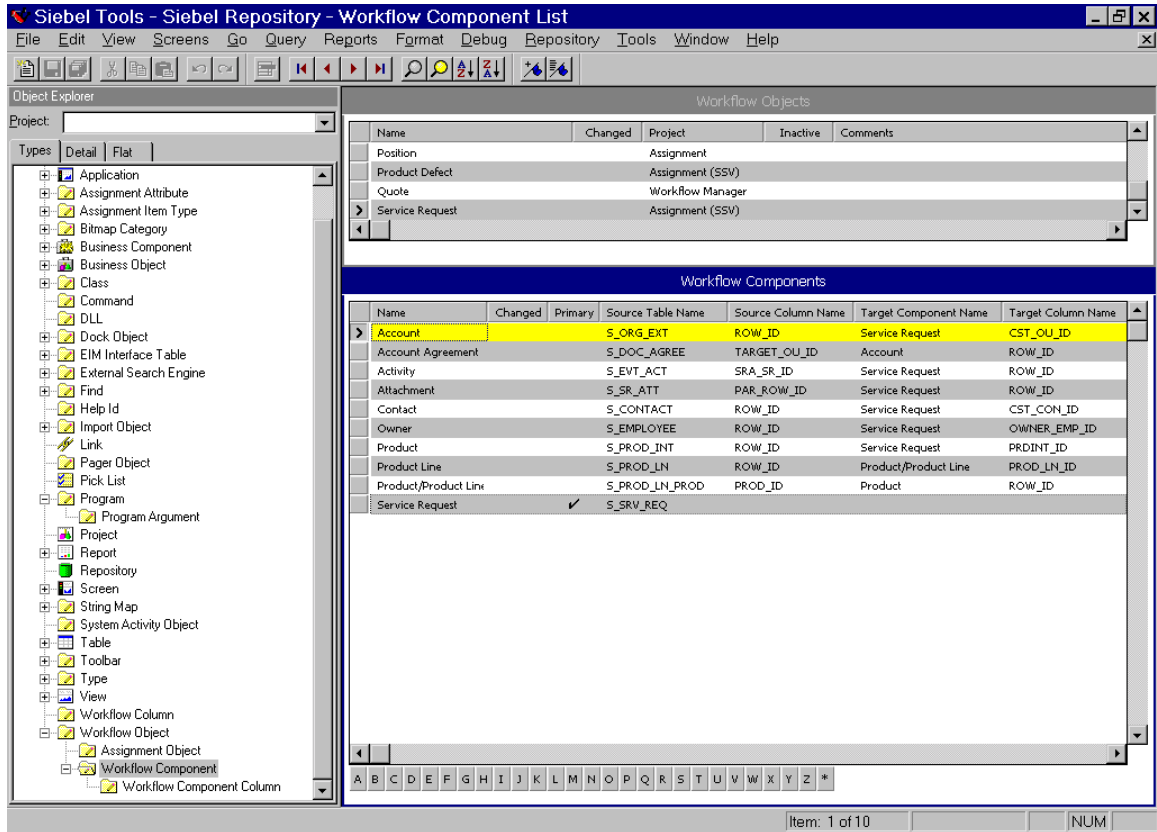


Figure 10-11. Workflow Policy Component List View

Table 10-3 describes all of the properties of a workflow policy component.

Table 10-3. Workflow Policy Component Properties Fields

Field	Description	Comments
Name	Name of the workflow policy component.	A descriptive name that is: <ul style="list-style-type: none">■ 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.
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. ¹
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.

1. Each workflow policy object must have only one primary workflow policy component.

Understanding 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, as shown in [Figure 10-12](#).

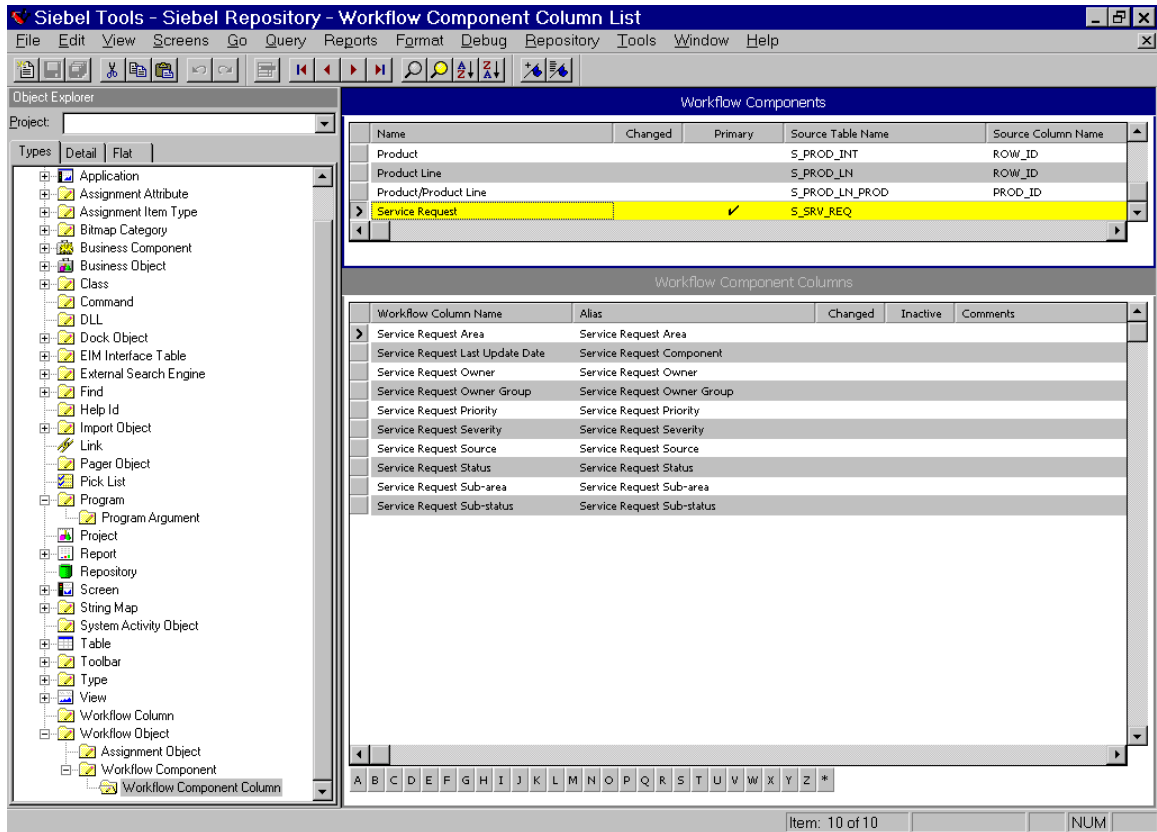


Figure 10-12. Workflow Policy Component Columns List View

Table 10-4 describes the workflow policy component columns values.

Table 10-4. 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 view.	The default is the workflow policy column name. A descriptive name that is: <ul style="list-style-type: none">■ 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 Policies view are called workflow policy component columns in Siebel Tools. The Column Instances available in the picklist in the Workflow Policies 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.

Figure 10-13 shows workflow policy components and columns added to the workflow policy object.

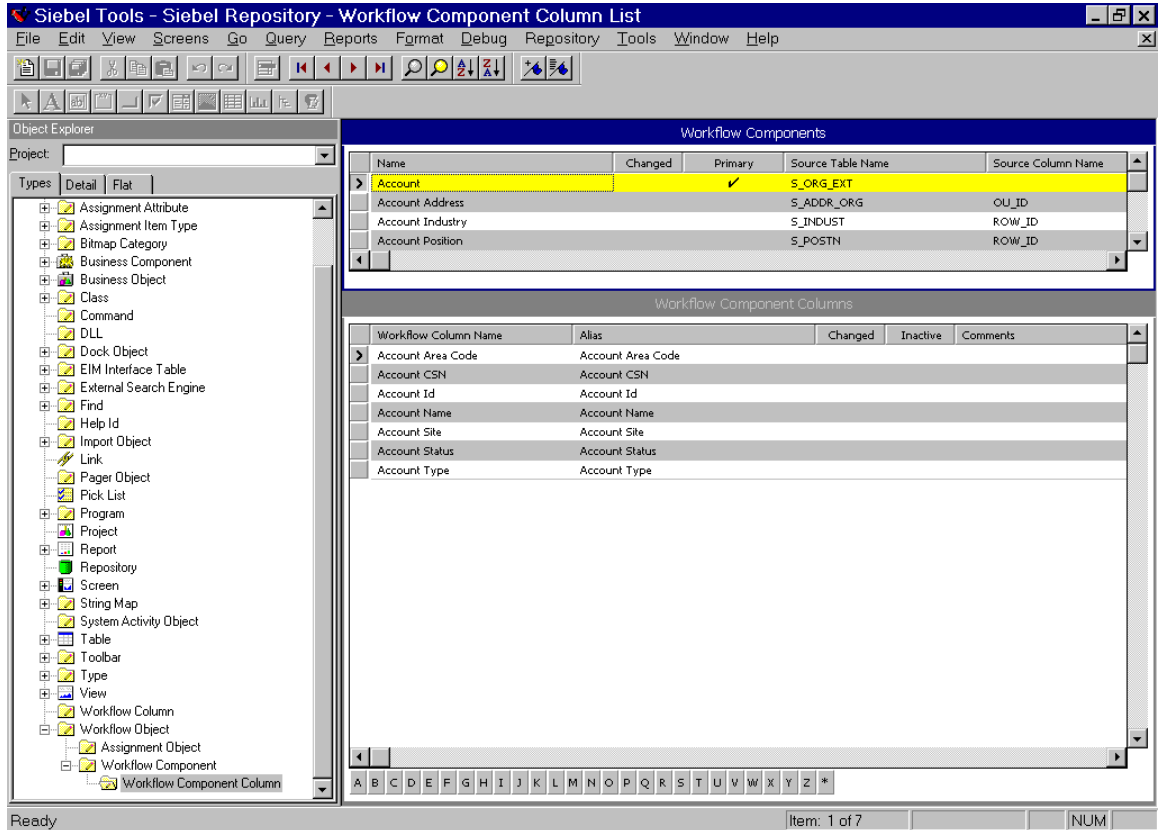


Figure 10-13. Adding Workflow Policy Components and Columns to the Workflow Policy Object View

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.

Using the Validate Tool

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, as shown in [Figure 10-14](#), right-click your mouse.

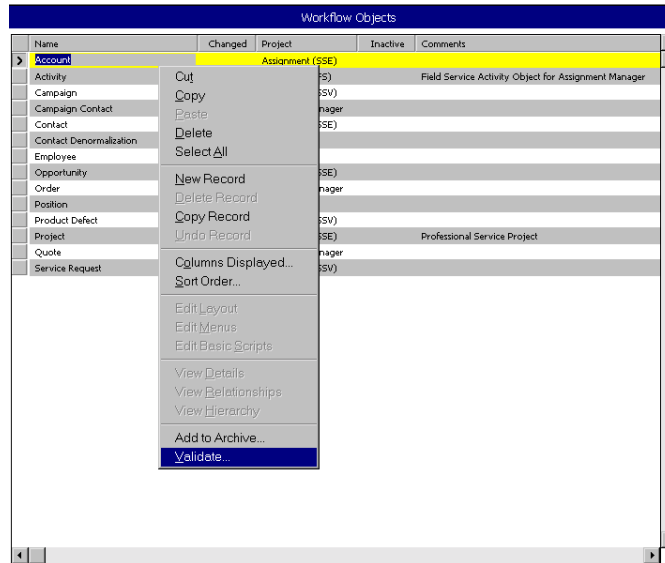


Figure 10-14. Validate Tool

Selecting Validate brings up the Validate screen. Clicking the Start button runs the validation process and returns information either as a caution shown in [Figure 10-15](#) or as error messages that appear in the Details box.

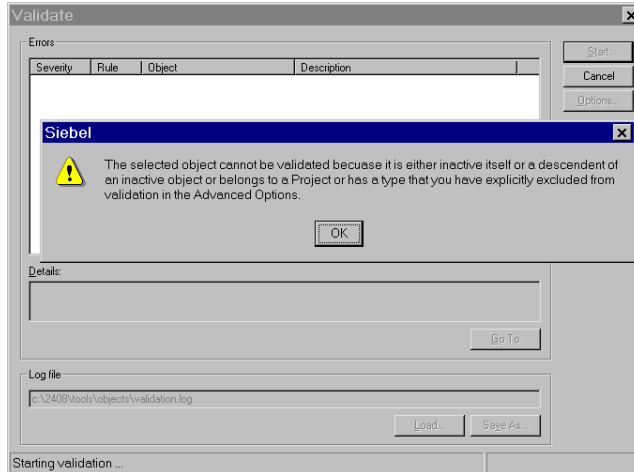


Figure 10-15. Validate Screen

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 Siebel 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

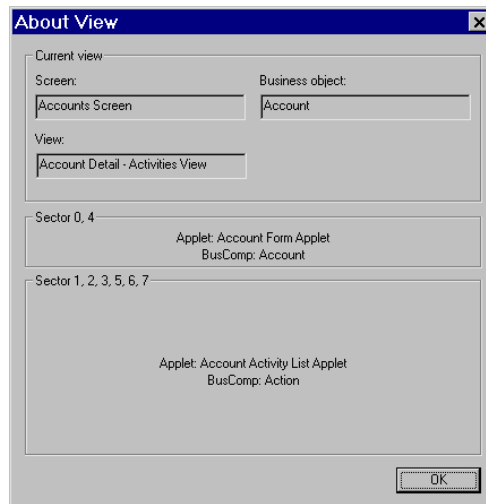
- 1 Start Siebel by entering the following from the command line:

```
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.



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.
- 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 show the Account/Action Link selected.

Name	Changed	Project	Parent Business Component	Child Business Component	Source Field	Destination Field
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 Summai	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)/Oppl		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/As		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 Acco
Account/Account AH		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 Account
Account/Account No		Account	Account	Account Note		Account Id
Account/Account Pri		Account	Account	Account Private Note		Account Id
Account/Account Syn		Link	Account	Account Synonym		Account Id
Account/Action		Link	Account	Action		Account Id
Account/Activity Plan		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.

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	

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.

Understanding Workflow Policy Programs

Workflow policies use workflow policy actions based on the workflow policy programs that are predefined in Siebel Tools. (See [Appendix B, “Predefined Programs,”](#) 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 Siebel:

- Send Message
- Send Page
- External Program
- Send Message Broadcast
- Database Operation

Using the Program List View

Figure 10-16 shows the Program List view with workflow policy programs predefined in Siebel Tools.

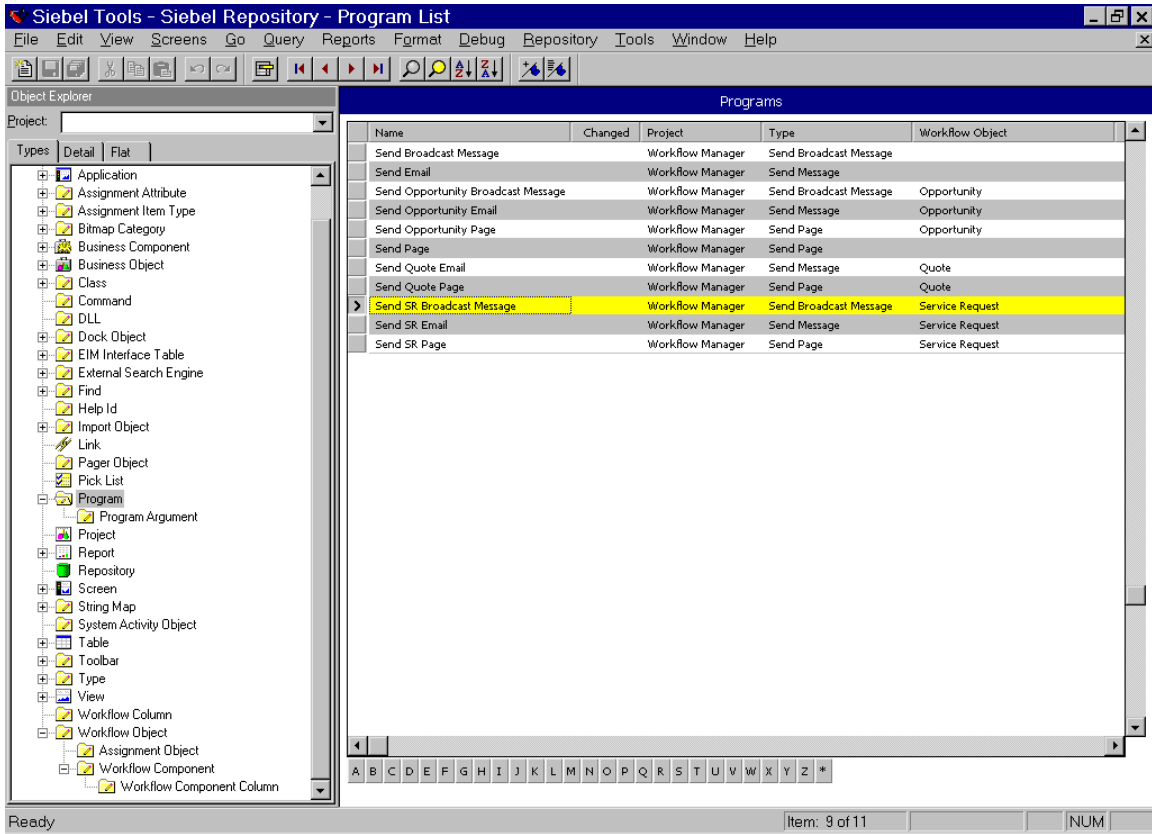


Figure 10-16. Program List View

Table 10-5 describes the Program properties fields.

Table 10-5. 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: <ul style="list-style-type: none"> ■ 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 user or 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.

Using 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. Figure 10-17 shows the Workflow Policy Program Argument List view. 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.

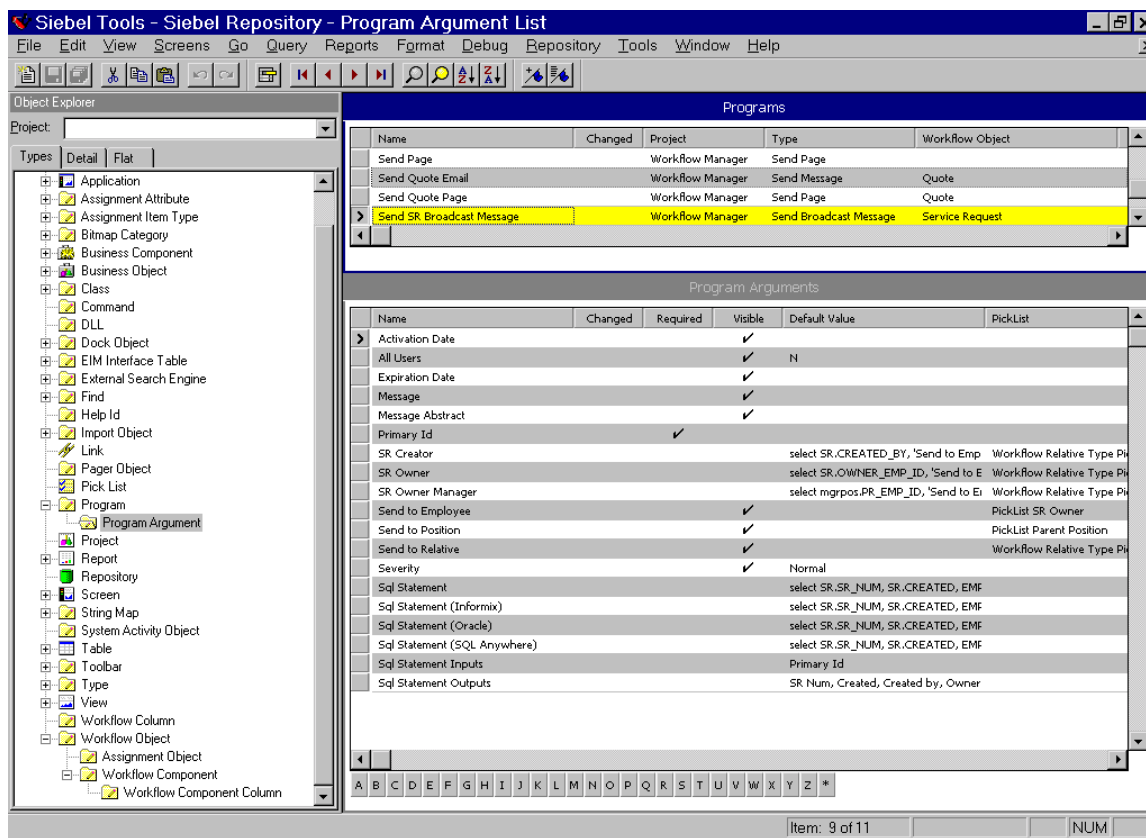


Figure 10-17. Workflow Policy Program Argument List View

Table 10-6 shows the Workflow Policy Program Argument properties values.

Table 10-6. 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, the following formats should be used:

- 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 program arguments 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 10-7](#).

Table 10-7. Valid Database Operations Workflow Policy Program Argument Values (1 of 2)

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 we now wish to insert/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.

Table 10-7. Valid Database Operations Workflow Policy Program Argument Values (2 of 2)

Name	Description	Allowable Default Value
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.
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 10-8](#) describes program arguments specific to the Send Message program.

Table 10-8. Send Message Program Argument Properties (1 of 2)

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.	

Table 10-8. Send Message Program Argument Properties (2 of 2)

Name	Description	Value
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 10-9](#) describes the Program Arguments particular to the Send Page program.

Table 10-9. 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.
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 10-10 describes the Program Arguments particular to the Run External Program program.

Table 10-10. 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.

Defining 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, MODIFICATION_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 than 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 10-7](#). 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.
- 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 Program Argument

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 Program → Send Opportunity Email → Program Arguments 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=?
```

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 the 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 Program Arguments

Before you begin to create the Recipient Type, Send to Relative, you must create an SQL statement in the Program Arguments 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 syntax is:
`&Table_Owner`
- The SQL statement must be valid for the specific database vendor being used.

