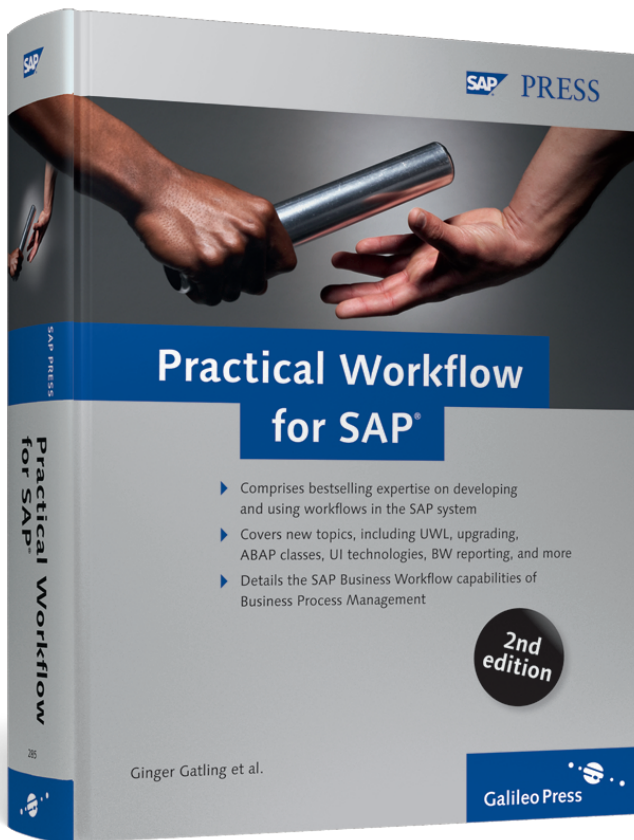


Jocelyn Dart, Erik Dick, Ginger Gatling, Oliver Hilss,
Somya Kapoor, Silvana Kempf, Susan Keohan, Thomas Kosog,
Paul Médaille, Mike Pokraka, Alan Rickayzen,
Shalini Sabnani, Jörn Sedlmayr, Ted Sohn

Practical Workflow for SAP®




Galileo Press

Bonn • Boston

Contents at a Glance

PART I Using SAP Business Workflow

1	Introduction	39
2	Requirements Gathering Strategy	57
3	Configuring the System	83
4	Work Item Delivery	95
5	Agents	133
6	Workflow Administration	161
7	Using SAP NetWeaver Business Warehouse for SAP Business Workflow Reporting	211

PART II Developing Your Own Workflow

8	Creating a Workflow	221
9	Advanced Workflow Design Techniques	263
10	Business Objects	289
11	ABAP Classes	353
12	Agent Determination Rules	413
13	Using Events and Other Business Interfaces	441
14	Custom Programs	475
15	Service-Enabling Workflows	501
16	Advanced Diagnostics	535
17	Upgrading SAP Business Workflow	555

PART III User Interface Technologies and SAP Business Workflow

18	User Interface Options	567
19	Using Web Dynpro ABAP	575
20	Using Web Dynpro Java	593
21	Using Business Server Pages	613
22	Using Forms	637
23	Alloy — Lotus Notes Integration	645
24	Duet — Microsoft Office Integration	661

PART IV Using SAP Business Workflow in SAP Business Suite Applications

25	SAP Supplier Relationship Management	679
26	SAP Customer Relationship Management	735
27	SAP ERP Human Capital Management —Processes and Forms	793
28	Setting Up an SAP-Provided SAP ERP Workflow	825
29	ArchiveLink	835

