



**SIEBEL BUSINESS PROCESS  
DESIGNER ADMINISTRATION  
GUIDE FOR FINANCIAL  
SERVICES**

*VERSION 7.5, REV. A*

12-EDXIOQ

*JUNE 2003*

Siebel Systems, Inc., 2207 Bridgepointe Parkway, San Mateo, CA 94404  
Copyright © 2003 Siebel Systems, Inc.  
All rights reserved.  
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.

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.

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

# Contents

## Introduction

How This Guide Is Organized . . . . .	17
Considerations for Upgrading . . . . .	19
Revision History . . . . .	21
Version 7.5, Rev. A . . . . .	21

## Chapter 1. Overview of Workflow

General Principles of Workflow . . . . .	26
Siebel Workflow Solutions . . . . .	28
Siebel Business Process Designer . . . . .	28
Activity Templates . . . . .	29
Siebel Assignment Manager . . . . .	29
Siebel SmartScript . . . . .	29

## Chapter 2. Introduction to Siebel Workflow

Siebel Business Process Designer Architecture . . . . .	32
Siebel Workflow Roles . . . . .	33
Siebel Business Process Designer . . . . .	34
Understanding Workflow Processes . . . . .	35
Workflow Processes Administration Overview . . . . .	35
Understanding Workflow Policies . . . . .	39
Workflow Policies Structure . . . . .	39
Workflow Policies Administration Overview . . . . .	42

Understanding State Models . . . . .	43
State Model Administration Overview . . . . .	43

### **Chapter 3. Planning for Workflow Processes**

Gathering Information . . . . .	46
Researching Existing Process Information . . . . .	46
Researching New Processes and Areas for Improvement . . . . .	47
Understanding Workflow Process Requirements . . . . .	48
Planning for Workflow Processes . . . . .	49
Defining a Primary Component for a Business Object . . . . .	49
Enabling a Business Service for Workflow Processes . . . . .	50
Defining a Test and Migration Strategy . . . . .	51
Verifying Workflow Policies Installation . . . . .	52
Verifying the Repository Setting . . . . .	52
Verifying the Workflow Setup . . . . .	52

### **Chapter 4. Designing Workflow Processes**

Using the Workflow Processes Views . . . . .	54
Using the All Processes View . . . . .	55
Using the Process Designer View . . . . .	56
Using the Process Properties View . . . . .	59
Working with Workflow Process Definitions . . . . .	62
Reviewing Existing Process Definitions . . . . .	62
Defining a New Process . . . . .	63
Modifying Existing Process Definitions . . . . .	64
Designing a Workflow Process . . . . .	65
Working with Process Properties . . . . .	68
Process Properties Versus Property Sets . . . . .	69
Defining Process Properties . . . . .	69
Concatenating Process Properties . . . . .	71

Working with Start Steps . . . . .	72
Using the Start View . . . . .	72
Defining a Start Step . . . . .	78
Working with Decision Steps . . . . .	83
Using the Decision Branches View . . . . .	83
Defining a Decision Step . . . . .	89
Working with Business Service Steps . . . . .	94
Using the Business Service View . . . . .	94
Defining a Business Service Step . . . . .	98
Working with Subprocess Steps . . . . .	102
Using the Subprocess View . . . . .	102
Defining a Subprocess Step . . . . .	106
Working with Siebel Operation Steps . . . . .	109
Defining a Siebel Operation Step . . . . .	112
Working with Wait Steps . . . . .	118
Defining a Wait Step . . . . .	123
Working with a User Interact Step . . . . .	126
Using the User Interact View . . . . .	127
Defining a User Interact Step . . . . .	134
Working with Stop Steps . . . . .	139
Using the Stop View . . . . .	140
Defining a Stop Step . . . . .	142
Working with End Steps . . . . .	145
Defining an End Step . . . . .	147
Working with Error Processes . . . . .	149
Passing User-Defined Process Properties and Property Sets to Error Processes . .	150
Assigning Error Processes to Subprocesses . . . . .	150
Working with Exceptions . . . . .	152
Using the Next Step Applet for Exceptions . . . . .	152
Defining an Exception . . . . .	153

Working with Expressions . . . . .	156
Using the Timestamp Argument . . . . .	156
Deleting a Step or Process . . . . .	157
Copying a Process . . . . .	157
Activating a Process . . . . .	158
Revising a Process Definition . . . . .	159
Working with Seeded Workflow Processes . . . . .	159
Configuring Workflows in a Multilingual Environment . . . . .	160
Defining Expressions for Workflows Running in a Multilingual Environment . . . . .	160
Wait Steps and Global Time Calculations in Workflow . . . . .	161
Using Workflow Persistence . . . . .	162
Enabling Workflow Persistence . . . . .	162
Usage Guidelines for Workflow Persistence . . . . .	163
Wait Steps and Workflow Persistence . . . . .	165
User Interact Steps and Workflow Persistence . . . . .	165
Using Runtime Events . . . . .	166
Creating Buttons that Invoke Workflow Processes . . . . .	167
Choosing Between Using Runtime Events and Using Workflow Policies . . . . .	169

## **Chapter 5. Testing Workflow Processes**

Testing Processes with the Process Simulator . . . . .	172
Using the Process Simulator View . . . . .	173

## **Chapter 6. Running Workflow Processes**

Invoking a Workflow Process . . . . .	178
Invoking From a Workflow Policy . . . . .	178
Invoking From a Script . . . . .	180
Invoking From a Runtime Event . . . . .	182
Defining a Configured Business Service . . . . .	182

Workflow Processing Modes . . . . .	184
Running a Workflow Process Within the Workflow Object Manager . . . . .	184
Running a Workflow Process On the Application Object Manager . . . . .	185
Running a Workflow Process in Batch . . . . .	185

## **Chapter 7. Migrating to the Production Environment**

Migrating Workflow Processes to Production . . . . .	188
Importing or Exporting a Process Definition . . . . .	188
Exporting Multiple Process Definitions . . . . .	189

## **Chapter 8. Administering Workflow Processes**

Monitoring Processes . . . . .	192
Correcting Processes . . . . .	192
Stopping Processes . . . . .	193
Deleting Workflow Process Instances . . . . .	194
Tracing and Event Log Levels . . . . .	195
Increasing Tracing Levels for Workflow Process Manager and Workflow Process Batch Manager . . . . .	195
Restricting Mobile Client Routing . . . . .	198
Modifying the Replication Field . . . . .	198
Running Workflow Processes on Regional Nodes . . . . .	199

## **Chapter 9. Planning for Workflow Policies**

Planning Workflow Policy Groups . . . . .	202
Planning for Workflow Policies . . . . .	203
Determining What to Monitor . . . . .	204
Planning Policies and Conditions . . . . .	204
Planning Workflow Policy Actions . . . . .	206
Planning Scenarios . . . . .	207
Defining a Test and Migration Strategy . . . . .	211

## **Chapter 10. Creating Workflow Policies**

Using the Workflow Policies Views .....	214
Working with Workflow Policy Actions .....	215
Understanding the Actions Applet .....	216
Understanding the Arguments Applet .....	217
Understanding the Recipients Applet .....	225
Creating a Workflow Policy Action .....	227
Workflow Policy Action Examples .....	228
Creating a Send Page Action .....	228
Creating a Send Email Action with a Repeating Message .....	230
Creating a Send Message Broadcast Action .....	232
Creating a Database Operation Action .....	233
Creating a Run External Program Action .....	234
Working with Workflow Policy Groups .....	237
Understanding the Group Applet .....	237
Understanding the Policies Applet .....	238
Creating a Workflow Policy Group .....	239
Working with Workflow Policies .....	240
Understanding the Policies List Applet .....	241
Understanding the Conditions Applet .....	243
Understanding the Actions Applet .....	248
Creating a Workflow Policy .....	249
Siebel Workflow Policy Examples .....	251
Creating a Send Page Workflow Policy .....	251
Creating a Send Email Workflow Policy .....	252

## **Chapter 11. Workflow Policies and Siebel Marketing**

Using Workflow Policy Programs for Campaign Execution .....	256
Using Send Campaign Email .....	256
Using Create Email Activity .....	257
Using Assign to Campaign .....	257



Understanding a Campaign Workflow Scenario . . . . .	258
--	-----

## **Chapter 12. Customizing Workflow Policies with Siebel Tools**

Understanding Siebel Tools . . . . .	264
Understanding Siebel Tools and Workflow Policies . . . . .	267
Working with Workflow Policy Objects . . . . .	268
Using the Siebel Tools Views . . . . .	271
Understanding the Workflow Policy Column List View . . . . .	272
Configuring a Workflow Condition Based on a Foreign Key . . . . .	273
Understanding the Workflow Policy Object List View . . . . .	274
Understanding the Workflow Policy Component List View . . . . .	274
Understanding the Workflow Policy Component Columns View . . . . .	277
Defining a Workflow Policy Column . . . . .	278
Defining a Workflow Policy Component . . . . .	279
Defining a Workflow Policy Object . . . . .	280
Modifying Policy Column Names . . . . .	281
Adding Policy Columns to a Workflow Policy Object . . . . .	281
Associating a Column with a Workflow Policy Component . . . . .	282
Using the Validate Tool . . . . .	282
Modifying an Existing Workflow Policy Object . . . . .	283
Understanding Workflow Policy Programs . . . . .	287
Using the Program List View . . . . .	288
Using the Workflow Policy Program Argument List View . . . . .	289
Defining a Workflow Policy Program . . . . .	293
Example of Creating a Program Argument . . . . .	295
Creating SQL Statements for Program Arguments . . . . .	296
Using Predefined Workflow Policy Programs . . . . .	297
Using Change SR Close Date to Today . . . . .	297
Using Change SR Owner . . . . .	298
Using Change SR Owner to Manager . . . . .	300

Using Send Quote Page . . . . .	302
Making Object Types Available in Siebel Client . . . . .	303

## **Chapter 13. Workflow Policies Server Administration**

Creating Database Triggers . . . . .	306
Database Triggers and Database Administration . . . . .	307
Running Generate Triggers . . . . .	307
Running the SQL Script File . . . . .	310
Database Triggers and Remote Users . . . . .	311
Setting Up the Siebel Server for Email Manager . . . . .	312
Setting Up the Communications Profile . . . . .	312
Setting Up the Siebel Server for Page Manager . . . . .	316
Troubleshooting the Email and Page Managers . . . . .	319
Working with Workflow Monitor Agent . . . . .	321
Using Workflow Monitor Agent . . . . .	323
Using Workflow Action Agent . . . . .	330
Starting Workflow Agent Processes Automatically with Siebel Server . . . . .	332
Using Workflow Policies and Siebel Server Task Trace Files . . . . .	333
Viewing Trace Files in Siebel Server Administration . . . . .	333
Viewing Trace Files in the Siebel Server Log Directory . . . . .	333
Tracing and Event Log Levels . . . . .	334
Using Workflow Policies Analysis Charts and Reports . . . . .	335
Using the Policy Frequency or Trend Analysis Chart . . . . .	335
Using Workflow Policies Reports . . . . .	335

## **Chapter 14. Testing Workflow Policies**

Testing New Policies and Monitoring the Results . . . . .	338
Troubleshooting . . . . .	340
Workflow Tracing . . . . .	341

## **Chapter 15. Migrating Policies to the Production Environment**

Migrating Workflow Policies . . . . .	344
---------------------------------------	-----

## **Chapter 16. Monitoring and Tuning Performance**

Monitoring Workflow Policies . . . . .	348
Using the Workflow Policy Log View . . . . .	349
Understanding the S_ESCL_REQ Table . . . . .	350
Understanding the S_ESCL_STATE Table . . . . .	351
Understanding the S_ESCL_ACTN_REQ Table . . . . .	352
Tuning Workflow Policies for Performance . . . . .	353
Workflow Policy Groups to Manage Server Load . . . . .	353
Multiple Workflow Monitor Agents and Workflow Action Agents . . . . .	353
Workflow Agents on Multiple Application Servers . . . . .	353
Running Workflow Agent Processes through Server Manager . . . . .	354
Optimal Sleep Interval for Workflow Policy Groups . . . . .	354
Optimal Action Interval for Each Workflow Action Agent . . . . .	355
Creating Multiple Workflow Agent Processes . . . . .	355
Performance Tuning Tips . . . . .	356

## **Chapter 17. Using State Models**

State Models Overview . . . . .	360
Creating State Models and State Transitions . . . . .	362
Creating State Models . . . . .	362
Creating State Transitions . . . . .	365
Restricting Transitions Between States . . . . .	368
Enabling Child Modification . . . . .	369
Using a State Model . . . . .	370
Understanding the State Model Cache . . . . .	371

## **Appendix A. Predefined Business Services**

Predefined Business Services	374
Financial Services Workflow User Interface Utilities	379
Data Transfer Utilities	382
Considerations for Data Transfer Utilities	388
Configuring Event-based Commands	390
SmartScript Integration with Workflow	392
Dynamic UI Service	395
Creating a Dynamic Applet	396
Creating a View Template for Dynamic Applets	398

## **Appendix B. Predefined Programs**

Predefined Programs	402
---------------------	-----

## **Appendix C. Siebel Workflow Terminology**

Siebel Workflow Terminology	406
-----------------------------	-----

## **Appendix D. Seeded Workflow Processes**

Seeded Workflows Provided with Siebel Applications	412
--	-----

## **Appendix E. Performance Tuning**

Tuning of Workflow Processes	430
Minimizing Usage of SearchSpec	430
Conditions Based on Parent and Child Business Components	431
Configuration Versus Workflow Comparison	431
Overheads	431
Tuning of Workflow Process Manager	433
Caching Business Services	433
MaxMTServers and MinMTServers	433
MaxTasks	434

Other Critical Parameters . . . . . 435

**Appendix F. Passing Parameters to and from Workflow and  
Data Manipulation within Workflows**

Manipulating Data Within Workflows . . . . . 438

    Accessing Runtime Event Parameters from Workflows . . . . . 438

Passing Parameters from a Business Service to a Workflow . . . . . 442

    Example Scripts for Passing Parameters . . . . . 442

Passing Parameters from Workflow to Global Variables (Profile Attributes) . .  
445

**Index 447**



# Introduction

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:

<b>Siebel Application Administrators</b>	<p>Persons responsible for planning, setting up, and maintaining Siebel applications.</p> <p>For Siebel Workflow, the Siebel Application Administrator is the department, service, or sales manager responsible for defining and changing business processes.</p>
<b>Siebel Application Developers</b>	<p>Persons who plan, implement, and configure Siebel applications, possibly adding new functionality.</p> <p>For Siebel Workflow, the Siebel Application Developer is responsible for configuring Workflow elements using Siebel Tools.</p>
<b>Siebel System Administrators</b>	<p>Persons responsible for the whole system, including installing, maintaining, and upgrading Siebel applications.</p> <p>For Siebel Workflow, the Siebel System Administrator is responsible for installing Siebel Server, managing Siebel Workflow, and creating database triggers.</p>

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.

### **Product Modules and Options**

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



## How This Guide Is Organized

This guide provides information necessary to plan, implement, and monitor Siebel Workflow effectively in a business environment. [Table 1](#) shows a summary of the information contained in each chapter:

**Table 1. Chapter Summary**

Chapter / Appendix	Summary of Contents
Introduction	Provides an overview of the <i>Siebel Business Process Designer Administration Guide</i>
<a href="#">Chapter 1</a>	Provides an overview of how you can use Siebel Workflow to automate the enforcement of your business policies
<a href="#">Chapter 2</a>	Provides an overview of how Siebel Business Process Designer works, including the architecture, concepts, roles, and the relationship between Siebel Tools and Siebel Business Process Designer
<a href="#">Chapter 3</a>	Provides a recommended sequence of tasks to plan for Workflow Processes
<a href="#">Chapter 4</a>	Describes how to use the Siebel Client to create Workflow Processes
<a href="#">Chapter 5</a>	Describes how to test Workflow Processes with the Process Simulator
<a href="#">Chapter 6</a>	Describes the various methods of invoking and running a Workflow Process
<a href="#">Chapter 7</a>	Describes how to migrate Workflow Processes from a test environment to a production environment
<a href="#">Chapter 8</a>	Describes how to administer Workflow Processes
<a href="#">Chapter 9</a>	Provides a recommended sequence of tasks to plan Workflow Policies implementation properly
<a href="#">Chapter 10</a>	Describes how to use the Siebel Client to create Workflow Policies and Workflow Actions
<a href="#">Chapter 12</a>	Describes how to create and modify Workflow Objects and Programs

**Table 1. Chapter Summary**

Chapter / Appendix	Summary of Contents
<a href="#">Chapter 13</a>	Describes how to create database triggers, start email and pager processes, and Workflow Agent processes
<a href="#">Chapter 11</a>	Describes how Workflow Policies and Siebel Marketing work together to automate campaign execution
<a href="#">Chapter 14</a>	Describes how to test your Workflow Policies before implementing them in a production environment
<a href="#">Chapter 15</a>	Describes the migration process from the test to the production environment
<a href="#">Chapter 16</a>	Describes how to monitor Workflow Policies so you can tune your system for optimal performance
<a href="#">Chapter 17</a>	Describes how to use the State Machine to manage state models and state transitions
<a href="#">Appendix A</a>	Provides a list of predefined Siebel Business Services
<a href="#">Appendix B</a>	Provides a list of predefined programs for use with Workflow Policies
<a href="#">Appendix C</a>	Provides terminology related to Siebel Business Process Designer
<a href="#">Appendix D</a>	Provides a list of seeded workflow processes that ship with Siebel eBusiness applications
<a href="#">Appendix E</a>	Describes ways to tune Workflow performance
<a href="#">Appendix F</a>	Describes how to manipulate data within workflows and how to pass parameters to and from Workflow

## Considerations for Upgrading

- The Workflow UI Utilities business service no longer exists. The functionality included in the Go To View method can be executed using the new User Interact step.
- For sending error notifications when the Workflow Monitor Agent or Workflow Action Agent exits with an error condition, a new parameter called Mail Server is required.
- Activity Type is a required field for all activity records. Make sure that all of your existing workflow policy actions and workflow processes that insert activity records include a value for Activity Type.
- If you used the Workflow Administration screen in a previous customization, replace this screen with the Front Office Workflow screen in Siebel Tools. The Workflow Administration screen has been inactivated in version 7.0.
- In the upgrade process, seeded workflows shipped with the latest Siebel release replace the seeded workflows from the previous release. Seeded workflows have inactive status and version 0. They cannot be modified. To use a seeded workflow, you must revise it and activate it. This causes a new copy of the seeded workflow to be created with the latest version number.
- If you want to maintain your customization from the previous Siebel release while incorporating the changes in the new seeded workflows, you must do a manual three-way merge. That is, you must revise and activate the upgraded seeded workflow (version 0) and manually merge it in your customizations.

---

**NOTE:** Nonseeded workflows are migrated as-is, carrying over their respective version number. If schema changes or other improvements have occurred in the new Siebel release, then nonseeded workflows may not work.

---

- For Siebel eBusiness Application Integration (eAI) workflows, there is a consideration regarding the < Value > field. Because of backward compatibility, the < Value > field can be defined as “String” even though it may contain binary data. After Unicode migration, the data in < Value > changes to type Binary, causing the passing of this data to no longer be a valid operation. When upgrading EAI workflows, you must do one of two things:

- Change the data type of the field to Binary, or
- Change the name of the process property from the existing name of the field to < Value > , since < Value > is a special process property that always has the data type Binary.

The following workflow policy components are inactivated in version 7.0. If necessary, you can reactivate these components:

- Indirect Account
- Indirect Account Address
- Indirect Account Industry
- Indirect Account Primary Address
- Indirect Account Synonym
- Opportunity/Indirect Account
- Primary Account Address

In addition, the following Sales Team workflow policy component columns are inactivated in version 7.0. You cannot reactivate these components:

- Forecast Close Date
- Forecast Probability
- Forecast Revenue
- Sales Team Component

# Revision History

*Siebel Business Process Designer Administration Guide for Financial Services,*  
Version 7.5, Rev. A

## Version 7.5, Rev. A

**Table 2. Changes Made in Version 7.5, Rev. A**

Topic	Revision
<a href="#">“Considerations for Upgrading”</a>	Revised this topic.
<a href="#">“Configuring Workflows in a Multilingual Environment”</a>	Added this new topic.
<a href="#">“Defining Conditions and Values for Next Step Branches”</a>	Added a note to this topic.
<a href="#">“Defining Exception Conditions and Values”</a>	Revised a note in this topic.
<a href="#">“Defining Siebel Operation Search Specifications”</a>	Added a caution to this topic.
<a href="#">“User Interact Steps and Workflow Persistence”</a>	Added this new topic.
<a href="#">“Using Runtime Events”</a>	Added new content and revised existing content within this topic.
<a href="#">“Using the Timestamp Argument”</a>	Added this new topic.
<a href="#">“Using the User Interact View”</a>	Added a note to this topic.
<a href="#">“Using the Workflow Processes Views”</a>	Revised this topic.
<a href="#">“Using Workflow Persistence”</a>	Added this new topic.
<a href="#">“Working with Business Service Steps”</a>	Added a note to this topic.
<a href="#">“Working with Decision Steps”</a>	Added new notes to this topic. Revised a table in this topic: Branches Applet Fields.
<a href="#">“Working with End Steps”</a>	Added a note to this topic.
<a href="#">“Working with Expressions”</a>	Added this new topic.
<a href="#">“Working with Start Steps”</a>	Revised a table in this topic: Next Step Branches Applet Fields.

**Table 2. Changes Made in Version 7.5, Rev. A**

Topic	Revision
<a href="#">“Working with a User Interact Step”</a>	Revised a table in this topic: Next Step Branches Applet Fields. Added a note to this topic. Added other new content and revised existing content within this topic.
<a href="#">“Working with Wait Steps”</a>	Revised tables in this topic: <ul style="list-style-type: none"><li>■ Next Step Branches Applet Fields</li><li>■ Wait Applet Fields</li></ul>
<a href="#">“Testing User Interact Workflows”</a>	Added this new topic.
<a href="#">“Invoking a Workflow Process”</a>	Added a note to this topic.
<a href="#">“Increasing Tracing Levels for Workflow Process Manager and Workflow Process Batch Manager”</a>	Added this new topic.
<a href="#">“Creating Database Triggers”</a>	Updated information on the Mode parameter in this table: Component-Specific Parameters for Generate Triggers.
<a href="#">“Setting Up the Siebel Server for Email Manager”</a>	Revised this topic.
<a href="#">“Troubleshooting the Email and Page Managers”</a>	Revised this topic.
<a href="#">“Working with Workflow Monitor Agent”</a>	Updated information in this table: WorkMon Command-Line Interface Parameters. Revised tasks within this topic. Revised note within this topic.
<a href="#">“Creating State Models”</a>	Revised this topic.
<a href="#">“Using a State Model”</a>	Revised a note in this topic.
<a href="#">“Predefined Business Services”</a>	Revised this appendix, removing the tables containing arguments and methods and replacing with a cross-reference to <i>Siebel Communications Server Administration Guide</i> .
<a href="#">“Performance Tuning”</a>	Added this appendix.

**Table 2. Changes Made in Version 7.5, Rev. A**

Topic	Revision
<a href="#">“Passing Parameters to and from Workflow and Data Manipulation within Workflows”</a>	Added this appendix.
<a href="#">“Seeded Workflow Processes”</a>	Added this appendix.

**Additional Changes**

Made other changes throughout the book, including global revisions to wording and naming, minor improvements to content, and minor revisions to book structure.





# Overview of Workflow

# 1

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

## General Principles of Workflow

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

- Allowing that response time objectives are met for customer callbacks and open service requests
- Specifying review policies for important processes like contracts, quotes, or product shipments
- Monitoring service requests or opportunities over time

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

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

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

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

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

- **Discount authority.** If a sales representative quotes a discount exceeding the maximum discount allowed, it requires the approval of the district sales manager or VP of Sales.

- **Pipeline management.** Each sales representative manages his or her pipeline to ensure sufficient levels of prospects at each stage of the sales cycle. If an area of the pipeline needs attention, the representative or manager should be alerted.
- **Forecasting accuracy.** Opportunities that are forecasted but never closed or forecasts having wide discrepancies with the actual revenue need to be flagged.

# 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 set of solutions through various software modules to address your workflow requirements. These modules are:

- Siebel Business Process Designer, referred to as Siebel Workflow
- Activity Templates
- Siebel Assignment Manager
- Siebel SmartScript

## Siebel Business Process Designer

Siebel Business Process Designer, 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, subprocesses, decision points, and tasks.
- **Workflow Policies.** Allows you to define policies that can act as triggers to execute a process. A policy consists of conditions and actions. When policy conditions are met, the policy action executes the relevant process.
- **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 *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 or 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 Administration Guide* for more information.

## Overview of Workflow

*Siebel Workflow Solutions*

# Introduction to Siebel Workflow

# 2

Siebel Workflow is an 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 Business Process Designer, a graphical user interface featuring a familiar flowcharting methodology for designing workflow processes.

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 Business Process Designer Architecture

Siebel Business Process Designer 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 Business Process Designer.
- **Siebel Client.** Workflow Processes, Workflow Policies, and State Models are administered through Business Process Administration 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 Server architecture, refer to *Siebel Server Administration Guide*.



## 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 *Integration Platform Technologies: Siebel eBusiness Application Integration Volume II*.

---

- The *Workflow Administrator* or *Business Analyst* defines or changes workflow processes and policies in the Siebel client using Siebel Business Process 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.

# Siebel Business Process Designer

Siebel Business Process 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.

## Understanding Workflow Processes

Workflow Processes is the module in Siebel Business Process Designer that you use to create workflow processes. It allows you to define your company's business processes using the Siebel Business Process 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, subprocesses, 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.

### Workflow Processes Administration Overview

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

#### Workflow Processes Views

Workflow Processes are administered through the Siebel Business Process Designer on the 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, a runtime event (such as an insert of a record or a button click), or a server component.

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

### Sample Workflow Process Scenarios

To help you understand how Workflow Processes works, see the usage scenarios “[Scenario 1: New Service Request](#)” on page 36 and “[Scenario 2: New Lead](#)” on page 38. You can view the sample 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 the Workflow Processes screen.
- 5 From the Show drop-down list, click All Processes.

#### **Scenario 1: New Service Request**

ABC Computing defines its business process for a new service request with Workflow Processes. 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 subprocess task. The service request is assigned to the appropriate agent based on the assignment rules defined in the Assign Service Request subprocess.
- **Severity.** This is a decision step. The service request 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. The process calls Assignment Manager to assign a lead and contains subprocess steps for High Quality Leads and Low Quality Leads.

The subprocess steps 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 subprocess step that calls the High Quality Lead workflow process.
- **Low Quality Lead.** This is a subprocess 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

The Workflow Policies module allows you to define policies that can act as triggers to execute a workflow process. A policy consists of one or more policy conditions. When the policy conditions are met, the policy action is executed.