Using Predefined Workflow Policy Programs

This section provides several examples of workflow policy programs and explains 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 [Appendix B, “Predefined Programs”](#)). These programs can be viewed in the Siebel Tools Object Explorer by clicking Program and then Program Argument.

Using Change SR Close Date to Today

If a Service Request has an activity of type Resolution, and the SR is open for more than five days, change the SR to today's date as shown in [Figure 10-18](#).

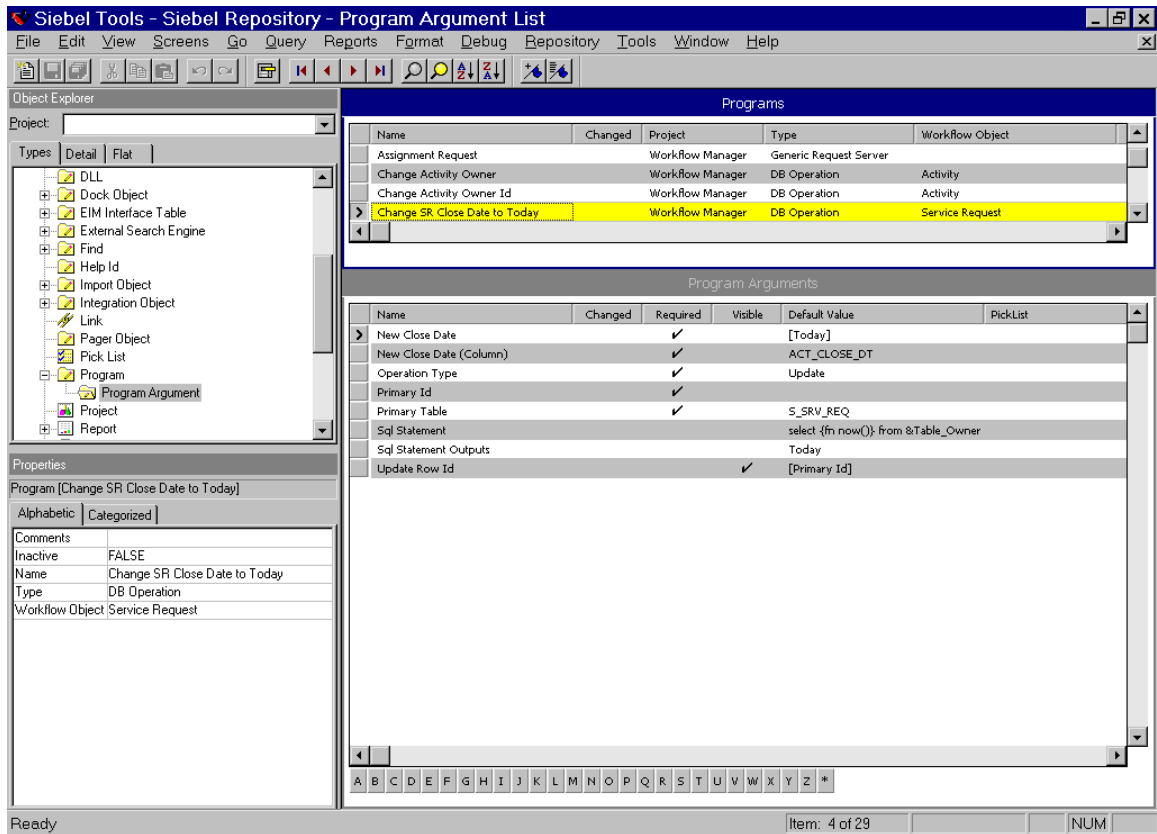


Figure 10-18. Change SR Close Date to Today

This workflow policy program enters today's system date into the field of the Service Request record, causing the workflow trigger.

Table 10-11. 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).
Sql Statement	<pre>select {fn now()} from &Table_Owner.S_DUAL</pre> <p>This statement calls 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 available in both Oracle and MS SQL and is used to hold temporary values.</p> <p>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.</p> <p>Different RDBMS have different formats for the same function (for example, in MS SQL, the function GetDate() is used to return the current date).</p>
Sql Statement Outputs	The Today argument creates a variable "Today" whose value is obtained from the SQL statement.
New Close Date (Column)	Specifies the field in the record to be updated (ACT_CLOSE_DT).
New Close Date	Specifies the field to be updated to the value of Today.

Using 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 ensure that the right people would see the incoming service request and assign it appropriately, as shown in [Figure 10-19](#).

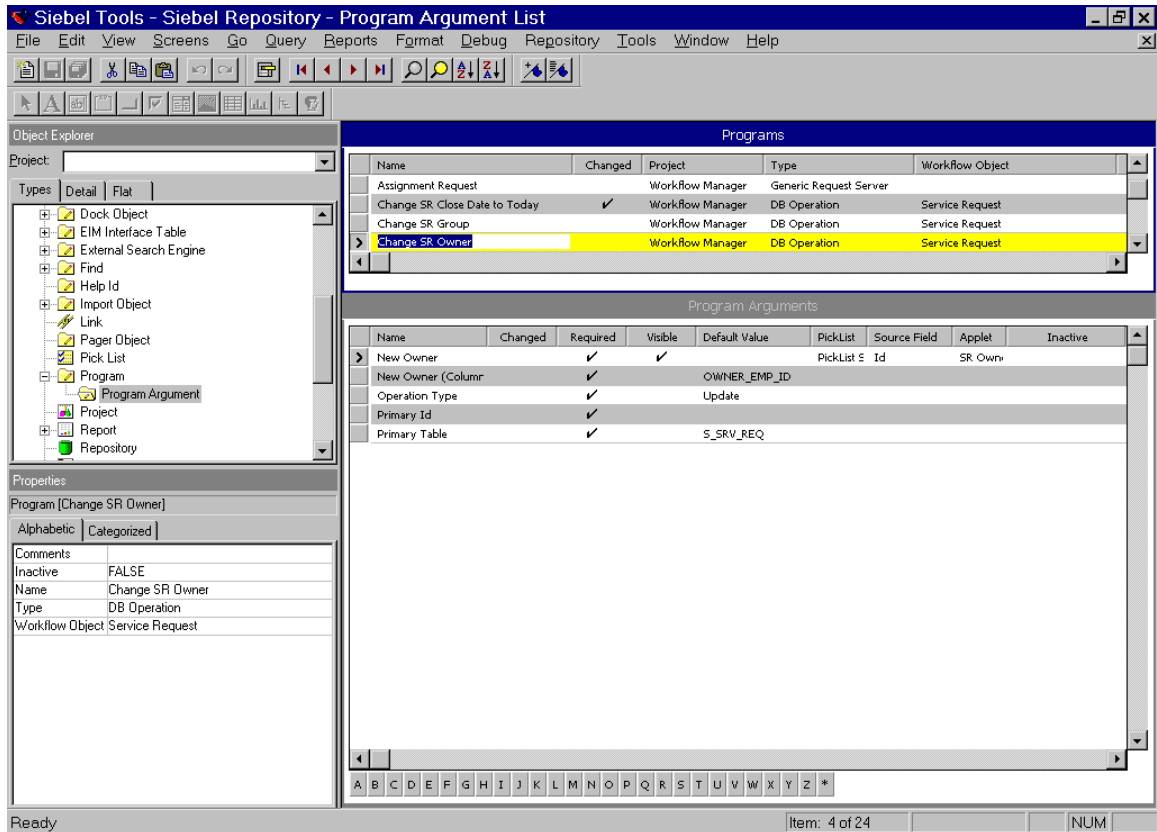


Figure 10-19. Change SR Owner

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 10-12. 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).
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.

Using 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 ensure a proper response time to service calls, as shown in [Figure 10-20](#).

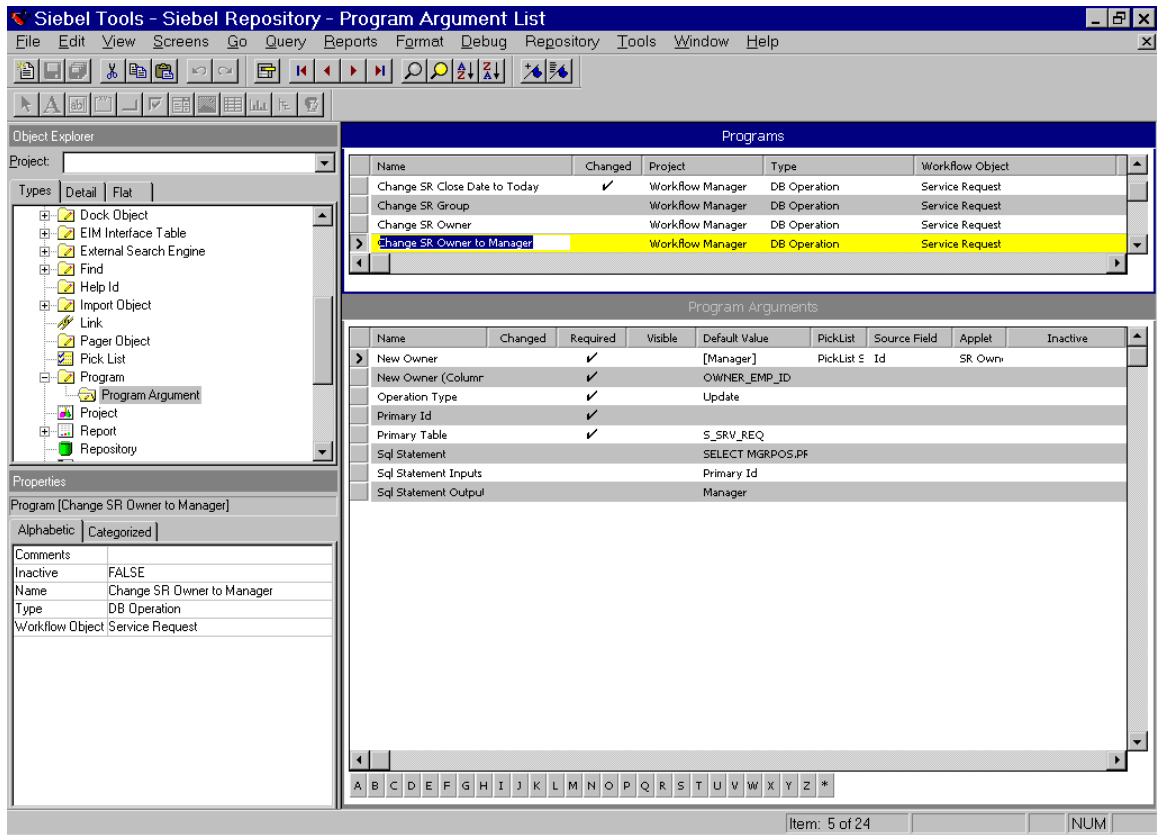


Figure 10-20. Change SR Owner to Manager

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 10-13. 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.
Sql Statement	<pre>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</pre> <p>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.</p> <p>This SQL statement joins four tables, giving access to data from all of them. In this example, only one field is retrieved.</p> <p>Policy Monitor requires the definitions 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.</p>
Sql Statement Inputs	Set to the value of Primary Id.
Sql Statement Outputs	Set to the value of Manager.

Using 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, as shown in [Figure 10-21](#).

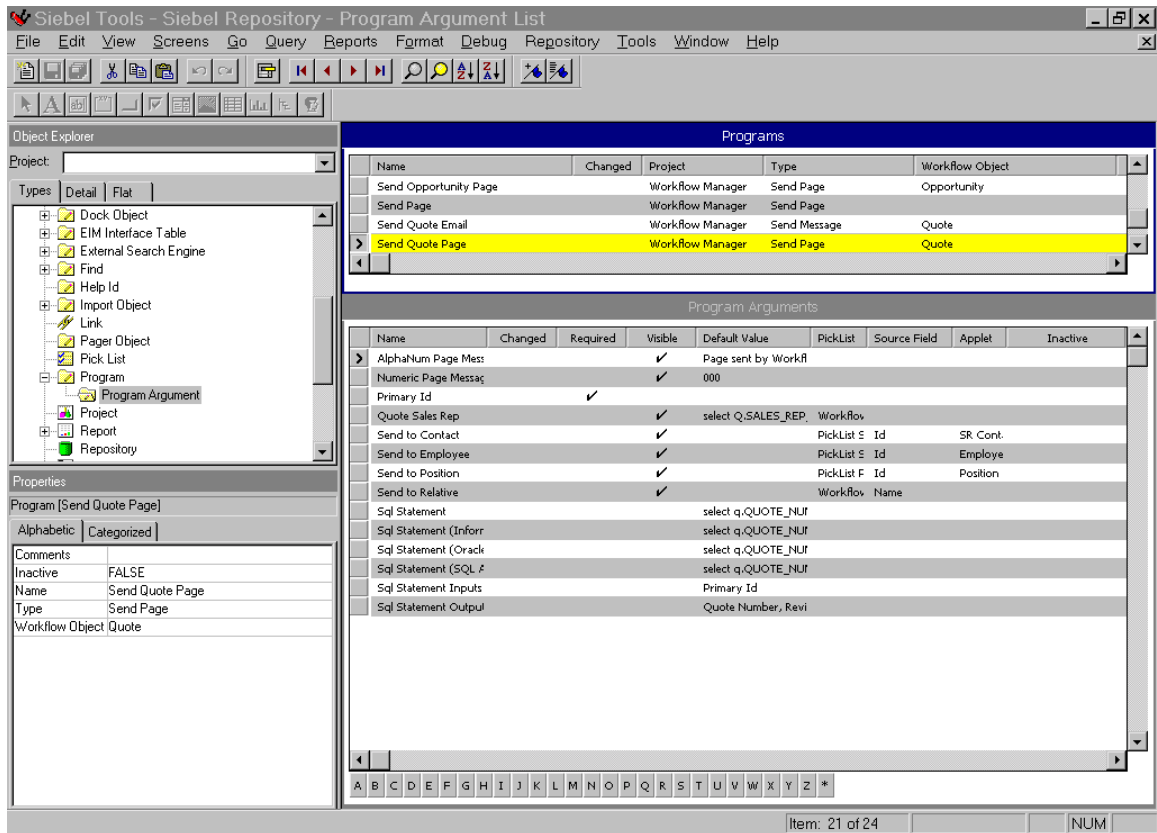


Figure 10-21. Send Quote Page

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 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.”

Workflow Policies Server Administration **11**

Verifying Workflow Policies Installation	11-2
Verifying the Repository Setting	11-2
Verifying the License Key	11-2
Creating Database Triggers	11-4
Database Triggers and Database Administration	11-5
Running Generate Triggers	11-5
Running the SQL Script File	11-7
Database Triggers and Remote Users	11-8
Setting Up the Siebel Server for Email Manager	11-8
Setting Up the Siebel Server for Page Manager	11-14
Troubleshooting the Email and Page Managers	11-18
Starting Workflow Monitor Agent	11-19
Using the Workflow Monitor Agent	11-20
Using Workflow Action Agent	11-25
Starting Workflow Agent Processes Automatically with Siebel Server	11-26
Using Workflow Policies and Siebel Server Task Trace Files	11-27
Viewing Trace Files in Siebel Server Administration	11-28
Viewing Trace Files in the Siebel Server Log Directory	11-29
Using Workflow Policies Analysis Charts and Reports	11-31
Using the Policy Frequency Analysis Chart	11-31
Using the Policy Trend Analysis Chart	11-31
Using Workflow Policies Reports	11-31

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. The installation process is described in *Siebel Installation Guide*.

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

Also note that Workflow should be running under the account (userid/password combination) of the user that starts it.

Verifying the Repository Setting

In the Siebel Client .cfg file used to configure workflow policies, check that the DockRepositoryName entry specifies the correct repository name.

Verifying the License Key

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

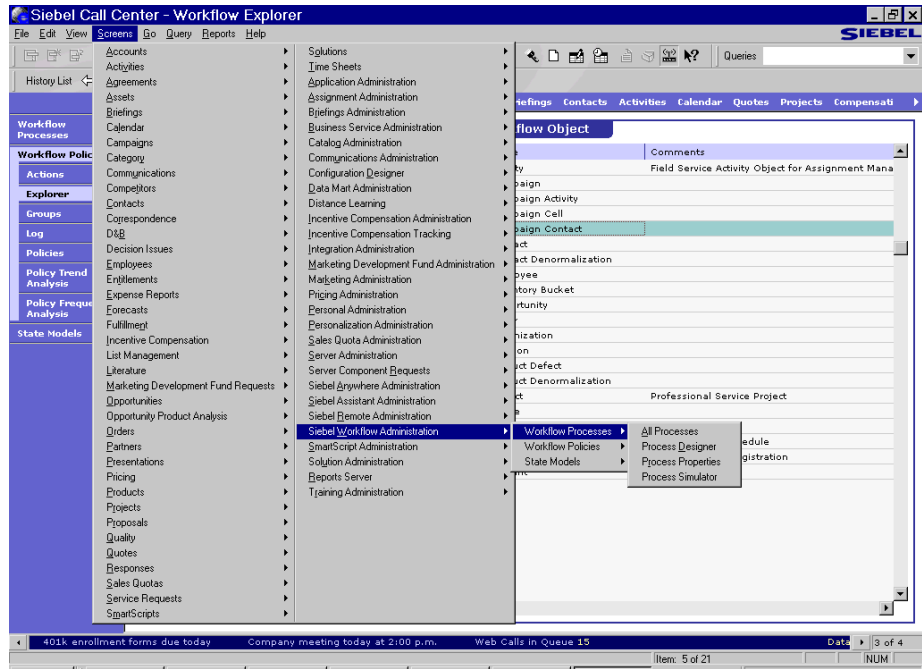
To do this, follow the procedure below to verify that you can access the Workflow Policies client screens and server screens.