Appendices

Contents

Foreword	27
Preface	29

PART I: Using SAP Business Workflow

1 Introduction	39
1.1 What Is SAP Business Workflow	39
1.2 What Is SAP NetWeaver Business Process Management	42
1.3 How Does SAP Business Workflow Fit Into the SAP Overall Business Process Management Strategy	43
1.4 In Which Business Situations Should I Use SAP Business Workflow	44
1.4.1 Factor 1: Human Involvement	45
1.4.2 Factor 2: Repetition	45
1.4.3 Factor 3: The Fruits of Success	46
1.4.4 Factor 4: The Cost of Failure	48
1.5 What Can Be Achieved Using SAP Business Workflow	49
1.6 Features of SAP Business Workflow	49
1.6.1 Robust Integration into SAP Applications	50
1.6.2 Workflow Builder	50
1.6.3 Graphical Status Visualization	51
1.6.4 Routing Mechanisms	52
1.6.5 Deadline/Escalation Management	53
1.6.6 Synchronization Management	53
1.6.7 Integration with Groupware Products	53
1.6.8 Ad-hoc Process Enhancers	53
1.6.9 Desktop Integration	54
1.6.10 Analytics with SAP NetWeaver Business Warehouse	54
1.6.11 Robust Extensible Architecture	55
1.7 Some Terminology	55
2 Requirements Gathering Strategy	57
2.1 Introduction	57
2.2 Understanding the Business Process	59

2.2.1	Business Drivers for the Process	59
2.2.2	Expected Benefits of Workflow	61
2.2.3	How Does the Process Work in Theory	63
2.2.4	How Does the Process Work in Practice	63
2.2.5	Unions and Workers Councils	66
2.3	Gathering Object Data	66
2.3.1	What Data Is Needed by Workflow	67
2.3.2	Where to Look for Data	68
2.3.3	What Data Already Exists	68
2.3.4	Making the Most of the Data Collected	69
2.4	Determining the Agents	72
2.4.1	Who Are Your Agents?	73
2.4.2	Criteria for Finding Agents	74
2.5	Determining Work Item Delivery Routes	77
2.6	Confirming the Workflow Design	78
2.7	Workflow Administration	80
2.8	Planning for Production Start	80
2.9	Planning for Future Enhancements	81

3 Configuring the System 83

3.1	Quick Start: Configure the SAP Workflow Engine in Seconds	85
3.1.1	Automatic Workflow Customizing	85
3.1.2	Specifics of Automatic Customizing	87
3.1.3	IMG Activities for Authorization Management	90
3.2	Task-Specific Customizing	91
3.2.1	Maintaining the Organizational Plan	91
3.2.2	Agent Assignment for Tasks	92
3.2.3	Activating the Triggering Events for a Workflow or a Task ...	92
3.3	Transport and Client Copy	92
3.3.1	Transport of Tasks	93
3.3.2	Transport of Settings for Tasks and Workflows	93
3.3.3	Transporting a Workflow Definition	93
3.4	Customizing for the SAP Workflow Engine	94

4 Work Item Delivery 95

4.1	The Human Factor	95
4.2	SAP Business Workplace	98

4.2.1	Work Item Text	99
4.2.2	Attachments	101
4.2.3	Reserving and Replacing a Work Item	102
4.2.4	Executing a Work Item	103
4.2.5	Synchronous Dialog Chains	105
4.2.6	Forwarding a Work Item	106
4.2.7	Priority	107
4.2.8	Resubmission	107
4.2.9	Logs	108
4.2.10	Outbox	110
4.2.11	Rejecting Work Items	110
4.2.12	Substitution	111
4.3	Universal Worklist	111
4.3.1	Major Features of the Universal Worklist	112
4.3.2	Getting Started with the Universal Worklist	114
4.3.3	Working with Tasks in the Universal Worklist	115
4.3.4	Personalization Options in the Universal Worklist	115
4.3.5	Customizing Options in the Universal Worklist	116
4.3.6	Action Handlers in the Universal Worklist	117
4.3.7	UWL Configuration Wizard	118
4.3.8	Universal Worklist API and the SONiC Connector	120
4.4	Examples of Universal Worklist Customizing	121
4.4.1	How Can I Add/Remove Columns	122
4.4.2	How Can I Create Custom Columns	122
4.4.3	How Can I Make a Mandatory Rejection Memo	125
4.5	Extended Notifications	125
4.5.1	Types of Notifications	125
4.5.2	Features of Extended Notifications	126
4.5.3	Configuration Required for Extended Notifications	127
4.6	Business Workflow Workplace	128
4.7	External Users	130
4.8	Other Considerations	131