---

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

---

---

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

---

### Workflow Policies Structure

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

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

### Workflow Policy Conditions

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

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

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

Policies frequently have more than one condition. All the conditions of the policy must be met (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
```

Siebel Workflow 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 243](#).

## Workflow Policy Actions

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



You can specify several actions for one workflow policy, such as sending a page to one person and an email to another. You can reuse actions in multiple workflow policies. See [Chapter 12, “Customizing Workflow Policies with Siebel Tools,”](#) for a discussion of actions and their parameters.

---

**NOTE:** In most cases, workflow policy actions should be used to run a workflow process.

---

### **Workflow Policy Action Program Types**

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

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

---

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

---

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

## **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 Workflow Agent process performance by grouping similar policies to run under one Workflow Agent process.

## **Workflow Policies Administration Overview**

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

## **Workflow Policies Views**

Workflow Policies is administered through the Siebel Business Process Designer in the Siebel Client. Instructions for accessing and using the Workflow Policies views are in [Chapter 12, “Customizing Workflow Policies with Siebel Tools.”](#)

## 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 Business Process Designer on the Siebel client. Instructions for accessing and using the State Model views are in [Chapter 17, “Using State Models.”](#)



## **Planning for Workflow Processes**

# **3**

This chapter describes the steps for planning a workflow process.

## 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. Usually, each workflow process is associated with a business object.

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

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

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

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



## Planning for Workflow Processes

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

- If your workflow process is associated with a business object, the business object should have a primary component defined in Siebel Tools. For more information, see [“Defining a Primary 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 Reference* 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.

## 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.”](#)

## Verifying Workflow Policies Installation

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

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

## Verifying the Repository Setting

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

## Verifying the Workflow Setup

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

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

### **To verify the workflow setup**

- 1** Log in to the Siebel Client as the Siebel administrator.
- 2** Select View > Site Map > Business Process Administration.
- 3** Under List Views, you should be able to see Workflow Policies groups, policies, and so forth. This indicates that your license key is correct.
- 4** Select View > Site Map > Server Administration > Component Group > Enterprise Component Groups from a client that is configured to manage the server component groups.

Check that the workflow management component group is enabled.

This chapter provides information about using the Workflow Processes views to design workflow processes in Siebel Business Process Designer. 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 Business Process Administration interface. Workflow Processes views include:

- **All Processes view.** For reviewing existing business processes. For more information, see [“Using the All Processes View” on page 55](#).
- **Process Designer view.** For designing the process flow diagram. For more information, see [“Using the Process Designer View” on page 56](#).
- **Process Properties view.** For defining the properties for a process. For more information, see [“Using the Process Properties View” on page 59](#).
- **Process Simulator view.** For testing the functionality of the processes you create. This is similar to a debugging application. For more information, see [“Using the Process Simulator View” on page 173](#).

To access more views, double-click the appropriate step icon in the flowcharted process on the Process Designer view. Views you can access include:

- **Business Service view.** For defining process steps that call a business service. For more information, see [“Working with Business Service Steps” on page 94](#).
- **Decision view.** For defining a decision point step, including decision branches, conditions, and values. For more information, see [“Working with Decision Steps” on page 83](#).
- **End view.** For defining the ending process step. For more information, see [“End Step Applet Field Descriptions” on page 145](#).
- **Siebel Operation view.** For defining an insert or update to a Siebel database record. For more information, see [“Working with Siebel Operation Steps” on page 109](#).
- **Start view.** For identifying the conditions that must be met for the process to be initiated. For more information, see [“Using the Start View” on page 72](#).
- **Stop view.** For defining the input and output arguments that result in a process termination. For more information, see [“Using the Stop View” on page 140](#).
- **Subprocess view.** For defining subprocess steps, including input and output arguments. For more information, see [“Working with Subprocess Steps” on page 102](#).

- **Wait view.** For defining a pause in the workflow process execution. For more information, see [“Working with Wait Steps” on page 118](#).
- **User Interact view.** For defining which view a user is navigated to, and which events to wait for before proceeding to the next step. For more information, see [“Using the User Interact View” on page 127](#).

## Using the All Processes View

This view displays a list of the currently defined workflow processes. The top portion of the view displays a list of all workflow processes. The lower portion is the Workflow Process form.

The tasks you can perform in this view are:

- **Define a new process.** See [“Defining a New Process” on page 63](#).
- **Delete a process.** See [“Deleting a Step or Process” on page 157](#).
- **Revise a process.** See [“Revising a Process Definition” on page 159](#).
- **Copy a process.** See [“Copying a Process” on page 157](#).
- **Activate or deactivate a process.** See [“Activating a Process” on page 158](#).
- **Export or import a process.** See [“Importing or Exporting a Process Definition” on page 188](#).

### Workflow Processes View Field Descriptions

[Table 3](#) describes the fields where you enter data in the Workflow Processes view.

### Using the Process Designer View

**Table 3. Fields in the All Processes View**

Field	Description	Possible Value
Name	The name of the process.	<i>Required.</i> A descriptive name that is: <ul style="list-style-type: none"><li>■ Consistent with your overall naming strategy.</li><li>■ Meaningful to the process designer.</li></ul>
Business Object	The name of the associated business object.	<i>Optional.</i> This value is selected from a picklist of business objects. Only business objects with a defined primary component appear in this picklist.  See <a href="#">“Planning for Workflow Processes” on page 49</a> for more information on defining the primary component for a business object.
Status	The current status of the process.	<i>Read-only.</i> 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.
Group	The name of the associated group for the business process.	<i>Optional.</i> 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. For more information, see <a href="#">“Defining a New Process” on page 63</a> .
Description	A text narrative describing the purpose of the process.	<i>Optional.</i>
Activation Date/Time	The activation start date and time.	<i>Optional.</i> This value is selected from a calendar widget.
Expiration Date/Time	The activation end date and time.	<i>Optional.</i> This value is selected from a calendar widget.
Created By	The login ID of the person who created the process.	<i>Read-only.</i> This value is automatically populated based on the logon name of the user.
Created	The date that the process is created.	<i>Read-only.</i> This value is automatically populated by the system.
Version	The version number of this process definition.	<i>Read-only.</i> The default version is 0. To increment the version, click the Revise button.



**Table 3. Fields in the All Processes View**

Field	Description	Possible Value
Replication	The replication level of the workflow process.	<p><b>All.</b> The workflow process definition is routed to all mobile clients and regional nodes.</p> <p><b>Regional.</b> The workflow process definition is routed to the regional nodes only.</p> <p><b>None.</b> (Default) The workflow process definition is not routed to the mobile client or the regional nodes.</p>
Persistence Frequency	The frequency to save state when this process is executed.	<p><i>Optional.</i></p> <p><b>On Pause.</b> State is stored to the database when the process instance changes status (such as by pausing or ending).</p> <p><b>Every Step.</b> State is stored at the end of every step execution.</p>
Persistence Level	The amount of state to save when this process is executed.	<p><b>Current Step.</b> Store the current step and the current process property values.</p> <p><b>All Steps.</b> Store all step instances as well as the runtime property values at every step instance.</p>
Error Process Name	The name of the error process to run when this process encounters an error.	<i>Optional.</i> This value is selected from a picklist of workflow processes.

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.

The tasks you can perform in this view are:

- **Define a new process.** See [“Defining a New Process” on page 63.](#)
- **Delete a process.** See [“Deleting a Step or Process” on page 157.](#)
- **Copy a process.** See [“Copying a Process” on page 157.](#)
- **Design a process flow.** See [“Designing a Workflow Process” on page 65.](#)

### Process Designer Palette Item Descriptions

Table 4 describes the items available on the Process Designer.

**Table 4. Process Designer Palette Items**

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, or a query.	Business object logic applies to all Siebel operations. A process can have one or more Siebel Operation steps.
Connector	Represents the direction of flow between steps in a business process.	A process can have one or more connectors.
Decision Point	Represents a type of step. A decision point 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 Point steps.
Subprocess	Represents a process embedded into another workflow process. A subprocess has its own process definition.	A process can have one or more subprocess steps.
End	Indicates when process execution is complete. An End step allows you to save output arguments as 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.
Exception	Represents a deviation from normal processing. An exception can be a system error or a user-defined error.	A process can have one or more Exception branches.
User Interact	Represents end user view navigation.	A process can have one or more User Interact steps.

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

From some detail views, you can select the Next Step form menu to view the Next Step form. The Next Step form lists steps that are accessible from the current step through connectors or exceptions. To return to the step detail, select Go Back from the form menu.

## 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.

Use the Process Properties view to:

- **Define a new process.** See [“Defining a New Process” on page 63](#).
- **Delete a process.** See [“Deleting a Step or Process” on page 157](#).
- **Copy a process.** See [“Copying a Process” on page 157](#).
- **Set process properties.** See [“Working with Process Properties” on page 68](#).

### Process Properties List Field Descriptions

[Table 5](#) describes the fields in the Process Properties view.

**Table 5. Process Properties View Fields**

Field	Description	Possible Value
Name	The name of the process property.	Free-form text.
Data Type	The type of data that can be stored in the process property.	<ul style="list-style-type: none"> <li>■ <b>String.</b> For alphanumeric data.</li> <li>■ <b>Date.</b> For dates.</li> <li>■ <b>Number.</b> For numeric data.</li> <li>■ <b>Hierarchy.</b> Data type used by eBusiness Application Integration to store data from a property set. For more information, see <i>Overview: Siebel eBusiness Application Integration Volume I</i>.</li> <li>■ <b>Binary.</b> For variant or binary information. Binary types must be assigned the None value in the In/Out field.</li> <li>■ <b>Integration Object.</b> For variant or binary information.</li> </ul>
Default String	Initial value if the process property is a string type.	Free-form text. If you enter < Value > , the process property is initialized with the value in the Value field of the workflow input property set.
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
Integration Object	Data type used by eBusiness Application Integration to store data from an integration object.	Account - Get Oracle Customer (Oracle).
In/Out	Describes whether or not the process property is passed in or out of the process, passed into the process and returned, or used only within the process.	<ul style="list-style-type: none"> <li>■ <b>In.</b> The process property is passed into the process. (Binary types cannot be assigned this value.)</li> <li>■ <b>Out.</b> The process property is passed out of the process. (Binary types cannot be assigned this value.)</li> <li>■ <b>In/Out.</b> The process property is passed into the process and returned. (Binary types cannot be assigned this value.)</li> <li>■ <b>None.</b> The process property is used only within the process.</li> </ul>

**Table 5. Process Properties View Fields**

Field	Description	Possible Value
Business Component	The name of the business component containing the virtual field.	Optional. This value is selected from a picklist of business components belonging to the workflow process business object.
Virtual Field	The name of the business component field mapped to the workflow process property.	Optional. This value is selected from a picklist of fields belonging to the business component. Only calculated fields with no calculated values appear in this picklist.
Comments	A text narrative describing the purpose of the process property.	Optional

# 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:

- [“Reviewing Existing Process Definitions.”](#)
- [“Defining a New Process” on page 63.](#)
- [“Modifying Existing Process Definitions” on page 64.](#)

## Reviewing Existing Process Definitions

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

The All Processes view provides a list of all the current process definitions. See [“Using the All Processes View” on page 55](#) for information about this view. Use the following procedure to review existing process definitions.

### ***To review process definitions***

- 1** From the Workflow Processes view, run the All Processes query.
- 2** 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, select Copy from the applet menu.

See [“Copying a Process” on page 157](#) 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 63](#) for more information on creating a new process.

## Defining a New Process

To define a new process, you must enter some basic information about the process, such as the process name, the associated business object, and the activation date. You can enter this information in the Workflow Process form in 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 also the view that you use to diagram the process steps. Diagramming process steps is the next part of creating a new process. See [“Using the Process Designer View” on page 56](#) for more information about this view.

### **To define a new process**

- 1** From the Business Process Administration screen, choose the All Processes view.
- 2** Select New Record 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 that have a defined primary business component. See *Siebel Tools Reference* for information about defining business objects. See [“Defining a Primary Component for a Business Object” on page 49](#) 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** From the application-level menu, choose View > Site Map > 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.

## Modifying Existing Process Definitions

You can modify active workflow process definitions without restarting the Workflow Process Manager.

The server parameter called Workflow Version Checking Interval controls how often the server component checks for new active versions of each process definition.

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

You can create several versions of the same process. When you create a new process definition, the version number defaults to 0. Each time you create a new version, the version number is incremented by 1.

To publish a version of a process definition, click Activate. This makes the version read-only, changes the status of the version to Active, and publishes the version. If there was a previous active version of that process definition, it changes the status of the previous version to Outdated.

To make changes to an active version, click Revise. This creates a copy of the process definition. The copy will have a new version number and a status of In Progress. For example, you have an active process called Service Request Escalation Process, Version 0. You can click Revise to create another version of this process. The second version will be called Service Request Escalation Process, Version 1.



## 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 access and fill in step details by double-clicking a step icon.

---

**NOTE:** See *Siebel SmartScript Administration 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 6](#) lists the palette icons and the corresponding detail view.

**Table 6. Step Icons and Detail Views**

Step Icons	Detail View
Start	Start
Business Service	Business Service Arguments
Siebel Operation	Siebel Operation Details
Decision	Decision Branches
Subprocess	Subprocess Arguments
Wait	Wait
End	End Arguments
Stop	Stop
Connector	Detail view of originating step

**Table 6. Step Icons and Detail Views**

Step Icons	Detail View
Exception	Detail view of originating step
User Interact	User Interact

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

**To diagram the process steps**

- 1 From the Business Process Administration screen, 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 72](#).

- 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, Subprocesses, Stop, Wait, Exceptions, and Siebel Operation. There can be multiples of each type of step. For details on each type of step, see:

- [“Defining a Decision Step” on page 89](#)
- [“Defining a Business Service Step” on page 98](#)
- [“Output Arguments Field Descriptions” on page 112](#)
- [“Working with Wait Steps” on page 118](#)
- [“Defining a Subprocess Step” on page 106](#)
- [“Defining a Wait Step” on page 123](#)
- [“Defining a Stop Step” on page 142](#)
- [“Defining an Exception” on page 153](#)

- [“Defining a User Interact Step” on page 134](#)
- 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 147](#).
- 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 89](#) and [“Defining Conditions and Values” on page 90](#) 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.

## Working with Process Properties

Process Properties are fields for storing values that you can use in steps, either as input and output arguments, or for performing evaluations.

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

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

## Process Properties Versus Property Sets

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

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

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

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

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

## Defining Process Properties

You define process properties in the Process Properties view.

### **To define a process property**

- 1 From the Business Process Administration screen, 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 (property set)
- **Binary.** If property holds a binary value
- **Integration Object.** If property holds an integration object

---

**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 on page 70](#) through [Step 6](#) to define additional properties.

## Concatenating Process Properties

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

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

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

### ***To concatenate process properties***

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

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

## Working with Start Steps

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

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

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

## Using the Start View

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

The tasks you can perform in this view are:

- **Define a start step.** See [“Defining a Start Step” on page 78](#).
- **Delete an existing start step.** See [“Deleting a Step or Process” on page 157](#).
- **Copy a start step.** See [“Copying a Process” on page 157](#).
- **Define next step branches.** See [“Defining Decision Branches” on page 89](#).
- **Define branch conditions and values.** See [“Defining Conditions and Values” on page 90](#).



### Start Applet Field Descriptions

Table 7 describes the fields where you enter data in the Start applet.

**Table 7. Start Applet Fields**

Field	Description	Possible Value
Name	The name of the Start step.	A descriptive name that is: <ul style="list-style-type: none"> <li>■ Consistent with your overall naming strategy</li> <li>■ Meaningful to the process designer</li> </ul>
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.
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.
Processing Mode	The mode in which the process will be run when triggered by runtime events.	Optional <ul style="list-style-type: none"> <li>■ <b>Local Synchronous.</b> Executes the process in the application object manager. This is the default.</li> <li>■ <b>Remote Synchronous.</b> Submits a synchronous request to the Workflow Process Manager server component to execute the process.</li> <li>■ <b>Remote Asynchronous.</b> Submits an asynchronous request to the Workflow Process Manager server component to execute the process.</li> </ul>

#### Next Step Branches Applet Field Descriptions

[Table 8](#) describes the fields in the Next Step Branches applet.

**Table 8. 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.	<p>The value can be one of the following choices:</p> <ul style="list-style-type: none"> <li>■ <b>Condition.</b> This value indicates that a condition is defined for the branch.</li> <li>■ <b>Default.</b> 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.</li> <li>■ <b>Connector.</b> Use this value when there is no condition branching involved.</li> <li>■ <b>Error Exception.</b> Use this value to define exception handling. This connector type captures system errors, such as an error noting that the Assignment Manager server component is not available. For more information, see <a href="#">“Working with Exceptions” on page 152</a>.</li> <li>■ <b>User Defined Exception.</b> Use this value to define exception handling. This connector type captures user-defined errors, such as an error noting that an order being submitted is incomplete. For more information, see <a href="#">“Working with Exceptions” on page 152</a>.</li> </ul>
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.

**Table 8. Next Step Branches Applet Fields**

Field	Description	Possible Value
Event Object Type	The type of the object the event refers to. This is used when defining a runtime event.	Optional <ul style="list-style-type: none"> <li>■ Application.</li> <li>■ Applet.</li> <li>■ BusComp.</li> </ul>
Event	The specific event the object supports.	Required if Event Object Type is specified. The set of events is different for different object types.
Event Object	The name of the object.	Required if Event Object Type is specified. This is the name as defined in Tools. The set of objects is different for different object types.
Sub Event	An options parameter for the event.	Optional <p>For InvokeMethod, the name of the method being invoked. For SetFieldValue, the name of the field being set.</p>
Event Cancel Flag	Abort the runtime event after executing the process.	Optional <p>This flag only applies to events that are cancelable. This flag functions like CancelOperation in scripting.</p>

### Conditions Applet Field Descriptions

[Table 9](#) describes the fields in the Conditions applet.

**Table 9. Conditions Applet Fields**

Field	Description	Possible Value
Compare To	Indicates where the comparison value is coming from.	This is a required field, with the following choices: <ul style="list-style-type: none"> <li>■ Business Component</li> <li>■ Process Property</li> <li>■ Expression</li> <li>■ Applet</li> </ul>

**Table 9. Conditions Applet Fields**

Field	Description	Possible Value
Operation	Identifies the comparison operation.	<ul style="list-style-type: none"> <li>■ <b>This Must Match.</b> The current value must match exactly, including case.</li> <li>■ <b>One Must Match.</b> One or more values must match exactly, including case.</li> <li>■ <b>All Must Match.</b> All of the values must match exactly, including case.</li> <li>■ <b>None Can Match.</b> None of the values can match exactly, including case.</li> <li>■ <b>This Must Match (ignore case).</b> The current value must match without regard to case.</li> <li>■ <b>One Must Match (ignore case).</b> One or more values must match without regard to case.</li> <li>■ <b>All Must Match (ignore case).</b> All of the values must match without regard to case.</li> <li>■ <b>None Can Match (ignore case).</b> None of the values can match without regard to case.</li> <li>■ <b>Greater Than.</b> Value must be greater than the comparison value.</li> <li>■ <b>Less Than.</b> Value must be less than the comparison value.</li> <li>■ <b>Between.</b> Value must be between a range of values.</li> <li>■ <b>Not Between.</b> Value cannot be between a range of values.</li> <li>■ <b>Is Null.</b> Value must be null.</li> <li>■ <b>Is Not Null.</b> Value cannot be null.</li> </ul>
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.
Business Component	Identifies the specific business component on which to base the condition.	<p>This is a required field when Business Component or Expression is the Compare To value.</p> <p>The picklist includes business components defined for the business object selected for the process.</p>

**Table 9. Conditions Applet Fields**

Field	Description	Possible 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.  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

You define a Start step in the All Processes view.

#### **To define a Start step**

- 1 In the All Processes view, select the process for which you would like to define a Start step.
- 2 Drill down on the process by clicking its name.

This brings you to the Process Designer view.

- 3 To add a Start step to the workspace, drag and drop the Start icon from the palette to the workspace.
- 4 Enter or modify the step name, and then enter a description of the purpose of the Start step.
- 5 See [“Defining Next Step Branches”](#) for instructions on setting up branches for the Start step.

### 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, Workflow will take the first condition that passes and execute it. Workflow will assume that the other conditions fail.

---

#### **To define a next step branch**

- 1 In the Process Designer 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 access 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 Choose a branch type. Type values are described in [Table 11 on page 85](#).

---

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

---

- 6 Enter the name of the next step and enter any comments.
- 7 See “[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. For example, you can define a condition based on the value of a priority field. If the priority is “high,” the process follows a branch that sends an email to a vice president. If the priority is “medium,” the email is sent to an engineer.

### To define conditions and values

- 1 Select the appropriate branch in the Next Step form in the Start view.
- 2 Click the Conditions form to activate the branch.
- 3 Choose a Compare To value:
  - **Process Property.** Compares a process instance’s process property value to a specified value.
  - **Expression.** Uses an expression to evaluate a specific value.

- **Business Component.** Uses the value in a business component field for the condition comparison or when you are defining an expression.
  - **Applet.** Uses the value in an applet field for the condition comparison.
- 4** If you chose Process Property, select the process property name from the Process Property picklist. If you chose Business Component or Expression, select the component and field names from the Business Component Name and the Business Component Field picklists.



5 Choose the operation to use for evaluating the values:

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

---

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

---

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

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

**6** Enter an applet name and applet field, if applicable.

Applets function like business components, except that applets have translatable names and picklists. Only list applets can be used in conditions.

**7** Enter any appropriate values in the Values form.

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

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

---

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

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

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

---

## 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

### 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. The lower portion has three applets: Branches, Conditions, and Values.

The tasks you can perform in this view are:

- **Define a decision step.** See [“Defining a Decision Step” on page 89](#).
- **Delete a decision step.** See [“Deleting a Step or Process” on page 157](#).
- **Copy a decision step.** See [“Copying a Process” on page 157](#).
- **Define branches.** See [“Defining Decision Branches” on page 89](#).
- **Define conditions and values.** See [“Defining Conditions and Values” on page 90](#).

#### Decision Applet Field Descriptions

[Table 10](#) describes the fields where you enter data in the Decision applet.

**Table 10. 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"><li>■ Consistent with your overall naming strategy</li><li>■ Meaningful to the process designer</li></ul>
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 11](#) describes the fields in the Branches applet.

**Table 11. 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.	<p>The value can be one of two choices:</p> <ul style="list-style-type: none"> <li>■ <b>Default.</b> This value indicates that if all conditional branches fail, this branch will be followed.</li> <li>■ <b>Condition.</b> Define zero or more conditions. If no conditions are defined, it means the branch always passes.</li> <li>■ <b>Connector.</b> Use this value whenever there is no condition branching involved.</li> <li>■ <b>Error Exception.</b> Use this value to define exception handling. This connector type captures system errors, such as an error noting that the Assignment Manager server component is not available. For more information, see <a href="#">“Working with Exceptions” on page 152</a>.</li> <li>■ <b>User Defined Exception.</b> Use this value to define exception handling. This connector type captures user-defined errors, such as an error noting that an order being submitted is incomplete. For more information, see <a href="#">“Working with Exceptions” on page 152</a>.</li> </ul>
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 12 describes the fields in the Conditions applet.

Table 12. Conditions Applet Fields

Field	Description	Possible Value
Compare To	Indicates where the comparison value is coming from.	<div>This is a required field, with the following choices:</div> <ul style="list-style-type: none"><li>■ Business Component</li><li>■ Process Property</li><li>■ Expression</li><li>■ Applet</li></ul>

**Table 12. Conditions Applet Fields**

Field	Description	Possible Value
Operation	Identifies the comparison operation	<ul style="list-style-type: none"> <li>■ <b>This Must Match.</b> The current value must match exactly, including case.</li> <li>■ <b>One Must Match.</b> One or more values must match exactly, including case.</li> <li>■ <b>All Must Match.</b> All of the values must match exactly, including case.</li> <li>■ <b>None Can Match.</b> None of the values can match exactly, including case.</li> <li>■ <b>This Must Match (ignore case).</b> The current value must match without regard to case.</li> <li>■ <b>One Must Match (ignore case).</b> One or more values must match without regard to case.</li> <li>■ <b>All Must Match (ignore case).</b> All of the values must match without regard to case.</li> <li>■ <b>None Can Match (ignore case).</b> None of the values can match without regard to case.</li> <li>■ <b>Greater Than.</b> Value must be greater than the comparison value.</li> <li>■ <b>Less Than.</b> Value must be less than the comparison value.</li> <li>■ <b>Between.</b> Value must be between a range of values.</li> <li>■ <b>Not Between.</b> Value cannot be between a range of values.</li> <li>■ <b>Is Null.</b> Value must be null.</li> <li>■ <b>Is Not Null.</b> Value cannot be null.</li> </ul>
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.
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.

**Table 12. Conditions Applet Fields**

Field	Description	Possible 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 Point icon to access the details view.

---

**NOTE:** To add a Decision step to the workspace, drag and drop the Decision Point 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.

---

## 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 65](#), 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 90](#) 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 Choose a branch type. Type values are described in [Table 11 on page 85](#).

---

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

---

- 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 on page 89](#) 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.

---

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

---

#### **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:
  - **This Must Match.** The current value must match exactly, including case.
  - **One Must Match.** One or more values must match exactly, including case.

- **All Must Match.** All of the values must match exactly, including case.

---

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

---

- **None Can Match.** None of the values can match exactly, including case.
  - **This Must Match (ignore case).** The current value must match without regard to case.
  - **One Must Match (ignore case).** One or more values must match without regard to case.
  - **All Must Match (ignore case).** All of the values must match without regard to case.
  - **None Can Match (ignore case).** None of the values can match without regard to case.
  - **Greater Than.** Value must be greater than the comparison value.
  - **Less Than.** Value must be less than the comparison value.
  - **Between.** Value must be between a range of values.
  - **Not Between.** Value cannot be between a range of values.
  - **Is Null.** Value must be null.
  - **Is Not Null.** Value cannot be null.
- 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. See [Table 8 on page 74](#).
- 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 Reference*.

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.** Assignment Manager can assign an object in a workflow process by calling the Synchronous Assignment Manager Request business service.
- **Server tasks.** You can run a server component task using the Asynchronous or Synchronous Server Requests business service.

For a list of some of the most commonly-used predefined business services, see [Appendix A, “Predefined Business Services.”](#)

You can use Siebel VB or Siebel eScript to define your own custom business services that you can invoke from a workflow process. You can define business services by navigating to Site Map > Business Service Administration, 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

## 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. The lower portion has two applets: Input Arguments and Output Arguments.

The tasks you can perform in this view are:

- **Define a business service step.** See [“Defining a Business Service Step” on page 98.](#)
- **Delete an existing business service step.** See [“Deleting a Step or Process” on page 157.](#)
- **Copy a business service step.** See [“Copying a Process” on page 157.](#)

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

### Business Service Applet Field Descriptions

[Table 13](#) describes the fields where you enter data in the Business Service applet.

**Table 13. 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"> <li>■ Consistent with your overall naming strategy</li> <li>■ Meaningful to the process designer</li> </ul>
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 <a href="#">“Enabling a Business Service for Workflow Processes” on page 50</a> 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.

#### 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 14](#) describes the fields in the Input Arguments applet.

**Table 14. Input Arguments Applet Fields**

Field	Description	Possible Value
Input Argument	The name of the input argument.	<p>This field is required. The picklist displays input arguments existing for the selected business service method.</p> <p>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.</p>
Type	The type of argument.	<p>This is a required field. The picklist contains the following choices:</p> <ul style="list-style-type: none"><li>■ Literal</li><li>■ Process Property</li><li>■ Business Component</li><li>■ Expression</li></ul>
Value	A string value.	<p>For Literal and Expression type input arguments. This could be a picklist, depending on the argument selected. String values can only be a maximum of 32,767 characters.</p>
Property Name	The name of the business process property.	<p>For Process Property-type input arguments.</p>
Property Data Type	The data type of the business process property.	<ul style="list-style-type: none"><li>■ String</li><li>■ Number</li><li>■ Data</li><li>■ Hierarchy</li></ul>



**Table 14. Input Arguments Applet Fields**

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 Field–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 15](#) describes the fields in the Output Arguments applet.

**Table 15. 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 <a href="#">“Defining Process Properties” on page 69</a> .
Type	The type or argument.	This is a required field. The picklist contains the following choices: <ul style="list-style-type: none"> <li>■ Literal</li> <li>■ Output Argument</li> <li>■ Business Component</li> <li>■ Expression</li> </ul>
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 15. Output Arguments Applet Fields**

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 Field type.

---

**NOTE:** Business component fields based on multi-value groups cannot be selected as values for input or output arguments. If you want to use a field based on a multi-value group, you need to define a business component for the field and link it to the appropriate business object. See *Siebel Tools Reference* for more information.

---

---

**NOTE:** Calculated fields 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 access 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 Reference* 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 100.

---

**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 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 Expression, enter the expression in the value field.

**8** Enter any appropriate comments.

---

**NOTE:** Business services, methods, and arguments all have Display Name and Hidden properties in Siebel Tools. For a business service, method, or argument to be displayed on any picklist, the Hidden flag for the object must be set to FALSE. For more information, see [“Enabling a Business Service for Workflow Processes” on page 50](#).

---

## Working with Subprocess Steps

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

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

- Define a subprocess step
- Define input arguments for the subprocess
- Define output arguments for the subprocess

## Using the Subprocess View

You use the Subprocess view to define a step for invoking a subprocess, including the input and output arguments.

The top portion of the view, the Subprocess applet, is where you can perform various tasks. The lower portion has two applets: Input Arguments and Output Arguments. Input and output arguments allow you to pass values to the subprocess properties.

The tasks you can perform in this view are:

- **Define a subprocess step.** See [“Defining a Subprocess Step” on page 106](#).
- **Delete an existing subprocess step.** See [“Deleting a Step or Process” on page 157](#).
- **Copy a subprocess step.** See [“Copying a Process” on page 157](#).

**Subprocess Applet Field Descriptions**

Table 16 describes the fields where you enter data in the Subprocess applet.

**Table 16. Subprocess Applet Fields**

Field	Description	Possible Value
Step	The name of the subprocess step.	A descriptive name that is: <ul style="list-style-type: none"><li>■ Consistent with your overall naming strategy</li><li>■ Meaningful to the process designer</li></ul>
Type	The type of step.	This value is automatically entered when you create the step on the Process Designer view. Read-only.
Description	A text narrative describing the purpose of the subprocess.	Free-form text.
Subprocess	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 17](#) describes the fields in the Input Argument applet.

**Table 17. Input Arguments Applet Fields**

Field	Description	Possible Value
Subprocess Input	The name of the subprocess property.	This is a required field. The picklist contains process properties for the selected subprocess.
Type	The type of argument.	<p><i>Required.</i> The drop-down list includes these choices:</p> <ul style="list-style-type: none"> <li>■ Business Component. Uses the value of the selected business component field. If the field is a multi-value field or if the business component is a child business component that has multiple rows, the first value retrieved is used.</li> <li>■ Expression. Evaluates the expression that is entered in the Value column and uses the result. If the expression references any business component fields, you must enter the business component name in the Business Component column.</li> <li>■ Literal. Uses the literal value that is entered in the Value column.</li> <li>■ Process Property. Uses the current runtime value from the process property.</li> </ul>
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.
Business Component Field	The name of a field within the business component.	For Business Component Field type input arguments.



## Output Arguments Applet Field Descriptions

[Table 18](#) describes the fields in the Output Arguments applet.

**Table 18. Output Arguments Applet Fields**

Field	Description	Possible Value
Property Name	The name of the Process Property to store the results.	<p>This is a required field.</p> <p>This is a picklist of properties that have been defined for the process. For more information about defining process properties, see <a href="#">“Defining Process Properties” on page 69</a>.</p>
Type	The type of argument.	<p><i>Required.</i> The drop-down list includes these choices:</p> <ul style="list-style-type: none"> <li>■ Business Component. Uses the value of the selected business component field. If the field is a multi-value field, or if the business component is a child business component that has multiple rows, the first value retrieved is used.</li> <li>■ Expression. Evaluates the expression that is entered in the Value column and uses the result. If the expression references any business component fields, you must enter the business component name in the Business Component column.</li> <li>■ Literal. Uses the literal value that is entered in the Value column.</li> <li>■ Output Argument. Evaluates the output argument entered in the Output Argument column and uses the result.</li> </ul>
Value	A string value	For Literal or Expression arguments.
Subprocess Output	The name of the output argument from the subprocess.	For Output Argument type.
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 Field type.

## Defining a Subprocess Step

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

### **To define a subprocess step**

- 1 In the All Processes view, select the process that is to be called as a subprocess to make it active.
- 2 Navigate to the Process Designer view and double-click the Subprocess icon to access the details view.

---

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

---

- 3 Enter a name and description for the step. The description should be a description of the purpose of the step.
- 4 Select the process that the subprocess step will call.

---

**NOTE:** You can display the process design for the selected subprocess by choosing Subprocess from the applet menu.

---

- 5 If you need to define input arguments for this subprocess, continue to [“Defining Subprocess Input Arguments” on page 106](#). If you need to define output arguments for this subprocess, continue to [“Defining Subprocess Output Arguments” on page 107](#).

## Defining Subprocess Input Arguments

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

***To define input arguments for a subprocess step***

- 1** With the appropriate subprocess name displayed in the Subprocess applet of the Subprocess view and the Input Arguments applet active, choose Edit > Add New Record.
- 2** Enter the name of the subprocess 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 Subprocess Output Arguments**

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

***To define output arguments for a subprocess step***

- 1** With the appropriate subprocess name displayed in the Subprocess applet of the Subprocess 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 Siebel Operation Steps**

The Siebel Operation view allows you to define Siebel Operation steps such as Insert, Update, or Query. These steps are performed on business components.

After you define a Siebel Operation step, you can use the Fields applet to define any field values for the step. For the Update step, you can use the Search Specifications view to define which records you want to update. For the Query step, you can use this view to query for a record with specific field values.

- To define a Siebel operation step, see [“Defining a Siebel Operation Step” on page 112](#).
- To delete a Siebel operation step, see [“Deleting a Step or Process” on page 157](#).
- To copy a Siebel operation step, see [“Copying a Process” on page 157](#).

### Siebel Operation Field Descriptions

[Table 19](#) describes the Siebel Operation form fields.

**Table 19. Siebel Operation Fields**

Field	Description
Name	The name of the Siebel Operation step. Enter a descriptive name that is: <ul style="list-style-type: none"><li>■ Consistent with your overall naming strategy</li><li>■ Meaningful to the process designer</li></ul>
Description	Enter a text description of the purpose of the Siebel Operation step.
Operation	<i>Required.</i> Enter the type of operation: Insert, Query, or Update.
Business Component	<i>Required.</i> The business component that will perform the action you specify. Choose from a list of business components that have been defined for the selected business object.
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 Field Descriptions

[Table 20](#) describes the fields in the Fields list.

**Table 20. Fields in the Fields List**

Field	Description
Name	Choose the name of the business component field from a list of all fields defined for the business component except multi-value groups and calculated fields.
Type	<i>Required.</i> Choose the type of value to enter in the field: Literal, Process Property, Business Component, or Expression.
Value	If you entered Literal or Expression in the Type file, enter a string value.
Property Name	If you entered Process Property in the Type file, enter the name of the business process property.

**Table 20. Fields in the Fields List**

Field	Description
Business Component	If you entered Business Component in the Type file, enter the name of the associated business component.
Business Component Field	If you entered Business Component in the Type file, enter the name of the field associated with the business component.

**Search Specifications Field Descriptions**

[Table 21](#) describes the fields in the Search Specifications list.

**Table 21. Search Specifications Fields**

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

## Output Arguments Field Descriptions

Table 22 describes the fields in the Output Arguments list.

**Table 22. Output Arguments Fields**

Field	Description
Property Name	<i>Required.</i> The name of the process property that will store the results. Choose from a list of all the properties that have been defined for the process. For more information about defining process properties, see <a href="#">“Defining Process Properties” on page 69</a> .
Type	<i>Required.</i> Choose the type of argument: Literal, Output Argument, Business Component, or Expression.
Value	If you entered Literal or Expression in the Type file, enter a string value.
Name	If you entered Output Argument in the Type file, enter the name of the output argument from the End step.
Business Component	If you entered Business Component in the Type file, enter the name of the business component.

## Defining a Siebel Operation Step

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

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

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



component called “Account Team” and then associate it with the Account business object using Siebel Tools. You could then select Account Team as the business component to update with the Siebel Operation step.

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

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

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

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

---

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

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

---

#### **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 access 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
  - Query

---

**NOTE:** You should verify that updates or inserts of fields that have dependencies are valid fields. For example, if you have a service request process and your process is updating the area and sub-area fields, you will need to verify that the values selected for the sub-area field are valid for that associated area.

---

- 6** Select the name of the business component.

---

**NOTE:** After you double-click Siebel Operation in the Process Designer, you must specify the Business Component Name in the Fields applet.

---

- 7** If you need to define fields for this Siebel operation, continue to [“Defining Fields for the Siebel Operation” on page 115](#).
- 8** If you need to define search specifications for this Siebel operation, continue to [“Defining Siebel Operation Search Specifications” on page 115](#).
- 9** If you need to define output arguments for this Siebel operation, continue to [“Defining End Step Output Arguments” on page 147](#).

## 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, select Add New Record from the applet menu.
- 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, select Add New Record from the applet menu.
- 2** Select a search specification type. The choices available are:
  - Literal
  - Expression
- 3** Enter search specifications.

---

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

---

- 4** If the search specification type is expression, select the applicable business component name.

---

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

---

## **Defining Siebel Operation Step Output Arguments**

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

### ***To define output arguments for an operation step***

- 1** With the appropriate Operation step name displayed in the Operation Step applet of the Operation view and the Output Arguments applet active, select Add New Record from the applet menu.
- 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 or until a specific event occurs. Workflow administrators can specify to pause a process instance in units of seconds, minutes, hours, or days. In addition, administrators can specify a service calendar to account for business hours and days when waiting a specified duration.

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

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

The top portion of the view, the Wait applet, is where you can perform various tasks. The lower portion is for Input Arguments.

The tasks you can perform in this view are:

- **Define a wait step.** See [“Defining a Wait Step” on page 123](#).
- **Delete an existing wait step.** See [“Deleting a Step or Process” on page 157](#).
- **Copy a wait step.** See [“Copying a Process” on page 157](#).

**Wait Applet Field Descriptions**

[Table 23](#) describes the fields where you enter data in the Wait applet.

**Table 23. 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"> <li>■ Consistent with your overall naming strategy</li> <li>■ Meaningful to the process designer</li> </ul>
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.	Note that when the maximum number of iterations is reached, an Object Manager error will be generated and the workflow process will return an In Error status. If you want the process to run to completion, you need to use a Workflow exception mechanism (such as an error process or exception branch) to catch and handle the error. For more information, see <a href="#">“Working with Exceptions” on page 152</a> .
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.
Workflow Process	The name of the process definition.	Read-only.
Business Object	The name of the associated business object.	Read-only.

**Table 23. Wait Applet Fields**

Field	Description	Possible Value
Service Hours	The name of the schedule used in calculating the wait end time.	Optional. This value is selected from a picklist of service calendars.
Processing Mode	The mode in which the process will be resumed when triggered by runtime events.	Optional <ul style="list-style-type: none"><li>■ <b>Local Synchronous.</b> Executes the process in the application object manager. This is the default.</li><li>■ <b>Remote Synchronous.</b> Submits a synchronous request to the Workflow Process Manager server component to execute the process.</li><li>■ <b>Remote Asynchronous.</b> Submits an asynchronous request to the Workflow Process Manager server component to execute the process.</li></ul>



### Next Step Branches Applet Field Descriptions

[Table 24](#) describes the fields in the Next Step Branches applet.

**Table 24. 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.	<p>The value can be one of the following choices:</p> <ul style="list-style-type: none"> <li>■ <b>Condition.</b> This value indicates that a condition is defined for the branch.</li> <li>■ <b>Default.</b> 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.</li> <li>■ <b>Connector.</b> Use this value whenever there is no condition branching involved.</li> <li>■ <b>Error Exception.</b> Use this value to define exception handling. This connector type captures system errors, such as an error noting that the Assignment Manager server component is not available. For more information, see <a href="#">“Working with Exceptions” on page 152</a>.</li> <li>■ <b>User Defined Exception.</b> Use this value to define exception handling. This connector type captures user-defined errors, such as an error noting that an order being submitted is incomplete. For more information, see <a href="#">“Working with Exceptions” on page 152</a>.</li> </ul>
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.

**Table 24. Next Step Branches Applet Fields**

Field	Description	Possible Value
Event Object Type	The type of the object the event refers to.	Optional <ul style="list-style-type: none"><li>■ Application.</li><li>■ Applet.</li><li>■ BusComp.</li></ul>
Event	The specific event the object supports.	Required if Event Object Type is specified. The set of events is different for different object types.
Event Object	The name of the object.	Required if Event Object Type is specified. This is the name as defined in Tools. The set of objects is different for different object types.
Sub Event	An options parameter for the event.	Optional <p>For InvokeMethod, the name of the method being invoked. For SetFieldValue, the name of the field being set.</p>
Event Cancel Flag	Abort the runtime event after executing the process.	Optional <p>This flag only applies to events that are cancelable. It functions like CancelOperation in scripting.</p>

## Input Arguments Applet Field Descriptions

Table 25 describes the fields in the Input Argument applet.

**Table 25. Input Arguments Applet Fields**

Field	Description	Possible Value
Input Argument	The name of the input argument.	This pick applet contains the following choices: <ul style="list-style-type: none"> <li>■ Sleep Unit (The unit of time to wait.)</li> <li>■ Sleep Quantity (The numeric value associated with sleep unit.)</li> </ul>
Type	The type of argument.	This is a required field. The picklist contains the following choices: <ul style="list-style-type: none"> <li>■ Literal</li> <li>■ Process Property</li> <li>■ Expression</li> <li>■ Business Component</li> </ul>
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 access 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:** For durations greater than 60 seconds, you should specify minutes or a greater unit of measure so that business component data is refreshed. Wait steps with durations measured in anything other than seconds are automatically persisted. For more information, see [“Wait Steps and Workflow Persistence” on page 165](#).

---

## 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, select Add New Record from the applet menu.
- 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.

## Defining Wait Step Next Steps Arguments

A wait step can wait for a specified runtime event to occur. If the previous Input Arguments applet is populated with a time duration, that time duration will be treated as a timeout parameter. The wait step will then wait until either the specified runtime event occurs or the timeout parameter has expired, whichever comes first. The one that occurs first will determine the branch that is taken.

### ***To define next steps arguments for a wait step***

- 1** In the Wait Step detail view applet menu, select Next Step.
- 2** Select the Next Steps applet.
- 3** If the branches are not yet defined, add a branch record by selecting New Record from the applet menu.

A branch with a runtime event specified will be taken when that event occurs. A branch with no runtime event specified will be taken if no runtime event is specified for other branches or if the timeout parameter has expired. For more information about defining runtime events, see [“Using Runtime Events” on page 166](#).

## Working with a User Interact Step

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

The user interact step has the following behaviors:

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

---

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

---

- The user interact step waits for a runtime event to resume processing. It waits in the memory of the user session. In cases where there is no runtime event defined, the workflow process will continue to the end.
- If, after a user interact step, the user manually navigates out of the view, the workflow process remains in the memory of the user session. The process is deleted when the user session is terminated or when another workflow process is instantiated in the same user session.

- If a workflow process containing a user interact step is run in the Process Simulator, incorrect behavior will result. To test workflows that contain user interact steps, you must test the workflow in the runtime environment. For more detailed information, see [“Testing User Interact Workflows” on page 176](#).

---

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

---

---

**CAUTION:** If using the workflow persistence feature with user interact steps, note that workflows with user interact steps should be persisted only for debugging purposes. A persisted workflow that contains a user interact step can resume in a different user session based on which user generated the event, causing unintended behavior. For more information on persistence and user interact steps, see [“User Interact Steps and Workflow Persistence” on page 165](#).

---

## Using the User Interact View

This view is for entering and viewing the details for a user interact step. A user interact step is used to represent the step in the process flow where a Siebel view is presented to the end user. The user can select an action associated with a runtime event that will then branch the process to the next appropriate step.

The tasks you can perform in this view are:

- **Define a user interact step.** See [“Defining a User Interact Step” on page 134](#).
- **Define next step branches.** See [“Defining User Interact Next Step Branches” on page 134](#).

- **Define branch conditions and values.** See [“Defining Conditions and Values for User Interact Next Step Branches”](#) on page 135.

---

**NOTE:** This release does not support backward navigation in a workflow process that involves user interact steps. Going backward in a workflow process with user interact steps causes the workflow process instance to lose context, which may result in errors with messages such as: “Cannot resume process 'PROCESS\_INSTANCE\_ID' for object 'OBJECT\_ID'. Verify that an instance does exist and has a 'Waiting' status.”

---



**User Interact Applet Field Descriptions**

[Table 26](#) describes the fields where you enter data in the User Interact applet.

**Table 26. User Interact Applet Fields**

Field	Description	Possible Value
Name	The name of the user interact step.	A descriptive name that is: <ul style="list-style-type: none"><li>■ Consistent with your overall naming strategy</li><li>■ Meaningful to the process designer</li></ul>
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.
Description	A text narrative describing the purpose of the user interact 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 and time that the process is created.	This value is automatically entered.

### Next Step Branches Applet Field Descriptions

[Table 27](#) describes the fields in the Next Step Branches applet.

**Table 27. 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.	<p>The value can be one of the following choices:</p> <ul style="list-style-type: none"> <li>■ <b>Condition.</b> This value indicates that a condition is defined for the branch.</li> <li>■ <b>Default.</b> 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.</li> <li>■ <b>Connector.</b> Use this value when there is no condition branching involved.</li> <li>■ <b>Error Exception.</b> Use this value to define exception handling. This connector type captures system errors, such as an error noting that the Assignment Manager server component is not available. For more information, see <a href="#">“Working with Exceptions” on page 152</a>.</li> <li>■ <b>User Defined Exception.</b> Use this value to define exception handling. This connector type captures user-defined errors, such as an error noting that an order being submitted is incomplete. For more information, see <a href="#">“Working with Exceptions” on page 152</a>.</li> </ul>
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.

**Table 27. Next Step Branches Applet Fields**

Field	Description	Possible Value
Event Object Type	The type of the object the event refers to. This is used when defining a runtime event.	Optional <ul style="list-style-type: none"> <li>■ Application.</li> <li>■ Applet.</li> <li>■ BusComp.</li> </ul>
Event	The specific event the object supports.	Required if Event Object Type is specified. The set of events is different for different object types.
Event Object	The name of the object.	Required if Event Object Type is specified. This is the name as defined in Tools. The set of objects is different for different object types.
Sub Event	An options parameter for the event.	Optional <p>For InvokeMethod, the name of the method being invoked. For SetFieldValue, the name of the field being set.</p>
Event Cancel Flag	Abort the runtime event after executing the process.	Optional <p>This flag only applies to events that are cancelable. This flag functions like CancelOperation in scripting.</p>

### Conditions Applet Field Descriptions

[Table 28](#) describes the fields in the Conditions applet.

**Table 28. Conditions Applet Fields**

Field	Description	Possible Value
Compare To	Indicates where the comparison value is coming from.	This is a required field, with the following choices: <ul style="list-style-type: none"> <li>■ Business Component</li> <li>■ Process Property</li> <li>■ Expression</li> <li>■ Applet</li> </ul>

**Table 28. Conditions Applet Fields**

Field	Description	Possible Value
Operation	Identifies the comparison operation.	<ul style="list-style-type: none"> <li>■ <b>This Must Match.</b> The current value must match exactly, including case.</li> <li>■ <b>One Must Match.</b> One or more values must match exactly, including case.</li> <li>■ <b>All Must Match.</b> All of the values must match exactly, including case.</li> <li>■ <b>None Can Match.</b> None of the values can match exactly, including case.</li> <li>■ <b>This Must Match (ignore case).</b> The current value must match without regard to case.</li> <li>■ <b>One Must Match (ignore case).</b> One or more values must match without regard to case.</li> <li>■ <b>All Must Match (ignore case).</b> All of the values must match without regard to case.</li> <li>■ <b>None Can Match (ignore case).</b> None of the values can match without regard to case.</li> <li>■ <b>Greater Than.</b> Value must be greater than the comparison value.</li> <li>■ <b>Less Than.</b> Value must be less than the comparison value.</li> <li>■ <b>Between.</b> Value must be between a range of values.</li> <li>■ <b>Not Between.</b> Value cannot be between a range of values.</li> <li>■ <b>Is Null.</b> Value must be null.</li> <li>■ <b>Is Not Null.</b> Value cannot be null.</li> </ul>
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.
Business Component	Identifies the specific business component on which to base the condition.	<p>This is a required field when Business Component or expression is the Compare To value.</p> <p>The picklist includes business components defined for the business object selected for the process.</p>

**Table 28. Conditions Applet Fields**

Field	Description	Possible 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.  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 User Interact Step

#### **To define a user interact step**

- 1 Select the appropriate process in the All Processes view.
- 2 Navigate to the Process Designer view and double-click the User Interact icon to access the User Interact view.

---

**NOTE:** To add a user interact step to the workspace, drag and drop the User Interact icon from the palette to the workspace.

---

- 3 Enter or modify the step name.
- 4 From the picklist, select the view name to which you would like to navigate the user. Only views associated with the business object will be available in the picklist.
- 5 Enter a description of the purpose of the user interact step.

---

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

---

### Defining User Interact Next Step Branches

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

---

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

---

**To define a next step branch**

- 1 From the Process Designer 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 user interact step, then proceed to step 4.

---

- 2 Drag and drop a connector arrow to the workspace, connecting the user interact step with the new next step.
- 3 Double-click the connector arrow to access the User Interact 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 Choose a branch type. Type values are described in [Table 27 on page 130](#).

---

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

---

- 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 User Interact Next Step Branches**

Conditions and values affect the flow of your process execution. Different actions may occur depending on which path is followed. For example, you can define a condition based on the value of a priority field, so that if the priority is equal to

High, the process execution follows a branch leading to an action that sends an email to a vice president. However, if the priority is equal to Medium, the email is sent to an engineer.

#### **To define conditions and values**

- 1** Select the appropriate branch in the Next Step applet in the User Interact 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:
  - **This Must Match.** The current value must match exactly, including case.
  - **One Must Match.** One or more values must match exactly, including case.



- **All Must Match.** All of the values must match exactly, including case.

---

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

---

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

- 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 Reference* 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 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

[Table 29](#) describes how the Stop step is handled, depending on how it is called and in which object manager it is running.

**Table 29. How Workflow Process Manager Handles Stop Steps**

Stop Step Conditions		Results
Workflow policy calls a process that contains a Stop step.		Workflow Process Manager: <ul style="list-style-type: none"><li>■ Exits</li><li>■ Writes an error message to the log file.</li></ul>
A script or runtime event calls a process that contains a Stop step.	■ Process is running in the Workflow Process Manager object manager.	Workflow Process Manager: <ul style="list-style-type: none"><li>■ Writes an error message to the log file.</li></ul>
	■ Process is running in the application object manager.	Workflow Process Manager: <ul style="list-style-type: none"><li>■ Flags an error message to the user.</li></ul>

### Using the Stop View

The top portion of the view, the Stop applet, is where you can perform various tasks. The lower portion has two applets: Input Arguments and Output Arguments.

The tasks you can perform in this view are:

- **Define a new stop step.** See [“Defining a Stop Step” on page 142.](#)
- **Delete a stop step.** See [“Deleting a Step or Process” on page 157.](#)
- **Copy a stop step.** See [“Copying a Process” on page 157.](#)

**Stop Applet Field Descriptions**

Table 30 describes the fields where you enter data in the Stop applet.

**Table 30. Stop Applet Fields**

Field	Description	Possible Value
Name	The name of the stop step.	A descriptive name that is: <ul style="list-style-type: none"><li>■ Consistent with your overall naming strategy</li><li>■ Meaningful to the process designer</li></ul>
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 31 describes the fields in the Input Argument applet.

**Table 31. 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"><li>■ Literal</li><li>■ Process Property</li><li>■ Expression</li><li>■ Business Component</li></ul>
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.

## 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 access 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. 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” on page 143](#).

## 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.



## 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. See [“Defining an End Step” on page 147](#).
- Define output arguments for the end step. See [“Defining End Step Output Arguments” on page 147](#).

---

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

---

### End Step Applet Field Descriptions

[Table 32](#) describes the fields in which you enter data in the End Step applet.

**Table 32. 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"><li>■ Consistent with your overall naming strategy</li><li>■ Meaningful to the process designer</li></ul>
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.

**Table 32. End Step Applet Fields**

Field	Description	Possible Value
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 33](#) describes the fields in the Output Arguments applet.

**Table 33. 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 <a href="#">“Defining Process Properties” on page 69</a> .
Type	The type of argument.	This is a required field. The picklist contains the following choices: <ul style="list-style-type: none"><li>■ Literal</li><li>■ Business Component</li><li>■ Expression</li></ul>
Value	A string value.	For Literal or Expression arguments.
Business Component Name	The name of the business component within the business object of the business process.	For Business Component type.
Business Component Field	The name of a field within the business component.	For Business Component Field type.

## 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 access 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
  - 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 Error Processes

You can use an error process for handling errors. An error process is defined in the form applet in the All Processes or Process Designer view. Like the subprocess step, the error process must have been predefined in order to be selected. When the ellipsis associated with the Error Process Name field is selected, a pick applet of all available processes will be shown. Once an error process is selected, this process will be called when the current process reaches an error state. Processing of the current process will stop and end, and will instead start the error process.

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

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

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

- The error process handles the error successfully.

The error handling is considered successful if the error process arrives at an End step. In this case, the error process will be in the Completed state. No error code will be returned to the caller of the workflow process. The workflow process remains in the In Error state.

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

The error handling is considered failed if the error process arrives at a Stop step. In this case, the error process will be in the In Error state. A new error code, which you have selected in the Stop step, is returned to the caller of the workflow process. The workflow process remains in the In Error state.

- The error process cannot handle the error.

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

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

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

- If you want the original process instance to pass any user-defined process properties to the error process, you have to explicitly recreate those user-defined process properties in your error process, giving them the same name and data type.
- If you want to pass a property set from the original process to the error process, you have to create a common user-defined hierarchy process property in the original workflow process and in the error process, and use this common hierarchy property to pass the property set.
- If you want the error process to get the name of the original workflow process, you have to create a common user-defined process property in the original workflow process and in the error process, and then pass the original process name through this common user-defined process property.

## Assigning Error Processes to Subprocesses

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

- The error process handles the error successfully.

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

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

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

## 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

## 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.



## 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 See [“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. Error codes can be text-based or number-based. Because error-code numbers change from release to release, use of Error Messages is preferred.

---

#### **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.

---

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

---

- **None Can Match.** None of the values can match exactly, including case.
- **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.

## Working with Expressions

Expressions can be used in workflow processes. The timestamp argument is of particular note, as its arithmetic varies from the time information used in workflow policy programs.

### Using the Timestamp Argument

You can use the timestamp argument to get the current system time and to do time arithmetic based on the current time.

The arithmetic involving time information for workflow processes is different from that for workflow policy programs. The second operand to the 'Timestamp ()' function must be provided in a scale of minutes, that is, as a fraction of the whole day.

For example, if the intended result of an arithmetic operation is 30 minutes, then the argument should appear as follows:

```
Timestamp()+0.021
```

The operation is explained as follows:

- $0.021 = 30 / (24 * 60)$
- $(24 * 60)$  represents a whole day in minutes
- 30 represents the required minutes

## Deleting a Step or Process

---

**CAUTION:** Deleting a process definition deletes all associated process instances, whether they are stopped, waiting, or running. Before you delete a process, be sure that it is not a subprocess in another process. To back up or save the records of the process, you can query the process definitions 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.

---

### ***To delete a process***

- 1 From the Business Process Administration screen, choose the All Processes view.
- 2 Select the process you want to delete.
- 3 Right-click the applet menu, then select Delete Process.

### ***To delete a step***

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

## Copying a Process

### ***To copy a process***

- 1 Locate and select the process you want to copy.
- 2 Click Copy.
- 3 Enter a new name for the process.
- 4 Modify the other definition fields as necessary for the new 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 checks the syntax for validity, registers runtime events if used, and changes the status of the process to Active. It also changes the status of the previous active version to Outdated.

When a process with runtime events specified is activated, you will also need to load the runtime events. Navigate to Runtime Events Administration, then click the applet menu and select Reload Personalization. This will load the runtime events in the current object manager session. For more information, see *Personalization Administration Guide*.

## Revising a Process Definition

### *To revise a process definition*

- 1 Select the target workflow process to revise and click Revise.

---

**NOTE:** The process should not have a status of In Progress. In Progress workflow processes are modifiable and do not need to be revised.

---

A copy of the process will be created. The name will remain the same, but the version number will increment by one.

- 2 Once the process definition has been thoroughly tested, the process can be made available. 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 Activate.

## Working with Seeded Workflow Processes

Many applications include seeded workflow processes for product functionality. By default, these seed workflow processes are inactive with a version number of 0. Seed data that is inactive must first be revised and then activated. Processes that include runtime events must be activated in the customer environment to create the appropriate runtime events. For information about seed workflow processes, see [Appendix D, “Seeded Workflow Processes.”](#)

## Configuring Workflows in a Multilingual Environment

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

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

- 1 Navigate to Site Map > Application Administration > List of Values.
- 2 Run the following query: Type = "WF\_\*".
- 3 Enable the database for multilingual lists of values by running the MLOV upgrade utility. For information on running the MLOV upgrade utility, see *Siebel Tools Reference*.
- 4 In the List of Values applet, make sure the Active flag is set.

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

## Defining Expressions for Workflows Running in a Multilingual Environment

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

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

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

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



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

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

## **Wait Steps and Global Time Calculations in Workflow**

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

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

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

---

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

---

## Using Workflow Persistence

Workflow persistence is a feature used to store the state of a workflow process instance and its steps, as well as its process properties.

### Enabling Workflow Persistence

Persistence is controlled by two settings within workflow definitions: persistence frequency and persistence level. You enable workflow persistence in the Workflow Processes view by adjusting these settings.

#### Persistence Frequency

Persistence frequency dictates how often the workflow engine stores state data. By default, persistence frequency is set to Never, meaning that persistence is not enabled for that particular process definition. The other options for the Persistence Frequency field are the following:

- **On Pause.** This frequency setting saves state data whenever the workflow is in the start step, the end step, and any other step that is waiting.
- **Every Step.** This frequency setting saves state data after every step in the process.

#### Persistence Level

Persistence level dictates the quantity of workflow state data that is saved to the database. The options for the Persistence Level setting are the following:

- **All Steps.** This level setting saves state data for all steps.
- **Current Step.** This level setting saves only the state data at the current step.

---

**NOTE:** If data of size greater than 16K is saved into the business component table, only 16K is retrieved and the rest is truncated.

---

#### *To enable workflow persistence*

- 1 Navigate to Site Map > Business Process Administration > Workflow Processes and select the process you want to work with.

- 2 On the All Processes tab, enter values for the Persistence Frequency field and the Persistence Level field using the drop-down picklists.

---

**NOTE:** If you are enabling persistence to suspend and resume a process, the recommended persistence settings are On Pause for persistence frequency and Current Step for persistence level. Only the current step data is used for resuming a process. Any higher setting has no effect on persisting and resuming the process.

---

## Usage Guidelines for Workflow Persistence

Use of the workflow persistence feature can impact performance significantly, depending on the setting combinations for frequency and level. You should give careful consideration to the persistence frequency and persistence level settings you apply to workflow processes to make sure you get the results you want.

The usage guidelines for workflow persistence are separated into the following categories:

- Long-running workflows
- In-memory workflows
- Troubleshooting of workflows

### Long-Running Workflows

Long-running workflows are typically flows that start, run some activities, suspend, and then resume at a later time. The lifecycle of such a workflow can span multiple suspend periods and resume periods.

For long-running workflows, the recommended settings are the following:

- Level: Current Step
- Frequency: On Pause

This setting combination should be used selectively, only on workflows that have long-running workflow characteristics. Use this setting combination only for workflow processes with execution spanning a significant time duration and which need to be suspended and resumed at a later time. Production workflows must have these settings to enable the long-running workflow feature.

### **In-Memory Workflows**

In-memory workflows—or short-duration workflows—are typically jobs that start and finish as a single execution of a series of steps in memory. For in-memory workflows, it is highly recommended that persistence be disabled on all production systems. Using workflow persistence for in-memory workflows causes performance to degrade significantly.

### **Troubleshooting of Workflows**

For both long-running and in-memory workflows, persistence can be used to debug and troubleshoot workflows. It is recommended that you use the persistence feature for troubleshooting of workflows *only* in the development environment.

In production, errors that occur when executing workflows are logged in server log files. This is the recommended approach for detecting errors in the production environment.

The various setting combinations impact performance at different levels. For example, the combination of a persistence level of Current Step and a persistence frequency of On Pause provides current-step-level debugging data with moderate performance impact. This is the minimum setting. As another example, the combination of persistence level: All Steps and persistence frequency: Every Step provides step-level debugging at all steps with very high performance impact. This is the maximum setting.

It is strongly recommended that these settings:

- Be used only when troubleshooting and testing.
- Be used only on a process-by-process basis rather than on all processes.

- Be disabled once the workflow is ready to be deployed. These settings should not be used on production systems.

---

**NOTE:** If persistence is enabled, large numbers of records can accumulate in Workflow instance tables over a period of time. To prevent this from creating space problems, unwanted records should be periodically purged using the Purge functionality in the Workflow Process Log view.

---

## **Wait Steps and Workflow Persistence**

For wait steps with wait units defined in minutes, hours, or days, persistence is implicitly enabled. No user setting is needed for this.

## **User Interact Steps and Workflow Persistence**

For workflow processes containing user interact steps, the persistence feature should be used for debugging purposes only. It is generally not recommended to use persistence for workflows that contain user interact steps.

## Using Runtime Events

Runtime events allow the Siebel application to respond in real time to user actions. Runtime events can be specified in the Start, Wait, or User Interact steps in the Next Steps applet to start or resume a workflow process. There are four columns in the Next Steps applet that are used to define a runtime event:

- **Event Object Type.** The type of object to which the event occurs. This can be an application, business component, or applet.
- **Event.** The specific event that happens to the object. The set of available events is different for different object types.
- **Event Object.** The name of the application, business component, or applet to which the event occurs.
- **Sub Event.** Used when the object type is BusComp or Applet and the event is InvokeMethod or SetFieldValue. The subevent is the name of the method or business component field to be monitored.

---

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

---

The following events are not supported with the user interact step:

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

Workflow policies should be used if it is necessary to detect database events. You should use workflow policies instead of events for defining integrations that occur on data change or write.

For more information on runtime events, see *Personalization Administration Guide*.

## Creating Buttons that Invoke Workflow Processes

### ***To create a button that invokes a workflow process***

- 1** From Siebel Tools, configure a button on an applet and specify the `MethodInvoked` property. For more information, see *Siebel Tools Reference*.
- 2** To enable the button, override the `PreCanInvokeMethod` property. Next, edit the server script and compile your changes to the Siebel repository file. The following example is specific to Siebel VB:

```
Function WebApplet_PreCanInvokeMethod (MethodName As String,  
CanInvoke As String) As Integer  
  
If MethodName = "<Name>" Then  
  
    CanInvoke = "True"  
  
    WebApplet_PreCanInvokeMethod = CancelOperation  
  
Else  
  
    WebApplet_PreCanInvokeMethod = ContinueOperation  
  
End If  
  
End Function
```

- 3** Define a workflow process. To invoke this workflow process from a button click, specify a runtime event in one of the following ways:
  - To start this workflow process, specify the runtime event on the Start step on a Condition branch.

- To resume this workflow process (if it has been paused) from a button click, specify the runtime event in a User Interact step or a Wait step on a Condition branch.

The fields values are described in [Table 34](#).

**Table 34. Runtime Event Field Values**

Field	Value
Event Object Type	BusComp
Event	PreInvokeMethod
Event Object	Name of the business component on which the applet that contains the button is based.
Sub Event	Name of the method set in step one. The Sub Event name must be unique.
Event Cancel Flag	True. If this flag is not checked, the error “The specialized method < Method Name > is not supported on this business component” will result when running the workflow process.

---

**NOTE:** If users click the button when there are no workflow process instances waiting, they will receive the error message “Cannot resume process < Process ID > for object < Object ID >”. You can prevent this error by requiring users to run a workflow process in order to access this view. For example, you can set up a workflow that includes the User Interact step. You can handle the error by defining an Error Exception branch or error process.

---

- 4 After you create the workflow process, activate the process and reload personalization. To reload personalization, navigate to Site Map > Runtime Events Administration > Events, and then select Reload Personalization from the applet menu. You must reload personalization after activating a workflow process that registers any runtime events in order for the process to take effect.



## **Choosing Between Using Runtime Events and Using Workflow Policies**

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



## **Testing Workflow Processes**

# **5**

This chapter describes testing a workflow process using the Process Simulator.

# Testing Processes with the Process Simulator

Testing your workflow processes before migrating them to your production environment verifies that resulting actions are accurate and useful and the results are exactly what you want.

You need to develop a test and migration procedure for introducing changes into the production environment. Some of the considerations for this procedure are discussed in [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 Business Process Designer. 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 application object manager. Actual invocation of the process may be run in the application object manager or in the Workflow Process Manager server session, depending on specific parameters.

---

The other methods involve invoking a workflow process outside of the Siebel Business Process Designer. For information on these methods of invoking a workflow process, see [“Invoking a Workflow Process” on page 178](#). For information about invoking a workflow process from a server component, see *Overview: Siebel eBusiness Application Integration Volume I* and *Siebel eMail Response Administration Guide*.

## 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.

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 access the detail steps by double-clicking the step icons.

This applet also contains the buttons for running the process. [Table 35](#) describes the buttons on the Simulator applet.

**Table 35. 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 36](#) describes the fields in the Step Details applet.

**Table 36. 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.
Process Property Hierarchy	Stores process property value of type hierarchy.

**To debug a process with the Process Simulator**

---

**NOTE:** The Process Simulator will not work with processes that have runtime 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.

---

---

**NOTE:** Workflow Process Simulator will simulate a wait period if a Wait step is specified in seconds. However, if the unit of time is specified in minutes or greater, the Simulator will simply move on to the next step.

---

- 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 use the Process Designer view at any time to make changes to the step details, and then return to the Process Simulator to debug the process.

### **Testing User Interact Workflows**

You cannot use the Process Simulator to test workflows that involve User Interact steps. If a workflow process containing a User Interact step is run in the Process Simulator, incorrect behavior will result. To test a workflow process that involves a User Interact step, you test the workflow in the runtime environment.

You must make sure that the corresponding view is assigned to the application that the user is using. For example, if the user session views are in Siebel Sales, make sure you are testing the workflow in the Siebel Sales environment, and not in the environment of another application, such as Siebel Call Center. Verify that the environment in which you are testing contains all the views you want to test.

#### ***To test a process involving a User Interact step***

- 1** Activate the workflow you want to test.
- 2** Verify that the corresponding view is assigned to the application used in the user session.
- 3** Reload personalization to load the runtime events in the current object manager session.
- 4** Test the workflow in the runtime environment.



# Running Workflow Processes

# 6

This chapter describes the different ways to invoke a workflow process and the different processing modes.

## 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 runtime event
- As a configured business service

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. *Overview: Siebel eBusiness Application Integration Volume I* and *Siebel eMail Response Administration Guide* discuss invoking a process from a server component. The other invocation methods are explained in this chapter.

---

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

---

## Invoking From a Workflow Policy

To invoke a workflow process from a workflow policy, you define a policy action that uses the workflow policy program Run Workflow Process. Alternatively, you can create a custom workflow policy program by copying the Run Workflow Process

program and then adding program arguments that correspond to workflow process properties. This way you can use the policy program to pass data to the workflow process properties.

---

**NOTE:** For complete information about defining workflow policies, see [Chapter 10, “Creating Workflow Policies.”](#)

---

**To invoke a process from a workflow policy**

- 1** Choose Site Map > Business Process Administration > Workflow Policy Actions, then define a new action.
  - a** In the Program field, use the picklist to select the Run Workflow 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** Navigate to the Workflow Policy Group view and define a new group.
- 3** Navigate to the Workflow 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 Generate Triggers.

See [“Creating Database Triggers” on page 306](#) for more information about running the trigger generator.
- 5** If you are using the Run Workflow Process program, verify that the Workflow Process Manager server component is online.
- 6** Run Workflow Monitor Agent.

See [“Using Workflow Monitor Agent” on page 323](#) for more information about monitoring workflow policies.
- 7** Violate 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 application 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.

```
// Example: Invoking a Workflow Process via scripting
function Invoke_Process()
{
  var svc = TheApplication().GetService("Workflow Process
  Manager");
  var Input = TheApplication().NewPropertySet();
  var Output = TheApplication().NewPropertySet();
  var bo = TheApplication().ActiveBusObject();
  var bc = bo.GetBusComp("Account");
  var rowId = bc.GetFieldValue("Id");

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

  svc.InvokeMethod("RunProcess", Input, Output);
}
```

**Example: 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.

```
//Example: Passing Field Values to Process Properties
function Invoke_Process()
{
    var svc = TheApplication().GetService("Workflow Process
Manager");
    var Input = TheApplication().NewPropertySet();
    var Output = TheApplication().NewPropertySet();
    var bo = TheApplication().ActiveBusObject();
    var bc = bo.GetBusComp("Opportunity");

    var rowId = bc.GetFieldValue("Id");
    var accountId = bc.GetFieldValue("Account Id");

    input.SetProperty("ProcessName", "My Opportunity Process");
    input.SetProperty("Object Id", rowId);

    // Pass the value of the Account Id field to the Account Id
process property
    Input.SetProperty("Account Id", accountId);