To verify the license key

- 1 Log in to the Siebel Client as the Siebel administrator.

2 Navigate to Screens → Workflow Administration.

Under List Views, you should be able to see Workflow Policies groups, policies, and so forth as shown below. This indicates that your license key is correct.



3 Navigate to Screens → Server Administration → Enterprise Operations → Enterprise Servers from a client that is configured to manage the server components.

You should see your Siebel Server listed with a state of Running. This indicates that Workflow Policies is correctly installed on the server.

Creating Database Triggers

The Generate Trigger (GenTrig) component on the Siebel Server allows you to create database triggers. Workflow Policies uses these database triggers to identify which records may match policy conditions. Generate Triggers needs to be run whenever any of the following conditions occur:

- New policies (including Assignment Policies) are created or deleted, with the exception of Workflow Policies which have Batch Flag set to TRUE.
- Policy conditions are amended (for example, when policy criteria are amended).
- Activation or expiration dates of policies (including Assignment Policies) are changed.

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 *Siebel Installation Guide* for more information on installing Server Manager.



Caution: Running Generate Triggers may result in invalid triggers if you have incorrectly defined a policy condition. An invalid trigger can prevent execution of normal user transactions. For this reason, you should 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:

- 1** 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.
- 2** 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.

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.

Running Generate Triggers

When running Generate Triggers, you need to 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 do not need to stop and restart other workflow processes 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, you should 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 Server Manager client, navigate to Screens → Server Administration → Servers → Server Tasks.
- 2** Click New.
- 3** Select Generate Triggers from the Server Components list. This creates a new line entry but does not start the task.

Click Parameters to modify parameter settings. The component-specific parameters for Generate Triggers are in [Table 11-1](#). See *Siebel Server Administration Guide* for a description of the generic and enterprise parameters.

NOTE: For Oracle and IBM DB2, you must specify the table owner password in the Parameter window.

- 4 Click Close to save the parameter settings and click Start to execute the task.

A new task appears at the bottom of the list with a State of Starting Up and a PID (process ID). The state changes as Generate Triggers runs.

- 5 To view changes to the state, refresh the screen by clicking Execute Query or choosing the Execute Query menu entry.
- 6 Upon completion, the State field contains either Completed or Exited With Error. It is recommended that you view the log details by clicking the task link, such as 4140, to verify the status of creating the triggers. More detailed information is available, using the Trace Flags described in *Siebel Server Administration Guide*.

Table 11-1. Component-Specific Parameters for Generate Triggers

Name	Value	Description
Trace Flags		Flags for component-specific tracing information. This parameter is used to turn on various types of component-specific tracing. See the chapters in <i>Siebel System Administration Guide</i> describing the individual server components for a description of how to set this parameter for each component.
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 file name 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.
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. You should make sure that no one is logged into 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.

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 file name 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 maker, 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.

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, a MAPI-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 MAPI-compliant client programs to verify your settings, including Microsoft Exchange or Microsoft Outlook.

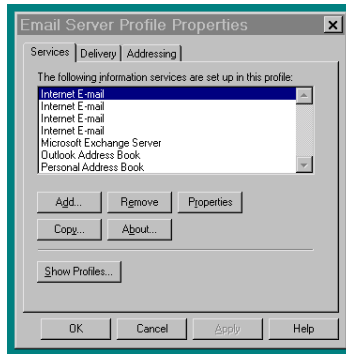
When you create your messaging profile, you must be logged into the same account that is used to start Siebel Server. This account is specified when Siebel Server is installed.

To find the account used to start Siebel Server

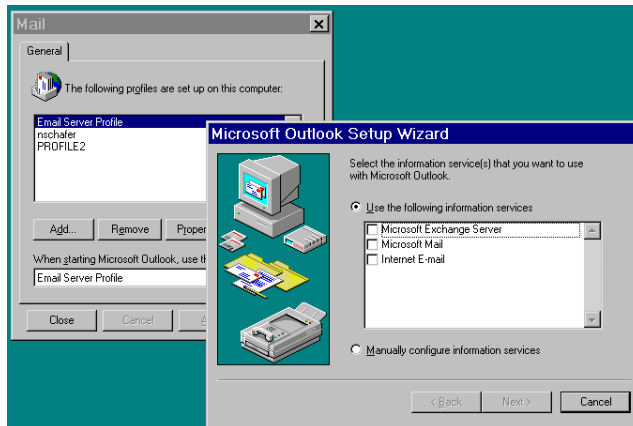
- 1** Under Settings, bring up the Control Panel.
- 2** Click Services.
- 3** Scroll down and select Siebel Server.
- 4** Click the “Startup...” button.
- 5** When the Service dialog box appears, locate the account used to run the Siebel Server Service in the This Account field.

To create your messaging profile

- 1** Create a new messaging profile by going to the Mail icon in the Windows Control Panel.
- 2** Bring up the Properties dialog box and click the Show Profiles button.



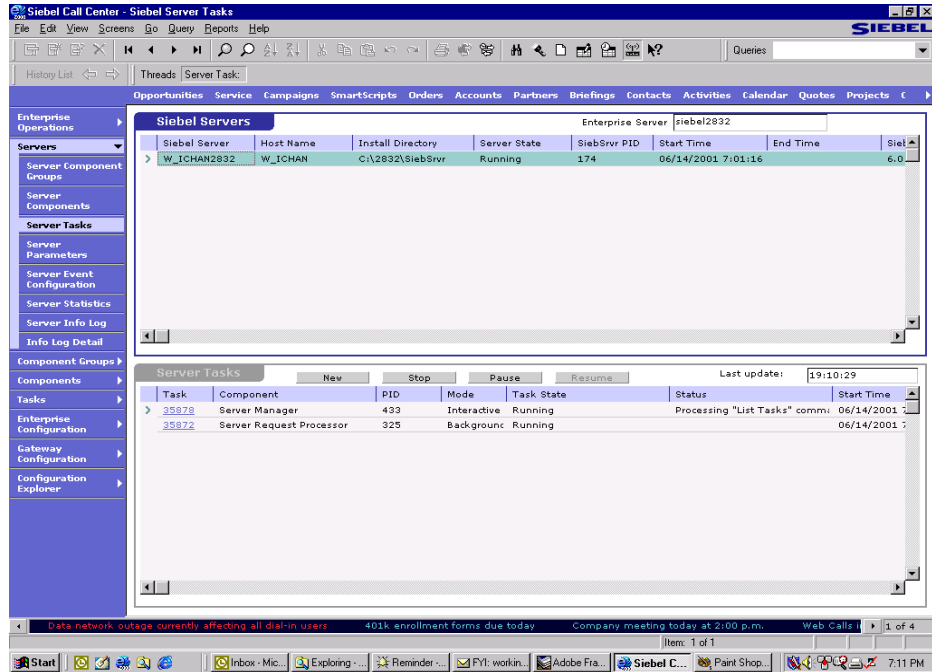
- 3** Click Add in the Mail dialog box to bring up Setup Wizard and follow that process to enter your mail profile.



After creating your mail profile, you need to start the Email Manager component on your Siebel Server. The Email Manager component executes email actions once the conditions of a Workflow policy are met. Go to the Siebel Server views to start and stop these components in the Siebel Server views.

To set up Email Manager on the Siebel Server

- 1 Navigate to Server Administration → Servers → Server Tasks.



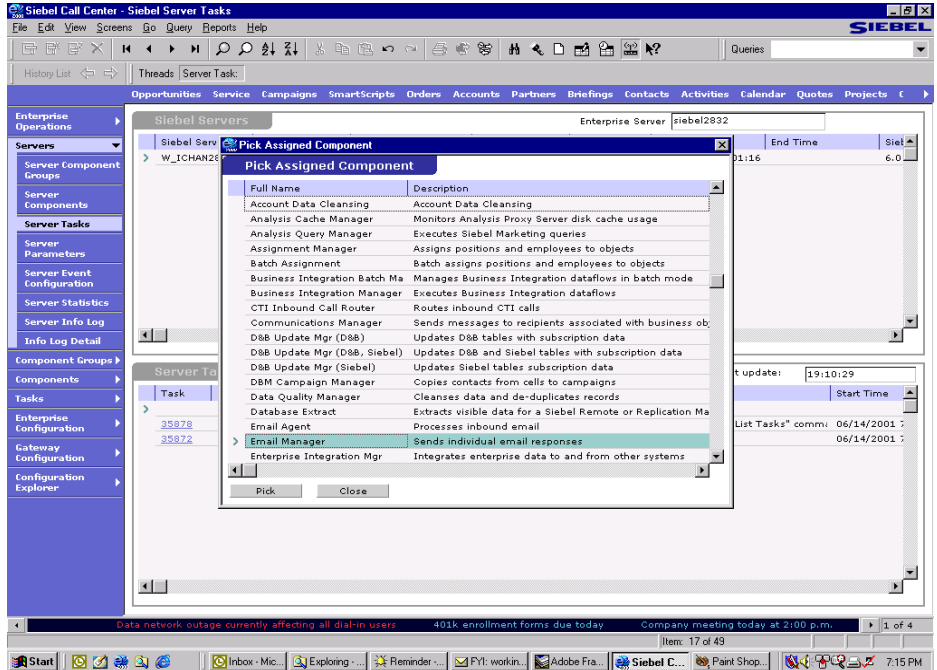
- 2 Go to the Server Tasks applet and click the New button.

An empty field appears under Component.

- 3 Scroll down the Pick Assigned Component picklist to Email Manager and select Email Manager.

4 Click Pick.

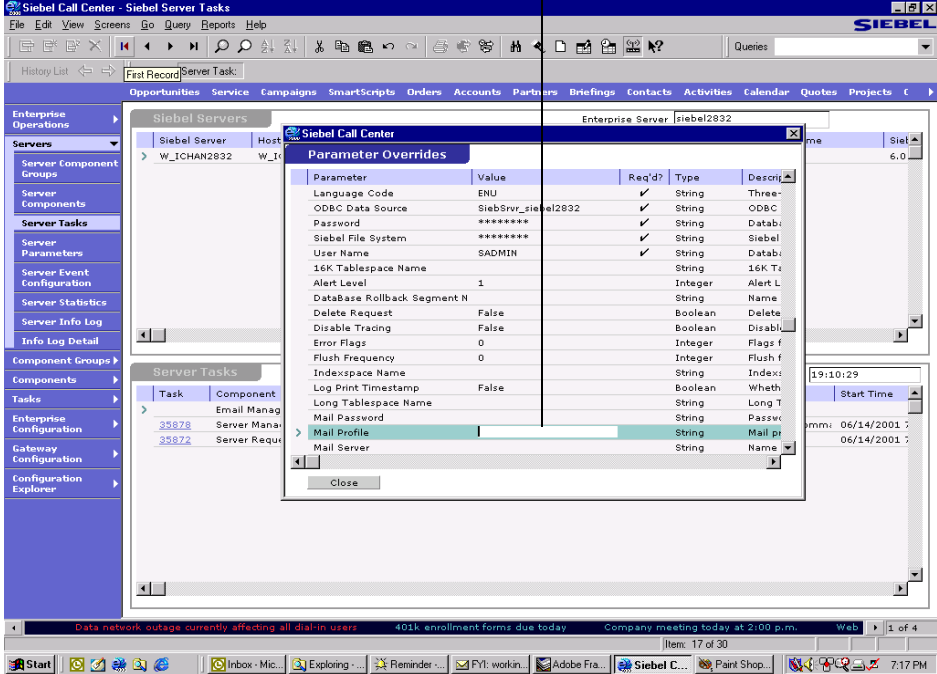
Email Manager is added to the Server Tasks applet.



- 5 Click Parameters in the Server Tasks applet, scroll down to Mail Profile, and enter the profile name under Values.

Enter the same profile name you used to create your messaging profile.

Enter Profile Name



Parameter	Value	Req'd?	Type	Descrip
Language Code	ENU	✓	String	Three-
ODBC Data Source	Siebsrvr_siebel2832	✓	String	ODBC
Password	*****	✓	String	Datab:
Siebel File System	*****	✓	String	Siebel
User Name	SADMIN	✓	String	Datab:
16K Tablespace Name			String	16K Tr
Alert Level	1		Integer	Alert L
Database Rollback Segment N			String	Name
Delete Request	False		Boolean	Delete
Disable Tracing	False		Boolean	Disabl
Error Flags	0		Integer	Flags f
Flush Frequency	0		Integer	Flush f
Indexspace Name			String	Index:
Log Print Timestamp	False		Boolean	Wheth
Long Tablespace Name			String	Long T
Mail Password			String	Passw
Mail Profile			String	Mail pr
Mail Server			String	Name

- 6 Click Start to start the Email Manager component.

The most important parameters for the Email Manager are Sleep Time, Mail Password, and Mail Profile. [Table 11-2 on page 11-13](#) describes parameters for the Email Manager.

Table 11-2. Parameters for the Email Manager Component (1 of 2)

Argument	Description
Error Flags	Flags to trace error information. This parameter is used to turn on various types of component-specific error tracing. It is recommended that you set this flag to at least 2.
Sql Trace Flags	Flags to trace SQL statements. If this parameter is set to 1, every SQL statement issued by the component tasks is logged to the information log file for each task. If this parameter is set to 2 (recommended), each SQL statement is logged in addition to information about the number of parse, execute, and fetch calls, and timing information about each type of call.
Trace Flags	<p>Flags for component-specific tracing information. This parameter is used to turn on various types of component-specific tracing. See the chapters describing the individual server components for a description of how to set this parameter for each component.</p> <p>The default is 0.</p> <p>By default, Email Agent does not log email successfully sent. If you want to log all email sent, set the trace flag to 2. Note that it is recommended you set this flag to at least 6.</p>
Sleep Time	Time to sleep between iterations (in seconds). How often the server checks the request queue.
Mail Password	Password for mail profile. (This is the password you use to log in to your mail system.) Can be NULL.
Mail Profile	Mail profile to use. Defined in Control Panel. 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.
Request Key	When you have more than one Email Manager, the request key serves as the ID for each Email Manager. You can then specify this key for the Workflow action in the Workflow Action Argument view. The request key can be any string.
Delete Request	<p>If this parameter is set to TRUE, Email Manager will delete email requests from the database after successfully processing them. Only email requests that fail will remain in the database with the appropriate error messages.</p> <p>If this parameter is set to FALSE, all email requests, whether successfully processed or not, will remain in the database.</p> <p>The default for this parameter is FALSE.</p>

Table 11-2. Parameters for the Email Manager Component (2 of 2)

Argument	Description
Save Sent Mail	<p>If this parameter is set to TRUE, copies of sent email will be kept in the Sent Items folder of the mailbox that email Manager uses to send email.</p> <p>If this parameter is set to FALSE, no record of the sent email will be retained.</p> <p>The default for this parameter is TRUE.</p>
Mail Server	<p>This parameter is used only when the mailbox that Email Manager will send the email from is specified on the email request. Currently this is supported only when the underlying email system is Microsoft Exchange Server. If an email request contains the "From" mailbox, Email Manager will create a dynamic profile on the email server specified by this parameter.</p>

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. You should verify that the settings are compatible with your modem.

To run the Page Manager component

1 Navigate to Server Administration → Servers → Server Tasks.

2 Go to the Server Tasks applet and click on the New button.

An empty field appears under Component.

3 Select the Page Manager component.

4 Scroll down the Pick Assigned Component picklist and select Page Manager.

5 Click Parameters in the Server Tasks applet and enter your parameters.

See [Table 11-3](#) for a list of parameters.

The most important parameters are Modem Port, Dial Prefix, Long Distance Prefix, and Local Area Code. You should change the values for these parameters to match your system. If you do not specify a parameter, the default values described in [Table 11-3](#) are used.

6 Click Start to start the Page Manager component.

Table 11-3. Parameters for the Page Manager Component (1 of 2)

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 <code>srvrmgr.exe</code> 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: <pre>SRVRMGR.EXE /g NT01022 /e SBLPRD_ENT502 /s SBLPRD_APP502 /u ***** /p ***** /c "START TASK FOR COMPONENT PageMgr WITH DialPrefix = '- ' "</pre>
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 doesn't 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 doesn't have a personal identification number (PIN). Default = 4
Modem Reset String	A modem command used to reset the modem. Default = ATZ Refer to your modem documentation for the correct command.

Table 11-3. Parameters for the Page Manager Component (2 of 2)

Parameter	Value
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 (MAPI-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.

You can look at the list of email and paging requests through the Application Administration → Server Request Monitor view. 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.

Starting Workflow Monitor Agent

To execute your Workflow policies, you need to start Workflow Monitor Agent. Workflow Monitor Agent checks to see when the conditions of policies are met, executes actions once those conditions are met, or submits a request to Workflow Action Agent when actions are to be taken.

You start and stop Workflow Monitor Agent and Workflow Action Agent in the Server Administration views.

Table 11-4. 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 matched.
S_ESCL_ACTN_REQ	This table holds the requests to execute actions.
S_ESCL_LOG	This table holds a history of base table rows that have matched policies.

Using the Workflow Monitor Agent

Workflow Monitor Agent performs several server processes for monitoring 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.

- 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. (See [Chapter 15, “Monitoring and Performance Tuning,”](#) for more information on workflow database tables.) Workflow Monitor Agent (Workmon) evaluates the request against the rules set up by the policies in the workflow policy group.

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 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. 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.

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. [Figure 11-1](#) describes the flow of Workflow Agent processes.

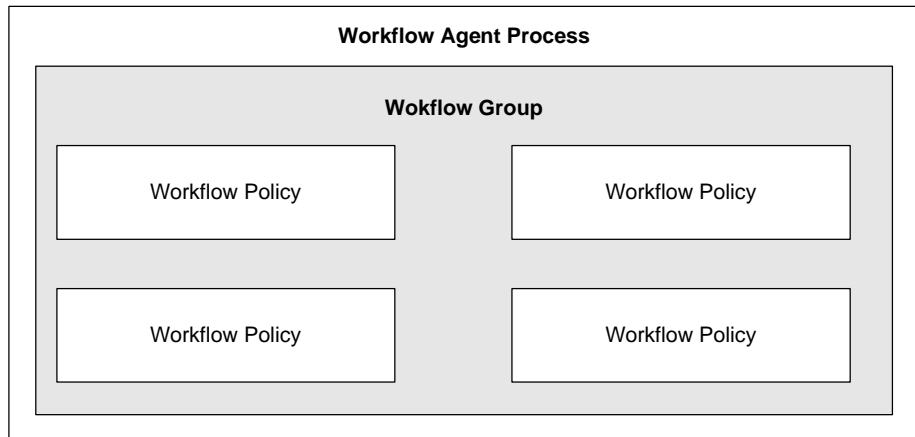


Figure 11-1. Workflow Agent Processes

To run the Workflow Monitor Agent component

- 1 Navigate to Screens → Server Administration → Servers → Server Tasks.
- 2 Click New.
- 3 In the Component field, pick the Workflow Monitor Agent component from the Pick Assigned Component picklist.
- 4 Click Parameters to bring up the Parameters Override dialog box.
- 5 Specify the parameters for the Workflow Monitor Agent process.
Group name is a required parameter.

NOTE: You should only run 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.

See [Table 11-5 on page 11-22](#) for a list of parameters.

- 6 Click Start to begin the Workflow Monitor Agent process.