5 Agents 133

5.1	Understanding Agent Assignment	134
5.1.1	Possible Agents	135
5.1.2	Responsible Agents	138
5.1.3	Excluded Agents	139

5.1.4	Recipients	140
5.1.5	Actual Agents	141
5.1.6	Assigning Multiple Agents per Work Item	142
5.1.7	Deadline and Notification Agents	143
5.2	Agent Assignment Using the Organizational Structure	144
5.2.1	The Basic Organizational Plan	144
5.2.2	The Extended Organizational Plan	148
5.2.3	Doing Without an Organizational Plan	150
5.3	Agent Assignment Using Task Groups	150
5.4	Implementing and Maintaining the Structure in Practice	151
5.4.1	Strategies for Organization Plan Maintenance	151
5.4.2	Transporting Organization Plans	153
5.4.3	Reassigning Work Items from One User to Another	154
5.4.4	Substituting Users	155
5.4.5	When Do You Choose to Refresh the Organizational Environment	157
5.4.6	Training and Encouraging Agents	158

6 Workflow Administration 161

6.1	Reporting on Workflows	162
6.1.1	Reporting on Workflow Progress	163
6.1.2	Reporting on Workflow Performance	164
6.1.3	Work Items by Processing Duration	165
6.1.4	Work Items with Monitored Deadlines	166
6.1.5	Reporting on Agent Behavior	166
6.1.6	Identifying and Justifying Potential Improvements	167
6.2	Error Resolution	168
6.3	General Techniques for Resolving Runtime Errors	171
6.3.1	Basic Settings for Error Monitoring	171
6.3.2	Finding and Fixing Work Items	172
6.3.3	Diagnosis of Workflows with Errors	172
6.3.4	Work Item Selection	173
6.3.5	Work Item Frequency	175
6.4	Working with the Work Item Display	176
6.4.1	Work Item Display — Standard View	177
6.4.2	Work Item Display — Technical View	179
6.4.3	How to Work with the Work Item Container Display	181

6.5	Working with the Workflow Log	182
6.5.1	User View	182
6.5.2	Technical View	185
6.6	Resolving Agent Determination Errors	188
6.6.1	Fixing Work Items with No Agents or Wrong Agents	188
6.6.2	Preventing Agent Determination Problems from Recurring	191
6.6.3	Support Tools for Agent Determination Problems	191
6.6.4	Resolving Buffering Errors	192
6.6.5	Fixing Work Items with Buffering Problems	193
6.6.6	Preventing Buffering Problems from Recurring	194
6.6.7	Support Tools for Buffering Problems	194
6.7	Other Support Tools	195
6.8	Help Desk in the Intranet	196
6.8.1	Web-Based Help Desks	196
6.8.2	Self Help	197
6.9	A Day in the Life of a Workflow Administrator	198
6.9.1	Who Should be the Workflow Administrator	199
6.9.2	Who Makes the Business Decisions in the Event of Failure	201
6.9.3	Crisis Response	203
6.9.4	Day-to-Day Monitoring	205
6.9.5	Periodic Reality Checks	206
6.9.6	Housekeeping and Archiving	208
6.9.7	Making the Most of a Precious Resource	209

7 Using SAP NetWeaver Business Warehouse for SAP Business Workflow Reporting 211

7.1	SAP NetWeaver Business Warehouse Basics for the Workflow Expert	212
7.2	Standard Workflow Analysis with SAP NetWeaver Business Warehouse	214
7.2.1	Data Flow and Overview of Provided Content	215
7.2.2	Aggregation Layer	215
7.2.3	InfoCube Aggregated Process Data	216
7.2.4	Using Standard BI Content	217