    svc.InvokeMethod("RunProcess", Input, Output);
}
```

## Invoking From a Runtime Event

Workflow Processes integrates with the Runtime Events Engine to provide a simplified event mechanism for automating business processes. This mechanism:

- Allows real-time monitoring of events
- Minimizes the need for scripting and workflow policy invocation

The following types of events are possible:

- Application
- Business Component
- Applet

---

**NOTE:** For more information about runtime events, see *Personalization Administration Guide* or *Siebel Tools Reference*.

---

## Defining a Configured Business Service

With a configured business service like Workflow Process Manager (Server Requests), you do not need to specify SRB parameters.

---

**NOTE:** To use Workflow Process Manager, SRP and SRB must be running. For more information, see *Siebel Server Administration Guide*.

---

When you invoke the Server Requests directly, you must specify SRB parameters.

### **To define a configured business service**

- 1 In Tools, navigate to Siebel Objects > Business Service. Add a new record with values:
  - Name: (this is the name that you can reference in scripting)
  - Class: CSSSRmService
  - Display Name: (this is the name that you see in Workflow views)

- 2 Click Business Service User Prop. Add the following records:
  - Name: Component
  - Value: (internal [or short] name of the server component, for example, 'WfProcMgr')
  - Name: Mode
  - Value: (mode of the server request, for example, 'Async')
- 3 (Optional) Enter additional user properties pertaining to the SRB (see [Appendix A, “Predefined Business Services”](#)).
- 4 Select Business Service Method. Add the following records:
  - Name: (this is the name that you can reference in scripting)
  - Display Name: (this is the name that you see in workflow views)
- 5 Click Business Service Method Arg. Add records specific to the component being invoked, for example, 'ProcessName' for WfProcMgr. Note that the name is the internal (or short) name of the server component parameter.

---

**NOTE:** SRB and SRP are required to run any business service that invokes server components.

---

## Workflow Processing Modes

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

### Running a Workflow Process Within the Workflow Object Manager

A workflow process can be run within the Workflow Process Manager server component. The methods of invoking a workflow process that support running a process on the server include:

- From a workflow policy that executes on the server
- From a script specifying the Server Request parameter
- From a Runtime event with Processing Mode set to Remote Synchronous or Remote Asynchronous

If a user invokes a process to be run on the server within the Workflow Process Manager server component, the process executes only if the user is connected to the server. If the user is not connected to the server, the request is queued and executes when the user synchronizes or the server becomes available.

If you compiled a custom .srf file using Siebel Tools, this file needs to be added to the Objects directory on the Siebel Application Server. In addition, you must update the siebel.cfg file referenced in the Server parameters to reflect the custom .srf file (note that the siebel.cfg file is the default configuration file for Workflow Process Manager server components).

---

**NOTE:** Business services calling UI functions, including navigation functionality such as the User Interact step, are not supported when processes run on the server.

---



## Running a Workflow Process On the Application Object Manager

Running a workflow process in the application object manager can be very useful for enforcing business processes with mobile users or for defining business processes that involve end-user navigation.

The methods that support invoking a workflow process in the application object manager include:

- From the Process Simulator
- From a script specified to run locally in the application object manager
- From a runtime event with processing mode specified as local synchronous

## Running a Workflow Process in Batch

Workflow processes can be run in batch mode by running the Workflow Process Batch Manager server component. Executing a process in batch allows you to execute the actions in a workflow process for multiple records. When you are running a process in batch, you may want to specify a search specification to limit the number of records that are evaluated.

Workflow Process Batch Manager takes the parameter SearchSpec, then executes the search specification on the primary business component of the process business object. For each fetched record, the Workflow Process Batch Manager invokes the workflow process and sets the Object ID process property as the current active row.

**Table 37. Workflow Process Batch Manager Parameters**

Display Name	Description
Workflow Process Name	Required. Name of the workflow process definition to execute.
Search Specification	Search specification that identifies the work items to process.

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

Component jobs are used for regularly scheduled components. Component requests are used in this example:

- 6** Navigate to Site Map > Server Administration > Enterprise Operations.
- 7** In the Component Request form, click New.
- 8** In the Component/Job field, click the ellipsis. The Component/Jobs dialog box appears.
- 9** Select Workflow Process Batch Manager.
- 10** In the Component Request Parameters form applet, click New.
- 11** In the Name field, click the ellipsis, then select Workflow Process Name from the dialog box.
- 12** In the Value field, type in the name of the Workflow Process to execute.
- 13** Click New to add another parameter.
- 14** In the Name field, click on the ellipsis.
- 15** Select Search Specification.
- 16** In the Value field, provide a search specification.

# Migrating to the Production Environment

# 7

This chapter describes how to migrate workflow processes to production.

# Migrating Workflow Processes to Production

Once you have tested your workflow processes 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 functions in the applet menu.

## Importing or Exporting a 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 Business Process Administration > Workflow Processes > All Processes.
- 2** Select the process or processes you want to export. To select more than one process, press and hold the CTRL key while selecting the processes.
- 3** From the Processes list menu, select Export Workflow.  
The XML workflow process definition appears.
- 4** From the XML dialog box menu, select File > Save As.
- 5** Enter the file path, file name, and the .xml file name extension, and then click Save.

The process or processes are exported. If you selected more than one process to export, all the processes are saved to one XML file.

---

**NOTE:** When exporting a process containing subprocesses, you must also export the subprocesses. Subprocesses are not exported automatically.

---

**To import a process definition**

- 1** Navigate to Business Process Administration > Workflow Processes > All Processes.
- 2** From the applet menu, select Import Workflow.
- 3** Select a path and file name of the process to import.
- 4** Click Open.

The process is imported with a status of In Progress.

---

**NOTE:** If a process definition of the same name exists in the target environment, the newly imported process definition's version number will be incremented by one.

---

## Exporting Multiple Process Definitions

**To export multiple process definitions**

- 1** Navigate to Business Process Administration > Workflow Processes > All Processes.
- 2** From the applet menu, select New Query.
- 3** Enter your query criteria, then select Run Query.
- 4** Select the processes you want to export.
- 5** Select Export Workflow, then click Export.
- 6** Select a path and file name for the processes you are exporting.

---

**NOTE:** When exporting a process containing subprocesses, you must also export the subprocesses. Subprocesses are *not* exported automatically.

---

## **Migrating to the Production Environment**

*Migrating Workflow Processes to Production*

# **Administering Workflow Processes**

# **8**

This chapter describes workflow administration tasks.

## Monitoring Processes

Workflow administrators can graphically view process instances and current steps if persistence is defined for a particular process definition.

Workflow process instances with persistence defined can be viewed in the Workflow Process Log view.

For more information on persistence, see [“Using Workflow Persistence” on page 162](#).

## Correcting Processes

Workflow administrators can correct and resume a process instance that has encountered errors. For example, if the Communications Server is not available, a workflow process sending an email notification will have a status of In Error. An administrator can enable the Communication Server component, then resume the workflow process. Resume will execute the last step called by the workflow engine.

A process must have a status of In Error to be resumed. If a workflow process has an error process defined for it, then the error process may handle the error without returning a status of “in error.” For example, a workflow process that has an error process defined for it encounters an error. It stops processing and passes all process properties to the error process. The error process does one of the following:

- **Returns the error.** If no Start conditions are satisfied, the error process returns the error and you can resume the original process.
- **Returns a different error.** If the error process arrives at a Stop step, it returns the new error and you can resume the original process.
- **Handles the error.** If the error process arrives at an End step, it does not return an error. If the error process handles the error, you cannot resume the original process.



## Stopping Processes

Workflow administrators can stop a process instance if persistence is defined. Process instances with a status of Running, Waiting, or Error can be stopped.

### **To stop a process instance**

- 1** Navigate to the Workflow Process Log View.
- 2** Select the process instance to stop.
- 3** Click Stop.

---

**CAUTION:** Stopped processes cannot be resumed.

---

## Deleting Workflow Process Instances

Using the purge function, an administrator can delete all process instances with a status of Stopped or Completed before the user-specified date. If you want to delete a paused instance, stop the instance first. Individual instances can be deleted using the delete function.

### ***To purge all process instances***

- 1** Navigate to the Workflow Process Log View.
- 2** Click Purge.
- 3** Select a date.
- 4** Click Purge.

### ***To delete an individual process instance***

- 1** Navigate to the Workflow Process Log View.
- 2** Select the process instance.
- 3** From the applet menu, select Delete Record.

---

**NOTE:** Instances with In Error cannot be deleted. An administrator will need to first stop the errored instance before it can be deleted.

---

## Tracing and Event Log Levels

Table 38 lists the events that Workflow Processes uses for logging:

**Table 38. Workflow Process Logging Events**

Event	Level	Description
Engine Invoked	3	Trace methods invoked
Engine Invoked	4	Trace methods invoked and arguments passed
Definition Loading	3	Trace process definitions loaded into memory
Definition Loading	4	Trace process and step definitions loaded into memory
Process Execution	3	Trace process instantiation
Process Execution	4	Trace process instance creation/completion Trace process properties get/set
Step Execution	3	Trace step instantiation
Step Execution	4	Trace step creation/completion Trace branch condition evaluation Trace business service invocation Trace business component insert/update

**NOTE:** Setting trace levels above default parameters will affect performance. Trace levels should be reset to default parameters after troubleshooting has been completed.

## Increasing Tracing Levels for Workflow Process Manager and Workflow Process Batch Manager

You can generate a more detailed trace file to assist in troubleshooting Workflow Process Manager and Workflow Process Batch Manager.

**NOTE:** These steps should be completed prior to executing the server process.

#### **To increase tracing levels**

- 1** From the application-level menu, choose View > Site Map > Server Administration > Components.
- 2** In the Components applet, choose the component for which you want to generate tracing, that is, either Workflow Process Batch Manager or Workflow Process Manager.
- 3** In the Component Event Configuration tab, the Log Level column is set to a default value of 1. Change this value from 1 to 4 for the following Event Types:
  - SQL Tracing
  - Object Manager SQL Log
  - Workflow Definition Loading
  - Workflow Engine Invoked
  - Workflow Process Execution
  - Workflow Step Execution

---

**NOTE:** More tracing information is generated as the Component Event Configuration Log Level value increases. For more information on the different Log Level values available for Component Event Configuration, see *Siebel Server Administration Guide*.

---

You must set the SQL Trace Flag parameter to 8 when starting a Workflow Process Manager task or a Workflow Process Batch Manager task. You can adjust this setting at the component level or when starting the component. See [“To set the SQL Trace Flag parameter” on page 197](#).

**To set the SQL Trace Flag parameter**

- At component level:

Navigate to View > Site Map > Server Administration > Components > Component Parameters.

---

**NOTE:** When making changes to Workflow Process Manager and Workflow Process Batch Manager in the Components view, the Siebel Server service does not need to be restarted. The changes take effect immediately.

---

- When starting a component request manually:

Navigate to View > Site Map > Enterprise Operations > Component Request.

---

**NOTE:** When making parameter changes at the component level in the Component Request view, the Siebel Server service does not need to be restarted for the changes to take affect.

---

The Workflow Process Batch Manager trace file (WfProcBatchMgr\_xxx.log) and the Workflow Process Manager trace file (WfProcMgr\_xxx.log) are located in the Siebel server/log directory. Once the process has completed, change the values that were modified back to their default value before running the same process again.

# Restricting Mobile Client Routing

The Replication field in the Workflow Processes view allows you to choose whether to route a workflow process definition to your mobile clients. Routing only the workflow process definitions your mobile clients need lets you reduce the amount of data in the local database. [Table 39](#) lists the possible values for the Replication field.

**Table 39. Workflow Processes View**

Field	Description
Replication	<p>Choose from these possible values:</p> <ul style="list-style-type: none"><li>■ <b>All.</b> The workflow process definition is routed to all mobile clients and regional nodes.</li><li>■ <b>Regional.</b> The workflow process definition is routed to the regional nodes only.</li><li>■ <b>None.</b> (Default) The workflow process definition is not routed to the mobile client or the regional nodes.</li></ul>

**NOTE:** Support of workflow processes on regional nodes is restricted. For more information, see [“Running Workflow Processes on Regional Nodes” on page 199](#).

## Modifying the Replication Field

Here are some things to remember when you are modifying the Replication field:

- Before you can modify the Replication field, the workflow process definition must have a status of In Progress. If the workflow process definition does not have this status, click the Revise button to create a new version and assign it In Progress status.
- Changing the Replication field value from None to All adds the workflow process definition and all related records to the mobile client or regional node when it synchronizes with the server.

- If your workflow has a status of In Progress, you can change the Replication field value from All to None (without creating a new version of the workflow) to delete the workflow process definition and all related records from the mobile client or regional node when it synchronizes with the server.

## **Running Workflow Processes on Regional Nodes**

You can execute workflow processes on regional nodes. The workflow process can be called from script or runtime events.

When executing a workflow process on a regional node, the workflow process must reside on the regional node. The settings and environment should be replicated entirely on the nodes. All the objects the workflow is referencing must be available on the regional node.





## **Planning for Workflow Policies**

# **9**

This chapter describes planning activities for workflow policies.

## 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 16, "Monitoring and Tuning Performance."](#)

## 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 12, “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 10, “Creating Workflow Policies,”](#) for more information on creating workflow policies.

---

**NOTE:** After planning a new workflow policy, you should test the policy definition by creating a query based on the policy. Then you can execute the query on your current production environment. The query response can help you determine the frequency of the workflow conditions. You may find that a policy creates excessive notification or insufficient visibility. See [Chapter 14, “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 that which is listed in [Table 40](#).

**Table 40. 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 41](#) 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 41. Planning Policies**

Name	Workflow Object	Workflow Group	Activation/ Expiration Date/ Time	Duration
Email Confirmation of SR	Service Request	Medium Frequency		0

**NOTE:** Duration indicates the time element that must be met before an action is performed. Each workflow policy has one duration, so if you need to cause an action to occur after one hour, two hours, and six hours, you must create a different policy for each duration.

After you determine your policy's workflow object and other properties, you need to define the workflow conditions, as shown in [Table 42](#). Conditions are in the form of an expression.

**Table 42. 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 43](#) illustrates the type of information you need to define a workflow policy action.

**Table 43. Workflow Policy Actions**

Action Name	Program	Workflow Object	Arguments
Send SR Email to Contact	Send SR Email	Service Request	Send to Contact

**NOTE:** Workflow Policies comes with a set of predefined actions and programs. You can use these or define your own actions or programs to suit your business needs.

## 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 44](#) lists a workflow policy that monitors quotes with a discount exceeding 30%, for which the purpose is to notify the sales manager to review and approve the quote.

**Table 44. 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 45](#) 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 45. 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 46 illustrates the type of information you need for the policy conditions.

Table 46. 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 47 lists definitions for the Send Email to Sales Manager action.

Table 47. Actions and Action Arguments

Action Name	Program	Workflow Object	Arguments and Substitutions
Send Email to Sales Manager	Send Quote Email	Quote	Subject: Please approve quote discount for [Account]  Message Template: Please approve the quote discount for quote [Quote Number] and notify [Last User First Name] [Last User Last Name]  Repeating Message: The following quotes also need approval [Quote Number]

Scenario 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 48 represents the general policy definition.

**Table 48. 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 49. Workflow Policies**

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

Table 49 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 50 shows the workflow condition definitions.

**Table 50. 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 51 on page 210 shows the action argument definitions.

**Table 51. Actions and Action Arguments**

Action Name	Program	Workflow Object	Arguments and Substitutions
Alert Agent of Open SR	Send SR Message Broadcast	Service Request	Abstract: You have over 20 service requests Message Template: You have over 20 service requests. Please review your service request queue.

## Defining a Test and Migration Strategy

Before implementing new workflow policies, you must verify them in a test environment consisting of a sample Siebel database and workflow policies test data. Testing new policies, conditions, and actions checks that the policy you release into the production environment properly executes and does not cause conflicts with your existing workflow policies.

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

- Make sure your test environment and production environment have identical versions of the software and that you are using realistic data in your database by using a partial or complete copy of the production database.
- Create a small group of workflow policies to implement as a first phase of implementation. After you have successfully implemented the first group, you can add more policies in a systematic manner.
- To verify a new workflow policy, go to your production environment, manually create a query based on the new policy, and check the response. This helps you determine if a policy creates excessive notification or misses the rows you want to monitor.

For more information on migrating your test environment to your production environment, see [Chapter 14, “Testing Workflow Policies,”](#) and [Chapter 7, “Migrating to the Production Environment.”](#)

## **Planning for Workflow Policies**

*Defining a Test and Migration Strategy*

# Creating Workflow Policies **10**

This chapter describes creating workflow policies.

# Using the Workflow Policies Views

The views you use to create and define workflow policies are a part of the Siebel Business Process Designer. To display the Workflow Policies view, you navigate to Workflow Policies in the Business Process Administration screens of Site Map.

Several views are used when you create Workflow policies. These views include:

- **Workflow Policies Action view.** Use to create the actions you want to use with your workflow policies.
- **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, to check policy trends, 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 Business Process Designer, you use the Workflow Policies Action view to define policy actions. This view and its associated fields are described below. For the procedure on creating a workflow policies action, go to [“Creating a Workflow Policy Action” on page 227](#).

The Workflow Policies Action view consists of three applets. These applets are:

- **Actions applet.** This is where you create a name for your action and choose the appropriate program.
- **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.

Understanding the Actions Applet

Table 52 describes the fields of the Workflow Policies Actions applet.

Table 52. 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"><li>■ Consistent with your overall naming strategy</li><li>■ Meaningful to the policy maker</li></ul>
Program	The workflow policies program associated with the action.	This program is chosen from a picklist. See <a href="#">Appendix B, “Predefined Programs.”</a>
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 217.](#)
- **Send Message.** See [“Using the Send Message Program Type” on page 220.](#)
- **Message Broadcast.** See [“Using the Message Broadcast Program Type” on page 221.](#)
- **Run External.** See [“Using the Run External Programs Type” on page 223.](#)
- **Database Operation.** See [“Using the Database Operation Program Type” on page 224.](#)

---

**NOTE:** Before using the email or paging functions, you need to perform the setup procedures described in [Chapter 13, “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.

## Send Page Arguments and Values

[Table 53](#) 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 53. Send Page Program Type**

Argument	Valid Values When Used by Action
Numeric Message Template	Numeric message when pager is numeric.
Alpha Message Template	Text message when pager is alphanumeric. “Current” is a reserved word in Siebel Workflow. Do not use this word in messages.
Available Substitutions	Dynamic fields that you can use in the Alpha Message Template. When the action executes, the substitution value is populated with the value from the record that meets all the workflow policy conditions.
Request Key	A string indicating which Page Manager should execute the action. You use this when multiple Page Managers are running. When you specify a request key string, it should match the Request Key parameter of the Page Manager that you want to execute the action. Leave this argument blank when you are running one Page Manager or when the Page Manager that executes the action is not important.

When setting the Send Page arguments, note the following:

- Siebel Workflow Policies automatically determines the correctly formatted message depending on what type of pager the person being paged has.

- If neither of the message arguments has a value, Workflow Policies logs an error message and the action is not completed.
- You can send only pages to employees. The pager information for an employee is stored in the Employee Administration view. The Siebel database currently does not store pager information for contacts.
- Messages support substitution of values that come from the Available Substitutions field.

## Using the Send Message Program Type

When you select the Send Email workflow policy program in the Actions applet, the Send Message Arguments applet displays along with the Recipients applet.

The Send Message Arguments applet allows you to create an email template used to build the message sent to the recipient specified in the Recipients applet.

## Send Email Arguments and Values

[Table 54](#) shows the arguments and valid values for the Send Email workflow policy program type.

**Table 54. Send Email Workflow Policy Program Type**

Argument	Valid Values When Used by Action
Subject	Subject line of email message.
Message Template	Text of message. Maximum length is 2000 characters, including variable substitutions. “Current” is a reserved word in Siebel Workflow. Do not use this word in a message.
Repeating Message	Message that is repeated when the Consolidate flag is checked on the Workflow Policies Policies view. “Current” is a reserved word in Siebel Workflow. Do not use this word in messages.
Available Substitutions	Dynamic fields that you can use in Subject, Message Template, and Repeating Message. When the action executes, the substitution value is populated with the value from the record that meets all the policy conditions.
Request Key	A string indicating which Email Manager should execute the action. You use this when multiple Email Managers are running. When you specify a request key string, it should match the Request Key parameter of the Email Manager that you want to execute the action. Leave this argument blank when you are running one Email Manager or when the Email Manager that executes the action is not important.

## Using the Message Broadcast Program Type

The Message Broadcast Arguments applet appears if the Send Message Broadcast workflow policy program is selected in the Actions applet.

Figure 1 shows the Send Message Broadcast Arguments applet.

**Figure 1. Send Message Broadcast Arguments Applet**

## Send Message Broadcast Arguments and Values

Table 55 shows the arguments and valid values for the Message Broadcast workflow policy program type.

**Table 55. 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 <a href="#">“Entering Date Calculations” on page 248</a> for more information.
Expiration	Date and time when the message broadcast expires. The variable CURRENT can be used when specifying the activation date. See <a href="#">“Entering Date Calculations” on page 248</a> 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.

**Table 55. Message Broadcast Workflow Policy Program Type**

Argument	Valid Values When Used by Action
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. An example of a Run External Program is described in [“Creating a Run External Program Action” on page 234](#).

### Run External Arguments and Values

[Table 56](#) shows the arguments and valid values for the Run External workflow policy program.

**Table 56. 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"><li>■ <b>Wait.</b> 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.</li><li>■ <b>No Wait.</b> Workflow Policies executes the external program in the background and then continues processing. The return code is not checked.</li></ul> <p>Note that 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.</p>
Available Substitutions	Dynamic fields that can be used as command line parameters. When the action executes, the substitution value is populated with the value from the record in violation.

If no path is supplied for the Executable Name argument, the executable is assumed to be in the current path of Workflow Policies running on the Siebel Server. For example, your Siebel Server may be installed on C:\siebsrvr. The default path for the executable name would be C:\siebsrvr\bin.

---

**NOTE:** The external program cannot be one that is interactive, requires a user interface, or accesses the Windows desktop.

---

### Using the Database Operation Program Type

Siebel Business Process Designer has a number of database operation programs already predefined. All you need to define are the parameters.

The Arguments applet appears if you select a database operation program such as Create Opportunity Activity in the Actions applet.

#### Database Operation Arguments and Values

Table 57 shows the arguments and valid values for the Database Operation workflow policy program.

**Table 57. 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 58 describes the fields in the Recipients applet.

**Table 58. 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"> <li>■ <b>Send to Employee.</b> Picklist of employees.</li> <li>■ <b>Send to Position.</b> Picklist of positions.</li> <li>■ <b>Send to Contact.</b> Picklist of contacts.</li> <li>■ <b>Send to Relative.</b> 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).</li> <li>■ <b>Send to Address.</b> Represents a direct email address for programs that send email.</li> </ul>
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).

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 Business Process 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 217](#) 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.

---

## 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 at least 20.

### **To create a Message Broadcast Action for open service requests**

- 1** In the Workflow Policies Actions view, fill in the Actions applet fields.
  - a** Create a new record in the Actions applet and enter the name of the action:  
`Alert Agent of Open SRs`
  - b** Select a predefined workflow policy program from the Program field picklist:  
`Send SR Message Broadcast`
  - c** Select a predefined workflow policy object from the Workflow Object field picklist:  
`Service Request`
- 2** Complete the Send Message Broadcast Arguments form using message arguments and typing in static text.
  - a** Enter text in Abstract:  
`You have over 20 Service Requests.`
  - b** Enter text in Message Template:  
`You have over 20 service requests. Please review your service request queue.`
- 3** Fill in the Recipients applet.
  - a** Select a predefined Recipient Type from the Recipient Type field picklist:  
`Send to Relative`
  - b** Select the Recipient Name from the Recipient Name picklist:  
`SR Owner`



## 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 Siebel Workflow whenever the parameters for calculating the lead change.

In the first of the following examples, a program named “leadcalc.exe” is in the C:\bin directory and the action is being defined to call and execute this program. The second example provides the procedure for running external programs on UNIX.

### **To run an external lead calculation program**

- 1** In the Workflow Policies Actions view, fill in the Actions applet fields as follows.

- a** Create a new record in the Actions applet and enter the name of the action:

Run Lead Calculation Program

- b** Select a predefined workflow policy program from the Program field picklist:

Run External Program

- c** Select a predefined workflow policy object from the Workflow Object field picklist.

---

**NOTE:** The workflow object field fills in automatically only when a workflow policy object is specified in the workflow policy program being selected. You pick a workflow policy object from the picklist when it does not automatically fill in.

---

**2** Fill in the Run External Program Arguments applet.

- a** Enter the name of the executable:

`leadcalc.exe`

- b** Enter any command line parameters.

These are the parameters you want to pass to the executable.

- c** Select an execute type.

- d** Select dynamic fields from Available Substitutions.

**3** Fill in the Recipients applet.

- a** Select a predefined recipient type from the Recipient Type field picklist:

`Send to Position`

- b** Select Recipient Name from the Recipient Name picklist:

`Support Manager`

This action is now available to use in a workflow policy.

#### **To run an external program on a UNIX platform**

The Run External Program workflow policy program is not supported on UNIX. However, you can use the following procedure as a workaround.

- 1** Define a business service that executes an external program.
  - a** From the application-level menu, choose View > Site Map > Business Service Administration > Business Service Methods.
  - b** Add a new Business Service, for example, Run Program.
  - c** Add a new Method, for example, Run.
  - d** Add a new Method Argument, for example, Program.
  - e** Select Proc: Service\_PreInvokeMethod.
  - f** Call Clib.system in the function body, for example:

```
var program = Inputs.GetProperty ("Program")
if (program)
{
Clib.system(program);
}
return (CancelOperation);
```
- 2** Create a workflow process calling the business service created in step 1.
  - a** Add and connect a Start step, a Business Service step, and an End step.
  - b** For the Business Service step, specify Run Program and Run.
  - c** For the input argument for Program, specify the external program you want to run. For example, /bin/mail hkim@pcs.com </home/users/hkim/letter.txt.
- 3** Run your workflow process.

## 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 16, “Monitoring and Tuning Performance.”](#)

## Understanding the Group Applet

[Table 59](#) describes the fields in the Workflow Groups applet.

**Table 59. 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 60 describes the fields in the Policies applet of the Workflow Policies Groups view.

**Table 60. 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 view to complete your workflow policy creation.

You create a new workflow policy after you create the policy action and the policy group.

The Workflow Policies view is made up of four applets:

- **Policies List applet.** Where you enter and view information about the workflow policy. The entry applet toggles with a list applet so that you can quickly move between working on an individual policy and viewing information about several policies or groups of policies.
- **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 List Applet

Table 61 defines the Policies List applet fields.

**Table 61. 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.

**Table 61. Policies Applet Fields**

Field	Description
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	<p>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.</p> <p>When this field is checked, run Workflow Monitor Agent with the Batch Mode flag set to TRUE.</p> <p>The default is unchecked.</p>

**NOTE:** Before you can move a Workflow Policy from one group to another group, all requests associated with that Workflow Policy must complete. If a Workflow Policy's group is changed while associated requests are pending, the Workflow Monitor Agent will fail with the error Rule Not Found. If this occurs, restore the Workflow Policy to its original group, wait for the requests to complete, and then proceed with the change.

## Understanding the Conditions Applet

[Table 62](#) defines the Conditions applet fields in the Workflow Policies Policies view.

**Table 62. 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 <a href="#">“Using Comparison Values in the Conditions Applet.”</a>
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 <a href="#">“Using Comparison Values in the Conditions Applet”</a> and <a href="#">“Entering Date Calculations” on page 248.</a>

### Using Comparison Values in the Conditions Applet

You use comparison values in the Operation field. The field exposes the Workflow policy component column for monitoring.

### Standard Comparisons

The Comparison field supports <, >, < >, > =, < =, =, LIKE, IN, NOT IN, BETWEEN, IS NULL, and IS NOT NULL operators. An 'AND' is implied between multiple conditions defined using these comparison values. 'AND' means that all conditions must be met before the action occurs.

When you specify values for the comparison operands LIKE, IN, NOT IN, and BETWEEN in the Value field of the Conditions applet of the Workflow Policies Policies view, it must be in a form that the underlying database expects. IN, NOT

IN, and BETWEEN require you to enter the database specific format for the field being examined, for example, IN ('a', 'b', 'c') or IN (1, 2, 3, 4) and BETWEEN 'A' and 'M' or BETWEEN 1 and 10.

---

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

---

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

Table 63 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 63. 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 64 describes the specialized comparisons for all database platforms that can be used in creating workflow conditions.

**Table 64. 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

Workflow Monitor considers both date and time when evaluating Workflow Policy conditions that perform a date comparison. CURRENT can be used when entering a value for a date comparison. The format for using CURRENT is CURRENT + /- d:h:m where “d” is day, “h” is hours, and “m” is minutes. You can use CURRENT in the comparison value for date fields. You can also use CURRENT when you specify the activation and expiration dates for a message broadcast action.

## Understanding the Actions Applet

[Table 65](#) defines the Actions applet fields on the Workflow Policies Policies view.

**Table 65. 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.



**Table 65. Actions Applet Fields**

Field	Description
Relative	The relative type when the recipient of the action is determined by the workflow object, for example, service request owner.
Consolidate Flag	Consolidates the action to one instance if more than one record meets all the conditions of the workflow policy during the same action interval. Default is FALSE. The consolidate flag is unavailable with actions that send pages.

Many of the choices in the fields in the Workflow Policies Policies view are predefined in other Siebel Business Process Designer views, either in Siebel Client or in Siebel Tools. You can modify the predefined choices or create new choices for these fields. These choices show up as a picklist in the Workflow Policies Policies view applets.

## Creating a Workflow Policy

### **To create a Workflow policy**

- 1 Navigate to Business Process 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 13, “Workflow Policies Server Administration.”](#)

---

## Creating Workflow Policies

*Siebel Workflow Policy Examples*

# **Workflow Policies and Siebel Marketing** **11**

This chapter describes workflow policy programs for campaign execution.

# Using Workflow Policy Programs for Campaign Execution

The Workflow Policy programs in Siebel Marketing were designed to allow a marketer to create complex campaign policies to automate the different stages of the campaign. Actions are based on the type of workflow policy program used and are used by Workflow Policies to create campaign policies.

Three workflow policy programs are designed for creating actions to execute campaigns:

- **Send Campaign Email.** Sends email to all contacts and prospects associated with a campaign.
- **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 Contacts/Prospects applet in the Campaign Administration 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 66](#) describes valid values for the Arguments applet.

**Table 66. 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 67](#) describes the value for the New Campaign argument.

**Table 67. 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 the Group field in the Workflow Policies Policies view.

## **Creating the Policies**

Once the workflow policy actions and the workflow policy group are ready, the policies can be created. Two policies are required in this scenario:

- Email for CD-ROM campaign—to trigger the sending of the offer email and the email activity record.
- Assign Non-Respondents—to trigger the reassignment of nonrespondents to a new campaign.

In the procedures below for creating the policies, it is important to note how the fields in the Conditions applet are set.

### ***To create the Email for CD-ROM Campaign policy***

- 1** Fill out the Policies Applet in the Workflow Policies Policies view.

- 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.

# **Customizing Workflow Policies with Siebel Tools 12**

This chapter describes how to customize workflow processes with Siebel Tools.

## 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 Reference*.

---

**NOTE:** When you use Siebel Tools to modify or create workflow policy objects on your local system, the changes are not available on the server until they are applied to the server.

---

Siebel Tools consists of an Object Explorer window and one or more Object List Editor windows. [Figure 2 on page 265](#) shows the Object Explorer and Object List Editor window. The Object List Editor window lists object definitions for each object type and allows you to edit object type properties. The Object Explorer provides navigation between each group of object definitions of a particular object type.

---

**NOTE:** Workflow policy objects are not included in the Object Explorer by default. Click View > Options > Object Explorer to add the workflow policy objects to the Object Explorer view.

---

*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.

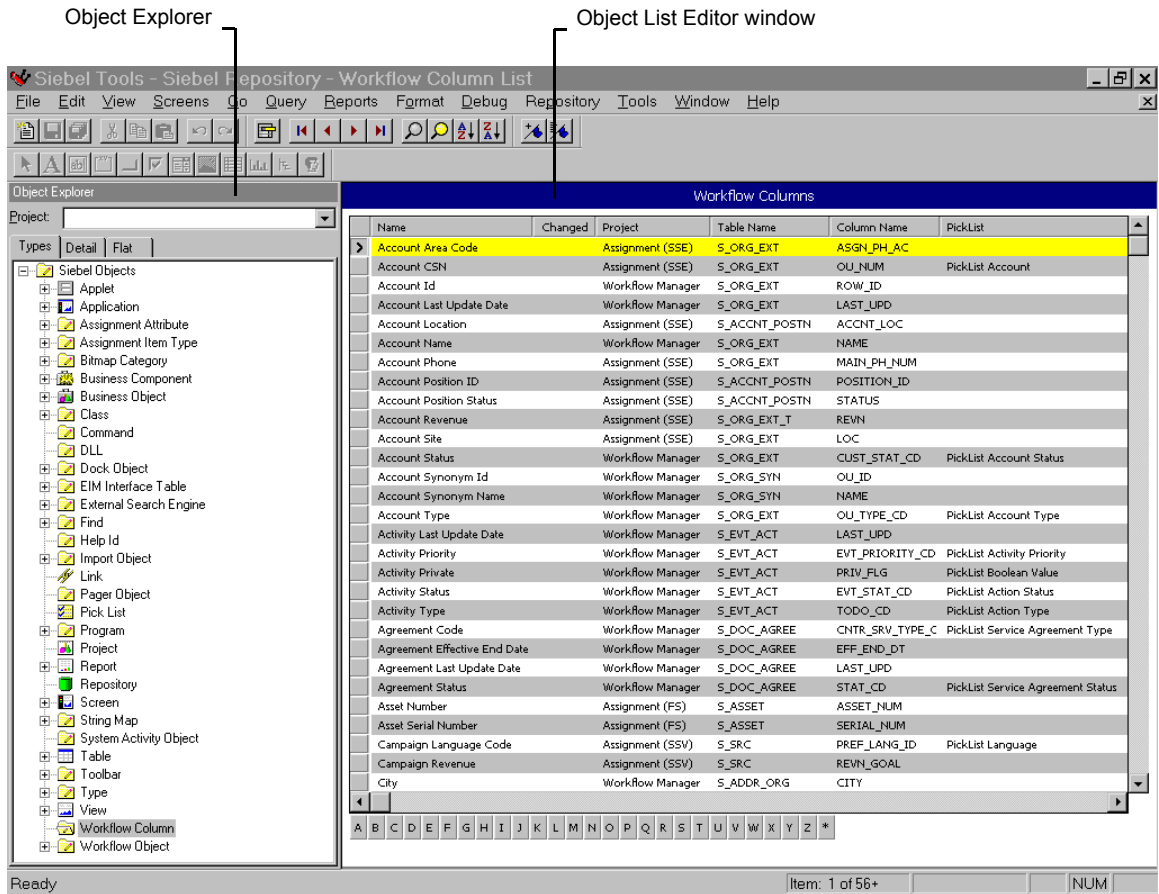
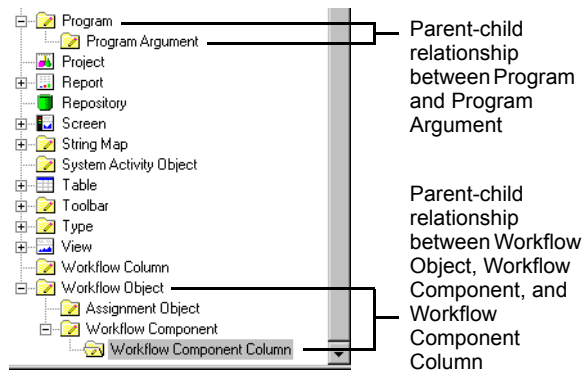


Figure 2. Object Explorer and Object List Editor Window

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



**Figure 3. 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 policies 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. The Workflow policy objects in Siebel Tools are exposed in the Workflow Object picklist in Workflow Policies.

The workflow policy component columns defined in Siebel Tools are available to Workflow Policies views. The workflow policy component columns in Siebel Tools are exposed in Workflow Policies in the Condition Field picklist in the Workflow Policies Policies view.

The policy programs defined in Siebel Tools are available in Workflow Policies views. The policy programs in Siebel Tools are exposed in Workflow Policies in the Program Field picklist in the Workflow Policies Actions view.

You use Siebel Tools to configure or create custom workflow policy objects and custom policy programs. The use of Siebel Tools is described in *Siebel Tools Reference*. 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.

---

**CAUTION:** You should not try to monitor Enterprise Integration Manager (EIM) table columns. To recognize EIM tables, look for table names that begin with EIM\_ or end with \_IF.

---

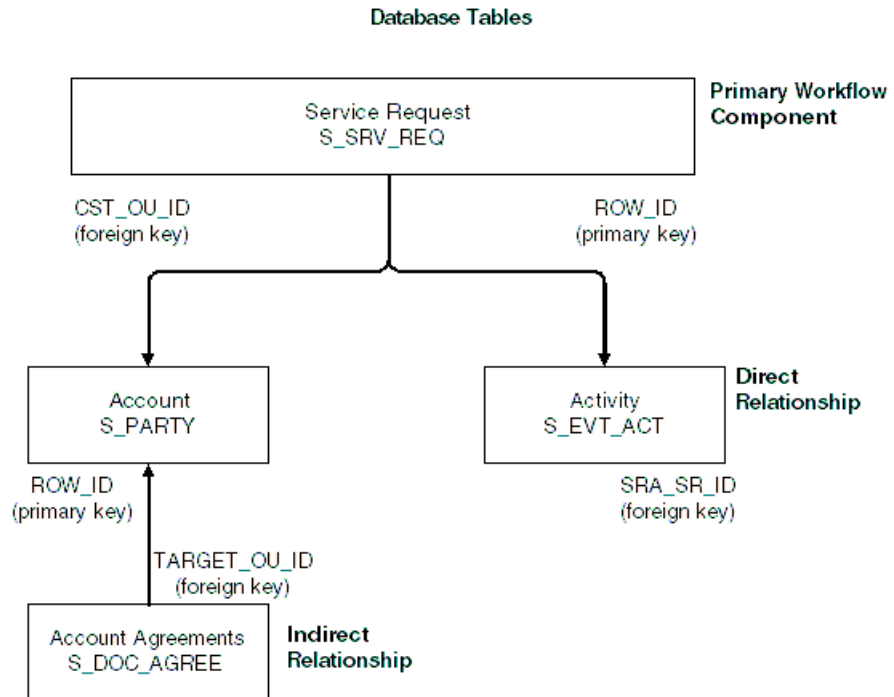
Creating a workflow policy object consists of four main steps:

- 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 4 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 4. Relationships of Workflow Policy Objects and Workflow Policy Component**

#### **Relationship Between Workflow Policy Object and Workflow Policy Component**

If the Service Request has the primary field checked, then it is the primary component. All the other components in the list are the nonprimary components of the Service Request workflow policy object.

## 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. You must activate extension columns in Siebel Tools in order to make them available for use in workflow database operations.
- **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 nonprimary 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

Table 68 describes the fields in the Workflow Policy Columns List view.

**Table 68. 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"> <li>■ Consistent with your overall naming strategy.</li> <li>■ Meaningful to the policy maker.</li> <li>■ Descriptive of how the column is used.</li> </ul>
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 Reference</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.



**Table 68. Workflow Policy Columns Applet Fields**

Field	Description	Comments
Inactive	Determines if this column is active or inactive. If column is inactive, the column is not compiled when you compile your .srf and is not accessible by any object.	A check mark indicates this is inactive and is not compiled or accessible.
Comments	Comments describing the purpose or use of column.	Any text.

## Configuring a Workflow Condition Based on a Foreign Key

You can configure a workflow condition that is based on a foreign key that exists in the primary table of the workflow object. For example, SOPTY.CURR\_STG\_ID, where S\_OPTY is the primary table of the Opportunity workflow object, and CURR\_STG\_ID is a foreign key from S\_STG.NAME.

### ***To configure a workflow condition based on a foreign key existing on the primary table***

- 1** Create a new Workflow column, S\_STG.NAME.
- 2** Make sure CURR\_STG\_ID is added under the Opportunity workflow component.
- 3** Create a new workflow component in the Opportunity workflow object based on S\_STG table:
  - Name = your choice
  - Source Table Name = S\_STG
  - Source Column Name = ROW\_ID
  - Target Component Name = Opportunity
  - Target Column Name = CURR\_STG\_ID
- 4** Add the new workflow column S\_STG.NAME (from Step 1) to the new workflow component.

You can now create a workflow condition that is based on the new workflow column.

## Understanding the Workflow Policy Object List View

[Table 69](#) describes the properties for a workflow policy object.

**Table 69. Workflow Policy Object Properties Fields**

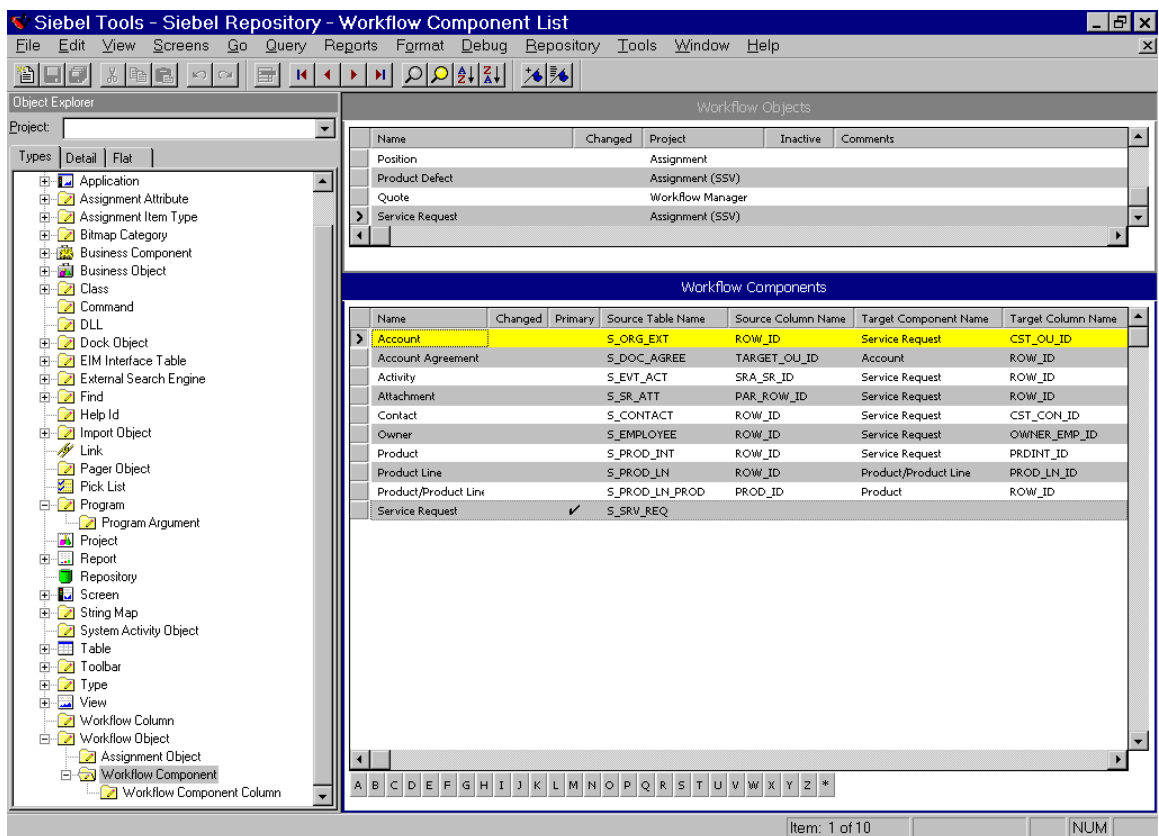
Field	Description	Comments
Name	The name of the workflow policy object.	A descriptive name that is: <ul style="list-style-type: none"> <li>■ Consistent with your overall naming strategy.</li> <li>■ Meaningful to the policy maker.</li> </ul>
Changed	The identifier for whether the record has been added 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. [Figure 5 on page 275](#) shows the Workflow Policy Component list view. Except for the primary workflow policy component, each workflow policy component defines a relationship to another workflow policy component. This relationship is defined by specifying a source policy column and a target policy column. The source and target columns on a workflow policy component identify foreign key relationships between the tables.

A *primary workflow policy component* is a workflow policy component that all other workflow policy components are directly or indirectly related to. From these workflow policy components, the workflow policy columns that are available for monitoring in the workflow policy can be defined.

To define a workflow policy object and its components, you should be familiar with the Siebel Data Model and *Siebel Data Model Reference*. *Siebel Data Model Reference* describes the tables and how the tables are related.



**Figure 5. Workflow Policy Component List View**

Table 70 describes all of the properties of a workflow policy component.

**Table 70. 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"><li>■ Consistent with your overall naming strategy.</li><li>■ Meaningful to the policy maker.</li></ul>
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.  Note: Each workflow policy object must have only one primary workflow policy component.
Source Table Name	The table that the workflow policy component is based on.	A table name from the picklist.
Source Column Name	The column in the source table that relates to another workflow policy component.	A picklist of columns from the table specified in the Source Table Name field. (Not required for the primary workflow policy component.)
Target Component Name	The target workflow policy component that this workflow policy component is related to.	A table name from the picklist. (Not required for the primary workflow policy component.)
Target Column Name	The column in the target workflow policy component that the source column in this workflow policy component is joined to.	A picklist of columns from the workflow policy component specified in the Target Component Name field. (Not required for the primary workflow policy component.)
Inactive	Indicates if the component is active or inactive.	A check mark indicates this field is inactive and is not compiled or accessible.  If the component is inactive, it is not compiled when you compile your .srf and is not accessible by any policy.
Comments	Any comments for the workflow policy component.	Descriptive text.

## 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.

[Table 71](#) describes the workflow policy component columns values.

**Table 71. Workflow Policy Component Columns Properties Fields**

Field	Description	Comment
Workflow Column Name	The name of the column defined in the workflow Policy Component Column view.	A picklist of all columns that were defined in the Workflow Policy Column view for the table that the workflow policy component is based on.
Alias	The name of the column as it appears in the Conditions Field picklist in the Workflow Policies Policies view.	The default is the workflow policy column name. A descriptive name that is: <ul style="list-style-type: none"><li>■ Consistent with your overall naming strategy.</li><li>■ Meaningful to the policy maker.</li><li>■ Descriptive of how the column is used.</li></ul>
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.

## 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, right-click your mouse.

Selecting Validate brings up the Validate screen. Clicking the Start button runs the validation process and returns information either as a caution or as error messages that appear in the Details box.

## Modifying an Existing Workflow Policy Object

When defining the types of workflow policies you need to operate your business, you may find that the predefined workflow policy objects do not contain the policy components you need. Use the procedural steps in this section as a general guideline for modifying a workflow policy object.

Before modifying a workflow policy object:

- Find out the name of its database table and column names. If you are going to add or modify a component, you need to know the relationship between the component and the primary workflow policy component.
- Make sure you do not have other records referencing this object that may be affected by your change. For example, before inactivating a component column, verify that no policy conditions are referencing the component column.

### **To determine a database table**

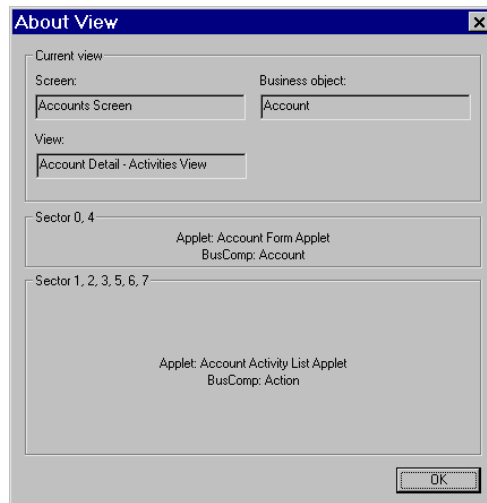
- 1 Start Siebel by entering the following from the command line:

```
C:\Siebel\bin\siebel.exe /x
```

- 2 In Siebel Client, navigate to the appropriate workflow policy object view. This is the view that contains the business data you want to monitor.

For example, if you need to modify the workflow for an account activity, you would navigate to the Account > Activities view.

- 3 Choose Help > About View.



About View identifies the Business object, Business components, and applets this view uses.

In the case of the Account Activities view, the dialog box identifies Action as the business component used by the Activities applet.

- 4 In Siebel Tools, select Business Component in the Object Explorer and find the appropriate component name.
- 5 Select the component (for example, Action) and find the table name. In the illustration above, the table name is S\_EVT\_ACT.

You use this table name when you create a workflow policy component.

#### ***To determine the relationship between a component and the primary workflow policy component***

- 1 In Siebel Client, navigate to the appropriate workflow policy object view.
- 2 Choose Help > About View.
- 3 Find the business component for the appropriate applet and the business object this view uses.

- 4 In Siebel Tools, select Business Object in the Object Explorer and search for the business component object name you noted in [Step 3 on page 284](#).
- 5 In Object Explorer, expand Business Object to Business Object Component and select the appropriate Business Object Component.

The attribute in the Link field identifies the link defining the relationship between the account and action business components.

- 6 In Object Explorer, select Link and find the applet/object link. The illustration below shows the Account/Action Link selected.

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 Summa	Id	Account ID
Account (SME)/Custc		Customer Briefing	Account (SME)	Customer Sales Performance	Id	Account ID
Account (SME)/Custc		Customer Briefing	Account (SME)	Customer Sales by Product I	Id	Account ID
Account (SME)/Custc		Customer Briefing	Account (SME)	Customer Top Products & R	Id	Account ID
Account (SME)/Oppc		SME Customer	Account (SME)	Opportunity Product Star	Id	Account ID
Account (SME)/Oppc		Customer Briefing	Account (SME)	Opportunity Star	Id	Account ID
Account (SME)/Oppc		Oppl (SME)	Account (SME)	Oppl 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/Ar		User Categories	Account Category	Account	Account Id	Id
Account Category/Ar		User Categories	Account Category	Position	Account Id	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 Type
Account Type (SME)		Oppl (SME)	Account Type (SME)	Oppl Product by Account	Account Type	Account Type
Account/Account		Link	Account	Account	Parent Acco	Account Id
Account/Account Att		Account	Account	Account Attachment	Id	Account Id
Account/Account Cal		User Categories	Account	Account Category	Id	Account Id
Account/Account Ext		Link (SSV)	Account	Account External Product	AEP Account	Account Id
Account/Account No		Account	Account	Account Note	Account Id	Account Id
Account/Account Pri		Account	Account	Account Private Note	Account Id	Account Id
Account/Account Sys		Link	Account	Account Synonym	Account Id	Account Id
Account/Action		Link	Account	Account	Account Id	Account Id
Account/Activity Plan		SA Activity Templa	Account	Activity Plan	Id	Account Id
Account/Agreement		Srvagres	Account	Agreement	Account Id	Account Id
Account/Asset Mgmt		Asset Management	Account	Asset Mgmt - Asset	Owner Acco	Account Id
Account/Assignment		Link	Account	Assignment Group	Account Id	Account Id
Account/Business Ad		Link	Account	Business Address	Account Id	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.

## Modifying an Existing Workflow Policy Object

- 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.

The screenshot displays the Siebel Tools interface. On the left is the Explorer pane showing a tree view of Siebel Objects. The 'Business Component' is expanded, and 'Field' is selected. On the right, there are two panes. The top pane, 'Business Components', shows a table with columns: Name, Changed, Project, Cache Data, Class, and Data Source. The bottom pane, 'Fields', shows a table with columns: Name, Changed, Calculated, Calculated Value, Join, Column, and Currency Code Field. The 'Account Id' field is selected in the Fields pane, and its 'Column' attribute is 'TARGET\_OU\_ID'.

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					ACCTN_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

[Table 72](#) describes the Program properties fields.

**Table 72. 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"><li>■ <b>DB Operation.</b> Insert or update a database table based on arguments.</li><li>■ <b>External Program.</b> Execute an external program in Windows.</li><li>■ <b>Send Message.</b> Compose and send an automatic email message.</li><li>■ <b>Send Page.</b> Send a page to a pager.</li><li>■ <b>Send Message Broadcast.</b> Send a message broadcast to a group of users.</li></ul>
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. The argument fields that display in this view depend on the type of workflow policy program you select. A workflow policy program argument is a child process of a workflow policy program.

[Table 73](#) shows the Workflow Policy Program Argument properties values.

**Table 73. 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 74](#).

**Table 74. Valid Database Operations Workflow Policy Program Argument Values**

Name	Description	Allowable Default Value
Primary ID	The row ID of the violating row that the Workflow policy program is acting on.	Should be empty.
Primary Table	The base table to which the action is applied. The base table can be unrelated to the record of the primary ID. Examples include: The violating row is in a child table and 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 74. Valid Database Operations Workflow Policy Program Argument Values**

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 75](#) describes program arguments specific to the Send Message program.

**Table 75. Send Message Program Argument Properties**

Name	Description	Value
Email Message	Body of the email message.	Any text with available substitutions.
Email Message Repeated	Text that is repeated when the Consolidate feature is used.	Any text with available substitutions.
Email Subject	Text in subject line of the email message.	Any text.
Send to Contact	All contacts available in Siebel.	
Send to Position	List of the positions available in Siebel.	
Send to Employee	List of all employees available in Siebel.	

## Send Page Program Arguments

[Table 76](#) describes the Program Arguments particular to the Send Page program.

**Table 76. 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.	

**Table 76. Send Page Program Argument Properties**

Name	Description	Value
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 77](#) describes the Program Arguments particular to the Run External Program program.

**Table 77. 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 74 on page 290](#). 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 Workflow Policy Program > Send Opportunity Email > Workflow Policy Program applet.

---

**NOTE:** To create a new workflow policy program argument for Send Opportunity Email, check the existing program arguments and make sure that the Send to Relative program argument exists.

---

- 2 Create a new record, `Primary Contact`.

---

**NOTE:** When creating new program arguments, they cannot have the same name as a SQL Statement Output or the Workflow Monitor Agent server task will hang with the message “Examining request for policy...” when inserting a record.

---

- 3 Bring down the box under Default Value and create your SQL statement. For example:

```
select O.PR_CON_ID, 'Send to Contact'
from &Table_Owner.S_OPTY O
where O.ROW_ID=?
```

---

**NOTE:** SQL statements are database vendor-specific. Use an external SQL tool to build and test your statements. When the test works, copy the statement into the field.

---

- 4 Select Workflow Relative Type Picklist in the PickList field.

This picklist describes this argument as a relative.

The Visible field is checked. The changed field becomes checked when you create a new program argument.

## Creating SQL Statements for Program Arguments

Before you begin to create the Recipient Type, Send to Relative, you must create a SQL statement in the Workflow Policy Program applet in Siebel Tools.

SQL statements written for Workflow Policies Program Arguments must have the following characteristics:

- The table name and column name you reference must be uppercase.
- The case-sensitive table name should be prefixed with:  
`&Table_Owner.`
- The SQL statement must be valid for the specific database vendor being used.



## 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

Using this program, you can define a policy such that if a Service Request has an activity of type Resolution, and the SR is open for more than five days, the SR close date is changed to today's date.

When the policy triggers the workflow policy program, the program enters the current system date into the Close Date field of the Service Request record. [Table 78](#) shows the arguments for the Change SR Close Date to Today program.

**Table 78. Change SR Close Date to Today Program Arguments**

Argument Name	Comment
Primary ID	Contains the row ID of the Service Request record meeting the policy condition.
Primary Table Operation Type	Specifies the table (S_SRV_REQ) and what action is to take place (Update).

**Table 78. Change SR Close Date to Today Program Arguments**

Argument Name	Comment
Sql Statement	<pre>select {fn now()} from &amp;Table_Owner.S_DUAL</pre> <p>This statement calls the Siebel function now() to obtain the current date and uses the table &amp;Table_Owner.S_DUAL to hold the value temporarily. The S_DUAL table is used to hold temporary values.</p> <p>Math functions can also be performed. For example, SQL Statement = select {fn now()} + 7 from &amp;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” variable obtains its value from the SQL statement.
New Close Date (Column)	Specifies the column in the record to be updated (ACT_CLOSE_DT).
New Close Date	Specifies the field to be updated to the value of Today.
Update Row ID	The row ID of the record you want to update. (The same as the value of the Primary ID.)

## 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 allow the right people to see the incoming service request and assign it appropriately.

This workflow policy program allows you to select a new owner from a picklist and put it into the field of the Service Request record matching the policy condition. [Table 79](#) shows the arguments for the Change SR Owner program.

**Table 79. 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 allow a proper response time to service calls.

This workflow policy program does the following:

- Uses the Primary ID as input into a SQL statement
- Uses a query SQL statement to retrieve the current value of the field Manager
- Sets the New Owner field to default to the current value of Manager
- Allows the end user to update the New Owner field optionally through a picklist

Table 80 shows the arguments for the Change SR Owner to Manager program.

**Table 80. Change SR Owner to Manager Program Arguments**

Argument Name	Comment
Primary ID	Contains the row ID of the Service Request record meeting the policy condition.
Primary Table Operation Type	Specifies the table (S_SRV_REQ) and what action is to take place (Update).
New Owner (Column)	Specifies the field in the record to be updated (Owner_EM_ID).
New Owner	Indicates that a picklist is to be displayed for assigning a new owner.

**Table 80. Change SR Owner to Manager Program Arguments**

Argument Name	Comment
Sql Statement	<pre>SELECT MGRPOS.PR_EMP_ID FROM &amp;TABLE_OWNER.S_POSTN POS, &amp;TABLE_OWNER.S_EMPLOYEE EMP, &amp;TABLE_OWNER.S_POSTN MGRPOS, &amp;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 contained in the workflow policy object, workflow policy components, and workflow policy columns. In working and coding workflow policy action programs using the Siebel tables, explicit joining of the base table through SQL code is required.</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.

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 13**

This chapter describes workflow policies server administration tasks.

## Creating Database Triggers

The Generate Trigger (GenTrig) component on the Siebel Server allows you to create database triggers. Workflow Policies uses database triggers to identify which records match policy conditions. You should run Generate Triggers when you:

- Create or delete new policies (including Assignment Policies), with the exception of Workflow Policies that have Batch Flag set to TRUE.
- Amend policy conditions or policy criteria.
- Change activation or expiration dates of policies, including Assignment Policies.

To run Generate Triggers, you must have installed Siebel Server, and the client you are using must be enabled to access the Siebel Server Administration screens. See the installation guide for the operating system you are using for more information on installing Server Manager.

---

**CAUTION:** If you have incorrectly defined a policy condition, running Generate Triggers may result in invalid triggers. An invalid trigger can prevent execution of normal user transactions. For this reason, 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.

---

**NOTE:** In this release, the Generate Triggers task now requires the Privileged User Name and Password instead of Table Owner ID and Password.

---

## Running Generate Triggers

When running Generate Triggers, remember the following tips, especially if you are deleting a policy:

- Deleting a policy and then running Generate Triggers does not remove the database trigger. When you delete a policy, you must run Generate Triggers with the remove parameter set to TRUE. This removes *all* triggers. You must then rerun Generate Triggers to reset the triggers for existing policies.
- You need to stop and restart the Workflow Monitor Agents when running Generate Triggers.
- Generate Triggers needs to be rerun whenever you change policy conditions. Generate Triggers does not need to be rerun when changing policy actions.
- For SQL Server, 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 Client, select View > Site Map > Server Administration > Enterprise Operations > Component Requests.

- 2** Click New.
- 3** Select Generate Triggers from the Component Job list. This creates a new line entry but does not start the task.
- 4** Click New in the Component Request Parameter Applet to modify parameter settings. The component-specific parameters for Generate Triggers are in [Table 81](#). See *Siebel Server Administration Guide* for a description of the generic and enterprise parameters.
- 5** Enter your Privileged User name and password.
- 6** Click the Component Request form applet menu, then select Submit Request.
- 7** To view changes to the state, refresh the screen by clicking Run Query from the applet menu.
- 8** Upon completion, the Status field contains either Success or Error. It is recommended that you view the log details. To do this, click Screen Enterprise Operations. In the Show field, select Tasks, then click the view tab called Task Info Log.

**Table 81. Component-Specific Parameters for Generate Triggers**

Name	Value	Description
Remove	TRUE or FALSE (default)	Set to TRUE to generate “DROP TRIGGER” statements to clean up the triggers. Remove does not generate “CREATE TRIGGER” statements.
Trigger File Name	Valid 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.

**Table 81. Component-Specific Parameters for Generate Triggers**

Name	Value	Description
EXEC	TRUE or FALSE (default)	<p>Determines if the SQL script file runs automatically or manually.</p> <p>If TRUE, the SQL script file runs automatically.</p> <p>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.</p> <p>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.</p>
Mode	ALL or WORK or ASGN	<p>Set to ALL to create both Workflow Policy triggers and Assignment Manager triggers.</p> <p>Set to WORK to create only Workflow Policy triggers.</p> <p>Set to ASGN to create only Assignment Manager triggers.</p>
Privileged User Name/ Privileged User Password	Assigned Privileged User name and password	All users must enter a Privileged User name and password. The Table Owner is considered a Privileged User, so you may enter the Table Owner name and password in the Privileged User name and password fields.

## Running the SQL Script File

Once Generate Triggers has completed, run the SQL script file if the EXEC parameter is FALSE.

### **To run the SQL script file**

- 1** Connect to the database server as the Siebel tableowner using your RDBMS vendor's SQL tool (for example, ISQL for Microsoft or SQL\*Plus for Oracle).
- 2** Run the SQL script file specified by the Trigger File Name parameter. The default 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 administrator, Bill Stevens, has finished creating policies in the test Siebel Client and wants the database triggers set in the Siebel database for the new policies. Using the Generate Triggers component, he sets the file output name.

This creates a file, TRIGGER.SQL, for the database administrator containing all the triggers that need to be modified or created in the test database for these policies.

The database administrator then runs the following command in SQL\*Plus to create the triggers in the Oracle database:

```
SQL>@<path>\mytrig.sql
```

The successful creation of each database trigger in the Oracle database is indicated on the screen. For information on the syntax required for other databases, refer to your database documentation.

---

**NOTE:** On an MS SQL server database, execute the script trigger.sql as the database owner (dbo) login for the Siebel database.

---

## **Database Triggers and Remote Users**

When a remote user synchronizes, the changes get incorporated into the database (for example, account information in the S\_ORG\_EXT table is updated on synchronization). If you run a workflow which creates database triggers that compare changes in the database against specific conditions, then the triggers will fire and rows get written to S\_ESCL\_REQ if the changes are of interest to the workflow conditions during synchronization.

## Setting Up the Siebel Server for Email Manager

Some workflow policy actions allow you to send email messages to specific individuals. To send email using Workflow Policies, an SMTP/POP3-compliant mail system must be working properly on your network and the Email Manager component of the Siebel Server must be running. You must also set the Mail Profile parameter to the name of the messaging profile you want to use for sending the email. You can use any SMTP/POP3-compliant client programs to verify your settings, including Microsoft Exchange or Microsoft Outlook.

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

### ***To find the account used to start Siebel Server***

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

For more information, see *Siebel Communications Server Administration Guide*.

## Setting Up the Communications Profile

Sending email through Workflow involves creating an SMTP/POP3 communications profile.

### ***To create a communications profile***

---

**NOTE:** In order to create a new communications profile, CompGrp “CommMgmt” must be enabled. Verify that it is enabled before beginning the steps below. For more information, see *Siebel Communications Server Administration Guide*.

---



- 1 Click View > Site Map > Communications Administration > Communications Drivers and Profiles.
- 2 In the Communications Drivers list applet, select the “Internet SMTP/POP3 Server” communications driver.
- 3 Click the Profiles tab.
- 4 Create a new profile called < Profile Name > .
- 5 Click the Driver Parameters tab, and complete the fields for the parameters described in [Table 82](#) in the Profile Parameters Overrides applet:

**Table 82. Driver Parameters for Creating a Communications Profile**

Parameter	Entry
From Address	< the sender’s email address for outbound communications >
POP3 Account Name	< the account name for the POP3 mailbox from which to retrieve inbound communications >
POP3 Account Password	< the password for the POP3 mailbox account >
POP3 Server	< the host name or IP address of the machine on which the Internet POP3 server is running, as appropriate for your network configuration >
SMTP Server	< the host name or IP address of the machine on which the Internet SMTP server is running, as appropriate for your network configuration >

**NOTE:** For details on the parameters listed in [Table 82](#), see *Siebel Communications Server Administration Guide*.

After creating your communications profile, you need to create the component definition of Email Manager. The Email Manager component executes email actions once the conditions of a Workflow policy are met.

### Starting Email Manager

You can start Email Manager from the command line, or from the Email Manager Component view.

#### **To start Email Manager from the command line**

To start the Email Manager task with the <Profile Name> profile, in the server manager command line, use the following command to start a task for Email Manager:

```
start task for comp MailMgr with MailProfile=<Profile Name>
```

#### **To start Email Manager from the Server Administration view**

- 1 Click View > Site Map > Server Administration > Components > Component Parameters.
- 2 Find the Email Manager component.
- 3 In the Component Parameter list, set Mail Profile = Test.
- 4 If you want to automatically start up a task for Email Manager whenever the component is restarted or the Siebel server services are restarted, set the component parameter Default Tasks = 1.

When the workflow policy is violated, Workflow Monitor Agent will insert a record into the S\_APSRVR\_REQ table for workflow actions that invoke any of the Send Email workflow policy programs. Email Manager will then pick up records from the S\_APSRVR\_REQ table, setting their status from QUEUED to ACTIVE and then to SUCCEEDED during the course of the execution. Subsequently, Outbound Communications Manager will be invoked to log onto the <Profile Name> profile and send the emails out to the recipients using the Outbound Communications Manager business service, Send Message method.

The most important parameters for the Email Manager are Sleep Time, Mail Password, and Mail Profile. [Table 83](#) describes parameters for the Email Manager.

**Table 83. Parameters for the Email Manager Component**

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 parsed by the component tasks is logged to the information log file for each task. If this parameter is set to 2 (recommended for debugging SQL), each executed SQL statement is logged.
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>
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	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.

# 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** Click View > Site Map > Server Administration > Enterprise Operations.
- 2** In the Component Requests view, click New.  
An empty field appears under Component.
- 3** From the Component/Job picklist, select the Page Manager component.
- 4** In the Component Request Parameters applet, click New.
- 5** Click on Parameters in the Server Tasks applet and enter your parameters.

See [Table 84](#) 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 84](#) are used.

**Table 84. Parameters for the Page Manager Component**

Parameter	Value
Modem Port	The component object model (COM) port where the modem is attached. Valid values are COM1, COM2, and so on. Default = COM1
Dial Prefix	A number or sequence of numbers that needs to be dialed to get an outside line. If no dial-out prefix is use, insert a “,” (comma). Default = 9  Note that when dialing 9 for an outside line is not required and you are using srvrmgr.exe from the command line, specifying a comma for this parameter returns an error. But, if you set it to a hyphen (or any other non-dialable character) it will work correctly. Example command:  SRVRMGR.EXE /g NT01022 /e SBLPRD_ENT502 /s SBLPRD_APP502 /u ***** /p ***** /c "START TASK FOR COMPONENT PageMgr WITH DialPrefix = '- ' "
Long Distance Prefix	The long-distance prefix. This value is added in front of all long-distance phone numbers. Set this parameter to equal an empty string if your location does not require a long-distance prefix to be dialed. Default = 1

**Table 84. Parameters for the Page Manager Component**

Parameter	Value
Local Area Code	The area code of your location. If the beginning digits of a phone number are equal to this code, they are removed before the phone number is dialed, and the long-distance prefix will not be added. Default = < empty >
Delay1	The number of seconds to wait between dialing a phone number and simulating key presses for the first set of numbers. This applies only to numeric paging. It is ignored for alphanumeric paging. Default = 12
Delay2	The number of seconds to wait between simulating key presses for the first and second set of numbers. This applies only to numeric paging. It is ignored for alphanumeric paging. This is also ignored if the numeric pager does not have a personal identification number (PIN). Default = 4
Modem Reset String	A modem command used to reset the modem. Default = ATZ Refer to your modem documentation for the correct command.
Modem Init String	A modem command used to initialize the modem. Default = AT&FQ0V1 Refer to your modem documentation for the correct command. For example, some modems require a numeric value after &F.
Modem Dial String	A modem command used to dial the modem. You should rarely need to change this parameter. Default = ATDT
Modem Hangup String	A modem command used to hang up the modem. You should rarely need to change this parameter. Default = ATH
Modem Restore String	A modem command used to restore the power-up settings of the modem. You should rarely need to change this parameter. Default = AT&F
Request Key	When you have more than one Page Manager, the request key serves as the ID for each Page Manager. You can then specify this key for the Workflow action in the Workflow Action Argument view. The request key can be any string.

## Troubleshooting the Email and Page Managers

Email Manager stops processing when it is unable to log on to the mail server (SMTP/POP3-compliant server) and logs an error message in the trace files.

Page Manager stops processing if the modem is not available. The requests continue to accumulate in the Requests table. After you fix your processing problems, you must restart the servers. The servers will continue processing from where they left off.

If Email Manager is able to log on but has a problem sending a particular email, it logs an error message and continues on to the next request. If Page Manager is able to interface with the modem but has a problem with a given page send, it will log an error and move on to the next request.

When Workflow Policies executes email and paging actions, it actually inserts email requests and paging requests into the database. These requests are inserted as records in the S\_APSRVR\_REQ table, which are then processed by Email Manager and Page Manager.

New requests have a status of "QUEUED." After a request is picked up by Email Manager or Page Manager, but before it is processed, it has a status of "ACTIVE." After a request is processed, its status becomes SUCCEEDED if the processing is successful, or FAILED if an error occurs.

To generate the sending emails, Siebel Server uses the UNIX "Mail" command. To verify that your server platform can run the command to a valid recipient and verify the email was successfully sent, do the following:

- 6** From the UNIX command prompt type:

```
> mail recipient email address
```

where *recipient email address* is a valid address.

- 7** 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.



## Working with Workflow Monitor Agent

To execute your Workflow policies, you need to start Workflow Monitor Agent. Workflow Monitor Agent checks when the conditions of policies are met, and executes actions once those conditions are met.

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

[Table 85](#) describes Workflow policies database tables.

**Table 85. Workflow Policies Database Tables**

Table	Description
S_ESCL_REQ	This table holds the potential matching requests caused by applications.
S_ESCL_STATE	This table holds the time-based policy matches.
S_ESCL_ACTN_REQ	Optional. This table holds the requests to execute actions. This is only used if you use Action Agent = TRUE.
S_ESCL_LOG	This table holds a history of base table rows that have matched policies.

Workflow Monitor Agent performs several server processes that monitor the Siebel database. Workflow Monitor Agent:

- Checks the Escalation Requests table to see when the conditions of a policy are met
- Monitors all policies within a single group

**NOTE:** You can only run one Workflow Monitor Agent process against a specific group at one time. You can run multiple Workflow Monitor Agent processes at the same time, but they must be against different groups; if you run two Workflow Monitor Agent processes against the same group, deadlocks will occur.

- If you use Action Agent = True, this generates requests for Workflow Action Agent in the Action Request table (S\_ESCL\_ACTN\_REQ)
- Purges requests from the S\_ESCL\_REQ table after processing. When a database trigger is activated because a workflow policy condition is met, a record is inserted into the Escalation Request table, S\_ESCL\_REQ. (See [Chapter 16, “Monitoring and Tuning Performance,”](#) 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 you use Action Agent = True, and if Workmon determines that the request in the S\_ESCL\_REQ table has no duration defined in the policy, Workmon either takes direct action and logs an entry into the S\_ESCL\_LOG table or sends it to the S\_ESCL\_ACTN\_REQ table.

If Workmon determines that the request has a time element that must be met, the request is sent to the S\_ESCL\_STATE table along with the expiration time. The request stays in the S\_ESCL\_STATE table until the expiration time is met, or the request is removed because the conditions of the policy are no longer met. Workmon evaluates each of the requests that remains in the S\_ESCL\_STATE table for a time duration match or to determine if the condition still matches in the S\_ESCL\_STATE table. If you use Action Agent = True, then as each match occurs WorkMon either takes direct action and logs an entry into the S\_ESCL\_LOG table or sends it to the S\_ESCL\_ACTN\_REQ table.

---

**NOTE:** If a workflow policy has a specified duration, the duration time is calculated from the time WorkMon detects that the row is in violation of the policy, not from the time the row was inserted into S\_ESCL\_REQ. For example, if you create a policy and set the duration as one week, but then WorkMon is not started until several days after Generate Triggers is run, the policy action will fire one week from when WorkMon is started, not one week from when the policy is created or Generate Triggers is run.

---

When the request for an action is made to the S\_ESCL\_ACTN\_REQ table, Workflow Action Agent executes the action and logs an entry into the S\_ESCL\_LOG table.

## Using Workflow Monitor Agent

Before you start Workflow Monitor Agent, you must create a separate server component definition for each Workflow Monitor Agent task. You can start Workflow Monitor Agent from the Server Manager command-line interface.

### Replication and Workflow Monitor Agent

Within the entire enterprise architecture of a Siebel deployment, there can be only one Workflow Monitor Agent monitoring a particular workflow group.

For example, a regional node can be running a Workflow Monitor Agent that monitors a group called Group 1. Meanwhile, in the headquarters, there is another WorkMon running, which monitors a group called Group 2. In this way, the organization is able to run Workflow Policies where needed, while working with the restriction of one WorkMon for one group.

---

**NOTE:** You cannot run more than one instance of Workflow Monitor Agent and Workflow Action Agent for a particular workflow group. However, you are allowed to have multiple Workflow Monitor Agent and Workflow Action Agent processes for different groups running at the same time.

---

## Starting Workflow Monitor Agent

### *To create a Workflow Monitor Agent component definition*

- 1 From the application-level menu, choose View > Site Map > Server Administration > Enterprise Configuration.

The Enterprise Configuration view appears.

- 2 Click the Component Definitions tab.

Two Component Definitions lists appear.

- 3 From the upper Component Definitions list menu, choose New Record.

A new record appears.

- 4 Complete the fields described in [Table 86](#):

**Table 86. Component Definitions Fields**

Field	Description
Name	Name of the component
Component Type	WorkMon
Component Group	Select an existing component group
Description	Description of the component
Alias	Alias for the component. The alias can not contain blank spaces.

- 5 From the upper Component Definitions list menu, choose Save Record.

The component definition is saved. To view the definition, you must perform a query.

**To set parameters and activate a Workflow Monitor Agent component definition**

- 1 In the upper Component Definitions list, perform a query for the component definition.
- 2 Optional. You may make additional changes to the component parameters. For a description of Workflow Monitor Agent parameters, see [Table 87 on page 326](#).
- 3 From the upper Component Definitions list menu, choose Enable Component Definition.

The definition state changes from “Creating” to “Active.”

- 4 Restart the Siebel server.

Your changes take effect.

**To stop or restart a Workflow Monitor Agent component**

- 1 From the application-level menu, choose View > Site Map.
- 2 Click Server Administration, then click Servers.

The Servers view appears.

- 3 Click the Server Components tab.
- 4 Select the component you would like to stop or restart, then click Shutdown or Startup.

**To start Workflow Monitor Agent using the Server Manager command-line interface**

- 1 Start the server manager by entering:

```
srvrmgr /g <gateway server address> /s <Siebel server name> /e  
<enterprise server name> /u <server administrator username> /p  
<server administrator password>
```

- 2 Start a new Workflow Monitor Agent task in background mode by entering:

```
start task for component WorkMon with  
SleepTime=<time>,GroupName=<group name>
```

For more information, see *Siebel Server Administration Guide*.

You can start the Workflow Monitor Agent from the command line. For more information, see *Siebel Server Administration Guide*.

---

**NOTE:** You will need to create a separate server component definition for each Workflow Monitor Agent task.

---

**To run the Workflow Monitor Agent task**

- 1 Navigate to View > Site Map > Server Administration > Enterprise Operations.
- 2 In the Component Request view, click New.
- 3 From the Component/Job picklist, select the name of the server component defined for this Workflow Monitor Agent task.
- 4 In the Component Request Parameters applet, click New.

Specify the parameters for the Workflow Monitor Agent. See [Table 87 on page 326](#) for a list of parameters.

- 5 Click Submit Request on the applet menu in the Component Requests Form to begin Workflow Action Agent task.

---

**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.

---

Table 87 shows Workflow Monitor Agent parameters.

**Table 87. WorkMon Command-Line Interface Parameters**

Parameter Name	Display Name	Description	Default Value
ActionAgent	Use Action Agent	Determines if Action Agent is automatically run with Monitor Agent.  If set to FALSE (the default setting), the Workflow Action Agent server component starts within Workflow Monitor Agent, and actions are then executed by Workflow Monitor Agent.  You must start Workflow Action Agent separately when using email consolidation and when the parameter Use Action Agent is set to TRUE.	FALSE
ActionInterval	Action Interval	Action execution interval in seconds. This argument determines when actions for a given policy are executed again on a given base table row. The purpose of this argument is to limit the number of times actions are executed if a row keeps going in and out of a matching condition.  In other words, if the same record repeatedly violates the same policy before the action interval expires, the record will be removed from the S_ESCL_REQ table and the action will not be performed again.  Note: The default is 3600 seconds. If this parameter is used, the value <i>must</i> be greater than 0 (zero) or unexpected behavior may result.	3600

**Table 87. WorkMon Command-Line Interface Parameters**

Parameter Name	Display Name	Description	Default Value
BatchMode	Processes the batch policies	Determines if Monitor Agent is running in batch mode. When the value is set to TRUE, only the policies that have the Batch flag set to TRUE will be evaluated. When FALSE, only the policies that have the Batch flag set to FALSE will be evaluated.  Note that when starting with Batch Mode set to TRUE, Workflow Monitor Agent will run once; that is, it will go through all records in the table and then exit out.	FALSE
CheckLogCacheSz	Cache size of Policy violations	Number of policy violations to store in cache.	100
DeleteSize	Request delete size	This indicates the number of records to commit at a time. The minimum is 1. If Workflow Monitor encounters deadlocks, you may reduce the default to 125 with minimal performance degradation. Note: to avoid call stack errors, do not set the Request Delete Size value to zero.	500
GenReqRetry	Number of seconds to retry	Number of seconds to retry sending a Generic Request message.	120
GroupName	Group Name	Required. Workflow policy group that Monitor Agent works on.	

**Table 87. WorkMon Command-Line Interface Parameters**

Parameter Name	Display Name	Description	Default Value
IgnoreError	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>	FALSE
KeepLogDays	Number of days to keep violation information	<p>Number of days worth of violation information that should be retained.</p> <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>	30
LastUsrCacheSz	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>	100
MailServer	Mail Server	<p>Name of email server to send notification of abnormal termination.</p>	
MailTo	Mailing Address	<p>Mail address to review notification of abnormal termination.</p> <p>Mail to &lt; mail ID &gt; 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>	



**Table 87. WorkMon Command-Line Interface Parameters**

Parameter Name	Display Name	Description	Default Value
ReloadPolicy	Reload Policy	Policy reload interval in seconds. This argument defines the frequency that policies are reloaded into the engine. This allows changes to be made on the screens and with the Generate Triggers component; the engine acts on the changes within some time frame.  The default is 600 seconds.	600
Requests	Requests per iteration	Maximum number of requests read per iteration.  This controls the maximum number of requests WorkMon reads from the requests queue within one iteration. Between iterations WorkMon deletes processed requests from the requests queue and commits, optionally reloads policies from the database, checks for shutdown request, and optionally sleeps. In other words, you can think of the Requests Per Iteration parameter as a way to control the maximum amount of work WorkMon performs before taking these between-iteration steps.	5000
Sleep Time	Sleep Time	The time in seconds that the Workflow Agent process “sleeps” after it has polled for events and fulfilled all obligations to notify. Once it has completed its obligations, the Workflow Agent process stops polling for the time period set by the sleep interval. This parameter affects both the performance of the Workflow Agent process and the responsiveness of the application server. See <a href="#">Chapter 16, “Monitoring and Tuning Performance,”</a> for more information.	60

---

**NOTE:** You can separate the processes for load balancing or run one process for ease of testing.

---

### Using Workflow Action Agent

The Workflow Action Agent process submits a request to Email Manager and Page Manager when actions are to be taken.

---

**NOTE:** You will need to create a component definition for each Workflow Action Agent task.

---

Workflow Action Agent:

- Processes requests logged in the action request table (S\_ESCL\_ACTN\_REQ) for a single group.
- Invokes all actions linked with the Workflow policy being processed.
- Logs email and page actions in the S\_APPSRVR\_REQ table for execution by Email Manager and Page Manager.
- Purges requests from S\_ESCL\_ACTN\_REQ after processing.

If the Use Action Agent parameter is set to TRUE in the Monitor Agent process, you need to perform the following steps to start the Action Agent process.

***To run the Workflow Action Agent process***

You start the Workflow Action Agent in the same way that you start the Workflow Monitor Agent. See [“Starting Workflow Monitor Agent” on page 323](#).

---

**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***

You shut down the Workflow Action Agent in the same way that you shut down the Workflow Monitor Agent. See [“To stop or restart a Workflow Monitor Agent component” on page 324](#).

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 View > Site Map > Server Administration > Enterprise Operations.
- 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 87 on page 326](#) 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 System 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 about the process. Trace files are created for the following Siebel Server processes:

- Generate Triggers
- Page Manager
- Email Manager
- Workflow Monitor Agent
- Workflow Action Agent

You can view trace file information in one of two places:

- The Siebel Server Tasks > Tasks Info Log in Server Administration
- The log directory of your Siebel Server.

### Viewing Trace Files in Siebel Server Administration

#### ***To view trace files in the Siebel Server Tasks view***

- Navigate to View > Site Map > Server Administration > Tasks > Tasks Info Log.

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

### Viewing Trace Files in the Siebel Server Log Directory

You can also use Windows Explorer to navigate to your Siebel Server log directory. Under \log, select the server's name to see a file that lists all the trace files for each server process.

You can double-click the Trace File icon to access the trace file. You can view the trace file for any application server task. See *Siebel Server Administration Guide* for more information on using trace files.

**Tracing and Event Log Levels**

Workflow Policies uses the events described in [Table 88](#) for logging.

**Table 88. Workflow Policies Logging Events**

Event	Level	Description
SqlParseAndExecute	4	Traces all SQL statements and execution times.
Object Assignment	3	Traces Workflow Monitor Agent while it is doing Dynamic Assignment. Use in conjunction with Rules Evaluation.
Rules Evaluation	4	Traces Workflow Monitor Agent while it is doing Dynamic Assignment. Use in conjunction with Object Assignment.

**NOTE:** Setting trace levels above default parameters will affect performance. Trace levels should be reset to default parameters after troubleshooting has been completed.

## 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 or Trend Analysis Chart

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

#### **To view the Policy Frequency or Trend Analysis chart**

- Navigate to View > Site Map > Business Process Administration > Workflow Policy Log. This view has two applets:
  - **Monitor Log.** This lists the workflow policies.
  - **Workflow Policy Frequency/Trend Analysis.** The Workflow Policy Frequency Analysis applet displays a chart illustrating the execution frequency of a selected policy. The Workflow Policy Trend Analysis applet shows the total number of workflow policy conditions met over a specified period of time. To toggle between the two analysis applets, use the toggle list on the chart applet.

### Using Workflow Policies Reports

In the Workflow Policies Policies and Log views, you can bring up a Reports page that you can print out. To bring up the report, select Reports from View.

The Reports page that appears provides summary information of the Workflow Policy.

If you need to review all the business rules for your organization, you can print out the Reports page for each of your workflow policies.





# Testing Workflow Policies 14

This chapter describes testing and troubleshooting procedures.

Testing your workflow policies before implementing them into your production environment allows action recipients to receive accurate and useful information and the results are exactly what you want.

You need to develop a testing and migration procedure for introducing changes into the production environment. Some of the considerations for creating a test and migration environment are discussed in [Chapter 9, “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 Server 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.

You verify your action by checking to see if the proper action occurs. That is, you can check that the email arrives or the pager goes off. You can monitor Workflow Agent progress using the Workflow Policies Log view.

The Workflow Policy Log view displays a log of all the records that meet a policy condition tracked by the Workflow Monitor Agent process. You access the Workflow Policy Log view from the Siebel Client.

The view contains the following fields:

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

Once you have verified that the workflow policies work as expected, you can migrate the workflow policies to your production environment.

## Troubleshooting

Because workflow policies are based on database triggers, a workflow policy can take effect on a database record only after the record is committed. If you have a policy that is based on multiple database tables, the policy takes effect only if the records on all tables are committed. For example, Opportunity Revenue is stored in the S\_OPTY\_POSTN table, and lead quality is stored in the S\_OPTY table. A policy with conditions Opportunity Revenue > 10M and Lead Quality = high takes effect only when the records are committed on both tables.

Also keep in mind that multiple business components can be created for the same database table using search specifications. If you are creating a workflow policy component to monitor a business component, be sure to include all the fields that are being used in search specifications as workflow policy columns. The workflow policy column can then be used in the policy conditions to allow appropriate behavior to be enforced.

If your workflow policy action does not occur, check the following:

- Verify that your test record meets *ALL* your workflow policy conditions.
- Verify that the client configuration file is pointing to the correct enterprise server (one error that can occur if the server is incorrect is ESC-00053, “Error loading rule definitions”).
- Check the workflow policy activation date/time.
- Check the monitor task:
  - Is the monitor awake and running against the correct group?
  - Search the Task Information log for the Row\_Id of your test record.
    - If Row\_Id does not exist, run GENERATE TRIGGERS.
    - Update your test record.
- Check the Action Agent task:
  - Is Action Agent awake and running against the correct workflow policy group?
  - Search the Task Information log for the Row\_Id of the test record.

- Make sure your triggers are generated.

## **Workflow Tracing**

Workflow Policies uses the General Events event for logging. To view informational messages, set the log level to 3. To view debugging information, set the log level to 4.



# Migrating Policies to the Production Environment **15**

This chapter describes how to migrate workflow policies to the production environment.

## 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 the upgrade guide for the operating system you are using.
- 3** Re-enter your workflow policy action types, workflow policies, and workflow policy groups exactly as they are in the test environment into the production environment.

---

**NOTE:** Information that you have entered using Siebel Tools does not need to be re-entered.

---

- 4** In the Siebel Server Manager Client, click Site Map > Server Administration > Enterprise Operations > Component Requests.
- 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 13, “Workflow Policies Server Administration.”](#) See *Siebel Server Administration Guide* for a description of the generic and enterprise parameters.



**7** Select Submit Query.

See [Chapter 13, “Workflow Policies Server Administration,”](#) for more information on trace files. For more information on using trace flags, see *Siebel Server Administration Guide*.

---

**NOTE:** To help prevent invalid triggers from being applied to your production environment, apply your database triggers to your test environment before you apply them to your production environment.

---

## **Migrating Policies to the Production Environment**

---

*Migrating Workflow Policies*

# Monitoring and Tuning Performance 16

This chapter describes workflow policies monitoring and performance tuning procedures.

## Monitoring Workflow Policies

You need to monitor Workflow Policies regularly to check 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.

Workflow Policies uses the General Events event for logging. To see informational messages, set the log level to 3. To see debugging information, set the log level to 4.

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

- **Workflow Policy 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 Policy Log View

The Workflow Policy 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 Policy Log view is described in *Siebel Online Help*.

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

---

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

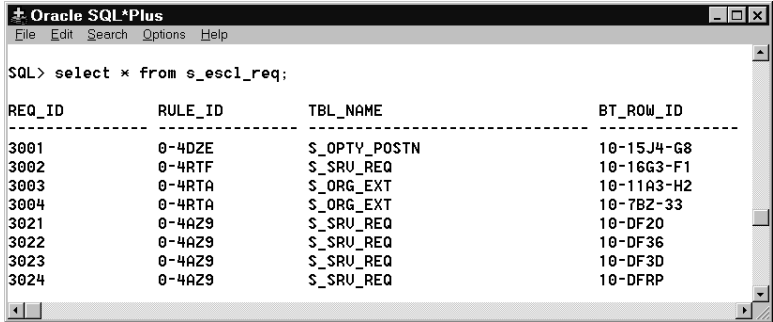
---

The Workflow Policy 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 6 shows the S\_ESCL\_REQ table.

A screenshot of the Oracle SQL\*Plus command-line interface. The window title is "Oracle SQL\*Plus". The menu bar includes "File", "Edit", "Search", "Options", and "Help". The command prompt shows "SQL> select \* from s\_escl\_req;". Below the command, the results of the query are displayed in a table format with four columns: REQ\_ID, RULE\_ID, TBL\_NAME, and BT\_ROW\_ID. The data is as follows:

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 6. S\_ESCL\_REQ Table

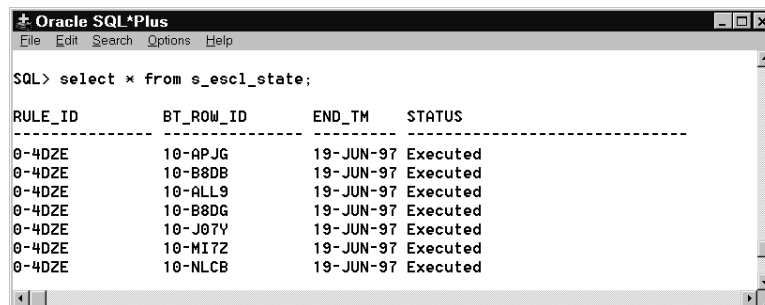
When a trigger fires against a Workflow policy condition, a record is inserted in the Escalation Request table, S\_ESCL\_REQ. This table contains all the rows in the database that could trigger a Workflow Policy to take action. After the workflow Monitor Agent processes a request, it removes the row 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 7 shows the S\_ESCL\_STATE time-based table.



The screenshot shows the Oracle SQL\*Plus interface with the command `SQL> select * from s_escl_state;` executed. The output displays a table with four columns: RULE\_ID, BT\_ROW\_ID, END\_TM, and STATUS. The data shows eight rows, all with a status of 'Executed' and an end time of '19-JUN-97'.

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-J07V	19-JUN-97	Executed
0-4DZE	10-MITZ	19-JUN-97	Executed
0-4DZE	10-NLCB	19-JUN-97	Executed

**Figure 7. 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.

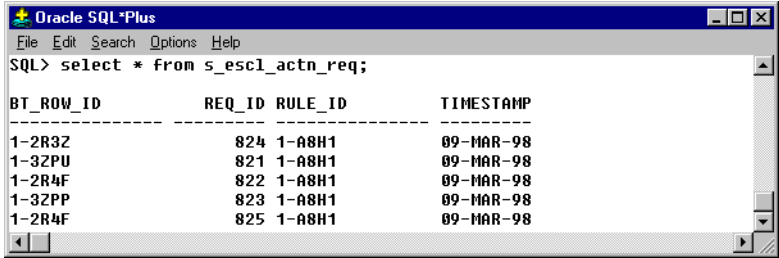
---

**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 8 shows the S\_ESCL\_ACTN\_REQ table.



The screenshot shows the Oracle SQL\*Plus interface with the command 'SQL> select \* from s\_esc1\_actn\_req;' executed. The resulting table has four columns: BT\_ROW\_ID, REQ\_ID, RULE\_ID, and TIMESTAMP. The data is 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 8. 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 allow 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 can 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, which helps make monitoring and action execution more effective.

### **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

To run Workflow Monitor Agent and Workflow Action Agent using a command-line interface, you need to supply the Component Alias. Workflow Monitor Agent is WorkMon, and Workflow Action Agent is WorkActn. For more information about Component Alias, see *Siebel Server Administration Guide*.

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 13, “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 13, “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.

## Monitoring and Tuning Performance

*Tuning Workflow Policies for Performance*

# Using State Models **17**

This chapter describes how to use state models.

## State Models Overview

The State Models module 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 the data of that object can be modified. For 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.

In this state model, the starting point state for all service requests is Open.



Allowable state transitions for this model are listed in [Table 89](#).

**Table 89. 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. To display this view, choose Site Map > Business Process Administration > State Models.

[Table 90](#) describes the fields in the State Models view.

**Table 90. Fields in the State Models View**

Field	Description
Activation	The date on which the state model begins to be enforced.
Business Component	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 <a href="#">“Enabling Business Components for State Models” on page 364</a> .
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 Date/Time	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	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 90. Fields in the State Models View**

Field	Description
State Description	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.
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 New 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"> <li>■ <b>No Delete.</b> 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.</li> <li>■ <b>No Update.</b> When checked, records in this state are read-only and cannot be updated. Note that No Update also affects a record's child records. For example, if an account becomes read-only due to use of the No Update restriction, the related contacts for the account also become read-only, so inserting and deleting contacts is no longer possible.</li> <li>■ <b>Restrict Transition.</b> 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.</li> </ul> <p>Note that despite No Delete and No Update restrictions specified for the state model, MVG fields that do not have a direct parent-child link between the parent field and the MVG applet will not be read-only when a record is in a No Update or No Delete state of a state model. To avoid this issue, you can use Siebel Tools to remove the MVG field from the user interface.</p> <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 Site Map > Business Process Administration > State Models.
- 2** In the State Models list applet, click New to create a new record for a state model.
- 3** Fill in the appropriate fields described in [Table 90 on page 362](#).

#### **Enabling Business Components for State Models**

You can enable state models on any business component (not enabled by default) that is based directly on the CSSBCBase class.

---

**NOTE:** Enabling state models for business components based on subclasses of CSSBCBase class is not supported.

---

To enable state models on a business component, the fields used:

- 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.

#### **To enable business components for state models 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 States view. To display this view, choose Site Map > Business Process Administration > State Models > States.

If a state model exists for a field in a business component, all possible transitions for the field must be defined in the States view. If a transition is not explicitly defined, it will not be available to users. If Restrict Transition is left unchecked, you can access all other states. Otherwise, only the defined transitions are accessible.

[Table 91](#) and [Table 92 on page 367](#) describe the fields in the States view. The State Model form applet at the top of the view contains data from the State Models view.

Table 91 describes the fields in the Transitions list applet.

**Table 91. Fields in the Transitions List Applet**

Field	Description
From State	<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	<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	<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 Reference</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>

**Table 91. Fields in the Transitions List Applet**

Field	Description
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	<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 92](#) describes the fields in the Authorized Positions list applet.

**Table 92. Fields in the Authorized Positions 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 Site Map > Business Process Administration > State Models.

- 2 In the State Models view, click the Name field in the appropriate state model record.

The States view appears.

- 3 Click the Transitions tab to access the Transitions list applet. In the Transitions list applet, click New to create a new state transition record.

You can create one or more state transitions. For each state transition, fill in the appropriate fields, as described in [Table 91 on page 366](#).

- 4 If you want to restrict the state transitions to a subset of users, uncheck the Public field and add records in the Authorized Positions list to specify one or more positions for these users.

---

**NOTE:** When the Public field is left unchecked, all other states are accessible and any defined transitions and their rules are ignored. Checking this field indicates that only defined transitions from this state are permitted. An end state is created if this field is checked, and no transitions are defined.

---

If a state is set to a default state in Tools, then it cannot be overridden in the State Models view.

For a description of the fields, see [Table 92 on page 367](#).

## Restricting Transitions Between States

Checking the Restrict Transition checkbox for one of the states defined for a State Model restricts transitions for all of the states defined for that State Model.

If you check the Restrict Transition checkbox for any of the states defined for a State Model, you must define valid transitions for each state defined for that State Model.

For example, if a State Model has five states defined for it, and you check the Restrict Transition checkbox for State 1, then you must define valid transitions from State 1 to State 2, State 3, State 4, and State 5.

Checking the Restrict Transition checkbox without defining any transitions for a state creates an End state. No transitions will appear in the state field drop-down list. Users cannot transition from an End state to any other state.



## Enabling Child Modification

By default, child records cannot be added, modified, or deleted when the parent record state is “No Delete.” You can change the default setting to enable child modification.

### ***To enable child modification for all business components***

- In Siebel Tools, set the State Model Cache business service Enable Child Modification user property to “Y.”

This allows you to add, modify, and delete child records, even when the parent record state is “No Delete.”

When the State Model Cache business service Enable Child Modification user property is set to “Y”, you can prevent child modification for a particular business component by defining the Enable Child Modification user property for that component, then setting the user property for that component to “N.”

### ***To prevent child modification for a particular business component***

- In Siebel Tools, define the Enable Child Modification user property for the business component, then set it to “N.”

This prevents child modification of the business component and all state models defined for it, even when the State Model Cache business service Enable Child Modification user property is set to Y.

## Using a State Model

After you create or modify a state model, your changes will not take effect until either a new server process is spawned or you restart the Siebel server.

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. State models are supported on regional nodes. For more information about support for regional nodes, see [“Modifying the Replication Field” on page 198](#) and *Siebel Remote and Replication Manager Administration Guide*.

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 check to see 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.

---

**NOTE:** User-defined business components are not supported. State models are enforced for updates made through Siebel Visual Basic, Siebel eScript, and workflow processes because these updates do not circumvent any business component logic. However, state models are not enforced for updates made through Enterprise Integration Manager or workflow policy programs.

---

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 field is checked for the transition.
- The Public 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.

As soon as a state model expires, it will no longer be used. This means that if you have two state models on the same field, the picklist will always behave according to the active state model. There is no need to bounce the server.

However, you should be careful when defining two state models on the same field. For example, suppose that state model A and B are on the same field. When A expires, B is activated. If A has a field called Open and B does not, then it is possible to be stuck in the Open state after A expires.

## **Understanding the State Model Cache**

This list describes some important characteristics of the State Model cache.

- The state model cache is read-only.
- Data in the state model cache never expires.
- The state model cache size is infinite.
- The state model cache is shared between users.
- Changes that you make to a state model using the State Model view will take effect when a new server process begins, or whenever you bounce the server. Usually, a new server process begins every few hours. When a new server process begins, all new client connections will be handled by the new process. The old server process will wait until all of its clients have disconnected or timed out, then it will destroy itself.

## Using State Models

*Using a State Model*

## **Predefined Business Services**

# **A**

This appendix describes how to use predefined business services with workflow policies.

# Predefined Business Services

This appendix describes these predefined business services:

- Outbound Communications Manager
- Synchronous Assignment Manager Requests
- Server Requests
- Workflow Utilities
- Financial Services Workflow User Interface Utilities
- Data Transfer Utilities
- SmartScript Integration with Workflow
- Dynamic Applets

For more predefined business services, see *Business Processes and Rules: Siebel eBusiness Application Integration Volume IV*.

## Outbound Communications Manager

The Outbound Communications Manager business service is for sending notifications, such as notifications to contacts or employees. For information on methods and arguments, see *Siebel Communications Server Administration Guide*.

## Synchronous Assignment Manager Requests

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 93](#) describes the Assign method arguments.

**Table 93. Assign Method Arguments**

Argument	Description
Assignment Object Name	Required. This is the object that you want to assign.
Object Row ID	Required. This is the row ID for the object you want to assign. To assign the work item for the workflow process, set this to the Object ID process property.

**TIP:** The Synchronous Assignment Manager Requests business service attempts to assign all records that meet the appropriate criteria, even if they are locked. To prevent errors in your process due to locked records, you should set up a condition in your workflow process or workflow policy to skip records that do not meet the condition `ASGN_USR_EXCLD_FLG = N`.

## Server Requests

The Server Requests business service is for sending generic requests to the server request broker. It can send it in three different modes: asynchronous, synchronous or schedule mode. While in synchronous mode, it will send the request to the server request broker and wait for a response. Otherwise, it will just send the request but does not wait for a response.

When invoking the Server Requests business service to submit a component request, you need to specify SRB parameters in the input property set and all component specific parameters in a child property set. There are two points to note:

There is no validation for component parameters passed in the child property set, and these arguments do not appear in the picklist in the workflow administration views.

**NOTE:** If you want to pass parameters to the server component that are not listed as available arguments, you can create a custom business service that contains the necessary parameters. Alternatively, you can create a component job that has the parameters defined as part of the job definition. For more information on creating jobs, see *Siebel Server Administration Guide*.

This service has two methods available:

- **Submit Request.** Use this method to submit a request to the server request broker.
- **Cancel Request.** Use this method to cancel any server request that is currently awaiting to be run.

**Submit Request Arguments**

Table 94 describes the Submit Request method arguments.

**Table 94. Submit Request Method Arguments**

Argument	Description
Component	Required (if Component Job is not entered). Enter the name of the server component to run.
Component Job	Required (if Component is not entered). Enter the name of the component job to run.
Delete After	Optional. Number of iterations before deleting the request. Works with Delete After Units. The default value is 0 (zero).



**Table 94. Submit Request Method Arguments**

Argument	Description
Delete After Units	<p>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).</p> <p>Other possible values are:</p> <ul style="list-style-type: none"> <li>■ ASAP</li> <li>■ SECONDS</li> <li>■ MINUTES</li> <li>■ HOURS</li> <li>■ DAYS</li> <li>■ WEEKS</li> <li>■ MONTHS</li> <li>■ YEARS</li> </ul>
Description	Optional. A description of the server request.
Hold Flag	Optional. For asynchronous requests only. Flag to indicate whether or not to hold the request.
Maximum Execution Time	For future use.
Method	Optional. Only applicable for service-based server components (for example, Workflow Process Manager, Communications Manager). Specify the business service method to invoke.
Mode of Server Request	<p>Required. This tells the server request broker how to handle the server request. While in auto mode, the server request broker will set the mode to either synchronous or schedule, depending if the client is connected or mobile.</p> <ul style="list-style-type: none"> <li>■ Sync: Synchronous</li> <li>■ Async: Asynchronous</li> <li>■ Schedule: Schedule</li> <li>■ Auto: Automatic configuration</li> </ul>

**Table 94. Submit Request Method Arguments**

Argument	Description
Request ID Needed	Optional. This is only applicable to asynchronous and schedule mode. If this is set to false, these two server requests will return even faster.
Request Key	For future use.
Repeat Interval	Optional. The interval for repeating requests.
Number of Repetitions	Optional. The number of repetitions for repeating requests.
Repeat From	Optional. Possible values are Scheduled Start, Actual Start, and End.
Repeat Interval Units	Optional. Unit of intervals for repeating requests.
Server Name	Optional. Enter the specific server that this request is to be run from.
Start Date	Optional. Start date and time.
Storage Amount	Optional. Enter the amount of time that the server request will be stored in the database in the event that the server is down.
Storage Units	Optional. Enter the units to measure the iterations for the Storage Amount argument. The units are the same as Delete After Units.

### Cancel Request Method Arguments

[Table 95](#) describes the Cancel Request method arguments.

**Table 95. Cancel Request Method Arguments**

Argument	Description
Request ID	Required. This is the ID of the server request to be cancelled.
Repeat Number	Optional. This is the number of repetitions of the repeating server requests that are to be cancelled.

### Workflow Utilities

The Workflow Utilities business service contains generic utilities that can be used in process definitions.

This business service has one available method, Return Property Values. This method returns a mirror image of the input arguments. The Return Property Values method is also referred to as the Echo method.

### Return Property Values Arguments

[Table 96](#) describes the Return Property Values method arguments.

**Table 96. Return Property Values Method Arguments**

Argument	Description
Input Arguments	This method accepts any input arguments.
Output Arguments	An exact copy of the input arguments.

## Financial Services Workflow User Interface Utilities

This business service allows you to design workflow processes that automate user actions such as navigating between records, clicking a hyperlink, creating or deleting records, or populating field values.

This business service operates at the applet level. It complements Siebel Operation steps, which operate at the business component level.

### Business Scenario

The scenario outlined in this section is an example of an insurance claim workflow. Your company may follow a different workflow that fits its business requirements.

In this scenario, you are a claims representative in an insurance company call center. You are filing a claim for a policyholder whose automobile was damaged in an accident.

First, you review the customer's policies in the Financial Services Summary view in the Contact screen. You select the auto policy, then click the New Claim button. This starts the Claim workflow.

The Claim workflow clicks the Policy Number hyperlink to navigate to the Claims view in the Auto Policies screen. Then it creates a new claim record and enters the Status and Loss Type information. You enter additional information, and then save the record. The Claim workflow navigates to the Loss Description view in the Claims screen. You enter information, and then save the record.

The Claim workflow navigates to the Insured Property view and enters information about the vehicle based on the information you entered in the Loss Description view. Now you can quickly enter additional information, such as the damage to the vehicle, the location of the vehicle, and so on.

The Claim workflow continues to guide you through other views in the Claims screen as you complete the claim filing process.

## Financial Services Workflow UI Utilities Methods

The Financial Services Workflow UI Utilities business service methods are described in [Table 97](#).

**Table 97. Financial Services Workflow UI Utilities Methods**

Method	Description / Arguments
Create Applet Record	<p>Creates a new record and completes a set of field values in the record.</p> <p>Input Arguments:</p> <ul style="list-style-type: none"> <li>■ Applet (Required): The name of the applet in which you want to create a record.</li> <li>■ Field Name &amp; Value Pairs (Optional): One or more pairs of field names and their associated values.</li> </ul> <p>Output Arguments:</p> <ul style="list-style-type: none"> <li>■ ObjId: The row ID of the new record.</li> </ul>
Do Applet Method	<p>Performs actions such as saving or copying the active record, or navigating to the next record.</p> <p>Input Arguments:</p> <ul style="list-style-type: none"> <li>■ Applet (Required): The name of the applet in which you perform this action.</li> <li>■ Method (Optional): The name of the method that specifies the action. Valid values are: <ul style="list-style-type: none"> <li>■ WriteRecord. Saves the active record.</li> <li>■ DeleteRecord. Deletes the active record.</li> <li>■ UndoRecord. Cancels the active record.</li> <li>■ CopyRecord. Copies the active record.</li> <li>■ GotoNext. Navigates to the next record.</li> <li>■ GotoPrevious. Navigates to the previous record.</li> <li>■ GotoFirstSet. Navigates to the first record set.</li> <li>■ GotoLastSet. Navigates to the last record set.</li> </ul> </li> </ul> <p>Output Arguments:</p> <ul style="list-style-type: none"> <li>■ ObjId: The row ID of the record that is active after the action was performed. For example, if you perform the action “GotoNext,” then ObjId will contain the row ID of the destination record.</li> </ul>

**Table 97. Financial Services Workflow UI Utilities Methods**

Method	Description / Arguments
Go To Applet	<p>Navigates to an applet.</p> <p>Input Arguments:</p> <ul style="list-style-type: none"><li>■ Applet (Required): The name of the applet to which you want to navigate.</li></ul>
Drill Down	<p>Clicks on a hyperlink of the active record.</p> <p>Input Arguments:</p> <ul style="list-style-type: none"><li>■ Applet (Required): The name of the applet that contains the hyperlink.</li><li>■ DrillDown (Required): The name of the field that contains the hyperlink.</li></ul> <p>Output Arguments:</p> <ul style="list-style-type: none"><li>■ ObjId: The row ID of the destination record.</li></ul>
Set Field Value	<p>Updates the values of a field or fields of the active record.</p> <p>Input Arguments:</p> <ul style="list-style-type: none"><li>■ Applet (Required): The name of the applet that contains the fields whose values you want to update.</li><li>■ Field Name &amp; Value Pairs (Required): One or more pairs of field names and their associated values.</li></ul>

## Data Transfer Utilities

This business service allows you to transfer data from a source business component to a destination component. For example, a user can enter data in one view, then use a toolbar command to navigate to another view. Data entered in the first view will be automatically entered in the second view.

## Data Map Objects

Data map objects indicate the data that is being transferred from the source business object to the destination business object. You can use the Application Administration > Data Map Administration screen to define data map objects. Only Siebel administrators should have access to this screen. Data map objects are described in [Table 98](#).

**Table 98. Data Map Objects**

Object	Description
Name	The name of the data map object. You should enter a unique name.
Source Business Object	The name of the source business object. You must define this name in the Siebel Repository file.
Destination Business Object	The name of the destination business object. You can specify the same business object as both source and destination business object.
Inactive	Check this box to make the data map object inactive.

**TIP:** Data map objects are cached in memory. Whenever you make changes to an existing data map object, click the Clear Cache button to refresh the cache so that your changes appear.

**Data Map Components**

Data map components define the mapping at the child business component level. Each data map object can contain multiple data map components. You can arrange data map components in a parent-child hierarchy, but you must specify a parent for all except one data map component. The parentless data map component is called the root data map component. Data map components are described in [Table 99](#).

**Table 99. Data Map Components**

Component	Description
Name	The name of the data map component. You should enter a unique name for each data map component in a data map object.
Source/Destination Business Component	The name of the source or destination business component. If you specify a parent for this data map component, you must define this business component in the Siebel repository file as a child of the source or destination business object to which the parent data map component is mapped.
Parent	The name of the parent data map component. If you specify a parent, the parent is mapped to particular source and destination business components. You must map the child data map component to a child of those source and destination business components. If you do not want to specify a parent, enter “Null” to indicate that this is the root data map component. Each data map object can have only one root data map component.
Inactive	Check this box to make the data map component inactive.



## Data Map Fields

Data map fields define the field-to-field mapping. Data map fields are described in [Table 100](#).

**Table 100. Data Map Fields**

Field	Description
Source Field	<p>The name of the field from which data will be transferred. You can transfer either:</p> <ul style="list-style-type: none"><li>■ The field value that is defined for the source business component in the Siebel repository file.</li><li>■ The result of a calculated expression. A calculated expression must include at least one of these characters:</li><li>■ [ ] Evaluated at the source. The result of the evaluation is transferred at run time.</li><li>■ ( ) Evaluated at the source. The result of the evaluation is transferred at run time.</li><li>■ “ ” Evaluated at the source. The result of the evaluation is transferred at run time.</li><li>■ + Evaluated at the source. The result of the evaluation is transferred at run time.</li><li>■ { } Evaluated at the destination business component.</li></ul>
Destination Field	<p>The name of the field to which data will be transferred. The destination field value can be either:</p> <ul style="list-style-type: none"><li>■ The field value that is defined for the source business component in the Siebel repository file.</li><li>■ The result of a calculated expression. If the destination field value is a calculated expression, then:</li><li>■ The source field value is transferred to the destination field. The destination field's calculated expression is evaluated at run time.</li><li>■ The result of the evaluation is transferred back into the source field. The destination business component is not updated.</li></ul>

**Table 100. Data Map Fields**

Field	Description
Key	<p>Matches destination records with source records.</p> <p>For example, the Update operation updates the record in the destination business component whose key destination fields all match those of the corresponding source fields.</p> <p>Each data map component should contain at least one key field. If the destination business component is populated with an associated list business component, at least one key field is required.</p>
Source /Destination Multi-Value Link	<p>This link indicates that the source and destination fields are multi-value fields.</p> <p>Data is transferred from one multi-value field to another by dividing data map fields into several subcomponents. All entries with the same source and destination multi-value link constitute a subcomponent. You should specify a key for each subcomponent.</p> <p>Note: Data transfer from a multi-value field to a single value field is not allowed.</p>

## Data Transfer Utilities Methods

The Data Transfer Utilities business service methods are described in [Table 101](#).

**Table 101. Data Transfer Utilities Methods and Arguments**

Method	Description / Arguments
Data Transfer	<p>Transfers data from the source business component to the destination business component.</p> <p>Input Arguments:</p> <ul style="list-style-type: none"><li>■ Data Map Object (Required): The name of the data map object that defines the mapping.</li><li>■ Operation (Required): Valid entries include Insert, Update, Delete, and Upsert.</li><li>■ Goto View (Optional): The name of a view that appears to users after the data transfer operation.</li><li>■ Option (Optional): A text field that allows you to specify additional options for the operation. The only option that is currently supported is NewBusObj. NewBusObj allows you to instantiate a new destination business object.</li></ul> <p>Output arguments:</p> <ul style="list-style-type: none"><li>■ MockMethodReplied: Y or N.</li></ul>
FAFireEventxxx	<p>A hidden method that you can use to create a toolbar command. “FAFireEventxxx” invokes the method “EventMethodxxx” on the primary business component of the active view. “EventMethodxxx” triggers the event manager, which invokes either a workflow or a business service.</p>

**Table 101. Data Transfer Utilities Methods and Arguments**

Method	Description / Arguments
GetActiveViewProp	<p>Auxiliary function to retrieve the active view's properties. It does not take any input arguments.</p> <p>Output arguments:</p> <ul style="list-style-type: none"><li>■ Business Object: The name of the business object.</li><li>■ View: The name of the active view.</li><li>■ Screen: The name of the active screen.</li><li>■ Thread Applet: The name of the thread applet.</li><li>■ Is Administration Mode: Y or N</li><li>■ View Mode Code: an integer representing Siebel view mode.</li><li>■ 0: SalesRep View</li><li>■ 1: Manager View</li><li>■ 2: Personal View</li><li>■ 3: AllView</li></ul>
TryMockMethod	<p>Advanced auxiliary function for administration of tool bar button workflow. It does not take any input arguments.</p> <p>Output arguments:</p> <p>MockMethodReplied. Y or N.</p>

## Considerations for Data Transfer Utilities

**CAUTION:** To verify that all operations are performed, SQL statements should be spooled during the development stage.

The Data Transfer Utilities business service follows these rules:

- Data Transfer Utilities execute inside each client's object manager.

- The Data Transfer Utilities business service reuses in-memory business objects. All operations are performed on the active source business object instead of a newly-instantiated source business object. This results in less memory use and better performance, but requires that you follow these constraints:
  - The active business component should not be in the query state.
  - In the following cases, the Update operation is the only operation allowed:
    - The destination business component is the active business component.
    - The destination business component is the source business component.
    - The destination business component is one of the source business component's ascendant business components.
- The Data Transfer Utilities business service requires a passed-in reference to the active business component. When you use the Data Transfer Utilities business service with other business services in a workflow, make sure that the other business services do not interfere with the passing of a reference to the active business component. Use spooled SQL statements to confirm that the operations are performed correctly.
- You should invoke the Data Transfer Utilities business service from Siebel Event Manager, or you should use it inside a workflow process that is invoked by Siebel Event Manager. Do not invoke the Data Transfer method from these business component events:
  - Query event. The business components are not in an updateable state.
  - DeleteRecord event. Siebel Event Manager does not pass in a reference to the active business component. Use PreDeleteRecord event instead.
- If the source business component is the active business component or one of its ascendant business components, data transfer is invoked only on the active row of the source business component. Otherwise, the operation is performed on all rows in the source business component.
- By default, DataTransfer operation cannot be used to invoke another DataTransfer operation. In other words, at any time within a client's object manager there is only one DataTransfer method in the call stack.

# Configuring Event-based Commands

The Data Transfer Utilities business service allows you to configure toolbar and menu commands based on Siebel Event Manager.

**To configure event-based commands**

- 1 Define a command in Siebel Tools. Business Service = "FINS Data Transfer Utilities". Method = "FAFireEventxxx". The method name can be anything that begins with "FAFireEvent".