Table 11-5. Workflow Monitor Agent Parameters (1 of 3)

Argument	Description
Action Interval	<p>Action execution interval in minutes. 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.</p> <p>In other words, if the same record repeatedly violates the same policy before the Action Interval expires, the record will be held until the Action interval expires. At the end of the Action Interval, if the record is still in violation, an action will take place.</p> <p>The default is 60 minutes. If this value is used, it <i>must</i> be greater than 0 (zero) or unexpected behavior may result.</p>
Group Name	<p>Workflow policy group that Monitor Agent works on (required).</p>
Mailing Address	<p>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.</p>
Reload Policy	<p>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.</p> <p>The default is 600 seconds.</p>
Requests per Iteration	<p>Maximum number of requests read per iteration.</p> <p>The default is 5000.</p> <p>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 doing these things.</p>

Table 11-5. Workflow Monitor Agent Parameters (2 of 3)

Argument	Description
Sleep Time	<p>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. See Chapter 15, “Monitoring and Performance Tuning,” for more information.</p> <p>The default is 60 seconds.</p>
Number of days to keep violation information	<p>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.</p> <p>The default is 30 days.</p>
Use Action Agent	<p>Determines if Action Agent is automatically run with Monitor Agent.</p> <p>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.</p> <p>You must start Workflow Action Agent separately when using email consolidation and when the parameter Use Action Agent is set to TRUE.</p>
Cache size of last user information	<p>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.</p> <p>The default is 100.</p>

Table 11-5. Workflow Monitor Agent Parameters (3 of 3)

Argument	Description
Process the batch Policies	<p>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. The default is FALSE.</p> <p>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.</p>
Ignore Errors	<p>Ignore errors while processing requests. By default, the Workflow Monitor and Action agents will not ignore errors that occur while processing the requests. If 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.</p> <p>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.</p>

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. 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_APPSRRV_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

- 1** Navigate to Screens → Server Administration → Servers → Server Tasks.
- 2** Click New.
- 3** In the Component field, pick the Workflow Action Agent component from the Pick Assigned Component picklist.
- 4** Click Parameters to bring up the Parameters Override dialog box.
- 5** Specify the parameters for the Workflow Action Agent process.

See [Table 11-5 on page 11-22](#) for a list of parameters. Only the group name is mandatory.
- 6** Click Start to begin Workflow Action Agent.
- 7** Repeat these steps for each agent you want to start.

You must start one Workflow Monitor Agent and one Workflow Action Agent for each workflow policy group.

NOTE: For more information on arguments for starting a Workflow Agent process, refer to *Siebel Server Administration Guide*.

To shut down the Workflow Agent process

- 1 Navigate to Server Administration → Servers → Server Tasks.
- 2 Select the Workflow process that is currently running and click Stop.

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 Navigate to → Screens Server Administration → Components → Component Parameters.
- 2 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 11-5 on page 11-22](#) 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** See *Siebel Server Administration Guide* for instructions on creating a defined component. Create the 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.

Using 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 on 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—either in the Siebel Server Tasks view in Server Administration or in the log directory of your Siebel Server.

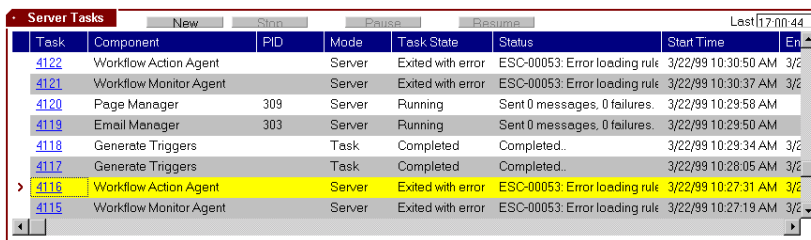
Viewing Trace Files in Siebel Server Administration

Find this view from Siebel Call Center.

To view trace files in the Siebel Server Tasks view

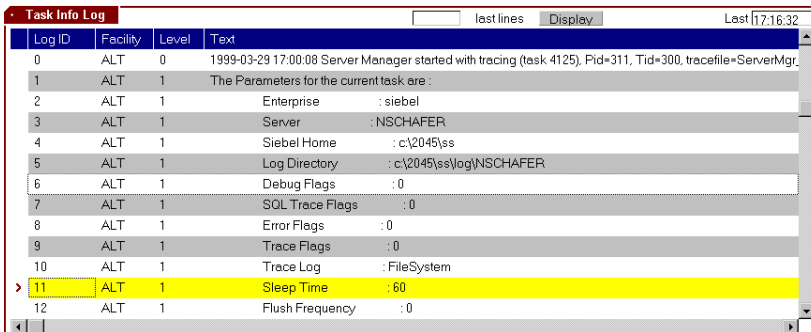
- 1 Navigate to Screens → Server Administration → Servers → Server Tasks.

The Server Tasks applet lists the status of all the Siebel Server tasks either running or started.



Task	Component	PID	Mode	Task State	Status	Start Time	End Time
4122	Workflow Action Agent		Server	Exited with error	ESC-00053: Error loading rule	3/22/99 10:30:50 AM	3/22/99 10:30:50 AM
4121	Workflow Monitor Agent		Server	Exited with error	ESC-00053: Error loading rule	3/22/99 10:30:37 AM	3/22/99 10:30:37 AM
4120	Page Manager	309	Server	Running	Sent 0 messages, 0 failures.	3/22/99 10:29:58 AM	
4119	Email Manager	303	Server	Running	Sent 0 messages, 0 failures.	3/22/99 10:29:50 AM	
4118	Generate Triggers		Task	Completed	Completed.	3/22/99 10:29:34 AM	3/22/99 10:29:34 AM
4117	Generate Triggers		Task	Completed	Completed.	3/22/99 10:28:05 AM	3/22/99 10:28:05 AM
4115	Workflow Action Agent		Server	Exited with error	ESC-00053: Error loading rule	3/22/99 10:27:31 AM	3/22/99 10:27:31 AM
4115	Workflow Monitor Agent		Server	Exited with error	ESC-00053: Error loading rule	3/22/99 10:27:19 AM	3/22/99 10:27:19 AM

- 2 Click on the hyperlink number in the Task field.
- 3 Review the error message information that appears in the Task Information Log applet.



Log ID	Facility	Level	Text
0	ALT	0	1999-03-29 17:00:08 Server Manager started with tracing (task=4125), Pid=311, Tid=300, tracefile=ServerMgr.
1	ALT	1	The Parameters for the current task are:
2	ALT	1	Enterprise : siebel
3	ALT	1	Server : NSCHAFER
4	ALT	1	Siebel Home : c:\2045\ss
5	ALT	1	Log Directory : c:\2045\ss\log\NSCHAFER
6	ALT	1	Debug Flags : 0
7	ALT	1	SOL Trace Flags : 0
8	ALT	1	Error Flags : 0
9	ALT	1	Trace Flags : 0
10	ALT	1	Trace Log : FileSystem
11	ALT	1	Sleep Time : 60
12	ALT	1	Flush Frequency : 0

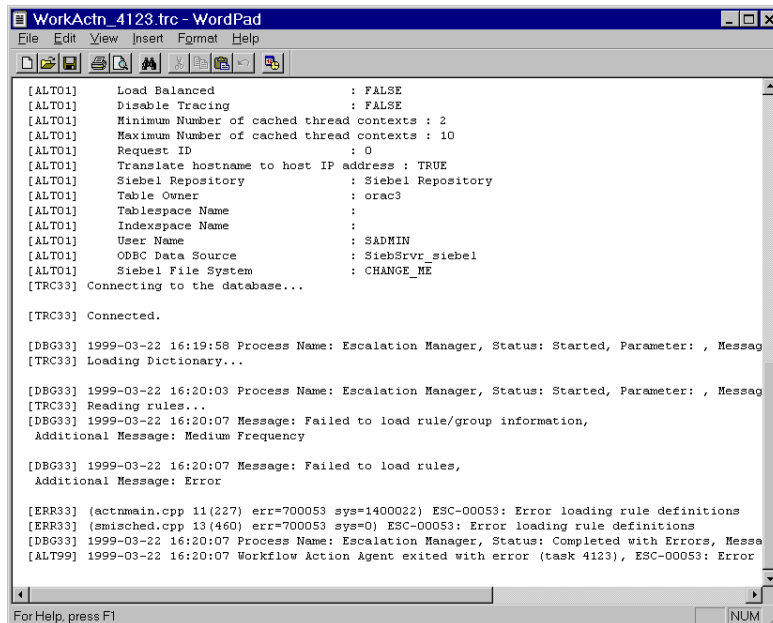
Viewing Trace Files in the Siebel Server Log Directory

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, as shown in [Figure 11-2](#).

Contents of 'C:\2045\ss\log\NSCHAFER'					
Name	Size	Type	Modified	Attribu.	
AsgnSrvr_4099.trc	4KB	TRC File	3/18/99 3:20 PM	A	A
AsgnSrvr_4110.trc	4KB	TRC File	3/18/99 3:20 PM	A	A
AsgnSrvr_4112.trc	5KB	TRC File	3/18/99 3:20 PM	A	A
BatchAssign_4100.trc	4KB	TRC File	3/18/99 3:20 PM	A	A
BatchAssign_4111.trc	4KB	TRC File	3/18/99 3:20 PM	A	A
BatchAssign_4113.trc	5KB	TRC File	3/18/99 3:20 PM	A	A
GenTrig_4117.trc	7KB	TRC File	3/22/99 10:28 AM	A	A
GenTrig_4118.trc	7KB	TRC File	3/22/99 10:30 AM	A	A
ServerMgr_4114.trc	3KB	TRC File	3/22/99 10:14 AM	A	A
SiebSrvr_4097.trc	1KB	TRC File	3/16/99 9:33 AM	A	A
SPMSynch_4104.trc	4KB	TRC File	3/16/99 9:34 AM	A	A
SPMSynch_4108.trc	4KB	TRC File	3/16/99 9:34 AM	A	A
SrvrSched_4098.trc	1KB	TRC File	3/16/99 9:33 AM	A	A
SynchMgr_4107.trc	3KB	TRC File	3/16/99 9:34 AM	A	A
SynchMgr_4109.trc	3KB	TRC File	3/16/99 9:34 AM	A	A
WorkActn_4116.trc	4KB	TRC File	3/22/99 10:28 AM	A	A
WorkActn_4122.trc	4KB	TRC File	3/22/99 10:30 AM	A	A
WorkActn_4123.trc	4KB	TRC File	3/22/99 4:20 PM	A	A
WorkMon_4115.trc	4KB	TRC File	3/22/99 10:28 AM	A	A
WorkMon_4121.trc	4KB	TRC File	3/22/99 10:30 AM	A	A
WorkActn_4124.trc	11KB	TRC File	3/23/99 3:56 PM	A	A
NSCHAFER.lrt	2KB	LRT File	3/23/99 3:56 PM	A	A
SiebSrvrWait_4098.trc	2KB	TRC File	3/23/99 3:56 PM	A	A
ServerMgr_4125.trc	3KB	TRC File	3/29/99 5:00 PM	A	A
siebel.NSCHAFER.log	6KB	Text Document	3/29/99 5:00 PM	A	A
MailMgr_4119.trc	12,816KB	TRC File	3/29/99 5:19 PM	A	A
ParaMgr_4120.trc	1? 816KB	TRC File	3/29/99 5:19 PM	A	A

Figure 11-2. Server Log Trace File

You can double-click the Trace File icon and access the trace file as shown in [Figure 11-3](#).



```
WorkActn_4123.trc - WordPad
File Edit View Insert Format Help
[ALTO1] Load Balanced : FALSE
[ALTO1] Disable Tracing : FALSE
[ALTO1] Minimum Number of cached thread contexts : 2
[ALTO1] Maximum Number of cached thread contexts : 10
[ALTO1] Request ID : 0
[ALTO1] Translate hostname to host IP address : TRUE
[ALTO1] Siebel Repository : Siebel Repository
[ALTO1] Table Owner : orac3
[ALTO1] Tablespace Name :
[ALTO1] Indexspace Name :
[ALTO1] User Name : SADMIN
[ALTO1] ODBC Data Source : SiebSrvr_siebel
[ALTO1] Siebel File System : CHANGE_ME
[TRC33] Connecting to the database...
[TRC33] Connected.
[DBG33] 1999-03-22 16:19:58 Process Name: Escalation Manager, Status: Started, Parameter: , Message
[TRC33] Loading Dictionary...
[DBG33] 1999-03-22 16:20:03 Process Name: Escalation Manager, Status: Started, Parameter: , Message
[TRC33] Reading rules...
[DBG33] 1999-03-22 16:20:07 Message: Failed to load rule/group information,
Additional Message: Medium Frequency
[DBG33] 1999-03-22 16:20:07 Message: Failed to load rules,
Additional Message: Error
[ERR33] (actnmain.cpp 11(227) err=700053 sys=1400022) ESC-00053: Error loading rule definitions
[ERR33] (smisched.cpp 13(460) err=700053 sys=D) ESC-00053: Error loading rule definitions
[DBG33] 1999-03-22 16:20:07 Process Name: Escalation Manager, Status: Completed with Errors, Message
[ALT99] 1999-03-22 16:20:07 Workflow Action Agent exited with error (task 4123), ESC-00053: Error
For Help, press F1 NUM
```

Figure 11-3. Server Log Trace File

You can view the trace file for any application server task. See *Siebel Server Administration Guide* for more information on using trace files.

Using 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.

Using the Policy Frequency Analysis Chart

Policy Frequency Analysis provides you with information about the number of times a Workflow policy executes.

To view the Policy Frequency Analysis

Navigate to Screens → Siebel Workflow Administration → Workflow Polices → Policy Frequency Analysis.

This view has two applets: Monitor Log and Workflow Policy Frequency Analysis. The Monitor Log applet lists the workflow policies. The Workflow Policy Frequency Analysis applet displays a chart illustrating the execution frequency of a selected policy.

Using the Policy Trend Analysis Chart

Policy Trend Analysis provides you with information about policy execution trends.

To view the Policy Trend Analysis

Navigate to Screens → Siebel Workflow Administration → Workflow Polices → Policy Trend Analysis.

This view has two applets: Monitor Log and Workflow Policy Trend Analysis. The Monitor Log applet lists the workflow policies. The Workflow Policy Trend Analysis applet shows the total number of workflow policy conditions met over a specified period of time.

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 in the menu bar of the Workflow Policies Policies view.

The Reports page that appears provides summary information of the Workflow Policy, as shown in [Figure 11-4](#).

The screenshot displays the Siebel Report Viewer interface. The main content area is titled 'Workflow Policy' and features the Siebel logo in the top right corner. Below the title, the report details for the 'ISS Activity Policy' are shown. The 'Workflow Object' is 'Service Request' and the 'Group' is 'ISS Group'. The 'Activation Date/Time' and 'Expiration Date/Time' are both blank, and the 'Duration' is also blank. The 'Created By' is 'SADMIN' and the 'Created On' is '1/1/80'. The 'Comments' section states: 'Policy to send EMail when a new activity is added to the Service Request'. The 'Conditions' section is a table with three columns: 'Column Instance', 'Comparison', and 'Value'. The 'Actions' section is a table with seven columns: 'Action', 'Sequence', 'Contact Last Name', 'Contact First Name', 'Employee Login', 'Position', and 'Relative'. The 'Status' at the bottom of the report is 'Report complete - 13 page(s)' and the page number is 'Page: 1 of 13'.

Column Instance	Comparison	Value
Activity Component	IS ADDED	
Activity Private	=	N
Receive Service Request Updates By EMail	=	Y

Action	Sequence	Contact Last Name	Contact First Name	Employee Login	Position	Relative
ISS Activity Action	1					SR Primary Contact

Figure 11-4. Workflow Policy Reports Page

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.

In a similar manner, you can select Reports from the menu bar in the Workflow Policies Log view and print out a Log report for each policy. This report provides a list of all the instances of that policy.

Workflow Policies and Siebel Marketing **12**

Using Workflow Policy Programs for Campaign Execution	12-2
Using Send Campaign Email	12-2
Using Create Email Activity	12-3
Using Assign to Campaign	12-3
Understanding a Campaign Workflow Scenario	12-4

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.
- **Create Email Activity.** Creates an activity record on all the contacts or prospects that were sent an email.
- **Assign to Campaign.** Takes a contact or a prospect and assigns it to a chosen campaign.

Using Send Campaign Email

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 Contact/Prospect applet on the Campaign management → Details view.

Using Create Email Activity

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 12-1](#) describes valid values for the Arguments applet.

Table 12-1. 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	
Value	Text or picklist.

Using Assign to Campaign

This workflow policy program adds the selected contact or prospect to the list of campaign contacts or prospects for the designated campaign.

Table 12-2. 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.

Understanding a Campaign Workflow Scenario

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
- Create a workflow policy group for the campaign
- Create the policies for the two tiers of the campaign

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 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.

a Enter the policy name:

Email for CD-ROM campaign

b Choose a workflow policy object from the picklist:

Campaign Contact

c Choose a workflow policy group from the picklist:

Campaign Group

d Enter a zero 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 start date.

c 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.

a Enter the policy name:

Non-Respondents of CD-ROM campaign

b Choose a workflow policy object from the picklist:

Campaign Contact

c Choose a workflow policy group from the picklist:

Campaign Group

d 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.

Testing Workflow Policies **13**

Overview	13-2
Testing New Policies and Monitoring the Results	13-3
Troubleshooting	13-5

Overview

Testing your workflow policies before implementing them into your production environment ensures that action recipients 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 [Chapter 8, “Planning for Workflow Policies.”](#)

Before you can test your new workflow policies, you must install the Siebel Server workflow policy components on the Siebel Server. See *Siebel System Administration Guide* for more information.



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

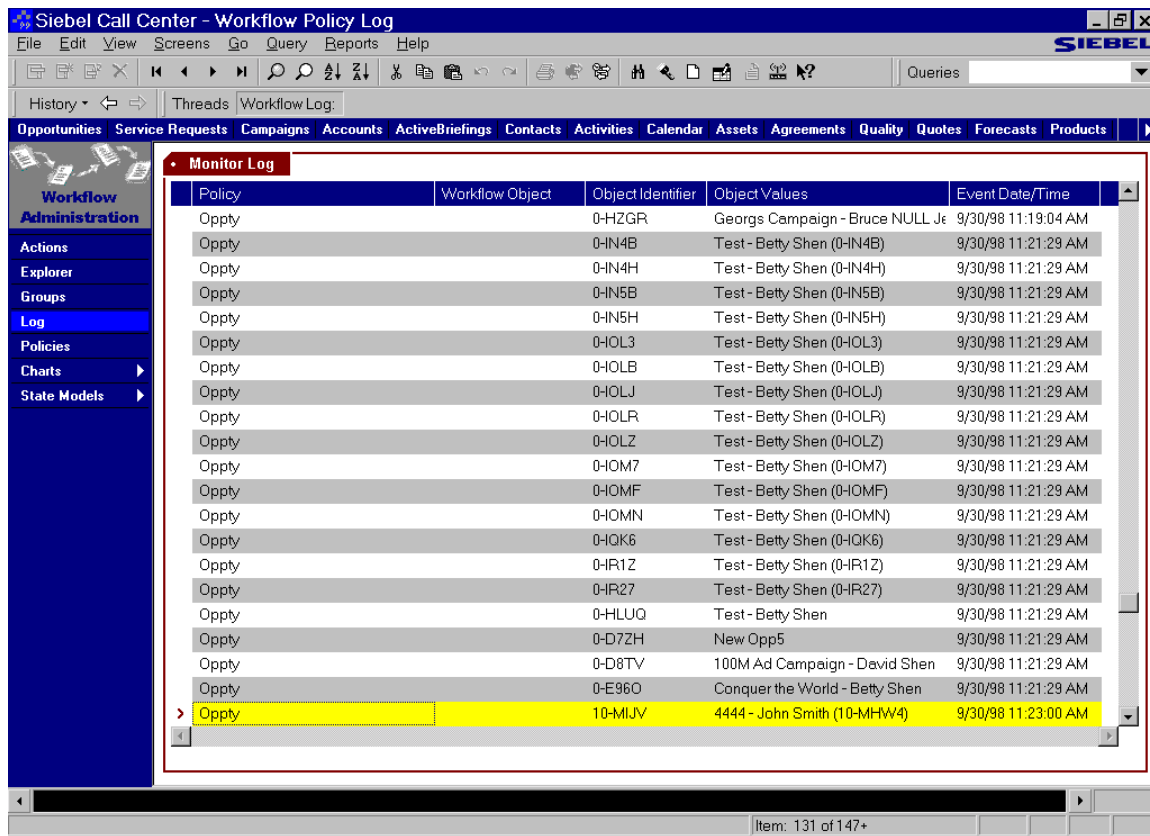
Correctly testing your workflow policies and eliminating any problems are critical before implementing the policies in your production environment.