PART II: Developing Your Own Workflow

8	Creating a Workflow	221
8.1	Workflow Builder Basics	222
8.1.1	Look and Feel of the Workflow Builder	222
8.1.2	Building Your First Workflow	226
8.1.3	Saving, Activating, and Testing	229
8.2	Enhancing Your Workflow	231
8.2.1	Deadline Monitoring	231
8.2.2	How to Create and Use Tasks	234
8.2.3	Using Asynchronous Tasks	241
8.2.4	How to Access Data and Activities	243
8.2.5	Notifications	245
8.3	Basics of Containers and Bindings	246
8.3.1	How to Create Containers and Bindings for Tasks	247
8.3.2	Creating Container Elements in the Workflow Container	247
8.3.3	Changing Container Elements	249
8.4	Steps	251
8.4.1	What Other Step Types Exist	252
8.4.2	How to Insert New Steps	254
8.4.3	What Kinds of Outcomes Exist	256
8.4.4	Which Task and Step Attributes Impact Work Item Execution	258
8.5	Documenting, Translating, Transporting, and Team Development ...	259
8.5.1	Documenting Workflow Definitions	259
8.5.2	Translating a Workflow into Other Languages	260
8.5.3	Transporting New Versions of a Workflow	260
8.5.4	How Do You Share Workflow Development Within a Team	261
9	Advanced Workflow Design Techniques	263
9.1	Step Conditions	263
9.1.1	Step Condition: Create Work Item	265
9.1.2	Step Condition: Complete Work Item	265
9.1.3	Step Condition: Completion Execution	266
9.2	How to Implement Parallel Processing	267
9.2.1	How to Implement Table-Driven Dynamic Parallel Processing	268

9.2.2	How to Implement a Fork	272
9.3	Reusing Workflows as Subworkflows	273
9.4	Modeled Deadlines	274
9.4.1	Notify Recipients Via Email That a Deadline Has Been Missed	275
9.4.2	Skip the Work Item If Deadline Has Been Missed	275
9.5	Containers	276
9.5.1	Containers and Bindings in Depth	277
9.5.2	Reimplementation of the Container	280
9.6	Binding	282
9.6.1	Binding Editor	282
9.6.2	Expressions	283
9.7	Blocks and Local Workflows	284
9.7.1	Local Workflows as a Practical Example of Block Implementation	284
9.7.2	Local Workflows and Other SAP-Provided Example Workflows	285
9.7.3	When to Use Local Workflows Versus Subworkflows	285
9.8	Ad-hoc Features and Review Workflows	286
9.8.1	Ad-Hoc Features to Change Workflows on the Fly	286
9.8.2	Review Workflows	287

10 Business Objects 289

10.1	Business Object Basics	291
10.1.1	Business Objects: Some Basic Terminology	293
10.1.2	Relationships Between Business Objects	294
10.1.3	Business Object Tools	297
10.1.4	Viewing Existing Business Object Types	299
10.2	Creating Your Own Business Object Types	307
10.2.1	Creating New Business Objects	310
10.2.2	Extending Existing Object Types Via Delegation	312
10.3	Creating Business Object Type Components	313
10.3.1	Adding an Interface	314
10.3.2	Creating Key Fields	314
10.3.3	Creating Attributes	315
10.3.4	Creating Methods	319
10.3.5	Creating Events	328

- 10.4 Business Object Type Programming 329
 - 10.4.1 Key Fields and the Object Declaration 330
 - 10.4.2 Programming Attributes 332
 - 10.4.3 Programming Methods 339
- 10.5 Some Useful Predefined Object Types 345
 - 10.5.1 Object Type SELFITEM 345
 - 10.5.2 Object Type WF_TASK 346
 - 10.5.3 Object Type SYSTEM 346
 - 10.5.4 Object Type FORMABSENC 346
 - 10.5.5 Object Type USR01 346
- 10.6 Real-World Examples for Business Object Customizing 347
 - 10.6.1 Custom Attribute 347
 - 10.6.2 Custom Method 349