- 2 Define a toolbar.
- 3 Define a toolbar item for the command you defined.
- 4 In the Siebel client, navigate to Site Map > Business Process Administration > Workflow Process, and then create a new workflow. The table below lists the entries you should make to configure the Start step for the workflow.

Field	Entry
Event Object Type	BusComp
Event	InvokeMethod
Event Object	The name of the business component in which the event is invoked. For a toolbar command, this is the primary business component in the view in which the command is invoked.
Sub Event	EventMethodxxx. Choose the same value for xxx that you chose for FAFireEvent.
Event Cancel Flag	Yes.

- 5 Define a Mock Event Sink step. For more information, see ["Mock Event Sink" on page 391](#).
- 6 Define the remaining workflow.

- 7 Define command visibility. In Siebel Tools navigate to Business Services > FINS Data Transfer Utilities. Define a user property in which Name = `MethodNameGotoView`, and Value = The name of a view. `MethodName` is the name of the command method. When you define this user property, this method is enabled only for users who have visibility to the view defined in the value. If the method does not contain a `GotoView`, visibility is not imposed on the method.

## Mock Event Sink

When a command is invoked from a toolbar button or menu, the Data Transfer Utilities business service invokes the method “`EventMethodxxx`” on the primary business component of the active view. To allow the invocation to be captured by Siebel Event Manager, the primary business component should be derived from `CSSBCBase`.

When the view is changed, Siebel framework polls each command for the application-level toolbar buttons and application menu to determine whether the button or menu items should be made read-only.

Data Transfer Utilities determines at runtime whether a “`FAFireEventxxx`” method should be disabled or not by initializing Mock Event Mode. It sets up a global mock event flag within the client’s object manager. It then invokes “`EventMethodxxx`” on the primary business component of the active view. If this “`EventMethodxxx`” is finally captured by a Mock Event Sink, a global response flag is set. When the Data Transfer Utilities finds out the response flag is set by a mock event sink, it enables the “`FAFireEventxxx`” method for that particular view. Otherwise, the method is disabled.

Mock Event Sinks are specialized business service methods that capture mock events. They check whether the client’s OM is in the mock event mode. If not, they do nothing. If so, they reply to the mock event by setting the response flag as well as the output argument.

`MockMethodReplied` = ‘Y’.

`TryMockMethod` in Data Transfer Utilities is a mock event sink. `DataTransfer` method has a built-in mock event sink.

## Toolbar Button

For information about configuring a toolbar button, see *Release Notes*.

If the toolbar button is mapped to a workflow, the last step before the End step in the workflow should contain a GoToView function, which can be implemented by either the DataTransfer Method in Data Transfer Utilities or the User Interact business service. If staying in the same view is desired, specify the target view name as the active view name. The purpose of the GoToView function is to force the user interface to refresh.

## SmartScript Integration with Workflow

You can invoke a SmartScript from a workflow process and pass parameters from the SmartScript to the workflow process. These business services support SmartScript integration with workflow:

- FINS Web Shopping Service
- FINS SmartScript Output Service

---

**NOTE:** In order to completely understand the concepts in this section, familiarity with SmartScript is required. For more information, see *Siebel SmartScript Administration Guide*.

---

### Business Scenario

The scenario outlined in this section is an example of how to insert a SmartScript in a workflow process. After the SmartScript ends, the workflow proceeds to the next step in the workflow process. Your company may follow a different workflow that fits its business requirements.

In this scenario, you are designing a SmartScript that allows customers to complete a loan application online. After the loan application is complete, your workflow verifies the customer's address in an external system.

## Integrating SmartScript with Workflow

### *To configure SmartScript integration with Workflow*

- 1** In the SmartScript, create Save User Parameters for the parameters you want to pass to the workflow. For more information on SmartScript user parameters, see *Siebel SmartScript Administration Guide*.



- 2** Create a Business Service step for your workflow using the FINS Web Shopping Service business service.
  - a** Enter the input arguments described in [Table 102 on page 394](#).
- 3** Create a subprocess that will invoke the SmartScript using the workflow User Interact step. The subprocess is based on the SmartScript Player business object and has three steps: a start step, a user interact step, and an end step.
  - a** In the user interact step, set the view name to “FINS SmartScript Player View (App)”.
  - b** In the Next Step’s applet, add two conditions, entering the information listed in the tables below:

Entries for Record 1 (first condition):

<b>Field</b>	<b>Entry</b>
Branch Name	Cancel
Type	Condition
Next Step	End
Event Object Type	Applet
Event	Invoke Method
Event Object	Smart Script Player Applet (FINS eSales App)
Subevent	CancelScript

Entries for Record 2 (second condition):

<b>Field</b>	<b>Entry</b>
Branch Name	Finish
Type	Condition
Next Step	End

Field	Entry
Event Object Type	Applet
Event	Invoke Method
Event Object	Smart Script Player Applet (FINS eSales App)
Subevent	FinishScript

- 4** Add a Business Service step using the FINS SmartScript Output Service business service. Enter the output arguments described in [Table 103 on page 395](#).

Now you may either end your workflow process or include additional steps, depending on your company’s business requirements. The SmartScript is integrated in your workflow process.

**FINS Web Shopping Service Methods**

The FINS Web Shopping Service business service methods are described in [Table 102](#).

**Table 102. FINS Web Shopping Service Methods**

Method	Description / Arguments
SetSmartScriptName	<p>Input Arguments:</p> <ul style="list-style-type: none"><li>■ Script Name (Required): The name of the SmartScript you want to invoke.</li><li>■ SSUserParameter# (Required): A SmartScript user parameter between 1 and 20. For example, “SSUserParameter1.” The value for the input argument must be the exact name of the user parameter specified in the SmartScript whose value needs to be passed into the workflow process at the conclusion of the SmartScript.</li></ul>

**FINS SmartScript Output Service Methods**

The FINS SmartScript Output Service business service methods are described in [Table 103 on page 395](#).

**Table 103. FINS SmartScript Output Service Methods**

Method	Description / Arguments
SetOutput	<p>Reads the SmartScript user parameters and outputs them as workflow properties:</p> <p>Output Arguments:</p> <ul style="list-style-type: none"><li>■ The name of the user parameter. Specifying the SmartScript user parameters as output arguments passes the value of the user parameters back to the workflow process when the SmartScript ends. These output arguments can be mapped to process properties.</li><li>■ SmartScript Result Action: Returns the action users take to end the SmartScript. Possible values are “Finish” or “Cancel.”</li></ul>

## Dynamic UI Service

This business service and associated administration views allow you to create and render views containing a single read-only applet in the Siebel Financial Services application. These views and applets, called dynamic applets, allow business analysts and developers to make frequent changes to the user interface without the need to recompile and redeploy a Siebel Repository File (SRF).

### Business Scenario

The scenario outlined in this section is an example of a Dynamic UI Service in a workflow process. Your company may follow a different workflow that fits its business requirements.

You would like to set up your company’s user interface to dynamically display read-only views at runtime, based on user input. For example, a customer requests a quote for an auto insurance policy. The policy includes the option to cover additional drivers. After the customer enters additional driver information, the workflow process uses a Dynamic UI applet to display a read-only list of the additional drivers for review. Customers can review the list, then select an option to continue the process.

To set up a similar user interface for your business, follow the instructions below.

## Creating a Dynamic Applet

### ***To create a dynamic applet***

- 1** Navigate to Site Map > Finance Administration > Dynamic Applet Administration.
- 2** Click the Dynamic Applet tab.
- 3** In the upper Dynamic Applet list, click New, then enter the view alias and view name. The view alias must be a unique name. The new name is a picklist of the views defined in the Template Definition view.
- 4** Step off the record. The Applet Instance form will automatically display the name of the applet template defined for this view.
- 5** In the Applet Instance form, enter the applet title, business object name, and business component name. The business object and business component will determine the fields to display for this applet.
- 6** If you want to set up a default search specification, enter:
  - Business component field. Specifies the field on which to base your search.
  - Default. Specifies a default value for which to search.
- 7** In the Field Control list, add a new record for each field you want to display. Each field control is displayed as a read-only text box with a text label. For each record, enter:
  - Sequence number. Indicates the display order for the field. For list applets, the order is left to right. For form applets, the order is left to right and top to bottom.
  - Field name. The business component field whose value you want to display.
  - Caption. The display label for the field.
- 8** In the Button Control list, add a new record for each button you want to display. Enter the sequence number, caption, and method for each button.
  - Sequence number. Indicates the display order for the button, from left to right.

- Caption. The display label for the button.
- Method. The method invoked when the button is clicked. The method name must begin with `FrameEventMethod`. For example, `FrameEventMethodContinue`.

## Displaying Dynamic Applets in a Workflow

### *To integrate the dynamic applet in a workflow*

- 1** Create a subprocess based on the FINS Dynamic UI business object.
- 2** Add a Dynamic UI Service business service step and invoke the Set View Name method described in [Table 104 on page 399](#).
- 3** (Optional) Add a second Dynamic UI Service business service step and invoke the AddRecord method described in [Table 104 on page 399](#).
- 4** Add a User Interact step to display the dynamic applet. For the view property of the User Interact step, enter the view template name defined in [Step 3 of “Creating a Dynamic Applet.”](#) For each button that may be clicked in the dynamic applet, add entries into the Next Steps list applet of the User Interact step. The following is an example of such an entry:

Field	Entry
Branch Name	Continue Button
Type	Condition
Next Step	End
Event Object Type	Applet
Event	InvokeMethod
Event Object	FINS Dynamic UI List Applet
Subevent	FrameEventMethodContinue

## Creating a View Template for Dynamic Applets

Creating a view template involves first creating views and applets based on specialized applet classes within Siebel Tools. These views and applets contain placeholder fields. Placeholder fields are applet controls that are not mapped to any particular business component. The views and applets will then form the basis of view templates and applet templates. A single view and applet template can be used to create any number of dynamic applets.

Two templates are provided as part of seed data: a view template containing a list applet template, and a view template containing a form applet template. The view templates are named as follows:

- FINS Dynamic UI Form view. A view template that contains a form applet template.
- FINS Dynamic UI view. A view template that contains a list applet template.

For more information about creating views and templates within Siebel Tools, see *Siebel Tools Reference*.

When creating dynamic applets in Siebel Tools, note the considerations listed below. (Use of the word “applet” in the bullets below refers to applets created in Siebel Tools.)

- When setting applet class, use:
  - CSSFrameListFADynamicUI for list applets
  - CSSFrameFADynamicUI for form applets
- If using an out-of-the-box business component, the applet business component should be FINS Dynamic UI.
- If you create your own Virtual Business Component (VBC) in Siebel Tools, the business class should be CSSBCVDynamicUI.
  - Dynamic Applet 7.5 supports 20 fields and five currency codes (that is, five types of currency value). In your new VBC, all field names should start with the prefix “Field” and a number value, (for example, “Field0”..., “Field19”). All currency code fields should start with the prefix “Currency Code” and a number value, (for example, “Currency Code0”..., “Currency Code4”).

- It is recommended that applets based on the same VBC have their corresponding views use a different business object.

Once the views and applets have been created in Siebel Tools, in order to define them as view templates and applet templates, they must be added to the Template Definition view.

**To add views and applets to the Template Definition view**

- 1** Navigate to Site Map > Finance Administration > Dynamic Applet Administration.
- 2** Click the Template Definition tab.
- 3** In the Template Definition applet, click New, and then enter the view name.
- 4** In the Applet Template Definition applet, click New, and then enter the applet name.

## Dynamic UI Service Methods

The Dynamic UI Service business service methods are described in [Table 104](#).

**Table 104. Dynamic UI Service Methods**

Method	Description / Arguments
Set View Name	Input Arguments: <ul style="list-style-type: none"> <li>■ View Alias (Required): The view alias defined in <a href="#">Step 3</a> of “Creating a Dynamic Applet” on page 396.</li> </ul>
AddRecord (optional)	Overrides the default search specification value. Input Arguments: <ul style="list-style-type: none"> <li>■ Business Component Name (Required): The name of the source business component.</li> <li>■ Object ID (Required): The row ID of the business component.</li> </ul>

## **Predefined Business Services**

*Predefined Business Services*



## Predefined Programs

# B

This appendix describes predefined programs.

# Predefined Programs

The following is a list of all predefined programs. These programs have been created from the five program types:

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

Table 105 contains common actions that you can use by inserting your own message text.

Table 105. Predefined Programs

Program	Description
<b>Send Page</b>	
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.
<b>Send Email</b>	
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.
<b>Message Broadcast</b>	
Send Broadcast Message	Send a generic message broadcast.
Send SR Message Broadcast	Send a message broadcast regarding a service request.

**Table 105. Predefined Programs**

<b>Program</b>	<b>Description</b>
Send Opportunity Message Broadcast	Send a message broadcast regarding an opportunity.
<b>Run External Program</b>	
Run External Program	Run an external program.
<b>Database Operation</b>	
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.

## Predefined Programs

*Predefined Programs*

# Siebel Workflow Terminology

# C

This appendix provides a glossary of workflow terminology.

## Siebel Workflow Terminology

[Table 106](#), [Table 107](#), and [Table 108](#) describe the common terms for Workflow Processes, Workflow Policies, and State Model.

**Table 106. Workflow Processes Terms**

Term	Definition
Arguments	Data passed to or received from a process or step.
Branch	A possible outcome of a workflow process step. A branch can have one or more conditions. A branch is followed by a step in the workflow process definition. If all of the conditions for the branch are met, the work item proceeds to the step following the branch.
Business object	A group of one or more business components. A business object represents an entity in the Siebel application that you would like to monitor. A workflow process is based on one and only one business object. Business objects are defined in Siebel Tools.
Business process	A process that is associated with operational objectives and business relationships. A business process is a set of one or more linked procedures, which collectively realize a business objective. An example of a business process is managing a new service request.
Business service	A type of step in a process in which an automated call is made to a service, such as the Outbound Communications service that handles inbound and outbound messaging. A workflow process definition can have one or more business service steps.
Connector	A definition of the relationship between two workflow process steps.
Decision point	A type of step in a workflow process definition in which the work item branches off to different steps depending on a set of conditions. A decision point consists of all possible branches for that point in the business process. Each branch consists of one or more conditions that must be met for a work item to follow that branch. A workflow process definition can have one or more decision steps.
End	A type of workflow process step that specifies when a process instance is finished.
Exception	A type of workflow process step that specifies when a process instance should follow an alternative branch instead of the normal branch path.
Process property	A storage field that contains values for use in steps as input and output arguments or for performing evaluations.

**Table 106. Workflow Processes Terms**

<b>Term</b>	<b>Definition</b>
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, query, 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.
Subprocess	A workflow process embedded into another workflow process as part of the workflow process definition. A subprocess has its own workflow process definition. A subprocess is a type of step. There can be one or more subprocess steps in a workflow process.
Wait	A type of workflow process step that specifies when a process instance should pause in execution and the duration of the pause.
Workflow process	The representation of a business process. A workflow process comprises one or more steps that indicate when a business process starts and ends and includes information about individual activities within the business process.
Workflow process instance	An instance of a workflow process that has been initiated. A process instance is initiated when the input conditions for a process definition have been met. A process instance consists of one or more step instances and contains one or more work items.
Work item	The representation of the work being processed in the context of a step within a process instance. A work item is an instance of a business object.

**Table 107. Workflow Policies Terms**

Term	Definition
Business object	A group of one or more business components. A business object represents an entity in Siebel that you would like to monitor. A workflow policy object is based on one and only one business object. Business objects are defined in Siebel Tools.
Business rule	The definition of how an organization wants to carry out a process in its operations.
Object type	An entity in Siebel Tools displayed as a node on the Object Explorer. For example, workflow policy objects, workflow policy components, workflow policy columns, and policy programs are all object types.
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.



**Table 107. Workflow Policies Terms**

Term	Definition
Workflow policy component	Components that define the Siebel database tables you would like to monitor. Workflow policy components also define the relationships between tables. Workflow policy components contain workflow policy columns. Workflow policy components are defined in Siebel Tools.
Workflow policy component column	A workflow policy column that is associated with a workflow policy component. Workflow policy component columns define the database columns that can be used in workflow policy conditions for a workflow policy. Workflow policy component columns are defined in Siebel Tools.
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 108. State Models 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 checks to see 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.



## Seeded Workflow Processes

# D

Seeded workflows are workflows that enable Siebel application logic in order to run specific Siebel applications. This appendix lists the seeded workflows shipped with Siebel applications.

# Seeded Workflows Provided with Siebel Applications

Seeded workflows have inactive status and version 0. They cannot be modified. To use a seeded workflow, you must revise it and activate it. This causes a new copy of the seeded workflow to be created with the latest version number.

**NOTE:** This is a list of all the seeded workflows provided with Siebel applications. For more information, see the application-specific documentation or contact Siebel Technical Support and provide the Group name (listed in [Table 109](#)) for the seeded workflow in question.

**Table 109. Seeded Workflow Processes Provided with Siebel Applications**

Group	Process Name
Analytics	Analytic Adapters
Content Center	Content Approval (Single Approver)
	Content Publish
	Content Publish to Files
	Content Task

**Table 109. Seeded Workflow Processes Provided with Siebel Applications**

Group	Process Name
Cross-Application	Account - New Order
	Account - New Quote
	Activity Filter
	Contact - New Order
	Contact - New Quote
	Contact Password Reset
	Import Assigned Fees Data
	Import Categories Data
	Import Charges
	Import Extension Data
	Import Fees Data
	Import Gallery Data
	Import Invoices
	Import Item Data
	Import Users Data

**Table 109. Seeded Workflow Processes Provided with Siebel Applications**

Group	Process Name
EAI	Order - Change SAP 4x Order Status
	Order - Create SAP 4x Order (SAP)
	Outbound Wireless Messaging Activity Dispatch
	Outbox Analysis Report Generator
	S2S Error Recovery Delete Root Process
	S2S Error Recovery Delete Root Sub-Process
	S2S Inbound Process
	S2S Outbound Process MQ
	S2S Outbound Process MQ - Pagination
	S2S Outbound Process MSMQ
	S2S Outbound Process MSMQ - Pagination
	S2S Outbound Sub-Process MQ
	S2S Outbound Sub-Process MQ - AMI
	S2S Outbound Sub-Process MQ - Pagination
	S2S Outbound Sub-Process MQ - Pagination - AMI
	S2S Outbound Sub-Process MSMQ
	S2S Outbound Sub-Process MSMQ - Pagination
	SAP 4x Account - Create SAP Customer (BAPI)

**Table 109. Seeded Workflow Processes Provided with Siebel Applications**

Group	Process Name
EAI/Oracle Connector 10.7	Account - Create or Update Oracle Customer
	Account - Receive Oracle Customer
	Account - Receive Oracle Customer Import Status
	Discount - Receive Oracle Discount
	InvLoc - Receive Oracle Organization
	Oracle Item (10.7) - Siebel Product
	Oracle Operating Unit (10.7) - Siebel Organization
	Oracle Organization (10.7) - Siebel InvLoc
	Oracle Receiver Dispatch Workflow
	Order - Create Oracle Order
	Order - Get Oracle Order Status
	Order - Receive Oracle Order
	Order - Receive Oracle Order Import Status
	Organization - Receive Oracle Operating Unit
	Pricing - Receive Oracle Price List
	Product - Receive Oracle Item

## Seeded Workflow Processes

*Seeded Workflows Provided with Siebel Applications*

**Table 109. Seeded Workflow Processes Provided with Siebel Applications**

Group	Process Name
EAI/Oracle Connector 11i	Account - Create or Update Oracle11i Customer
	Account - Receive Oracle11i Customer
	Account - Receive Oracle11i Customer Import Status
	InvLoc - Receive Oracle11i Organization
	Oracle Item (11i) - Siebel Product
	Oracle Operating Unit (11i) - Siebel Organization
	Oracle Organization (11i) - Siebel InvLoc
	Order - Create Oracle11i Order
	Order - Get Oracle11i Order Status
	Order - Receive Oracle11i Order
	Organization - Receive Oracle11i Operating Unit
	Product - Receive Oracle11i Item
eChannel	eChannel - Cancel Order Process
	eChannel Go to Current Quote View (eSales)
	eChannel Shopping Cart Scenarios Workflow
eEvents	Event Cancellation Email (eEvents)
	Event Confirmation Email (eEvents)
	Event Invitation Email (eEvents)
	Event Waitlisted Email (eEvents)



**Table 109. Seeded Workflow Processes Provided with Siebel Applications**

Group	Process Name
eLearning	Class Confirmation Email (eTraining)
	Class Reminder Email (eTraining)
	Class Waitlisted Email (eTraining)
	Course Confirmation Email (eTraining)
	eTraining - Approve Third Party Registration
	eTraining - Checkout Enrollment Process
	eTraining - Class Enrollment Process
	eTraining - Class Reg Enroll Process
	eTraining - Course Enrollment Process
	eTraining - Course Reg Enroll Process
	eTraining - Decline Third Party Registration
	eTraining - Drop Class Process
	eTraining - Drop Course Process
	eTraining - Drop Enrollment Process
	eTraining - Goto Shopping Cart Process
	eTraining - Show Course Details Process
	eTraining - Third Party Registration - Submit
	eTraining Enrollment Process
	eTraining Leftover Course View
	Show Course Detail (Int Prod) Process
	Show Course Detail Add To Cart Process
	Show Course Detail Favorite List
	Show Course Detail Quote Comparison
	Show Course Detail Quote Template

## Seeded Workflow Processes

*Seeded Workflows Provided with Siebel Applications*

**Table 109. Seeded Workflow Processes Provided with Siebel Applications**

Group	Process Name
eMail Response	eMail Response - Analyze Message
	eMail Response - Append Thread Id
	eMail Response - Client Send Email
	eMail Response - Create Activity
	eMail Response - Get Entitlement Id
	eMail Response - Identify Language
	eMail Response - Parse Junk Email
	eMail Response - Process Message
	eMail Response - Process Service Request
	eMail Response - Response Workflow
	eMail Response - Route Email
	eMail Response - Search Spec
	eMail Response - Send Acknowledgement
	eMail Response - Send Auto Response
	eMail Response - SR Help
	eMail Response - SR Query
	eMail Response - SR Status
	eMail Response - SR Submit
	eMail Response - SR Update
	eMail Response - Update Activity Status
ERM	Inbound Communication - Voice Route Request
	Group News Approver Workflow

**Table 109. Seeded Workflow Processes Provided with Siebel Applications**

Group	Process Name
ISS/eAuction	eAuction Alert Notice
	eAuction Autobid Notice
	eAuction Bid Notice
	eAuction Close Invoice
	eAuction Export XML file
	eAuction Privilege Setting
	eAuction Lister Bid Accepted Notice
	eAuction Lister Close Notice
	eAuction Lister Invoiced Notice
	eAuction Outbid Notice
	eAuction Partially Outbid Notice
	eAuction User Status Notice
	eAuction User Welcome Notice
	eAuction Win Notice
	eAuction Wireless Notice
	eAuction XML Bulk Upload
	Import Auction Rating Data
	Import Auction Watch Data
	Import Bids Data

## Seeded Workflow Processes

*Seeded Workflows Provided with Siebel Applications*

**Table 109. Seeded Workflow Processes Provided with Siebel Applications**

Group	Process Name
ISS/eSales	Credit Check - Orders
	Credit Check - Quotes
	eSales - Add To Template Process
	eSales - Approve Order Process
	eSales - Authorize Order Process
	eSales - Cancel Order Process
	eSales - Checkout Process
	eSales - Complete Checkout Process
	eSales - Current User Type
	eSales - Delete Old Anonymous Quotes
	eSales - Display Error Process
	eSales - Edit Quote Process
	eSales - Enter Payment Details Process
	eSales - Enter Shipping Details Process
	eSales - Login View
	eSales - Modify Order Process
	eSales - Order Confirmation View
	eSales - Order Quote Process
	eSales - Power Customer Quotes View
	eSales - Reject Order Process
	eSales - Reorder Process
	eSales - Reprice Process
	eSales - Save Account Shipping Address
	eSales - Save Cart Process
	eSales - Save Credit Card

**Table 109. Seeded Workflow Processes Provided with Siebel Applications**

Group	Process Name
ISS/eSales, cont.	eSales - Save Shipping Address
	eSales - Set Primary Account Address
	eSales - Setting PO Status
	eSales - Shopping Cart View
	eSales - Silent Authorize Order Process
	Send Order Accepted Email (eSales)
	Send Order Awaiting Approval Email (eSales)
	Send Order Failed Email (eSales)
	Send Order Rejected Email (eSales)
	Send Order Shipped Email (eSales)
	Shipping Calculation Process
License Keys	LicCustomerNotifyNU
	LicCustomerNotifyUH
	LicUsageSummaryArchive
Marketing	Automatic Data Retrieval
	Campaign Launch
	Campaign List Association
	Campaign Load
	Import Marketing Contact
	Marketing Plan Recalculation
	Program Execution

## Seeded Workflow Processes

*Seeded Workflows Provided with Siebel Applications*

**Table 109. Seeded Workflow Processes Provided with Siebel Applications**

Group	Process Name
Order Management	Goto_Order
	Goto_Quote
	SIS OM Active Quote Sub-process
	SIS OM Apply Completed Service Order Line Item to Service Profile
	SIS OM Auto Select Order Billing and Service Accounts
	SIS OM Auto Select Quote Billing and Service Accounts
	SIS OM Disconnect Asset Sub-process
	SIS OM Disconnect Products & Services Process
	SIS OM Edit Complex Asset Workflow
	SIS OM Edit Delta Quote Line Item
	SIS OM Edit Service Order Line Item
	SIS OM Go to Products & Services Sub-Process
	SIS OM Go to Quote Detail View Sub-Process
	SIS OM Modify Products & Services Process
	SIS OM New Products & Services Process
	SIS OM Profile Process
	SIS OM Quote To Order Workflow - PMT Version
	SIS OM Submit Order Process
	SIS OM Ungroup Quote

**Table 109. Seeded Workflow Processes Provided with Siebel Applications**

Group	Process Name
PeopleSoft Connector	Account - Receive PeopleSoft Customer
	Account - Send PeopleSoft Customer
	Employee - Receive PeopleSoft Employee
	Employee - Send PeopleSoft Employee
	Position - Receive PeopleSoft Position
	Position - Send PeopleSoft Position

**Table 109. Seeded Workflow Processes Provided with Siebel Applications**

Group	Process Name
PRM ASI	Opportunity Sharing Build Siebel Message
	Opportunity Sharing External Create
	Opportunity Sharing External Query
	Opportunity Sharing External Update
	Opportunity Sharing Inbound or Outbound
	Opportunity Sharing Pick Partner
	Opportunity Sharing Receive Create Response
	Opportunity Sharing Receive Update Response
	Opportunity Sharing Set Update Status
	Opportunity Sharing Siebel Create
	Opportunity Sharing Siebel Query
	Opportunity Sharing Siebel Update
	PRM ANI Inbound Addressing Change
	PRM ANI Inbound Create Account Process
	Transfer Cart Inbound Create Cart Process
	Transfer Cart Inbound Create Contact Process
	Transfer Cart Inbound Create Quote Process
	Transfer Cart Inbound Receive Process
	Transfer Cart Outbound Create and Append Process
	Transfer Cart Outbound Create Header Process
	Transfer Cart Outbound Initial Process
	Transfer Cart Outbound Receive Acknowledgment Process
	Transfer Cart Outbound Request Process



**Table 109. Seeded Workflow Processes Provided with Siebel Applications**

Group	Process Name
Sales	Add Time Item to Time Sheet
	Create Rerouted Oppty Activity
	eAdvisor Opportunity Confirmation Nav
	eAdvisor Opportunity Creation
	eAdvisor Opportunity Find Product
	eAdvisor Opportunity Integration
	eAdvisor Opportunity Integration (Homepage Nav)
	eAdvisor Opportunity User Profile
	eAdvisor Opportunity User Registration
	Expense Report AutoGen
	Expense Report Process
	Get Manager Email
	Go to My Time Sheets view
	Opportunity Assigned Notification Process
	Opportunity Inactive Notification Process
	Opportunity Lost Notification Process
	Opportunity Pending Notification Process
	Opportunity Won Notification Process
	Time Sheet AutoGen
	Time Sheet Process
	Update Expense Status
	Update Time Sheet Status

## Seeded Workflow Processes

*Seeded Workflows Provided with Siebel Applications*

**Table 109. Seeded Workflow Processes Provided with Siebel Applications**

Group	Process Name
SAP 4x Account	SAP 4x Account - Update SAP Customer (4x IDOC)
	SAP 4x Account - Update SAP Customer (IDOC)
	SAP 4x Account - Update Siebel Account (4x IDOC)
	SAP 4x Account - Update Siebel Account (IDOC)
SAP Account	Account - Get SAP Order List
	Account - Get SAP Order List (MO)
	Account - Import SAP Order
	Account - Import SAP Order (MO)
	Account - Import SAP Order and Get SAP Order Status
	Account - Import SAP Order and Get SAP Order Status (MO)
	Account - Receive SAP Customer
SAP Connector	Product - Receive SAP 4x Material (4x IDOC) - DDTE
	Quote To Order SAP 4x Submit
	Quote To Order SAP 4x Validate
	SAP 4x Account - Create or Update SAP Customer
	Send4xIdoc_UpdateSAPCustomer_MQAMI
SAP IDOC	SAP 4x IDOC Processor
	SAP IDOC Processor

**Table 109. Seeded Workflow Processes Provided with Siebel Applications**

Group	Process Name
SAP Order Entry	Order - Create SAP 4x Order
	Order - Create SAP Order
	Order - Create SAP Order and Get SAP Order Status
	Order - Get SAP 4x Order Status
	Order - Get SAP Order Status
	Order - Get SAP Order Status (MO)
	Order - Simulate SAP 4x Order
	Order - Simulate SAP Order
SAP Product	Product - Receive SAP 4x Material
	Product - Receive SAP 4x Material (4x IDOC)
	Product - Receive SAP Material
Service	Siebel Message send email
	Siebel Message send email for activity owner
Siebel Service Wireless	Inbound Wireless Messaging Activity Dispatch
UAN	Get Account ASI
	Get Order Status ASI
	Submit Order ASI
	Submit Quote ASI
	Synchronize Account ASI
	Synchronize Contact ASI
	Synchronize Employee ASI
	Synchronize Household ASI
	Synchronize Position ASI
	Synchronize Simple Product ASI

## Seeded Workflow Processes

*Seeded Workflows Provided with Siebel Applications*

**Table 109. Seeded Workflow Processes Provided with Siebel Applications**

Group	Process Name
Universal Queuing	UQ HTTP Processing
User Registration	User Registration Company Information (SCW)
	User Registration Forgot Password Process
	User Registration Individual Information (SCW)
	User Registration Initial Process
	User Registration Process
	User Registration SubProcess
WCS Connector	Contact - Receive WCS Customer
	Order - Receive WCS Order
	Product Catalog - Create WCS Catalog
	WCS MQ Processor
Wireless	Wireless Activity Alert
	Wireless eService SR Alert
	Wireless Opportunity Alert
	Wireless Service Request Alert

## Performance Tuning

# E

This appendix describes ways to tune Workflow performance.

## Tuning of Workflow Processes

In order to improve performance when running workflow processes, you can follow the guidelines explained in this section.

---

**NOTE:** This performance tuning information is provided as general guidelines for tuning and optimizing performance of workflow processes. Every implementation of Siebel applications is unique, so every use of workflow processes is also unique.

---

### Minimizing Usage of SearchSpec

Although SearchSpec is a feature of Workflow, it is recommended that you minimize your use of this feature with workflow processes that are invoked very frequently. Minimizing your use of SearchSpec, especially for frequently invoked processes, improves Workflow engine performance during runtime because the engine does not have to construct the SearchSpec string.

It is important to note, however, that you should not completely avoid using SearchSpec, because using no SearchSpec can indicate actions taking place on the current row in some cases, and on all rows in other cases. For specific guidelines, note the following:

- For Siebel operations, minimize usage of SearchSpec.
- For batch process requests, use a SearchSpec that matches as few rows as necessary to get the job done.

### Indexing Fields in SearchSpec

If you determine that SearchSpec indeed needs to be used, it is recommended that you make sure all the fields being used are properly indexed. Proper indexing of the fields helps Workflow and the underlying database to efficiently build queries.

## Conditions Based on Parent and Child Business Components

When a condition is being evaluated at a decision step or any other step using a combination of parent and child business components, it is recommended that you closely benchmark the expression or the condition. In some cases, this will require spooling the SQL.

---

**NOTE:** The explain plan of the SQL might show an extended and poorly performing query. In such cases, it is better to break the conditions up into multiple decision steps and evaluate the conditions separately.

---

## Configuration Versus Workflow Comparison

In some cases, you may need to perform a comparison between different objects.

For example, you assign a service request to a candidate depending on the industry of the account associated with it. In this case, it is necessary to perform a query against Account to fetch the appropriate industries, or to check an industry against all the industries with which the account is associated.

If the workflow process in this example is going to be evaluated frequently, consider exposing Account Industry on Service Request by the appropriate configuration in order to enhance workflow performance.

## Overheads

Overhead varies depending on whether you are running workflows locally or in Workflow Process Manager.

### Running Workflows Locally

A workflow instance—that is, one run of a workflow definition—can run within a UI object manager (UI/OM). In this case, the workflow runs locally, within the current thread that the logged-in user is using. This means that if N users are connected and they all need to run a workflow definition, the definition would run in that user thread. In this mode, Workflow adds a fixed overhead (100-200K) to the user memory (usually referred to as the model) plus memory taken up by other objects (such as business components) contained in the tasks within that workflow.

## Running Workflows in Workflow Process Manager

The workflow itself runs within a separate object manager (the application server process), which uses a fixed set of resources (MaxMTS, MaxTasks) to schedule the workflow. The Workflow Process Manager (WFProcMgr) is a multithreaded process that runs multiple workflows and is more scalable because it uses a pool of threads and models.

Generally, the mode of the workflow used depends on what the application is trying to achieve. It is generally recommended that you try to schedule a workflow task in the WFProcMgr, especially if the results of a run are not immediately needed.

### Asynchronous Mode

Use of the asynchronous mode comes with the following pros and cons:

#### Pros

- All user threads are not loaded.
- More scalable as long as:
  - There are maximum N simultaneously connected users
  - There are maximum X simultaneous running workflows
  - If X is smaller than N, then a WFProcMgr with X tasks can handle a much larger pool (N) of users

#### Cons

- On error, you must look at the log files because there is no automatic notification.
- The SRM could have a timeout or retry feature.
- Slightly more latency. Additional cost (minimal) of one request per response.



## Tuning of Workflow Process Manager

This section provides general approaches to tune and optimize performance of Workflow Process Manager. It is imperative to remember that every implementation of Siebel applications is unique, and so every use of workflow processes is also unique. It is in your best interest to test, continually monitor, and tune your workflow processes to achieve optimal throughput.

---

**NOTE:** The information provided in this section should be considered general background information. No attempt is made to detail the many variables that will affect tuning at specific customer sites, so this is not a substitute for specific tuning recommendations made by Siebel Technical Support.

---

### Caching Business Services

The business services invoked through Workflow Process Manager should have the Cache property set to TRUE. This makes it possible for the Workflow engine to not reload and re-parse the business service. This enhances the performance of workflows that invoke business services.

---

**NOTE:** Predefined Siebel business services that have the Cache property set to OFF should *not* be reset to TRUE.

---

### MaxMTServers and MinMTServers

Maximum MT Servers (MaxMTS) is the maximum number of Instances of Component (processes) that can be run for a particular component on a particular server. MaxMTS is also used to calculate how many tasks can be run on each Instance of Component.

Minimum MT Servers (MinMTS) is only used when the Siebel Server or a particular component is started. MinMTS determines how many processes to start up for this component.

---

**NOTE:** You should set MinMTServers = MaxMTServers. Setting MinMTServers less than MaxMTServers can cause a delay of service for new users as additional MTServers get initialized.

---

Suggested settings are as follows:

- $\text{MaxMTServers} = \text{< target \#users >} / \text{< users/OM >}$

- $\text{MaxTasks} = 20\% \text{ target \#users}$

For example, for 500 users, set MaxMTServers = 5 and MaxTasks = 100.

- $\text{MinSharedDbConns} = 10\% \text{ target \#users}$

- $\text{MaxSharedDbConns} = 10\% \text{ target \#users}$

- $\text{MinTrxDBConns} = 10\% \text{ target \#users}$

- If applicable, you can set BypassHandler = TRUE for any single component to attain approximately 8% throughput gain.

---

**NOTE:** With the setting BypassHandler = TRUE, there will be no regular component logging unless the component experiences difficulties and generates an error message.

---

---

**NOTE:** You should closely monitor the WFProcMgr process usage during a steady state run, and then adjust the parameters according to your observations.

---

## MaxTasks

MaxTasks is the maximum number of concurrent user sessions that can run on a server at one time. An MTServer is equivalent to a single process of the Object Manager.

MaxTasks/Max Server is the maximum number of tasks running in one instance of MT Server, as in the following example:

Min MT Server = Max MT Server = 5

Max Tasks = 20

Max Task per MT server =  $20 / 5 = 4$  tasks on each MT Server

### **SIEBEL\_OSD\_LATCH and SIEBEL\_OSD\_NLATCH**

It is very important that you make sure to keep the server running under 80% CPU utilization. If there are any changes on MaxTasks, change SIEBEL\_OSD\_LATCH and SIEBEL\_OSD\_NLATCH based on the following operations:

- $\text{SIEBEL\_OSD\_LATCH} = 2 * (\text{Accumulated MaxTasks of all components})$
- $\text{SIEBEL\_OSD\_NLATCH} = 10 * (\text{Accumulated MaxTasks of all components})$

For example, on an application server box, you have enabled two multithreaded server components: CallCenter\_enu, and WFProcMgr. MaxTasks = 500 for the component CallCenter\_enu and MaxTasks = 100 for the WFProcMgr component. So SIEBEL\_OSD\_LATCH should be set accordingly to 1200 (as a result of  $2 * [500 + 100]$ ) and SIEBEL\_OSD\_NLATCH should be set accordingly to 6000.

## **Other Critical Parameters**

To improve scalability, use the parameter information in this section. For more detailed information on the parameters in this section, see *Siebel Server Administration Guide*.

### **OM-Model Cache Maximum**

This parameter determines the size of the cache for model objects in Object Manager-based server components, such as Business Service Manager and Workflow Process Manager. Each model in the cache creates two database connections for the life of the model (one connection for insert, update, and delete operations; the other connection for read-only operations).

The model cache improves performance for the OM-based server components. More cache helps throughput, but only in asynchronous mode. Keep in mind, however, that this costs memory. The volume of the model cache should be determined case by case. In general, use the default value.

---

**NOTE:** It is generally recommended that you do not disable the OM-Model Cache Maximum parameter. The default value is 10. A value of 0 disables this parameter, and the maximum setting for this parameter is 100.

---

### **HonorMaxTask=TRUE**

If HonorMaxTasks is set to TRUE, then once MaxTasks is reached, all further requests are rejected instead of being queued.

### **Recycle Factor**

Recycle Factor allows WfProcMgr processes to be restarted periodically.

# **Passing Parameters to and from Workflow and Data Manipulation within Workflows**

# **F**

This appendix explains how to manipulate data within workflows and how to pass parameters to and from Workflow.

## Manipulating Data Within Workflows

You can manipulate data within workflows. This section explains how to access runtime event parameters from workflows.

### Accessing Runtime Event Parameters from Workflows

#### ***To access runtime event parameters from workflows***

- 1 Set up a runtime event for each applet that is required to trigger a workflow process.

Fill in the fields of the Next Steps applet. Example field values are shown in the following table.

Field	Value
Branch Name	Enter a name for the branch.
Type	Condition
Next Step	Enter a name for the next step.
Event Object Type	Applet
Event	InvokeMethod
Event Object	Choose the name of the event object that will trigger the workflow.
Sub Event	NewRecord
Comments	Optional
Event Cancel Flag	The default is blank.

- 2 Set up your process properties in Process Designer as usual. For information on using process properties, see [“Using the Process Properties View” on page 59](#).

- 3 Drill down into the first step in your workflow and add a new output argument for each process property you want to populate using runtime events. Add new output arguments by filling in the fields of the Output Arguments applet. Example field values are shown in the following table.

Field	Value
Property Name	Enter a name for the property.
Type	Expression
Value	GetProfileAttr("RestrucOut")  Note: As you are filling in the Profile Attribute field, make note of the name you give to each of the Profile Attributes property values, for use in <a href="#">Step 8 on page 440</a> .
Output Argument	Leave as default.
Business Component Name	Leave as default.
Business Component Field	Leave as default.
Comments	Optional

- 4 Activate the workflow process. For more information, see [“Activating a Process” on page 158](#).
- 5 Since the process includes runtime events, you must load the runtime events.
  - a Navigate to Runtime Events Administration.
  - b From the applet drop-down menu, select Reload Personalization.
- 6 From the Events view tab, find the runtime event that is associated with the workflow runtime event that you created. As an example, you can query for the InvokeMethod event on the Opportunity List Applet object.

- 7 Drill down on the Action Set Name. The Actions view is shown in the following table.

Actions					
New Query 1 - 5 of 5					
Name	Action Type	Sequence	Active	Start Date	End Date
Set Trans Id	Attribute Set	1	✓		
Set Response Has Row	Attribute Set	2	✓		
Set Restructure Out	Attribute Set	3	✓		
Set Restructure In	Attribute Set	4	✓		
Workflow_1IS-1DJRT	BusService	5	✓		

More Info		
Query 2 of 5		
*Name:	Profile Attribute:	Business Service Name:
Set Response Has Row	RespHaveRows	
*Sequence:	Set Operator:	Business Service Method:
2	Set	
*Active:	Value:	Business Service Context:
<input checked="" type="checkbox"/>	0	
Conditional Expression:	Set Minimum:	
Start Date:	Set Maximum:	

- 8 Add a new Action for each process property that will be populated using runtime events. For example, you can create a new action to set the ACU Transaction ID and call it Set ACU Trans ID.

- a For each new action:
- ❑ Set the Action type to Attribute Set.
  - ❑ Set the Profile Attribute to match the value you used in the GetProfileAttr call in the workflow designer in [Step 3 on page 439](#), such as TransType.
  - ❑ Set the Set Operator value to Set.
  - ❑ Assign the Value field the appropriate value, such as the literal string FA-0001. You assign the value using the standard Siebel expression builder.
  - ❑ Set the Sequence for the action to be less than the default sequence (for the Workflow\_XXXXXXX Action).



- b** Make sure the Sequence setting of the Workflow\_XXXXXXX Action is the highest number so that this action happens after all the other actions.

---

**NOTE:** Whenever you modify the workflow, you must repeat this step because modifying a workflow resets the workflow action's sequence to 1.

---

- c** From the applet drop-down menu, select Reload Personalization.

## Passing Parameters from a Business Service to a Workflow

The workflow engine can be invoked as a standard Siebel business service. You can run a workflow by invoking the Workflow Process Manager business service and passing an input property set to the business service. The input property set is required to contain a property named `ProcessName`, which specifies the name of the workflow process to be run. In addition to the `ProcessName` property, you can put other values, such as strings, numbers, and property sets, into the property set. These values will be passed to the workflow process by the Workflow Process Manager business service.

You can also use scripts to pass parameters.

### Example Scripts for Passing Parameters

The examples in this section show script that you can use to pass parameters.

#### Example 1: Invoking Workflow Programmatically and Constructing an Input Property Set

The following is a sample script which programmatically invokes the Workflow Process Manager business service and which constructs an input property set, `psInputs`, for the business service. This script defines strings that will be put into the input property set as properties.