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.

Testing Workflow Policies

Testing New Policies and Monitoring the Results

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 (see [Figure 13-1](#)).



The screenshot shows the Siebel Call Center interface with the Workflow Policy Log view open. The window title is "Siebel Call Center - Workflow Policy Log". The interface includes a menu bar (File, Edit, View, Screens, Go, Query, Reports, Help) and a toolbar with various icons. A navigation pane on the left shows "Workflow Administration" with sub-items: Actions, Explorer, Groups, Log (selected), Policies, Charts, and State Models. The main area displays a table titled "Monitor Log" with the following columns: Policy, Workflow Object, Object Identifier, Object Values, and Event Date/Time. The table contains 20 rows of log entries, with the last row highlighted in yellow.

Policy	Workflow Object	Object Identifier	Object Values	Event Date/Time
Oppty		0-HZGR	Georgs Campaign - Bruce NULL Jc	9/30/98 11:19:04 AM
Oppty		0-IN4B	Test - Betty Shen (0-IN4B)	9/30/98 11:21:29 AM
Oppty		0-IN4H	Test - Betty Shen (0-IN4H)	9/30/98 11:21:29 AM
Oppty		0-IN5B	Test - Betty Shen (0-IN5B)	9/30/98 11:21:29 AM
Oppty		0-IN5H	Test - Betty Shen (0-IN5H)	9/30/98 11:21:29 AM
Oppty		0-HOL3	Test - Betty Shen (0-HOL3)	9/30/98 11:21:29 AM
Oppty		0-HOLB	Test - Betty Shen (0-HOLB)	9/30/98 11:21:29 AM
Oppty		0-HOLJ	Test - Betty Shen (0-HOLJ)	9/30/98 11:21:29 AM
Oppty		0-HOLR	Test - Betty Shen (0-HOLR)	9/30/98 11:21:29 AM
Oppty		0-HOLZ	Test - Betty Shen (0-HOLZ)	9/30/98 11:21:29 AM
Oppty		0-IOM7	Test - Betty Shen (0-IOM7)	9/30/98 11:21:29 AM
Oppty		0-IOMF	Test - Betty Shen (0-IOMF)	9/30/98 11:21:29 AM
Oppty		0-IOMN	Test - Betty Shen (0-IOMN)	9/30/98 11:21:29 AM
Oppty		0-IQK6	Test - Betty Shen (0-IQK6)	9/30/98 11:21:29 AM
Oppty		0-IR1Z	Test - Betty Shen (0-IR1Z)	9/30/98 11:21:29 AM
Oppty		0-IR27	Test - Betty Shen (0-IR27)	9/30/98 11:21:29 AM
Oppty		0-HLUQ	Test - Betty Shen	9/30/98 11:21:29 AM
Oppty		0-D7ZH	New Opp5	9/30/98 11:21:29 AM
Oppty		0-D8TV	100M Ad Campaign - David Shen	9/30/98 11:21:29 AM
Oppty		0-E96O	Conquer the World - Betty Shen	9/30/98 11:21:29 AM
Oppty		10-MIJV	4444 - John Smith (10-MHW4)	9/30/98 11:23:00 AM

Figure 13-1. Workflow Policies Log View

The Workflow Policies 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 Policies 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 of 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

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 ensure that the appropriate behavior is 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.

Migrating Workflow Policies

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 *Siebel Upgrade Guide*.
- 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 Server Manager Client, navigate to Screens → Server Administration → Servers → Server Tasks.
- 5** Click New. Select Generate Triggers from the Server Components list. This creates a new line entry but does not start the task.
- 6** Click Parameters to modify parameter settings.

For a description of the component-specific parameters for Generate Triggers, see [Chapter 11, “Workflow Policies Server Administration.”](#) See *Siebel Server Administration Guide* for a description of the generic and enterprise parameters.

- 7** Click Close to save the parameter settings and click Start to execute the task.
A new task appears at the bottom of the list with a state of Starting Up and a PID (process ID). The state changes as Generate Triggers runs.
- 8** To view changes to the state, refresh the screen by clicking Execute Query or choosing the Execute Query menu entry.
- 9** Upon completion, the State field contains either Completed or Exited With Error. Clicking the Task link (a task ID such as 4140) provides a general log.
See [Chapter 11, “Workflow Policies Server Administration,”](#) for more information on trace files. For more information on using trace flags, see *Siebel Server Administration Guide*.

NOTE: To ensure that invalid triggers are not applied to your production environment, apply your database triggers to your test environment before you apply them to your production environment.

Monitoring and Performance Tuning **15**

Monitoring Workflow Policies	15-2
Using the Workflow Policies Log View	15-3
Understanding the S_ESCL_REQ Table	15-5
Understanding the S_ESCL_STATE Table	15-6
Understanding the S_ESCL_ACTN_REQ Table	15-7
Tuning Workflow Policies for Performance	15-8
Workflow Policy Groups to Manage Server Load	15-8
Multiple Workflow Monitor Agents and Workflow Action Agents	15-8
Workflow Agents on Multiple Application Servers	15-8
Running Workflow Agent Processes through Server Manager (svrmgr)	15-9
Optimal Sleep Interval for Workflow Policy Groups	15-9
Optimal Action Interval for Each Workflow Action Agent	15-10
Creating Multiple Workflow Agent Processes	15-10
Performance Tuning Tips	15-11

Monitoring Workflow Policies

You need to monitor Workflow Policies regularly to ensure that all events are handled correctly and that the server uses its resources optimally. Purging your log files periodically prevents them from becoming too large.

You can monitor a Workflow Agent process using the following logs and files:

- **Workflow Policies Log view.** Provides a list of all executed policies. The Policy Frequency Analysis view and Policy Trend Analysis views are available to analyze how frequently policies are executed over time.
- **Workflow Agent Trace logs.** These include:
 - Workflow Monitor Task Log. Workflow Monitor provides detailed information about its processing in its trace file.
 - Workflow Action Agent Task log. Workflow Action Agent provides detailed information about its processing in its trace file.
 - Email and Page Server Trace logs.
 - Run Email and Page Server components with Trace Flag set to 1 for detailed reporting on email activity.
 - Query S_APSRVR_REQ for status information on email and page requests that were logged by Action Agent.
- Admin logging Facility
- Admin Task Info Log
- S_ESCL_REQ table
- S_ESCL_STATE table
- S_ESCL_ACTN_REQ table

Using the Workflow Policies Log View

Figure 15-1 shows the Workflow Policies Log view that is available in the Siebel Client.

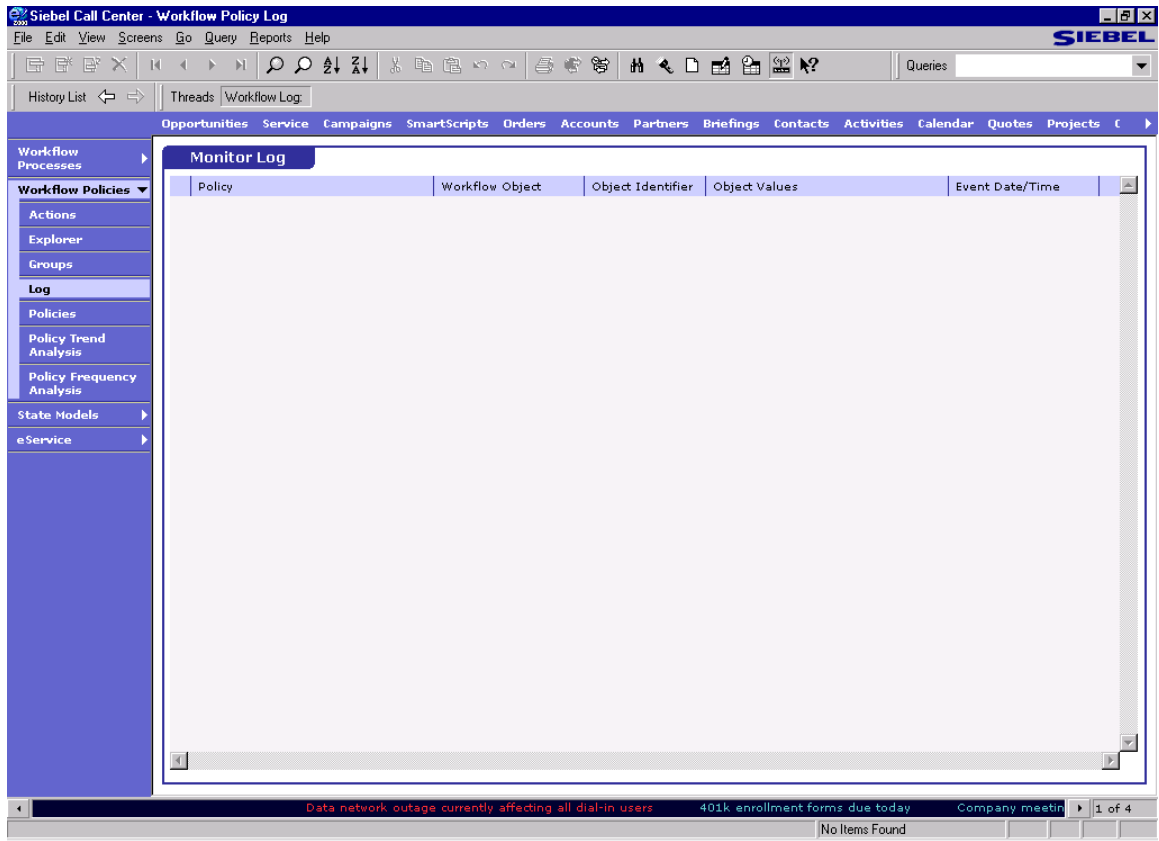


Figure 15-1. Workflow Policies Log View

The Workflow Policies Log view displays a log of all the policies executed as evidenced by a Workflow Monitor Agent process. The policy maker can monitor Workflow Agent process activity to determine if the current policies are adequate, if new policies need to be created, or if policies need to be refined. The Workflow Policies Log view is described in *Siebel Online Help*.

You access the Workflow Policies Log view from the Siebel Client. The log information is generated by the server components of Siebel Workflow Policies.

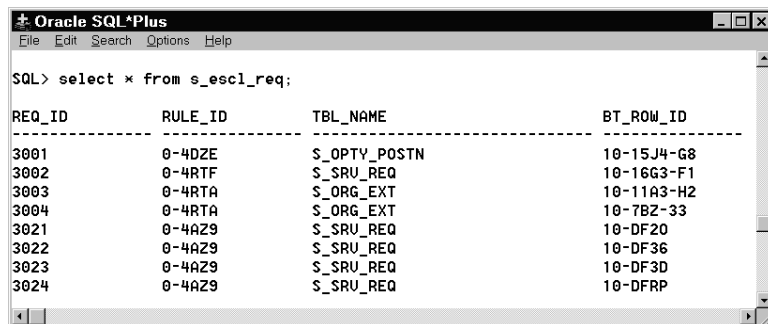
NOTE: The Policy Frequency Analysis and Policy Trend Analysis views let you view Policy Log data in a graphical format.

The Workflow Policies Log view contains the following fields:

- **Policy.** The name of the policy that was executed
- **Workflow Object.** The name of the assigned workflow policy object
- **Object Identifier.** The ID of the workflow policy object for which the policy was executed
- **Object Values.** Identifying information for the row that executed the policy
- **Event Date/Time.** The date and time of the policy execution event

Understanding the S_ESCL_REQ Table

Figure 15-2 shows the S_ESCL_REQ table.



The screenshot shows the Oracle SQL*Plus interface with the following data displayed in the S_ESCL_REQ table:

REQ_ID	RULE_ID	TBL_NAME	BT_ROW_ID
3001	0-4DZE	S_OPTY_POSTN	10-15J4-G8
3002	0-4RTF	S_SRU_REQ	10-16G3-F1
3003	0-4RTA	S_ORG_EXT	10-11A3-H2
3004	0-4RTA	S_ORG_EXT	10-7BZ-33
3021	0-4AZ9	S_SRU_REQ	10-DF20
3022	0-4AZ9	S_SRU_REQ	10-DF36
3023	0-4AZ9	S_SRU_REQ	10-DF3D
3024	0-4AZ9	S_SRU_REQ	10-DFRP

Figure 15-2. S_ESCL_REQ Table

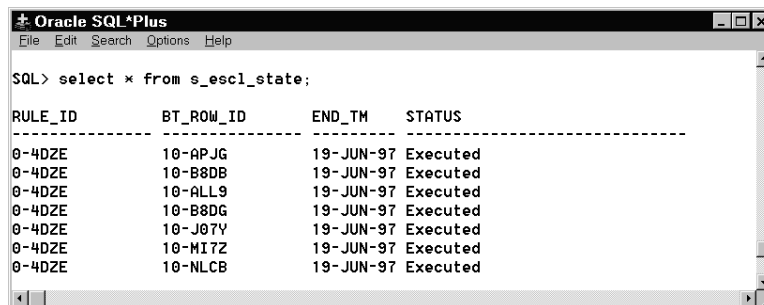
When a trigger fires against a Workflow policy condition, a record is inserted in the Escalation Request table, S_ESCL_REQ. This table contains all the rows in the database that could trigger a Workflow Policy to take action. After the workflow Monitor Agent processes a request, it removes the row from this table.

Use your database tools to monitor the size and efficiency of the table. If the table becomes very large, this could indicate that the number of policies being monitored is too large and a new Workflow Policies process needs to be created to share the load. If rows are being monitored and not being removed after the time interval is met, this could indicate that a policy was deactivated without removing the database triggers. The triggers are continuing to send data that is not being acted on by a Workflow Policies instance.

NOTE: If you expire or delete any active Workflow Policies, confirm that no outstanding records are in the S_ESCL_REQ, S_ESCL_ACTN_REQ, or S_ESCL_STATE tables.

Understanding the S_ESCL_STATE Table

Figure 15-3 shows the S_ESCL_STATE time-based table.



The screenshot shows an Oracle SQL*Plus window with the following data:

RULE_ID	BT_ROW_ID	END_TM	STATUS
0-4DZE	10-APJG	19-JUN-97	Executed
0-4DZE	10-B8DB	19-JUN-97	Executed
0-4DZE	10-ALL9	19-JUN-97	Executed
0-4DZE	10-B8DG	19-JUN-97	Executed
0-4DZE	10-J07Y	19-JUN-97	Executed
0-4DZE	10-MI7Z	19-JUN-97	Executed
0-4DZE	10-NLCB	19-JUN-97	Executed

Figure 15-3. S_ESCL_STATE Table

The S_ESCL_STATE time-based table contains all the rows that have been executed (all conditions are true) and are waiting for the time duration element to expire.

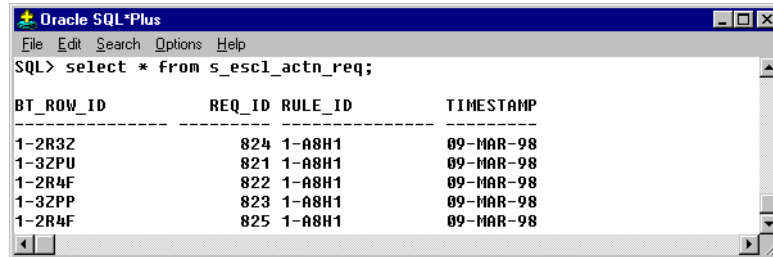
Use your database tools to monitor the size and efficiency of the S_ESCL_STATE table. If the table becomes very large, this could indicate that the number of policies being monitored is too large for the current number of Workflow Policies processes and a new Workflow Policies process needs to be created to share the load.

If rows are being monitored, but are not being removed after the time interval is met, this could indicate that a policy was deactivated without removing the database triggers. The triggers are continuing to send data that is not being acted on by a Workflow Policies process.

NOTE: If you expire or delete any active Workflow Policies, confirm that no outstanding records are in the S_ESCL_REQ, S_ESCL_ACTN_REQ, or S_ESCL_STATE tables.

Understanding the S_ESCL_ACTN_REQ Table

Figure 15-4 shows the S_ESCL_ACTN_REQ table.



The screenshot shows the Oracle SQL*Plus interface with a query executed: `SQL> select * from s_escl_actn_req;`. The results are displayed in a table with the following columns: BT_ROW_ID, REQ_ID, RULE_ID, and TIMESTAMP. The data rows are as follows:

BT_ROW_ID	REQ_ID	RULE_ID	TIMESTAMP
1-2R3Z	824	1-A8H1	09-MAR-98
1-3ZPU	821	1-A8H1	09-MAR-98
1-2R4F	822	1-A8H1	09-MAR-98
1-3ZPP	823	1-A8H1	09-MAR-98
1-2R4F	825	1-A8H1	09-MAR-98

Figure 15-4. S_ESCL_ACTN_REQ Table

The S_ESCL_ACTN_REQ table contains all the rows that are awaiting action execution. These rows have violated the policy; and the time duration element, if any, has expired.

Use your database tools to monitor the size and efficiency of the S_ESCL_ACTN_REQ table. If the table becomes very large, this could indicate that the number of policies being monitored is too large for the current number of Workflow Policies processes and a new Workflow Policies process needs to be created to share the load.

If rows are being monitored, but are not being removed after the time interval is met, this could indicate that a policy was deactivated without removing the database triggers. The triggers are continuing to send data that is not being acted on by a Workflow Policies process.

NOTE: If you expire or delete any active Workflow Policies, confirm that no outstanding records are in the S_ESCL_REQ, S_ESCL_ACTN_REQ, or S_ESCL_STATE tables.

Tuning Workflow Policies for Performance

Workflow Policies can be tuned to optimize your resources and also meet the policy's timing requirements by grouping similar policies and assigning these policy groups to servers that can handle the workload.

Performance tuning can be handled in the following interrelated ways:

Workflow Policy Groups to Manage Server Load

Workflow policy groups allow you to group policies with similar polling intervals. This distributes the load to ensure efficient processing. For example, if you have very critical policies that must be responded to within minutes of the policy trigger event and you have other policies that need a response within a day, you would assign them to different workflow policy groups.

The advantage of selective grouping is that a Workflow Agent's polling resources are focused on a smaller number of policies, ensuring effective monitoring and action execution.

Multiple Workflow Monitor Agents and Workflow Action Agents

For each Workflow policy group, you need to define a Workflow Monitor Agent and Action Agent combination. Each Workflow Agent combination monitors the policies within its assigned workflow policy group.

If you are a high-volume call center or you have a large number of policies that need very short polling intervals, you may want to create multiple groups with Workflow Agent processes to run in parallel. A single Workflow Agent process that is monitoring and handling a large number of events may become slow to respond and not meet the time interval commitments set by the policy.

Workflow Agents on Multiple Application Servers

You can run Workflow Agent processes on different servers to ease the workload on each server. You can then adjust the polling interval for each group so that polling for noncritical policies does not prevent efficient processing of critical policies.

Running Workflow Agent Processes through Server Manager (srvrmgr)

To run Workflow Monitor Agent and Workflow Action Agent using a command-line interface, you need to supply the Component Type. You can get the Component Type by navigating to Screens → Server Administration → Enterprise → Component Definitions. Workflow Monitor Agent is WorkMon, and Workflow Action Agent is WorkActn.