11 ABAP Classes 353

- 11.1 ABAP Classes Basics 354
 - 11.1.1 History 354
 - 11.1.2 Why Use Classes 355
 - 11.1.3 More Resources 357
- 11.2 Comparing ABAP Classes and BOR Objects 358
 - 11.2.1 Interfaces 358
 - 11.2.2 Object Identification 359
 - 11.2.3 Class/Component Level 360
 - 11.2.4 Attributes 360
 - 11.2.5 Methods 361
 - 11.2.6 Events 361
- 11.3 Utility Classes 361
 - 11.3.1 Creating a Utility Class 362
 - 11.3.2 Component Visibility 364
 - 11.3.3 Methods 365
 - 11.3.4 Using a Utility Class in a Task 366
 - 11.3.5 Attributes 368
 - 11.3.6 Functional Methods 370
- 11.4 Business Classes 374
 - 11.4.1 Local Persistent Object Reference 376
 - 11.4.2 Implementing FIND_BY_LPOR 379
 - 11.4.3 Implementing LPOR 380
 - 11.4.4 Instantiating ABAP Class Objects 381

11.4.5	Alternate Ways to Instantiate Objects	383
11.4.6	Attributes	385
11.4.7	Attributes Versus Functional Methods	385
11.5	Exception Classes	387
11.6	Events	392
11.6.1	Responding to Events	392
11.6.2	Raising Events	393
11.7	Using BOR Objects in Classes	401
11.8	Recommendations and Advanced Topics	406
11.8.1	Performance and Instance Management	406
11.8.2	Persistent Classes	409
11.8.3	Class Design Considerations	410
12 Agent Determination Rules		413
12.1	Determining Agents Through Rule Resolution	414
12.2	Rule Basics	415
12.2.1	Testing and Simulating Rules	417
12.2.2	Rule to Locate the Plant Manager	417
12.3	Agent Determination Rule Resolution in the Workflow Step	420
12.4	What Happens If Rule Resolution Has No Result	421
12.5	Responsibility Rules	423
12.5.1	Responsibility Rules with Multiple Criteria	423
12.5.2	Secondary Priorities	425
12.6	Agent Determination Using SAP Organizational Data	427
12.7	Function Modules as Rules	429
12.7.1	Creating a Rule Based on a Function Module	429
12.7.2	Linking Rule Function Modules from Different SAP Components	431
12.8	Evaluation Paths as Rules	434
12.9	Other Options for Responsible Agent Assignment	437
12.10	Real-World Example of Agent Determination Rules	438
13 Using Events and Other Business Interfaces		441
13.1	Understanding Events	442
13.2	How to Know What Events Exist	444
13.2.1	Look in the Object	444
13.2.2	Look in the Event Linkage Table	444

13.2.3	Remember to Turn the Trace Off When You Are Done	445
13.3	Defining Events	447
13.4	Raising Events from Business Applications	447
13.4.1	Raising Events Via Change Documents	449
13.4.2	Raising Events Via Status Changes	451
13.4.3	Raising Events Via Message Control	453
13.4.4	Raising Events Via Changes to HR Master Data	454
13.4.5	Raising Events Via Business Transaction Events	455
13.4.6	Raising Events by Calling a Workflow API	457
13.5	Using Events in Workflows	457
13.5.1	How to Use Events in Workflows	457
13.5.2	Triggering Events	458
13.5.3	Start Conditions for Workflows	462
13.5.4	Terminating Events	464
13.5.5	Workflow Header Events	467
13.5.6	Parked Events	468
13.5.7	Event Queues	469
13.6	Generic Object Services	470
13.6.1	Starting Workflows Manually	471
13.6.2	Viewing the Logs of Related Workflows	473
13.6.3	Subscribe to an Object Instance	473
13.6.4	Sending an Email	473
13.7	Starting Workflows from Messages	474