```
var msgName = "Siebel Agent Authorization Retrieval";  
var reqSubType = "CICS Services Request";  
var reqType = "AgentAuthorizationReq";  
var CICSServiceName = "Consumer Auto Agent Authorization Retrieval";  
var processName = "Consumer Auto VBC VBC Template";  
var reqFileName =  
"C:\\sea752\\XMLMessages\\AgentAuthorizationVBCReq-final.xml"  
var resFileName =  
"C:\\sea752\\XMLMessages\\AgentAuthorizationVBCResponse-final.xml"
```

### **Example 2: Defining Property Sets for the Input Property Set**

The following is a sample script that defines property sets, which will be put into the input property set as child property sets.

```
//Request PS  
  
var psRequest = app.NewPropertySet();  
  
var psAgentNumTag = app.NewPropertySet();  
  
var psType = app.NewPropertySet();  
  
var sAgentID;
```

### **Example 3: Constructing Property Sets**

The following is a sample script that constructs property sets.

```
//Build property set hierarchy  
  
sAgentID = app.LoginName();  
  
psRequest.SetType("XMLHierarchy");  
  
psAgentNumTag.SetType("DataAgentNumber");  
  
psAgentNumTag.SetValue(sAgentID);  
  
psRequest.AddChild(psAgentNumTag);
```

### **Example 4: Assembling Properties and Child Property Sets into the Input Property Set**

The following is a sample script that assembles properties and child property sets into the input property set.

```
psInputs.AddChild(psRequest);//Pass in Property Set  
  
psInputs.SetProperty("RequestURLTemplate", requestURLTemplate);//  
Pass in string  
  
psInputs.SetProperty("RequestSubType", reqSubType);  
  
psInputs.SetProperty("ReqType", reqType);  
  
psInputs.SetProperty("MessageName", msgName);
```

```
psInputs.SetProperty("CICSServiceName", CICSServiceName);  
psInputs.SetProperty("ProcessName", processName);  
psInputs.SetProperty("Request File Name", reqFileName);  
psInputs.SetProperty("Response File Name", resFileName);
```

### **Example 5: Invoking the Workflow Process Manager Business Service and Passing It the Input Property Set**

The following is a sample script that invokes the Workflow Process Manager business service and passes it the input property set.

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

## Passing Parameters from Workflow to Global Variables (Profile Attributes)

You can use a business service to access and pass parameters from Workflow to global variables.

### **To access workflow parameters for a running workflow process**

- 1** Define a business service with relevant methods and parameters.
- 2** Access the business service from the workflow process.
- 3** In the business service step in the workflow process, pass the workflow process properties to the business service method arguments. For more information, see [“Working with Business Service Steps” on page 94](#).
- 4** Use the following script to take the business service argument values and assign them to Profile Attributes.

```
function Service_PreInvokeMethod (MethodName, Inputs, Outputs)
{
if( MethodName == "XXX" ) {
var isWorkflowRunning, viewValidCurrent, viewValidNext;
// read the input arguments into profile attributes
isWorkflowRunning = Inputs.GetProperty("Workflow Running");
viewValidCurrent = Inputs.GetProperty("Valid View Current");
viewValidNext = Inputs.GetProperty("Valid View Next");
TheApplication().SetProfileAttr("WFRunning", isWorkflowRunning);
TheApplication().SetProfileAttr("WFViewCurrent", viewValidCurrent);
TheApplication().SetProfileAttr("WFViewNext", viewValidNext);
}
```

- 5 Use the profile attributes for further processing. All the necessary information has been put into the profile attributes of the application. Users can use the standard procedures for accessing the profile attributes to extract this information. For more information, see *Personalization Administration Guide*.

# Index

## A

- action parameters, defined 40
- ActionAgent parameter 326
- ActionInterval parameter 326
- actions
  - creating for workflow policies 227
  - examples of for workflow policies 228
  - working with for workflow policies 215
- Actions applet
  - field descriptions, Actions view 216
  - field descriptions, Policies view 248
- activating processes, procedure 158
- activity templates module, described 29
- All Processes view, screen example and list of tasks 55
- architecture of workflow components 32
- arguments
  - action parameters for policies 40
  - common values for policy programs 290
  - Create E-mail Activity program 257
  - creating argument values 294
  - creating SQL statements for program 296
  - Database Operation program 224
  - definition 406
  - Policy Program List Argument view, about 289
  - programs, example of creating 295
  - Run External program 223
  - Send E-Mail program 220
  - Send Message Broadcast program 221
  - Send Page program 218
- Assign Non-Respondents policy, creating 262
- Assign to Campaign E-mail action 260

- Assign to Campaign program 257
- Assignment Manager module, described 29
- Assignment Request, Workflow Policy program about 41
- asynchronous server requests 375
- audience for guide 15

## B

- Batch feature 230
- Batch field for workflow policies 250
- batch mode, running workflow processes 185
- BatchMode parameter 327
- branch
  - Decision step, defining 89
  - definition 406
  - next step branches, defining 78
  - Start step types 74, 121, 130
  - user interact next step branches, defining 134
  - using the Next Step applet, Start steps 79, 135
- Branches applet, Decision view fields 85
- business analyst, role of 33
- business objects
  - definition of 406
  - primary component, defining 49
- business policies
  - creating policy actions 227
  - creating policy groups 239
  - creating workflow policies 249
  - gathering information on 46
- business process, definition of 406
- business processes

*See also* workflow processes, designing  
 Process Properties applet fields, table  
     of 59  
 Process Properties view 59  
 runtime events, about using to  
     automate 182  
 business rule, defined 408  
 business service  
     definition 406  
     using in processes 94  
 Business Service applet  
     field descriptions 95  
     using to define a Business Service  
         step 98  
 Business Service step  
     basic details, defining 98  
     input arguments, defining 99  
     output arguments, defining 100  
 Business Service view  
     Business Service applet fields 95  
     Input Arguments applet fields 96  
     Output Arguments applet fields 97  
     using 94  
 Business Service, Workflow Policy Action,  
     executing from, note 227  
 business services, enabling workflow  
     for 50  
 business services, predefined  
     asynchronous server requests 375  
     Outbound Communications  
         Manager 374  
     synchronous server requests 375  
     Workflow Utilities, arguments 378

## C

cache, state model 371  
 calculated fields, updating 112  
 campaigns  
     Create Email Activity program 257  
     marketing campaigns, scenario with  
         Workflow policies 258  
 comparison operations  
     Conditions applet, Decision step 87

    Conditions applet, Start step 76, 132  
 comparison values  
     entering date calculations 248  
     specialized 245  
     standard 243  
     using in the Conditions applet 243  
 conditions  
     defining in Decision steps 90  
     defining in Exceptions 153  
     defining in Start steps 79, 135  
     expressions, format of 93  
 Conditions applet  
     field descriptions, Decision view 86  
     field descriptions, Policies view 243  
     field descriptions, Start view 75, 131  
     using comparison values 243  
     using specialized comparisons 245  
     using standard comparisons 243  
 Configurator workflow role, described 33  
 connector  
     defining for Decision branches 89  
     defining for Start branches 78, 134  
     definition of 406  
     diagraming a process, using for 67  
     Exceptions, for 153  
     point in connector, removing or  
         adding 67  
 copying, workflow process or step 157  
 Create Email Activity  
     about and arguments 257  
     creating 260  
 creating  
     database triggers for workflow  
         policies 306  
     workflow policies 249  
     workflow policy actions 227  
     workflow policy groups 239

## D

database  
     sample database, launching 36  
     Siebel database, described 32  
     specialized comparisons 247



- triggers, creating 306
- Workflow Policies database tables 321
- Database Operation, Workflow Policy
  - program about 41
- date calculations, comparison values 248
- debugging, workflow processes 175
- Decision applet
  - field descriptions 84
  - using to define a Decision step 89
- Decision Branches view
  - Branches applet fields 85
  - Conditions applet fields 86
  - Decision applet fields 84
  - tasks 83
  - using to define workflow process
    - Decision steps 89
- Decision steps
  - basic details, defining 89
  - branches, defining 89
  - conditions, defining 90
  - definition of 406
  - values, defining 90
  - working with 83
- default SQL statements 302
- DeleteSize parameter 327
- deleting
  - processes 157
  - workflow process instances 194
  - workflow process steps 157

## E

- email
  - consolidating for recipient 230
  - consolidating for recipient, about using
    - batch mode 250
  - creating Workflow policy for 252
  - testing 253
- Email for CD-ROM Campaign policy,
  - creating 261
- Email Manager
  - parameters for 315
  - setting up on Siebel Server 314
  - troubleshooting 319

- End Arguments view
  - End Step applet fields 145
  - Output Arguments applet fields 112, 146
  - using to define workflow process End
    - steps 147
- End step
  - about working with 145
  - applet field descriptions 145
  - defining an end step 147
  - definition of 406
- End User, workflow role, described 33
- error code
  - conditions, not using in 153
  - defined 68
- Error Message
  - conditions, using in 153
  - defined 68
- errors, correcting process and
  - resuming 192
- events, tracing and logging 195
- Exceptions
  - defining 153
  - definition of 406
- exporting
  - multiple process definitions 189
  - process definition 188
- expressions, format of 93
- External Program, Workflow Policy program
  - about 41

## F

- field name modifications 281
- Fields applet
  - defining workflow process Siebel
    - Operation steps 115
  - field descriptions 110
- flowchart tool
  - See Process Designer view

## G

- Generate Trigger (GenTrig)
  - component-specific parameters for 308

- functions of 306
- tips for running 307
- Generic Request Server, Workflow Policy program, about 41
- GenReqRetry parameter 327
- GroupName parameter 327
- groups
  - creating for workflow policies 239
  - planning policy groups 202
  - working with for workflow policies 237

## I

- IgnoreError parameter 328
- importing, process definitions 189
- input arguments
  - defining for business service steps 99
  - Stop steps, defining 143
  - Sub Process step, defining 106
  - Wait step, defining 124
- Input Arguments applet
  - field descriptions, Business Service view 96
  - field descriptions, Stop view 142
  - field descriptions, Sub Process view 104
  - field descriptions, Wait view 123

## K

- KeepLogDays parameter 328

## L

- LastUsrCacheSz parameter 328
- license key validation 52
- logging events, table of 195

## M

- MailTo parameter 328
- marketing campaigns, scenario with Workflow policies 258
- Message Broadcast
  - Arguments applet, about and arguments and values 221
  - workflow policy program type 217

- migrating
  - policies, suggested strategies for 211
  - production environment, workflow policies to 344
  - workflow processes 188
- migration strategy, defining workflow processes 51
- modifying
  - policy column and policy field names 281
  - Workflow policy object 283
- monitoring
  - Workflow Monitor Agent, viewing executed policies 349
  - Workflow Policies processes 348
- multiple conditions, defining for each branch 82, 138
- multiple records, executing actions in a workflow processes 185
- multi-value groups, updating fields based on 112

## N

- name, modifying policy column and policy field names 281
- Next Step applet
  - defining conditions and values for Start steps 79, 135
  - field descriptions, Start view 74, 121, 130
  - using for exceptions 152

## O

- Object Explorer view
  - workflow policy object, defining 280
  - workflow policy objects, adding to 264
- Object Id, default process property 68
- Object Manager
  - running workflow processes 185
  - workflow process, invoking process example 180
- object type

- definition in relation to Siebel Tools 264
- program types, table of 288
- workflow policies terms, defined 408
- workflow policy objects, modifying 283
- workflow policy program, viewing with
  - Program List view 288
- object types for workflow policy 409
- organization of guide 17
- Outbound Communications Manager
  - available methods, described 374
- output arguments
  - Business Service steps, defining 100
  - End step, defining 117, 147
  - Sub Process step, defining 107
- Output Arguments applet
  - field descriptions, Business Service
    - view 97
  - field descriptions, End Step view 112, 146
  - field descriptions, Sub Process view 105

## P

- Page Manager
  - parameters for 317
  - setting up 316
  - troubleshooting 319
- parallel processing, Workflow processes,
  - supported by 78, 134
- performance
  - tuning tips for Workflow Policies 356
  - tuning workflow policies 353
  - Workflow Agent processes, creating
    - multiple 355
- Persistence Level, about setting 192
- planning
  - policy scenarios 207
  - workflow policies, about 203
  - workflow policies, determining what to
    - monitor 204
  - workflow policies, planning policies and
    - conditions, about 204
  - workflow policy groups, about 202
- policies

- creating 249
- creating actions for 227
- creating groups for 239
- monitor, determining what 204
- planning for 203
- planning policies and conditions,
  - about 204
- Policies applet
  - described, Workflow Policies Groups
    - view 237
  - field descriptions, Groups view 238
  - field descriptions, Policies view 241
  - Workflow Policies Groups view, field
    - descriptions 238
- policy action
  - See also* programs
  - creating 227
  - definition of 408
  - examples of 228
  - working with 215
- policy condition
  - definition of 408
  - role in workflow policies 39
- Policy Frequency Analysis chart,
  - viewing 335
- policy groups
  - creating 237
  - planning, about and reasons for
    - using 202
- predefined programs
  - assigning manager as owner 300
  - close date, changing 297
  - Message Broadcast program, screen
    - example and arguments 221
  - owner, changing 298
  - Run External Program 223, 234
  - Send Email, using and arguments 220
  - Send Quote Page 302
  - table of 402
  - workflow action, using a predefined
    - program to create 41
  - workflow policy programs 297

primary component, defining for business object 49

Process Designer view

- about, screen example, and list of tasks 57
- Designer applet items 58
- diagramming a process 65

Process Properties

- definition of 406
- workflow process, passing field values to example 181
- workflow processes, about passing to 69

Process Properties view

- defining process properties 69
- Process Properties applet fields, table of 59
- using 59

Process Property values, using in expressions 82, 138

Process Simulator view

- button functions 173
- debugging workflow processes 175
- definition of 407
- Step Details applet fields 173
- testing Workflow Process, caution 175
- using 173
- workflow processes, using to invoke 172

process steps

- copying 157
- Decision 83
- deleting 157
- End step, about 145
- Siebel Operation, about working with 109
- Start step, about working with 72
- Stop 139
- Stop step, about working with 139
- Sub Process step, about working with 102
- testing with Process Simulator, about 172
- Wait steps 118

processes

- existing process definitions, modifying 64
- existing process definitions, reviewing 62
- new process, defining 63
- production environment, migrating to 344

program

- definition of 408
- property fields descriptions, workflow policies 288
- workflow policy types 217

Program Argument common properties, table of 289

Program Argument List view, about 289

Program List view, using to list workflow policy programs 288

programs

- predefined, table of 402
- workflow policy action types 41

properties

- described 264
- Program Argument common, table of 289
- Run External Program Argument 293
- Send Message Argument 292
- workflow policy programs 288

property sets, passing to workflow processes 69

## R

recipient types, table of 225

Recipients applet, field descriptions 225

ReloadPolicy parameter 329

repository setting, verifying 52

Requests parameter 329

row id, locating for a object id property 175

Run External

- Program Argument properties, table of 293

- workflow policy program, creating
  - action 234
- workflow policy program. about and
  - example and arguments 223
- Run Workflow Process, about using to
  - define a policy 178

**S**

- S\_ESCL\_ACTION table 321, 322
- S\_ESCL\_ACTN\_REQ table 322, 352
- S\_ESCL\_LOG table 321, 322
- S\_ESCL\_REQ table 321, 322, 350
- S\_ESCL\_STATE table 321, 322, 351
- sample database, launching 36
- scripts, invoking workflow processes,
  - examples 180
- Search Specification applet, field
  - descriptions 111
- Send Campaign Email, using 256
- Send Email
  - creating for Workflow policy 252
  - using and arguments 220
- Send Message Argument, properties, table
  - of 292
- Send Message Arguments applet 220
- Send Message Broadcast, Workflow Policy
  - program 41
- Send Message, Workflow Policy
  - program 41
- Send Page
  - about and arguments 217
  - Workflow Policy program about 41
- Send Quote Page, using and SQL statement
  - examples 302
- Send to Relative recipient type, about
  - sending email or page 225
- server requests, submit request
  - arguments 376
- Service 26
- Service Request Priority, updating 233
- service requests
  - assigning manager as owner 300
  - close date, changing to today's date 297

- owner, changing to today's date 298
- Siebel administrators
  - correcting processes and resuming 192
  - deleting workflow process
    - instances 194
  - stopping workflow processes 193
- Siebel Client, description 32
- Siebel Client, making object types
  - available 303
- Siebel database, described 32
- Siebel eScript, examples of invoking from
  - workflow process 180
- Siebel Operation
  - applet, field descriptions 110
  - definition 407
- Siebel Operation Object Id, described 68
- Siebel Operation step
  - about working with 109
  - defining 112
  - fields, defining 115
  - search specifications, defining 116
  - workspace, adding to 114
- Siebel Operation view
  - Fields applet fields 110
  - Search Specification applet fields 111
  - Siebel Operation applet fields 110
  - using 109
- Siebel Server
  - Email Manager, setting up 314
  - setting up for Page Manager 316
  - startup account, locating 312
- Siebel Server Administration, viewing trace
  - files 333
- Siebel Server log directory 333
- Siebel Server task trace file
  - created for listed processes 333
  - viewed in Siebel Server
    - Administration 333
  - viewed in Siebel Server log
    - directory 333
- Siebel Server, description 32
- Siebel SmartScript
  - See SmartScript module

- Siebel Tools
    - customizing Workflow Policies 264
    - five types of programs in 287
    - Program List view, using 288
    - Validate Tool provided by 282
    - Workflow Administration and 267
  - Siebel Tools, description 32
  - Siebel VB, examples of invoking from
    - workflow process 180
  - Siebel Workflow Administration, about and
    - screen example 34
  - Simulator applet, button functions 173
  - sleep intervals, determining for workflow
    - policy groups 354
  - SmartScript module, described 29
  - specialized comparisons, descriptions 247
  - SQL script file 310
  - SQL statements
    - created for program arguments 296
    - types of 302
  - Start applet, field descriptions 73
  - Start step
    - about working with 72
    - defining a start step 78, 134
    - definition 407
  - Start view
    - Conditions applet fields 75, 131
    - Next Step applet fields 74, 121, 130
    - tasks 72
    - Values applet 77, 133
  - state machine 43, 409
  - state model
    - See also* Workflow Policies
    - cache 371
    - creating 362
    - definition 409
    - new state model, creating 364
    - overview 43
    - overview and screen diagram 360
  - state transitions
    - creating 365
    - definition 43, 409
    - new state transition, creating 367
  - Step Details applet, field descriptions 173
  - step instance, defined 407
  - steps
    - copying 157
    - definition 407
    - deleting 157
  - Stop applet, field descriptions 141
  - Stop step
    - about working with 139
    - defining a stop step 142
    - definition 407
    - input arguments, defining 143
  - Stop view
    - Input Arguments applet fields 142
    - Stop applet fields 141
    - tasks 140
  - Sub Process applet, field descriptions 103
  - Sub Process steps
    - about working with 102
    - defining 106
  - Sub Process view
    - Input Arguments applet 104
    - Output Arguments applet fields 105
    - Sub Process applet fields 103
    - tasks 102
    - using 102
  - sub-process, defined 407
  - synchronous server requests 375
  - System Administrator, workflow role 33
- ## T
- testing
    - creating strategy for 211
    - workflow process, creating strategy for 51
    - workflow processes, Process Simulator 172
  - testing workflow policies 338
  - trace file
    - Siebel Server log directory, viewing in 333
    - Siebel Server task trace file, created for listed processes 333

- viewing trace files 333
- tracing and logging events 195
- trigger
  - database triggers, creating 306
  - Generate Trigger (Gen Trig), component-specific parameters for 308
  - Generate Trigger (Gen Trig), functions 306
  - Generate Trigger (Gen Trig), tips of running 307
  - invalid trigger, avoiding in production environment 345
- troubleshooting
  - Email and Page Manager 319
  - notes on 340

## U

- User Interact step, viewing and defining step 126
- user, workflow role described 33

## V

- Validate Tool, using 282
- values
  - Decision steps, defining 90
  - described 265
  - Message Broadcast program type 221
  - Recipients applet fields 225
  - Run External program type 223
  - Send E-mail program type 220
  - Send Message Broadcast program type, table of 221
  - valid database operations program argument 290
- Values applet, about
  - Start view 77, 133

## W

- Wait applet, field descriptions 119
- Wait steps
  - about working with and screen example 118

- defining a wait step 123
- definition 407
- input arguments, defining 124

### Wait view

- Input Arguments applet fields 123
- Wait applet fields 119

- wildcards, using in standard comparisons 244

- work item, defined 407

### workflow

- definition 26
- gathering information on 46
- user roles 33

### Workflow Action Agent

- action interval, about 355
- caution in assigning 356
- command-line interface 354
- functions of 330
- starting 321
- to run processes of 331
- to shut down 331

### Workflow Action Arguments

- applets, described 215

### Workflow Actions

- creating Assign to Campaign E-mail 260
- creating Create Email Activity 260
- creating Database Operation Program type 233
- creating Message Broadcast Program type 221
- creating Run External Program type 223
- creating Send Campaign E-mail 259
- creating Send Message Program type 220
- creating Send Page Program type 217
- using Database Operation program type 224

### Workflow Administration

- Siebel Tools and 267

### Workflow Administration views

- Siebel Tools programs available in 267

- workflow administrator, role of 33

### Workflow Agent process

- monitoring 348
- multiple processes, creating 355
- starting automatically 332
- Workflow Groups, starting multiple for 332
- Workflow columns
  - adding 278
  - adding to Workflow objects 281
  - applet fields in 272
  - associating Workflow component with 282
  - described 408
- Workflow Columns applet fields, values of 272
- Workflow component columns
  - described 409
  - functions of 268
  - properties fields 277
- Workflow Component Columns List view
  - described 277
  - function of 271, 274
- Workflow components
  - architecture 32
  - associating with Workflow column 282
  - defining 279
  - described 268, 409
  - relationship between 269
  - relationship between component and primary 284
  - relationship between Workflow object and 270
- Workflow Components List view
  - displayed 275
  - function of 271
- Workflow condition, defined 408
- Workflow Conditions applet
  - using specialized comparisons in 245
- workflow groups
  - creating 202
  - defining a workflow policy group 261
  - description 42
  - Workflow Agent processes, starting multiple 332
  - workflow policy group, described 409
- Workflow Groups applet
  - about using 237
  - field descriptions 237
- Workflow Groups view, function of 214
- Workflow Manager
  - See Workflow Policies
- Workflow Monitor Agent
  - caution in assigning 356
  - command-line interface 354
  - executed policies, viewing 349
  - parameters 326
  - starting 323
  - testing workflow policies 338
- Workflow object properties fields 274
- Workflow objects
  - adding columns to 281
  - adding new 280
  - defined using Object Explorer views 280
  - described 409
  - functions of 268
  - relationship between Workflow components and 270
- Workflow Objects List view 271
- Workflow Policies
  - definition 39
  - overview 39
  - policy condition, about 39
  - program types, list of 41
  - structure, about rule structure (diagram) 39
  - views, administered by and example 42
  - workflow groups, about 42
  - workflow policy action, parts of (diagram) 40
- workflow policies
  - See also performance
  - Assign Non-Respondents policy, creating 262
  - conditions, table of specialized comparisons 247
  - created for Send Email 252
  - creating 240, 249



---

- described 408
- Email for CD-ROM Campaign policy,
  - creating 261
- invoking a process 178
- license key validation 52
- monitoring processes 348
- monitoring, about 348
- moving from one group to another,
  - note 242
- performance tuning tips 356
- planning policies and conditions,
  - about 204
- planning, about 203
- planning, determining what to
  - monitor 204
- production environment, migrating
  - to 344
- program types 217
- repository setting, viewing 52
- required components 52
- Siebel Operation steps, using different
  - object layers, described 113
- sleep interval, determining for
  - groups 354
- testing 211, 338
- using specialized comparisons 245
- using standard comparisons 243
- views, list of 214
- Workflow Policies Actions view
  - applet, field descriptions 216
  - applets, description of 215
  - Message Broadcast Arguments
    - applet 221
  - Recipients applet 225
- Workflow Policies Arguments applet 217
- Workflow Policies Explorer view 214
- Workflow Policies Groups view
  - Policies applet fields, table of 238
  - Policies applet, described 237
  - Workflow Groups applet 237
- Workflow Policies Log view 214
- Workflow Policies module, described 28
- Workflow Policies Policies view
  - Actions applet field, table of 248
  - working with 240
- Workflow Policies Report page, about 335
- Workflow Policies view, function of 214
- Workflow Policy Action Agent 202
- workflow policy actions
  - creating 227
  - examples 228
  - working with 215
- Workflow Policy applet fields 241
- workflow policy column, defined 408
- workflow policy component column,
  - defined 409
- workflow policy components, defined 409
- workflow policy groups
  - creating 239
  - definition 409
  - planning 202
- Workflow Policy Log 338
- Workflow Policy Monitor Agent 202
- workflow policy objects
  - defined 409
  - modifying 283
  - Object Explorer, adding to 264
- workflow policy program
  - property field descriptions 288
  - viewing with Program List view 288
- Workflow Process applet, field
  - descriptions 55
- workflow process instance
  - about monitoring settings 192
  - defined 407
- Workflow Process Manager Batch, about
  - running workflow processes in batch
    - mode 185
- workflow processes
  - about designing views 54
  - activating 158
  - business object, defining for a primary
    - component 49
  - business services, enabling 50
  - copying 157
  - deactivating 158

---

- definition 407
- deleting 157
- exporting\importing 188
- gathering information for 46
- invoking from a runtime event,
  - about 182
- invoking from a script, examples 180
- invoking from a workflow policy 178
- overview 35
- planning tips 49
- process properties and property sets 69
- requirements 48
- running in batch mode 185
- running on Object Manager 185
- testing caution 175
- testing, defining and migration
  - strategy 51
- tracing events 341
- Workflow Processes module, described 28
- workflow processes, designing
  - All Processes view, screen example and
    - list of tasks 55
  - definitions, working with 62
  - Designer applet palette items 58
  - process steps, about diagramming 65
  - process steps, diagramming
    - procedure 66
  - step icons, table of 65
  - Workflow Processes applet fields, table
    - of 55
- workflow processes, generating 57
- workflow processes, monitoring
  - about 192
  - correcting processes and resuming 192
  - deleting workflow process
    - instances 194
  - stopping 193
  - tracing and logging events, table of 195
- Workflow programs
  - Assign to Campaign 257
  - configuring predefined 297
  - Create Email Activity 257
  - creating or modifying 293
  - described 287, 408
  - five types of 287
  - Send Campaign Email, using 256
- Workflow Utilities, arguments 378