For example, to get parameters for WorkMon, list parameters for server Siebel_Server_Name component WorkMon.

NOTE: You need to enclose parameter values that contain a space in quotes. For example, if you run the workflow monitor at the command line with `GROUPNAME=Assignment Group`, you will get an error. You should use `GROUPNAME="Assignment Group"`.

See *Siebel Server Administration Guide* for complete `srvrmgr` syntax.

Optimal Sleep Interval for Workflow Policy Groups

By creating groups with similar polling intervals, you can assign the workflow policy group to a Workflow Agent process with a polling rate that matches the workflow policy group. Different polling intervals can be assigned to each workflow policy group using the Sleep Time parameter (see [Chapter 11, “Workflow Policies Server Administration,”](#) for more information on Workflow Policies).

After Workflow Agents process all requests, the agent processes sleep for the interval specified by this argument before processing begins again. You should set the sleep intervals as large as is possible, but at an interval that still meets your business requirements.

You should adjust the sleep interval for each Workflow Agent process to meet the requirements of each workflow policy group. For example, workflow policy group A contains accounts that require a response to a Severity 1 service request within 10 minutes. Workflow policy group B contains policies that require a customer follow-up call within 14 days. Workflow policy group A is very time-critical, so you could set the sleep interval to 60 seconds so that the assigned Workflow Policies instance polls frequently. Workflow policy group B is not as time-critical, so you could set the sleep interval to 48 hours and the Workflow Policy instance can still meet its commitments.

Another example where optimal configuration of the Sleep Time parameter may be required is in the case of multiple users who may need to update the same record. If you have, for example, a workflow policy that monitors service requests and you have multiple users that retrieve and modify open service request records, you need to set the sleep time parameter so that users will have enough time to update the text fields.

If the sleep interval is not set high enough, you may encounter an error message stating “The selected record has been modified by another user since it was retrieved. Please continue.” and you will lose your changes as the new field values for this record are displayed.

Optimal Action Interval for Each Workflow Action Agent

For each Workflow Action Agent, you can set an action interval, which determines when actions for a given policy are re-executed 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. For example, if a service request severity is set to critical and triggers a policy, you do not want to re-execute the policy action if it is changed and has been reset to critical during this interval.

Creating Multiple Workflow Agent Processes

To create multiple Workflow Agent processes

- 1** Go to the Workflow Policies Groups view to create the workflow policy groups and assign policies to the groups.

In this first step, group all your policies according to their similar polling intervals. If you have a large number of policies in a group with short polling intervals, divide them into multiple groups.

- 2** Go to the Server Tasks view under Siebel Server.
- 3** Click New.
- 4** Select the Workflow Monitor Agent or Workflow Action Agent component.
- 5** Click Parameters.
- 6** Set the group for Workflow Monitor Agent or Workflow Action Agent to monitor.

- 7 Set an appropriate sleep interval parameter. See [Chapter 11, “Workflow Policies Server Administration,”](#) for a list of parameters.
- 8 Click Start to begin Workflow Monitor Agent or Workflow Action Agent.

Repeat these steps to start the Workflow Monitor Agent and Workflow Action Agent process for each Workflow group.
- 9 Test your groups and sleep intervals.



Caution: Assign a workflow policy group to only one Workflow Monitor Agent and Workflow Action Agent process. Multiple Workflow Monitor Agent and Workflow Action Agent processes running the same workflow policy group cause unpredictable completion times and possible multiple actions to be created for one trigger.

Performance Tuning Tips

The following are some tips for dealing with common performance problems:

- Monitor the size of S_ESCL_REQ and S_ESCL_STATE tables.

If these tables are too large, they may slow down your system’s performance.

In general:

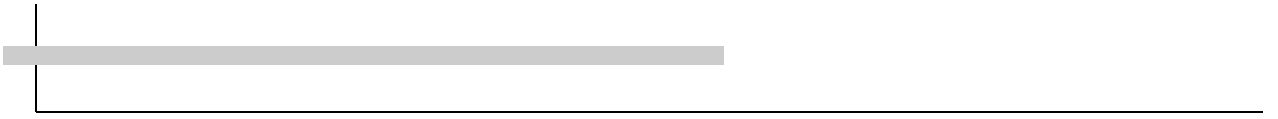
- Too many records indicate that the number of workflow policies monitored is too large. In this case, add additional Workflow Agents to handle the load.
- If rows are not removed after the time interval is met, the workflow policy may have been deactivated but the triggers were not removed. These tables will become very large if you do not restart Generate Triggers.
- Run multiple Workflow Monitor and Workflow Action Agents in parallel. Doing so:
 - Focuses an agent’s polling resources on a smaller number of workflow policies.
 - Allows faster throughput by shortening the time between when the workflow policy event is triggered and when the agent notices the event.

- Distribute workflow policy processes across Siebel Servers. You can distribute processes so that:
 - High-maintenance policies can be grouped on a server with sufficient resources to handle the workflow CPU requirements.
 - Low-maintenance policies can be run on a server that shares resources with other Siebel processes.
- If you find that Siebel Workflow Policies runs significantly slower during a certain time period, you should investigate what other processes may be contending for CPU resources on the Siebel Application Server. You may discover that the server has certain time periods with high activity that interfere with the ability of the Workflow Policies process to monitor or act. Arrange the Workflow Policies processes on the servers so that the polling periods are compatible with the resources available.

State Models

Part 4

Chapter 16. Using State Models



Using State Models **16**

State Model Overview	16-2
Creating State Models and State Transitions	16-5
Creating State Models	16-5
Creating State Transitions	16-10
Using a State Model	16-13

State Model Overview

The State Models views provide a data-driven method for extending workflow control based on the status of an object, such as a service request or product defect. A state represents the status of an object, such as Open, Closed, or Pending. The State Models module consists of three key elements:

- **State Model.** A blueprint of acceptable states and state transitions that are enforced by the state machine.
- **State Machine.** An engine that enforces the transitions between states for an object during its lifetime. The state represents where the object is in its lifetime. The state can also control whether or not the data of that object can be modified. As an example, a service request that is in a Closed state may be considered “frozen,” such that its attributes cannot be modified.
- **State Transition.** Defines the allowable migration of an object from one state to the next. For instance, a service request that has been closed but must be re-opened may go from the Closed state to an Open state, and may go from Open to Pending, but may not transition directly from Closed to Pending.

The state transition can also enforce other behavior. It can control which individuals or groups of individuals can initiate the transition. It also can verify that specific conditions have been met before the transition occurs.

Figure 16-1 provides an example of a state model for a service request.

The screenshot displays the Siebel Call Center - State Models application. The main window shows a list of state models with the following data:

Name	BusComp Name	Field Name	Activation Date/Time	Expiration Date
Customer Service Request Status	Service Request	Status	09/11/1998 1:58:12 PM	
Defect Status	Product Defect	Status	10/05/1998 3:13:54 PM	
Defect Sub-Status	Product Defect	Sub-Status	10/05/1998 3:29:00 PM	
Expense Report Approval	Expense	Status	01/01/1970 12:00:00 AM	
Fund Request Status	Fund Request	Status	12/15/1999 8:41:24 PM	
Incentive Compensation Plan Status	Incentive Compensation Plan	Status Code	05/27/1999 7:59:35 PM	
SrvReqMgr Status	SrvReqMgr Request	Status Displayed	09/03/1998 5:07:38 PM	
SrvAdmin Svcddef State	Enterprise Service Definitions	State	11/02/1998 6:44:55 PM	
Time Sheet Approval	Time Sheet	Status	01/01/1970 12:00:00 AM	

Below the list is a detailed view of the 'Customer Service Request Status' model:

- State Model:** Customer Service Request Sta
- BusComp:** Service Request
- Field:** Status
- Activation Date/Time:** 09/11/1998 1:58:12 PM
- Expiration Date/Time:**
- States:** Open
- State Description:** Open
- Comments:** State model to control who can cancel or close a service request
- Created By:** ADMIN
- Created On:** 09/11/1998 1:58:12 PM

Figure 16-1. Example of a State Model for a Service Request

In this state model, the starting point state for all service requests is Open.

Allowable state transitions for this model are listed in [Table 16-1](#).

Table 16-1. State Transitions for a Service Request

From State	To State
Cancelled	Open
Closed	Open
Open	Cancelled
Open	Closed
Open	Pending
Pending	Cancelled
Pending	Closed
Pending	Open

The state transition can define the required authorization for transitioning an object. For example, a transition of moving service requests from Open to Cancelled may be restricted to the service center manager. By default, anyone with update authority on an object can change the state, unless this authority is explicitly overridden by the state transition definition within the state model.

The state transition can be preempted if required conditions have not been met. The state machine reviews all required conditions defined for the state model before the state transition is completed.

Creating State Models and State Transitions

This section describes how to create and define state models and state transitions.

Creating State Models

Creating and defining a state model is handled through the State Models view, shown in [Figure 16-2](#). To display this view, choose Screens → Siebel Workflow Administration → State Models → State Models.

Name	BusComp Name	Field Name	Activation Date/Time	Expiration Date
Customer Service Request Status	Service Request	Status	09/11/1998 1:58:12 PM	
Defect Status	Product Defect	Status	10/05/1998 3:13:54 PM	
Defect Sub-Status	Product Defect	Sub-Status	10/05/1998 3:29:00 PM	
Expense Report Approval	Expense	Status	01/01/1970 12:00:00 AM	
Fund Request Status	Fund Request	Status	12/15/1999 8:41:24 PM	
Incentive Compensation Plan Status	Incentive Compensation Plan	Status Code	05/27/1999 7:59:35 PM	
SrvReqMgr Status	SrvReqMgr Request	Status Displayed	09/03/1998 5:07:38 PM	
SrvAdmin Svcddef State	Enterprise Service Definitions	State	11/02/1998 6:44:55 PM	
Time Sheet Approval	Time Sheet	Status	01/01/1970 12:00:00 AM	

New	Delete	Copy	Cancel
State Model	Customer Service Request Sta	States	Open
BusComp	Service Request	State Description	Open
Field	Status	Created By	SADMIN
Activation Date/Time	09/11/1998 1:58:12 PM	Created On	09/11/1998 1:58:12 PM
Expiration Date/Time		Comments	State model to control who can cancel or close a service request

Figure 16-2. State Models View

Table 16-2 describes the fields in the State Models view.

Table 16-2. Fields in the State Models View (1 of 2)

Field	Description
Activation Date/Time	The date on which the state model begins to be enforced.
BusComp Name	The business component that the state model is based on. The pick applet for this field displays all of the business components that have been enabled for the state model. To enable a business component that is not enabled by default, see “Enabling Business Components for State Model” on page 16-8.
Created By	The Siebel user name for the user who created the state model. This read-only field is not displayed in the list applet.
Created On	The date and time that the state model was created. This read-only field is not displayed in the list applet.
Comments	Text that you can add to describe the state model.
Expiration Date/Time	The date on which the state model is no longer enforced.
Field	The name of the field that the state transitions apply to. The pick applet for this field displays all of the fields that have been defined for the business component selected in the BusComp Name field.
Name	Name that uniquely identifies the state model. This is a free-form text field.

Table 16-2. Fields in the State Models View (2 of 2)

Field	Description
State Description	<p>A description of the state that is named in the State Name field. You can specify or modify the description for the current state in the States multi-value group applet.</p>
State Name	<p>A list of all possible values for the business component field that the state model is based on. This is a multi-value group. All values that are used in the state transition must be added to the values for State Name.</p> <p>To add a new value, click the ellipsis button, and then click the New button in the States multi-value group applet. This applet also allows you to change which state name is the default, and to modify the state description.</p> <p>If there is a list of values for the business component field, you can select the value from a pick applet. Otherwise, enter the appropriate value in the State Name field.</p> <p>Optionally, you may specify restrictions for the state model by activating the following flags:</p> <ul style="list-style-type: none"> ■ No Delete. When checked, records in this state cannot be deleted unless they are referenced by a parent record in which the Cascade Delete property is set to Delete. Cascade Delete will always override the State Model restrictions. ■ No Update. When checked, records in this state are read-only and cannot be updated. ■ Restrict Transition. When checked, records in this state that do not have a state transition defined cannot change states. This effectively turns the record's state into an end state. <p>Once all states are defined, you should select one state as the default. The default state name specifies a beginning value for the business component field. For example, in the Service Request business component, the default value for the Status field is Open.</p> <p>To specify or change which state name is the default, click in the Default column on the States multi-value group applet.</p>

To create a new state model

- 1** Choose Screens → Siebel Workflow Administration → State Models → State Models.
- 2** Create a new record for a state model by choosing Edit → Add New Record.
- 3** Fill in the appropriate fields described in [Table 16-2 on page 16-6](#).

NOTE: If you are using multiple sales methods, do not create a state model in the Sales Stage field in the Opportunity business component. The Sales Stage field is dependent on the sales method selected for the Opportunity. If a state model is created in this field and multiple sales methods are being used, the sales methods will not function properly.

Enabling Business Components for State Model

You can enable state model on any business component (not enabled by default) that is based directly on the CSSBCBase class.

NOTE: Enabling state model for business components based on subclasses of CSSBCBase class is not supported.

To enable state model on a business component, the fields used for State Model:

- must be of single value
- can only have a specific set of values as defined by the set of records returned by querying Screens/Application/List of Values/List of Values on the corresponding LOV_TYPE. For example, TODO_TYPE is the set of action types. The corresponding field in the Activity business component can only take one of those values
- should have no dependencies on other fields as described in the note above

To enable business components for state model functionality

- 1** Lock the project for the selected business component.
- 2** Create a new business component user property with the following properties:
 - Name = State Model
 - Value = Y
- 3** Recompile the Siebel Repository.

NOTE: For these Siebel Tools objects to be accessible in the Siebel client, the Siebel Repository must be updated in the Siebel database. State Models 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”.



Caution: Do not create a state model for the business components Enterprise Service Definitions or SRMSynch Request. These are used for server administration. If a state model for one or both of these business components exists, do not modify it.

Creating State Transitions

Creating and defining a state transition is handled through the State Transitions view, shown in [Figure 16-3](#). To display this view, choose Screens → Siebel Workflow → State Models → State Transitions.

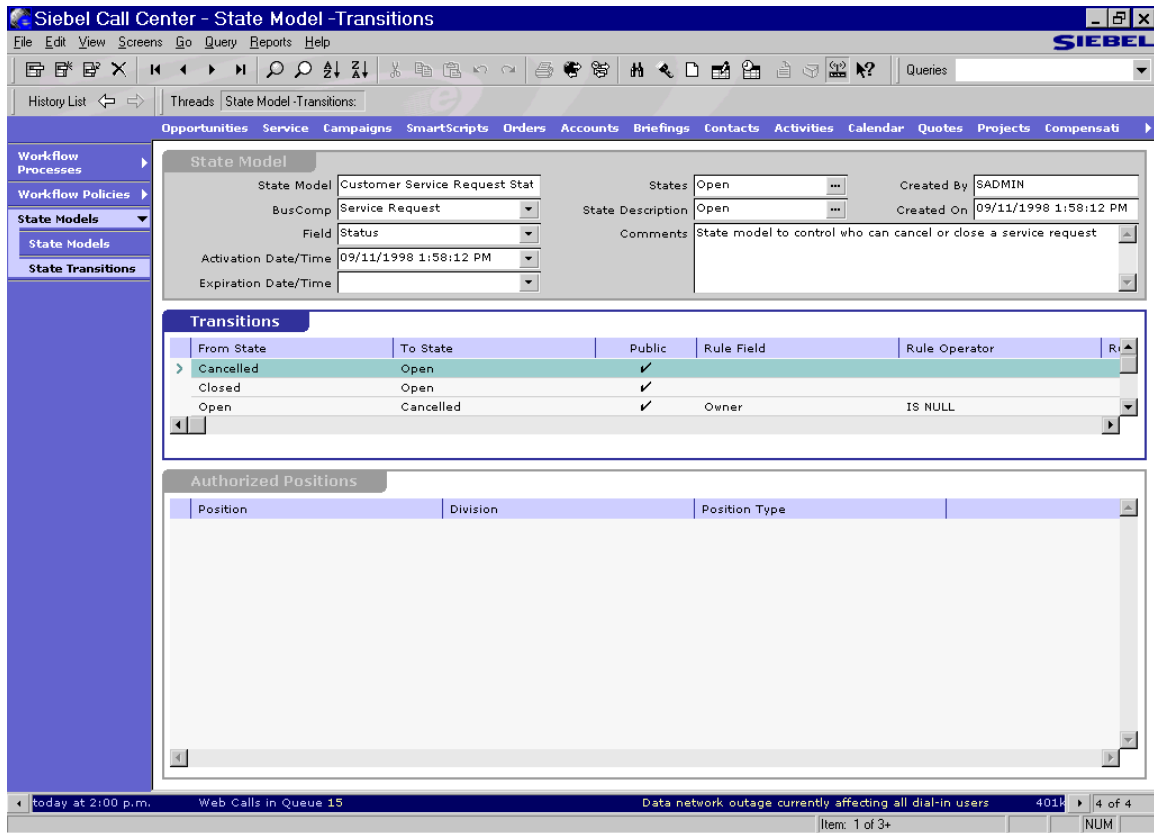


Figure 16-3. State Transitions View

If a state model exists for a field in a business component, all possible transitions for the field must be defined in the State Transitions view. If a transition is not explicitly defined, it will not be available to users.

[Table 16-3](#) and [Table 16-4](#) describe the fields in the State Transitions view. The State Model form applet at the top of the view contains data from the State Models view.

Table 16-3 describes the fields in the Transitions list applet.

Table 16-3. Fields in the State Transitions View, Transitions List Applet

Field	Description
From State Name	<p>The original field value for the state transition. The picklist for this field will display all of the state names that have been defined for the state model.</p> <p>Note that if a state transition is not defined for the default State Name value, the business component field that the state model is based on cannot be changed.</p>
Public Flag	<p>Indicates if the transition is available to all users.</p> <p>By default, the Public Flag is checked to indicate that all users will be allowed to make the transition. Positions specified in the Authorizations list applet will be ignored.</p> <p>If you want to restrict the transition to a subset of users, uncheck this field and add the positions for these users in the Authorizations list applet.</p> <p>If Public Flag is unchecked but no positions have been added in the Authorizations list applet, no users will be able to make the transition.</p>
Rule Field Name	<p>Allows you to specify a condition that must be satisfied for the transition to occur.</p> <p>The pick applet displays a list of all fields for the business component that the state model is based on. To create a condition, select a field whose value should be checked before the transition occurs.</p>
Rule Expression	<p>Allows for the creation of complex or multiple conditions that must be satisfied for users to make the transition.</p> <p>The syntax for the Rule Expression field is the same as the syntax for calculated field values and field validation in Siebel Tools. See <i>Siebel Tools Guide</i> for more information on the syntax.</p>
Rule Operator	<p>A comparison operator for the Rule Field Name field. If a rule field name was selected, this operator will determine how the rule field name is checked in relation to the Rule Value field.</p>
Rule Value	<p>Specifies the value to be checked in relation to the Rule Field Name and Rule Operator fields.</p> <p>For example, a state transition may be created that allows users to change the status of a service request from Open to Closed when the sub-status of the service request has been set to Resolved. In this example, the Rule Field Name value would be Sub-Status, the Rule Operator value would be =, and the rule value would be 'Resolved'.</p> <p>The rule value must be enclosed in single quotes, as shown in the preceding example.</p>
To State Name	<p>The new field value for the state transition, changing from the value indicated in From State Name.</p> <p>The picklist for this field will display all of the state names that have been defined for the state model.</p> <p>All conditions for the transition must have been satisfied before the transition is allowed.</p>

Table 16-4 describes the fields in the Authorizations list applet.