14 Custom Programs 475

14.1	The Engine	475
14.1.1	Work Items	476
14.1.2	Events	481
14.1.3	Work Item States	481
14.2	Workflow APIs	484
14.2.1	The Simple WAPI Container	484
14.2.2	SAP_WAPI_CREATE_EVENT	485
14.2.3	SAP_WAPI_START_WORKFLOW	485
14.2.4	SAP_WAPI_WORKITEMS_TO_OBJECT	487
14.2.5	SAP_WAPI_READ_CONTAINER	487
14.3	Advanced Business Interface Techniques	488
14.3.1	When Should a Workflow with an Event Be Started	488

14.3.2	Raising Events by Calling a Function Module	489
14.3.3	Advanced Event Linkage	494
14.3.4	Adding Generic Object Services to Your Own Transactions	497
14.3.5	Implementing the Workflow Toolbox in Your Own Transactions	498
14.4	Office Document Interfaces	499
14.4.1	Business Communications Services	500
14.4.2	SO_*_API1 function modules	500
15 Service-Enabling Workflows		501
15.1	Workflow in the Context of a Business Process	501
15.2	Web Services and Service-Oriented Architecture	502
15.2.1	What Are Web Services	503
15.2.2	Web Services Description Language	503
15.2.3	Web Services and SOAP	505
15.2.4	Other Web Service Standards	506
15.3	What Is Service-Oriented Architecture	507
15.4	SOA at SAP	510
15.5	Implementing the Service Interface for a Workflow	512
15.5.1	The Function Module	512
15.5.2	Service-Enabling in the Application (Inside-Out Approach)	513
15.5.3	Service Enabling from the Enterprise Services Repository (Outside-In Approach)	520
16 Advanced Diagnostics		535
16.1	The Tools of the Trade	535
16.2	The Diagnosis Logs	537
16.2.1	The Workflow Log	537
16.2.2	The Diagnosis Transaction	538
16.2.3	The Event Trace	540
16.2.4	The RFC Log	542
16.3	Debugging with the ABAP Debugger	543
16.3.1	The Workflow Trace	545
16.3.2	When to Use the Workflow Trace	549
16.4	Preparing a Task to Go to Production	549

17 Upgrading SAP Business Workflow 555

- 17.1 First Steps to Upgrading Workflow 555
- 17.2 Steps to Take Before the Upgrade Begins 557
 - 17.2.1 Complete All Running Workflow Instances 557
 - 17.2.2 Create Workflow Test Plans 557
 - 17.2.3 WF and T Tasks Versus WS and TS Tasks 557
 - 17.2.4 Clean Runtime Tables 558
- 17.3 Steps to Take During and After the Upgrade 559
 - 17.3.1 Conversion of Event Linkage Tables 559
 - 17.3.2 Basis Support Package 559
 - 17.3.3 Configure Your System for Workflow 559
 - 17.3.4 Workflow Definition and Binding 559
 - 17.3.5 Tables SWW_CONTOB and SWW_CONT 560
 - 17.3.6 Changed Locking Behavior of Runtime 561
 - 17.3.7 Other Issues After the Upgrade 563
- 17.4 Important SAP Notes 563

PART III: User Interface Technologies and SAP Business Workflow

18 User Interface Options 567

- 18.1 Inbox and the User Interface 568
- 18.2 Containers, Bindings, and Data Flow 570
 - 18.2.1 Where the Customized UI Fits in with the Data Flow
When Using Universal Worklist 570
 - 18.2.2 Common Steps When Using Web Dynpro and Business
Server Pages with Universal Worklist 572

19 Using Web Dynpro ABAP 575

- 19.1 Example Based on User Decision 576
 - 19.1.1 Standard User Decision 576
 - 19.1.2 User Decision with Note in Universal Worklist 577
 - 19.1.3 User Decision with Web Dynpro ABAP 580
- 19.2 Required Settings in Web Dynpro ABAP 583
 - 19.2.1 Web Dynpro ABAP Application 583
 - 19.2.2 Window 586
 - 19.2.3 Main View 587