Table 16-4. Fields in the State Transitions View, Authorizations List Applet

Field	Description
Division	The name of the division in which the selected position is defined. This read-only value is populated from the Position field.
Position	<p>If the Public Flag field in the Transitions list applet is unchecked, the Position field reflects the name of the position that can make the state transition. Any employee with the selected position will have the authority to make the state transition.</p> <p>If the Public Flag field is checked, the ability to make the state transition is not restricted to users whose positions may be specified.</p> <p>The pick applet for this field will display all of the positions that have been defined in the organization.</p>
Position Type	The type for the selected position. This read-only value is populated from the Position field.

To create a new state transition

- 1 Choose Screens → Siebel Workflow Administration → State Models → State Models.
- 2 In the State Models view, click the Name field in the appropriate state model record.

The State Transitions view appears.
- 3 In the Transitions list applet, create a new state transition record by choosing Edit → Add New Record.

You can create one or more state transitions. For each state transition, fill in the appropriate fields, as described in Table 16-3 on page 16-11.
- 4 If you want to restrict the state transitions to a subset of users, uncheck the Public Flag field and add records in the Authorizations list applet to specify one or more positions for these users.

For a description of the fields, see Table 16-4 on page 16-12.

Using a State Model

After a state model is created, it can be immediately activated. No additional steps are required to activate a state model. As soon as the activation date and time are reached, the state model will be in effect.

The state model is not stored in the Siebel Repository. State models can be created at any point through the Siebel user interface without compiling a new Siebel Repository file. For mobile users who connect through Siebel Remote, the state model will be enforced once a user has synchronized with the database server.

The state model is enforced on the Siebel client, not on the Siebel Server. When a user activates one of the business components that has a state model, the Siebel client will ensure that any state models created for the business component are enforced. A business component is activated if a user navigates to an applet that is based on that business component.

User-defined business components are not supported.

NOTE: State models are enforced for updates made through Siebel Visual Basic because Siebel VB does not circumvent any business component logic. However, state models are not enforced for updates made through Enterprise Integration Manager or Siebel Workflow.

When a user tries to change a field that a state model is based on, such as a Status field, the user sees only the field values that have been defined as possible transitions. For example, if the state transition from Open to Closed has been defined, the value Closed will be available when the current state is Open.

The transition will be available for the user only if either of the following is true:

- The Public Flag field is checked for the transition.
- The Public Flag field is not checked for the transition, and the user's position is specified as having the authorization to make this transition.

If the transition has been defined with a condition (rule), this condition is checked when the user selects the new field value. If the condition has not been satisfied, the user will be notified that the condition was not satisfied, and the transition will not take place.

Using State Models

Using a State Model

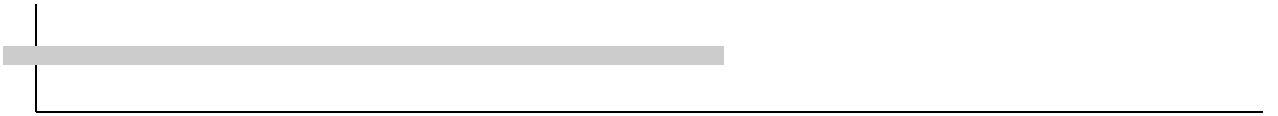
Appendixes

Part 5

Appendix A. Predefined Business Services

Appendix B. Predefined Programs

Appendix C. Siebel Workflow Terminology



This appendix contains a description of some of the predefined business services.

The services covered here are:

- Outbound Communications Manager.
- Synchronous Assignment Manager Requests.
- Synchronous (or Asynchronous) Server Requests.
- Workflow UI Utilities.
- Workflow Utilities.

For more predefined business services, see *Siebel eBusiness Application Integration Guide*.

Outbound Communications Manager

The Outbound Communications Manager business service is for sending notifications to contacts or employees. For more information, see *Siebel Communications Server Administration Guide*.

This service has three available methods:

- **Create and Submit Request.** Use this method to create and submit a request to the communications server. This method assumes that you want to send a previously created package.
- **Send Message.** Use this method for sending a notification without predefining a package.
- **Submit Request.** Use this method for submitting a request that has already been created.

Create and Submit Request Arguments

Table A-1. Create and Submit Request Method Arguments

Argument	Description
Comments	Optional. For sending notifications to contacts or employees.
Default Preference	Optional. If the contact does not have a contact preference and Only Send Preference flag is true, use this method for sending the notification.
Only Send Preference Flag	Optional. Set this flag to true if you want only to send a notification using the contact preference.
Package Name List	Required. Enter one or more packages to send to the recipient. You can either select a package from the picklist or enter one or more package names separated by semicolons.
Process Mode	Optional. This is the mode in which you want the communications server to run. The mode choices are Local or Remote. Local runs the communications server on the object manager, and Remote runs the communications server on the server. If this is not specified, the communications server runs on the server.
Request Name	Required. This is the name of the communications request that you want to submit.
Source Id List	Required. Enter one or more source ids to send to the recipient. You can either select a source from the picklist or enter one or more source id names separated by commas.
Recipient Group	Required. These are the recipients for the notification. You can select the recipient group from a picklist.

Send Message Arguments

Table A-2. Send Message Method Arguments

Argument	Description
Communication Profile	Required. The profile is used for delivering the notifications.
Bcc Address List	Optional. This is a list of email addresses that you want to include on the BCC line.
Message Body	Optional. This is the text message.
Cc Address List	Optional. This is a list of email addresses that you want to include on the CC line.
Message Subject	Optional. This is the subject line of the message.
To Address List	Required. This is a list of email addresses that you want to include on the TO line.
Process Mode	Optional. This is the mode in which you want the communications server to run. The mode choices are Local or Remote. Local runs the communications server on the object manager, and Remote runs the communications server on the server. If this is not specified, the communications server runs on the server.

Submit Request Arguments

Table A-3. Submit Request Method Arguments

Argument	Description
Request Id	Required (if Request Name is not entered). This is the row id for the communications request that you want to submit.
Request Name	Required (if Request Id is not entered). This is the name of the communications request that you want to submit.

Synchronous Assignment Manager Requests

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 A-4. 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.

Synchronous (or Asynchronous) Server Requests

The Synchronous (or Asynchronous) Server Requests business services are for sending generic requests to the server request manager. The Synchronous Server Request business service sends a request to the server request manager and waits for a response. The Asynchronous Server Request business service sends a request but does not wait for a response.

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. For more information on creating jobs, see *Siebel Server Administration Guide*.

This service has one method available, Submit Request. This method submits a request to the server request manager.

Submit Request Arguments

NOTE: In additions to the arguments described below, you must also specify required parameters, if any, for the particular component or component job of your request.

Table A-5. Submit Request Method Arguments (1 of 2)

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: <ul style="list-style-type: none"> ■ ASAP ■ SECONDS ■ MINUTES ■ HOURS ■ DAYS ■ WEEKS ■ MONTHS ■ YEARS
Description	Optional. A description of the server request.
Expiration Date	Optional. The date on which the request expires.
Hold Flag	Optional. For asynchronous requests only. Flag to indicate whether or not to hold the request.

Table A-5. Submit Request Method Arguments (2 of 2)

Argument	Description
Method	Optional. Only applicable for service-based server components (for example, Workflow Process Manager, Communications Manager). Specify the business service method to invoke.
Notify	For future use.
Priority	Optional. The priority of the request.
Request Key	For future use.
Requestor	Optional. The employee submitting the request.
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.
Start Date	Optional. Start date and time.

Workflow UI Utilities

The Workflow UI Utilities business service is for navigating a user to a specific Siebel view. This business service is supported only when you are running a business service in the object manager.

This business service has one available method, Go To View. This method navigates a user to a view.

NOTE: There can be only one step in a workflow process that calls the Go To View method. When a process contains this method, the method is executed at the end of the process and the process terminates.

Go To View Argument

Table A-6. Submit Request Method Arguments

Argument	Description
View Name	Required. This is the name of the view to which you want to navigate the user.

Workflow Utilities

The Workflow Utilities business service contains generic utilities that can be used in process definitions.

This business service has one available method, Sleep. This method is for pausing a workflow process for a specific amount of time. This method is similar to using a Wait step in a process definition.

Sleep Argument

Table A-7. Submit Request Method Arguments

Argument	Description
Seconds to Sleep	Required. This argument is used for entering the sleep duration in seconds.

Predefined Programs

B

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 B-1](#) contains common actions that you can use by inserting your own message text.

Table B-1. Predefined Programs (1 of 2)

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.

Table B-1. Predefined Programs (2 of 2)

Program	Description
Message Broadcast	
Send Broadcast Message	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.
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.

Siebel Workflow Terminology

C

The following three tables describe the common terms for Workflow Processes, Workflow Policies, and State Model.

Table C-1. Workflow Processes Terms (1 of 2)

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	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 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.

Table C-1. Workflow Processes Terms (2 of 2)

Term	Definition
Process property	A storage field that contains values for use in steps as input and output arguments or for performing evaluations.
Process Simulator	A Workflow Processes view featuring a graphical flowchart interface used for debugging workflow processes.
Siebel Operation	A type of workflow process step that handles database operations such as insert or update of a business component record or field.
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.
Sub-process	A workflow process embedded into another workflow process as part of the workflow process definition. A sub-process has its own workflow process definition. A sub-process is a type of step. There can be one or more sub-process 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 C-2. Workflow Policies Terms (1 of 2)

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 business object is based on one and only one workflow policy 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.
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.

Table C-2. Workflow Policies Terms (2 of 2)

Term	Definition
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.
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.

Table C-3. State Model Terms

Term	Definition
State machine	The engine that enforces the transitions between states for an object during its lifetime. A state represents the status of an object or where the object is in its lifetime, such as Open, Closed, or Pending. The state can also control whether or not the data of that object can be modified. As an example, a service request that is in a Closed state may be considered “frozen,” such that its attributes cannot be modified.
State model	The blueprint of acceptable states and state transitions that the state machine enforces. The state machine then ensures that these objects go through the desired process defined in the state model.
State transition	A transition that defines the allowable migration of an object from one state to the next. For instance, a closed service request that must be re-opened may go from the Closed state to an Open state, and may go from Open to Pending, but may not transition directly from Closed to Pending. The allowable migration of a service request from Closed to Open, or Open to Pending, represents defined state transitions.

Index

Symbols

- .cfg file, about updating for custom .srf file 6-6
- .srf file
 - Objects directory, about adding to 6-6

A

- action parameters
 - defined 2-15
 - See also* arguments
- actions
 - creating for workflow policies 9-15
 - examples of for workflow policies 9-16 to 9-23
 - working with for workflow policies 9-3
- Actions applet
 - description of, Actions view 9-3
 - description of, Policies view 9-26
 - field descriptions, Actions view 9-5
 - field descriptions, Policies view 9-34
- activating processes, procedure 4-79
- activity templates module, described 1-4
- All Processes view
 - reviewing process definitions 4-13
 - tasks 4-6
- arguments
 - action parameters for policies 2-15
 - common values for policy programs 10-38
 - Create E-mail Activity program 12-3
 - creating argument values 10-41
 - creating SQL statements for program 10-44
 - Database Operation program 9-13
 - definition C-1

- Policy Program List Argument view,
 - about 10-36
 - programs, example of creating 10-43
- Run External program 9-11
- Send E-Mail program 9-8
- Send Message Broadcast program 9-10
- Send Page program 9-7
 - storing in process properties 4-18
- Arguments applet
 - description of, Actions view 9-3, 9-5
 - description of, Policies view 9-26
- Assign to Campaign E-mail
 - action 12-6 to 12-8
- Assign to Campaign program 12-3
- Assignment Manager module,
 - described 1-4
- audience for guide Intro-2

B

- Batch feature 9-17 to 9-18
- Batch field for workflow policies 9-35
- batch mode, running workflow processes 6-8
- branch
 - Decision step, defining 4-34
 - defining for Start steps 4-25 to 4-26
 - definition C-1
 - Start step types 4-22
 - using the Next Step applet, Start steps 4-26 to 4-28
- Branches applet, Decision view fields 4-31
- Business Analyst, workflow role
 - described 2-3
- business components, enabling for state model 16-8 to 16-9

- business object
 - defining a primary component 3-4
 - definition C-1, C-3
- business policies
 - creating policy actions 9-15
 - creating policy groups 9-26
 - creating workflow policies 9-35
 - gathering information on 3-2 to 3-3
- business process
 - definition C-1
 - segregating by group 4-7
- business rule, defined C-3
- business service
 - definition C-1
 - enabling for workflow 3-5 to 3-6
 - using in processes 4-37
- Business Service applet
 - field descriptions 4-39
 - using to define a Business Service step 4-42 to 4-43
- Business Service step
 - basic details, defining 4-42 to 4-43
 - input arguments, defining 4-43
 - output arguments, defining 4-44
- Business Service view
 - Business Service applet fields 4-39
 - Input Arguments applet fields 4-40
 - Output Arguments applet fields 4-41
 - tasks, list of 4-39
 - using 4-38
- Business Service, Workflow Policy Action,
 - executing from, note 9-15
- business services, predefined
 - Outbound Communications
 - Manager A-1 to A-3
 - synchronous (or asynchronous) server requests A-4 to A-6
 - synchronous assignment Manager requests A-4
 - Workflow UI Utilities, about and arguments A-6 to A-7
 - Workflow Utilities, about and arguments A-7

C

- calculated fields, updating 4-48
- campaigns
 - Create Email Activity program 12-3
 - marketing campaigns, scenario with
 - Workflow policies 12-4 to 12-8
- columns. *See* Workflow columns
- comparison operations
 - Conditions applet, Decision step 4-32
 - Conditions applet, Start step 4-23
- comparison values
 - entering date calculations 9-34
 - specialized 9-32 to 9-33
 - standard 9-30 to 9-31
 - using in the Conditions applet 9-30
- conditions
 - defining in Decision steps 4-35 to 4-36
 - defining in Exceptions 4-76 to 4-77
 - defining in Start steps 4-26
 - expressions, format of 4-36
- Conditions applet
 - description for Workflow Policies Policies view 9-26
 - field descriptions, Decision view 4-32
 - field descriptions, Policies view 9-29
 - field descriptions, Start view 4-23
 - field names in Siebel Tools 10-24
 - using comparison values 9-30
 - using specialized
 - comparisons 9-32 to 9-33
 - using standard
 - comparisons 9-30 to 9-31
- Configurator workflow role, described 2-3
- connector
 - defining for Decision branches 4-34
 - defining for Exceptions 4-73
 - defining for Start branches 4-25 to 4-26
 - definition of C-1
 - diagraming a process, using for 4-17
 - Exceptions, for 4-75
 - point in connector, removing or adding 4-17

- copying, workflow process or step 4-78
- Create Email Activity
 - about and arguments 12-3
 - creating 12-5
- creating
 - database triggers for workflow policies 11-4
 - workflow policies 9-35
 - workflow policy actions 9-15
 - workflow policy groups 9-26
- CURRENT, using to compare date only 9-34
- D**
- database
 - sample database, launching 2-7
 - Siebel database, described 2-2
 - specialized comparisons 9-33
 - triggers, creating 11-4
 - Workflow Policies database tables 11-19
- Database Operation, Workflow policy program 2-16
- date
 - calculations, entering a value for 9-34
 - comparing date only 9-34
- debugging, workflow processes 5-5 to 5-6
- Decision applet
 - field descriptions 4-30
 - using to define a Decision step 4-33
- Decision Branches view
 - Branches applet fields 4-31
 - Conditions applet fields 4-32
 - Decision applet fields 4-30
 - tasks 4-29
 - using to define workflow process Decision steps 4-33
 - working with 4-29
- Decision steps
 - basic details, defining 4-33
 - branches, defining 4-34
 - conditions, defining 4-35 to 4-36
 - definition of C-1
 - values, defining 4-35 to 4-36
 - working with 4-28
- default SQL statements 10-52
- deleting
 - processes 4-78
 - workflow process steps 4-78
- designated contact, becoming Intro-7
- Designer applet
 - palette items, table of 4-9 to 4-10
 - using to define workflow processes 4-16
- designing, workflow processes. 4-3
- documentation,
 - additional Intro-5 to Intro-6
- E**
- editing. *See* modifying
- email
 - consolidating for recipient 9-17 to 9-18
 - consolidating for recipient, about using batch mode 9-35
 - creating Workflow policy for 9-37 to 9-38
 - testing 9-38
- Email Manager
 - parameters for 11-13 to 11-14
 - setting up 11-8 to 11-13
 - troubleshooting 11-18 to 11-19
- End Arguments view
 - End Step applet fields 4-60
 - Output Arguments applet fields 4-61
 - tasks 4-60
 - using to define workflow process End steps 4-61 to 4-62
- End step
 - about working with 4-59
 - applet field descriptions 4-60
 - defining an end step 4-61 to 4-62
 - definition of C-1
- End User, workflow role, described 2-3
- error code
 - conditions, not using in 4-76
 - defined 4-18
- Error Message
 - conditions, using in 4-76
 - defined 4-18

- Exceptions
 - about working with 4-73
 - defining 4-75
 - definition of C-1
- exporting, process definitions 7-2
- expressions, format of 4-36
- F**
- features, new features in this
 - release Intro-5
- field name modifications 10-24
- Fields applet
 - defining workflow process Siebel
 - Operation steps 4-50
 - field descriptions 4-47
- flowchart tool. *See* Process Designer view
- G**
- Generate Trigger (GenTrig)
 - component-specific parameters for 11-6
 - functions of 11-4
 - tips for running 11-5
- Go To View method, about A-6
- groups
 - creating for workflow policies 9-26
 - defining for workflow processes 4-15
 - planning policy groups 8-2
 - working with for workflow policies 9-23
- H**
- help, contacting technical
 - support Intro-7 to Intro-8
- HTTP Transport business service, invoking
 - from 3-6
- I**
- importing, process definitions 7-2
- input arguments
 - defining for business service steps 4-43
 - Stop steps, defining 4-72
 - storing in process properties 4-18
 - Sub Process step, defining 4-57
 - Wait step, defining 4-66
- Input Arguments applet
 - field descriptions, Business Service
 - view 4-40
 - field descriptions, Stop view 4-69
 - field descriptions, Sub Process
 - view 4-54
 - field descriptions, Wait view 4-65
- Invoke Workflow Process, Workflow policy
 - program 2-16
- L**
- license key validation 11-2
- M**
- marketing campaigns, scenario with
 - Workflow policies 12-4 to 12-8
- Message Broadcast
 - Arguments applet, about and arguments
 - and values 9-9 to 9-10
 - workflow policy program type 9-5
- Method Execute Batch Process, running
 - workflow processes 6-8
- migrating
 - policies, suggested strategies for 8-9
 - production environment, workflow
 - policies to 14-1 to 14-2
 - workflow processes 7-2
 - workflow processes, suggested strategies
 - for 3-6
- modifying
 - policy column and policy field
 - names 10-24
 - Workflow policy object 10-29 to 10-30
- monitoring
 - Workflow Policies
 - processes 15-2 to 15-7
 - Workflow Policy Log view,
 - using 15-3 to 15-4
- multiple conditions, defining for each
 - branch 4-28
- multiple records, executing actions in a
 - workflow processes 6-8
- multi-value groups, updating fields based
 - on 4-48

N

- name, modifying policy column and policy field names 10-24
- Next Step applet
 - defining conditions and values for Start steps 4-26 to 4-28
 - field descriptions, Start view 4-22
 - using for exceptions 4-75

O

- Object Explorer views, defining a workflow policy object 10-23 to 10-26
- Object Id, default process property 4-18
- Object Manager
 - running workflow processes 6-7
 - workflow process, invoking process example 6-4
- object type
 - definition in relation to Siebel Tools 10-3
 - making available in Siebel Client 10-53
 - program types, table of 10-35
 - workflow policies terms, defined C-3
 - workflow policy objects,
 - modifying 10-29 to 10-30
 - workflow policy program, viewing with Program List view 10-34
- object types
 - workflow policy objects, defined C-4
- organization of guide Intro-3
- Outbound Communications Manager
 - available methods, described A-1
 - create and submit request method arguments, table of A-2
 - send message method arguments, table of A-3
 - submit request method arguments, table of A-3
- output arguments
 - Business Service steps, defining 4-44
 - End step, defining 4-62
 - Stop step, defining 4-72
 - storing in process properties 4-18
 - Sub Process step, defining 4-58

Output Arguments applet

- field descriptions, Business Service view 4-41
- field descriptions, End Step view 4-61
- field descriptions, Stop view 4-70
- field descriptions, Sub Process view 4-55

P

Page Manager

- parameters for 11-16 to 11-17
- setting up 11-14 to 11-17
- troubleshooting 11-18 to 11-19
- palette, Designer applet 4-9 to 4-10
- parallel processing, Workflow processes, supported by 4-25
- parameters, passing to the server component that are not listed A-4
- performance
 - tuning tips for Workflow Policies 15-11 to 15-12
 - tuning workflow policies 15-8 to 15-10
 - Workflow Agent processes, creating multiple 15-10
- planning
 - policy scenarios 8-6 to 8-8
 - workflow policies, about 8-3
 - workflow policies, determining what to monitor 8-4
 - workflow policies, planning policies and conditions, about 8-4
 - workflow policy groups, about 8-2
 - workflow processes 3-4
- policies
 - creating 9-35
 - creating actions for 9-15
 - creating groups for 9-26
 - monitor, determining what 8-4
 - planning for 8-3
 - planning policies and conditions, about 8-4
- Policies applet
 - described, Workflow Policies Groups view 9-23

- field descriptions, Groups view 9-25
- field descriptions, Policies view 9-28
- Workflow Policies Groups view, field descriptions 9-25
- Workflow Policies Policies view, described 9-26
- policy action
 - creating 9-15
 - definition of C-3
 - examples of 9-16 to 9-23
 - program types 2-16
 - working with 9-3
- policy condition
 - definition of C-3
 - role in workflow policies 2-14
- Policy Frequency Analysis chart, viewing 11-31
- policy groups
 - creating 9-23
 - planning, about and reasons for using 8-2
- Policy Manager. *See* workflow policies
- Policy Trend Analysis chart, viewing 11-31
- predefined programs
 - Message Broadcast program, screen example and arguments 9-9 to 9-10
 - Run External Program 9-10 to 9-12, 9-21 to 9-23
 - Send Email, using and arguments 9-8 to 9-9
 - Send Quote Page 10-51 to 10-52
 - table of B-1 to B-2
 - workflow action, using a predefined program to create 2-16
 - workflow policy programs 10-44 to 10-52
- primary component, defining for business object 3-4
- process definitions, reviewing 4-13
- Process Designer view
 - defining a process 4-14
 - Designer applet items 4-9 to 4-10
 - diagramming a process 4-15
 - tasks 4-9
 - using 4-8
- Process Properties
 - applet, field descriptions 4-12
 - definition of C-2
 - workflow process, passing field values to example 6-5
 - working with 4-18
- Process Properties view
 - about using and screen example 4-11
 - defining process properties 4-18 to 4-19
 - Process Properties applet fields 4-12
 - tasks, list of 4-12
- Process Property values, using in expressions 4-28
- Process Simulator view
 - button functions 5-4
 - debugging workflow processes 5-5 to 5-6
 - definition of C-2
 - Step Details applet fields 5-4 to 5-5
 - testing Workflow Process, caution 5-5
 - using 5-3
 - workflow processes, using to invoke 5-2
- process steps
 - copying 4-78
 - Decision 4-28
 - deleting 4-78
 - End step, about 4-59
 - Exceptions, about working with 4-73
 - Siebel Operation, about working with 4-45
 - Start step, about working with 4-20
 - Stop 4-67
 - Stop step, about working with 4-67
 - Sub Process step, about working with 4-52
 - testing with Process Simulator, about 5-2
 - Wait 4-63
- production environment, migrating to 14-1 to 14-2

- program
 - See also* Workflow programs
 - definition of C-3
 - predefined, table of B-1 to B-2
 - property fields descriptions, workflow policies 10-35
 - workflow policy action types 2-16
 - workflow policy types 9-5
- Program Argument common properties, table of 10-37
- Program Argument List view, about 10-36
- Program List view, using to list workflow policy programs 10-34
- properties
 - described 10-3
 - Program Argument common, table of 10-37
 - Run External Program Argument 10-41
 - Send Message Argument 10-39 to 10-40
 - workflow policy programs 10-35
- R**
 - recipient types, table of 9-13
 - Recipients applet
 - description of 9-3
 - field descriptions 9-13
 - release, new features Intro-5
 - repository setting, verifying 11-2
 - reviewing process definitions 4-13
 - row id, locating for a object id property 5-5
 - rule structure, described and diagram 2-13
 - Run External
 - described 2-16
 - Program Argument properties, table of 10-41
 - workflow policy program, creating action 9-21 to 9-23
 - workflow policy program. about and example and arguments 9-10 to 9-12
 - Run Workflow Process, about using to define a policy 6-2
- S**
 - S_ESCL_ACTION table 11-19, 11-20
 - S_ESCL_ACTN_REQ table 11-20 to 11-21
 - S_ESCL_ACTN_REQ table, understanding 15-7
 - S_ESCL_LOG table 11-19, 11-20
 - S_ESCL_REQ table 11-19, 11-20
 - S_ESCL_REQ table, understanding 15-5
 - S_ESCL_STATE table 11-19, 11-20
 - S_ESCL_STATE table, understanding 15-6
 - sample database, launching 2-7
 - scripts, invoking workflow processes, examples 6-4 to 6-5
 - Search Specification applet, field descriptions 4-48
 - Send Campaign Email, using 12-2
 - Send Email
 - creating for Workflow policy 9-37 to 9-38
 - using and arguments 9-8 to 9-9
 - Send Message Argument, properties, table of 10-39 to 10-40
 - Send Message Arguments applet 9-8 to 9-9
 - Send Message Broadcast, Workflow policy program, about 2-16
 - Send Message, Workflow policy program, about 2-16
 - Send Page
 - about and arguments 9-6 to 9-7
 - Workflow policy program, about 2-16
 - Send Quote Page, using and SQL statement examples 10-51 to 10-52
 - Send to Relative recipient type, about sending email or page 9-14
 - server component, passing parameters that are not listed A-4
 - Service Request Priority, updating 9-20
 - Siebel Client
 - description 2-2
 - making object types available in 10-53
 - Siebel database, described 2-2

- Siebel eScript, examples of invoking from
 - workflow process 6-4 to 6-5
- Siebel Operation
 - applet, field descriptions 4-46
 - definition C-2
- Siebel Operation Object Id, described 4-18
- Siebel Operation step
 - about working with 4-45
 - defining 4-48 to 4-50
 - fields, defining 4-50
 - navigating through steps 4-50
 - search specifications, defining 4-51
 - workspace, adding to 4-49
- Siebel Operation view
 - Fields applet fields 4-47
 - Search Specification applet fields 4-48
 - Siebel Operation applet fields 4-46
 - using 4-46
- Siebel Server
 - description 2-2
 - setting up for Email
 - Manager 11-8 to 11-13
 - setting up for Page
 - Manager 11-14 to 11-17
- Siebel Server Administration, viewing trace files 11-28
- Siebel Server log directory 11-29 to 11-30
- Siebel Server task trace file
 - created for listed processes 11-27
 - viewed in Siebel Server
 - Administration 11-28
 - viewed in Siebel Server log
 - directory 11-29 to 11-30
- Siebel technical support Intro-7 to Intro-8
- Siebel Tools
 - customizing Workflow Policies 10-3
 - description 2-2
 - five types of programs in 10-33
 - Program List view, using 10-34
 - Validate Tool provided by 10-27 to 10-28
 - Workflow Administration
 - and 10-6 to 10-8
- Siebel VB, examples of invoking from
 - workflow process 6-4 to 6-5
- Siebel Workflow Designer, overview of interface 2-4
- Simulator applet, button functions 5-4
- SmartScript module, described 1-4
- specialized comparisons, descriptions
 - of 9-33
- SQL script file 11-7
- SQL statements
 - created for program arguments 10-44
 - types of 10-52
- Start applet, field descriptions 4-21
- Start step
 - about working with 4-20
 - defining a start step 4-24
 - definition C-2
- Start view
 - Conditions applet fields 4-23
 - Next Step applet fields 4-22
 - tasks 4-21
 - Values applet 4-24
- state machine
 - definition C-4
 - function 2-19
- state model
 - business components, enabling
 - for 16-8 to 16-9
 - creating 16-5 to 16-7
 - definition C-4
 - new state model, creating 16-8
 - overview and screen
 - diagram 16-2 to 16-4
 - understanding 2-19
 - views 2-20
- State Models module, described 1-4
- state transitions
 - creating 16-10 to 16-12
 - definition 2-19, C-4
 - new state transition, creating 16-12
- Step Details applet, field descriptions 5-4 to 5-5
- step icons, table of 4-15

- step instance, defined C-2
 - steps
 - copying 4-78
 - definition C-2
 - deleting 4-78
 - Stop applet. field descriptions 4-68
 - Stop step
 - about working with 4-67
 - defining a stop step 4-71
 - definition C-2
 - input arguments, defining 4-72
 - output arguments, defining 4-72
 - Stop view
 - Input Arguments applet fields 4-69
 - Output Arguments applet fields 4-70
 - Stop applet fields 4-68
 - tasks 4-68
 - storing values 4-18
 - Sub Process applet, field descriptions 4-53
 - Sub Process steps
 - about working with 4-52
 - defining 4-56
 - Sub Process view
 - Input Arguments applet 4-54
 - Output Arguments applet fields 4-55
 - Sub Process applet fields 4-53
 - tasks 4-53
 - using 4-53
 - sub-process, defined C-2
 - synchronous (or asynchronous) server requests
 - about A-4
 - submit request method arguments, table of A-5 to A-6
 - synchronous assignment Manager requests, about and assign method arguments A-4
 - System Administrator 2-3
- T**
- technical support Intro-7 to Intro-8
 - testing
 - creating strategy for 3-6, 8-9
 - workflow process strategy 3-6
 - workflow processes, Process Simulator 5-2
- trace file
- Siebel Server log directory, viewing in 11-29 to 11-30
 - Siebel Server task trace file, created for listed processes 11-27
 - viewing trace files 11-28
- trigger
- database triggers, creating 11-4
 - Generate Trigger (Gen Trig), component-specific parameters for 11-6
 - Generate Trigger (Gen Trig), functions 11-4
 - Generate Trigger (Gen Trig), tips of running 11-5
 - invalid trigger, avoiding in production environment 14-2
- troubleshooting
- Email and Page Manager 11-18 to 11-19
 - notes on 13-5
- U**
- user, workflow role described 2-3
- V**
- Validate Tool, using 10-27 to 10-28
 - values
 - comparison values, using in the Conditions applet 9-30
 - Decision steps, defining 4-35 to 4-36
 - described 10-4
 - Message Broadcast program type 9-10
 - Recipients applet fields 9-13
 - Run External program type 9-11
 - Send E-mail program type 9-8
 - Send Message Broadcast program type, table of 9-10
 - valid database operations program argument 10-38
 - Workflow component properties fields 10-18
 - Values applet, about Start view 4-24

W

- Wait applet, field descriptions 4-64
- Wait steps
 - defining a wait step 4-65 to 4-66
 - definition C-2
 - input arguments, defining 4-66
 - working with 4-63
- Wait view
 - Input Arguments applet fields 4-65
 - tasks 4-64
 - Wait applet fields 4-64
- WF_GROUP_CD, groups for business processes defined 4-7
- wildcards, using in standard comparisons 9-30
- work item, defined C-2
- workflow
 - definition 1-2
 - gathering information on 3-2 to 3-3
 - general functions of 1-2
 - sample database, launching 2-7
 - user roles 2-3
- Workflow Action Agent
 - caution in assigning 15-11
 - functions of 11-25
 - starting 11-19
 - to run processes of 11-25
 - to shut down 11-26
- Workflow Action Arguments
 - applets, described 9-3
 - view, displayed 9-4
- Workflow Actions
 - creating Assign to Campaign E-mail 12-6 to 12-8
 - creating Create Email Activity 12-5
 - creating Database Operation Program type 9-20
 - creating Message Broadcast Program type 9-9 to 9-19
 - creating Run External Program type 9-11
 - creating Send Campaign E-mail 12-4 to 12-5
 - creating Send Message Program type 9-8 to 9-18
 - creating Send Page Program type 9-6 to 9-7
 - using Database Operation program type 9-13
 - using preconfigured program to create 2-16
- Workflow Administration
 - program picklist, screen example 10-8
 - Siebel Tools and 10-6 to 10-8
 - Workflow Component Columns picklist in 10-7
- Workflow Administration views
 - Siebel Tools programs available in 10-8
 - Workflow component columns available in 10-7
- Workflow Administrator, workflow role described 2-3
- Workflow Agent process
 - illustrated 11-21
 - multiple processes, creating 15-10
 - starting automatically 11-26
 - Workflow Groups, starting multiple for 11-27
- Workflow columns
 - adding 10-21
 - adding to Workflow objects 10-24 to 10-25
 - applet fields in 10-14
 - associating Workflow component with 10-26
 - described C-3
- Workflow Columns applet fields, values of 10-14
- Workflow component columns
 - available in Workflow Administration views 10-7
 - described C-4
 - functions of 10-9
 - picklist, diagram 10-7
 - properties fields 10-20

- Workflow Component Columns List view
 - described 10-19 to 10-20
 - function of 10-12, 10-16
- Workflow component properties
 - fields 10-18
- Workflow components
 - associating with Workflow
 - column 10-26
 - defining 10-22
 - described 10-9, C-3
 - relationship between 10-10 to 10-11
 - relationship between component and primary 10-30 to 10-32
 - relationship between Workflow object and 10-11 to 10-18
- Workflow Components List view
 - displayed 10-17
 - function of 10-12
 - relationships displayed in 10-11
- Workflow condition, defined C-3
- Workflow Conditions applet
 - using specialized comparisons in 9-32
 - using standard comparisons in 9-31
- workflow groups
 - creating 8-2
 - defining a workflow policy group 12-6
 - description 2-17
 - Workflow Agent processes, starting
 - multiple 11-27
 - workflow policy group, described C-4
- Workflow Groups applet
 - about using 9-23
 - field descriptions 9-25
- Workflow Groups view, function of 9-2
- Workflow Monitor Agent
 - caution in assigning 15-11
 - functions of 11-20 to 11-21
 - parameters of 11-22 to 11-24
 - to run component of 11-21
- Workflow Object picklist (Workflow Administration) 10-6
- Workflow object properties fields 10-16
- Workflow Object view 10-25
- Workflow objects
 - adding columns to 10-24 to 10-25
 - adding new 10-23
 - defined using Object Explorer
 - views 10-23 to 10-26
 - described C-4
 - displayed in Workflow Objects List view 10-15
 - functions of 10-9
 - in Siebel Tools 10-6
 - relationship between Workflow components and 10-11 to 10-18
- Workflow Objects List view
 - displayed 10-15
 - function of 10-12
- workflow policies
 - See also* performance
 - conditions, table of specialized comparisons 9-33
 - created for Send Email 9-37 to 9-38
 - creating 9-26, 9-35
 - definition 2-14
 - described C-3
 - invoking a process 6-2 to 6-3
 - key parts 2-15
 - monitoring processes 15-2 to 15-7
 - monitoring, about 15-2
 - moving from one group to another, note 9-29
 - performance tuning tips 15-11 to 15-12
 - planning policies and conditions, about 8-4
 - planning, about 8-3
 - planning, determining what to monitor 8-4
 - production environment, migrating to 14-1 to 14-2
 - program types 9-5
 - Siebel Operation steps, using different object layers, described 4-49
 - structure 2-13
 - testing 3-6, 8-9, 13-3 to 13-5

- using specialized
 - comparisons 9-32 to 9-33
- using standard
 - comparisons 9-30 to 9-31
- views, list of 9-2
- Workflow groups, organized into 2-17
- Workflow Policies Actions view
 - applet, field descriptions 9-5
 - applets, description of 9-3
 - Message Broadcast Arguments applet 9-9 to 9-10
 - Recipients applet 9-13
- Workflow Policies Arguments applet using 9-5
- Workflow Policies Explorer view
 - function of 9-2
- Workflow Policies Groups view
 - Policies applet fields, table of 9-25
 - Policies applet, described 9-23
 - Workflow Groups applet 9-23
- Workflow Policies Log view
 - function of 9-2
- Workflow Policies module
 - architectural components 2-2
 - conditions of 2-14 to 2-15
 - described 1-4
 - key parts of, diagram 2-15
 - overview 2-12
 - views, administered by and example 2-17
- Workflow Policies Policies view
 - Actions applet field, table of 9-34
 - Column Instances in picklist names 10-24
 - Policies applet 9-26
 - working with 9-26
- Workflow Policies Policy Frequency Analysis
 - functions of 9-2
- Workflow Policies Policy Trend Analysis, function of 9-2
- Workflow Policies Report page, about and screen example 11-31 to 11-32
- Workflow Policies view, function of 9-2
- Workflow Policy Action Agent, about
 - assigning to workflow policy groups 8-2
- workflow policy actions
 - creating 9-15
 - described 2-15
 - examples 9-16 to 9-23
 - working with 9-3
- Workflow Policy Agent, about assigning to workflow policy groups 8-2
- Workflow Policy applet fields 9-28
- workflow policy column, defined C-3
- workflow policy component column, defined C-4
- workflow policy component columns, Conditions applet 10-24
- workflow policy components, defined C-3
- workflow policy groups
 - creating 9-26
 - definition C-4
 - planning 8-2
- Workflow Policy Log view, using Workflow Policies log view to monitor 15-3 to 15-4
- Workflow Policy Monitor Agent, about
 - assigning to workflow policy groups 8-2
- workflow policy object
 - defined C-4
 - modifying 10-29 to 10-30
- workflow policy program
 - property field descriptions 10-35
 - viewing with Program List view 10-34
- Workflow Process applet, field
 - descriptions 4-6
- workflow process instance, defined C-2
- Workflow Process Manager
 - verifying process 6-6
- workflow processes, about running in batch mode 6-8

- Workflow Process Manager Batch, about
 - running workflow processes in batch mode 6-8
- workflow processes
 - activating 4-79
 - copying 4-78
 - deactivating 4-79
 - definition C-2
 - deleting 4-78
 - designing 4-3
 - enabling business services 3-5 to 3-6
 - exporting\importing 7-2
 - gathering information for 3-2
 - Go To View method, about A-6
 - groups, defining 4-15
 - invoking from a script,
 - examples 6-4 to 6-5
 - invoking from a workflow
 - policy 6-2 to 6-3
 - new process, defining 4-14 to 4-15
 - overview of module 2-5
 - planning tips 3-4
 - requirements 3-3
 - running in batch mode 6-8
 - running on Object Manager 6-7
 - running on the server 6-6
 - testing caution 5-5
 - views, description of 4-3 to 4-4
- Workflow Processes module,
 - described 1-3
- Workflow programs
 - See also* program
 - Assign to Campaign 12-3
 - configuring predefined 10-44 to 10-52
 - Create Email Activity 12-3
 - creating or modifying 10-41
 - described 10-33, C-3
 - five types of 10-33
 - Send Campaign Email, using 12-2
- workflow role, described 2-3
- Workflow UI Utilities, about and
 - arguments A-6 to A-7
- Workflow Utilities, about and
 - arguments A-7