19.3	Configuration for the Universal Worklist	590
19.4	Ensuring the User Only Executes the Work Item in the Universal Worklist	592
20	Using Web Dynpro Java	593
20.1	Real-World Example of Web Dynpro Java and Workflow	593
20.2	What Must Be Done on the Workflow Side	595
20.3	What Must Be Done on the Java Side	603
20.3.1	Presentation Layer	603
20.3.2	Model Layer	605
20.4	What Must Be Done in the Universal Worklist	608
21	Using Business Server Pages	613
21.1	Building the Example BSP and Workflow	613
21.1.1	Laying the Groundwork	613
21.1.2	Setting Up the Web Service Handler	621
21.2	Launching the BSP and Transferring Data	622
21.2.1	Defining the Workflow Service and Creating a Task	622
21.2.2	Creating a Test Workflow	624
21.2.3	Launching a BSP Application from the Business Workplace	625
21.2.4	Launching a BSP Application from Another Web Application	626
21.2.5	Launching a BSP Application from the Universal Worklist	626
21.3	Completing Workflow Tasks from BSP Applications	629
22	Using Forms	637
22.1	SAP Interactive Forms by Adobe	637
22.2	SAP Business Workflow Forms	638
22.2.1	Simple Forms Creation	638
22.2.2	Document Templates	641
23	Alloy — Lotus Notes Integration	645
23.1	The Design Paradigm	646
23.1.1	Sidebar	646

23.1.2	Mail Inbox	647
23.1.3	Decision Views	648
23.1.4	Calendar Items	650
23.2	Prerequisites for Alloy	652
23.3	Generic Workflow Capabilities	653
23.4	Adding New Decisions to Alloy	653
23.4.1	Adding New Decision Steps	655
23.4.2	Design a New User Interface for the Decision	656
23.4.3	Add Additional Data from the SAP Context	658
23.4.4	Adding a Dialog Task	659
23.4.5	Sidebar Enhancements	659
23.5	Standard Alloy Decisions	660

24 Duet – Microsoft Office Integration 661

24.1	Design Time Layout	663
24.2	How an Approval Workflow Application Works in Duet	666
24.3	Prerequisites to Use the Duet Workflow Approval Template	669
24.4	Creating Your Approval Workflow Application	670
24.4.1	Adding New Decisions to Duet	672
24.4.2	Design a New User Interface for the Decision	674
24.4.3	Add Additional Data from the SAP Context	675
24.4.4	Adding a Dialog Task	675
24.4.5	Sidebar Enhancements	676

PART IV: Using SAP Business Workflow in SAP Business Suite Applications

25 SAP Supplier Relationship Management 679

25.1	Approval Processes	681
25.1.1	Simple Approval Processes	682
25.1.2	Complex Approval Processes	682
25.1.3	Key Concepts and Requirements That Make Up the Foundation of SAP SRM Approval Processing	683
25.2	SAP SRM Approval Frameworks	688
25.3	Process-Controlled Workflow in Detail	690
25.3.1	Technical Background	693
25.3.2	Basic Assumptions for Document Processing	694

25.3.3	Configuration of Approval Levels	694
25.3.4	Business Rules Framework	697
25.3.5	Agent Determination	698
25.3.6	Approval Process Overview: User-Focused Process Visualization	701
25.4	Real-World Example of a Process-Controlled Workflow	704
25.4.1	Simplest Scenario, Execute Without Approval	705
25.4.2	Configuring Two-Step Approval	706
25.4.3	Details on BRF Integration	707
25.5	Application-Controlled Workflow in Detail	710
25.5.1	One-/Two-Step Approval	710
25.5.2	N-step BAdI	711
25.5.3	SAP-Provided BAdIs and Workflow Templates	712
25.5.4	Line Item Based BAdI	713
25.5.5	Approvers Changing Documents in Application- Controlled Workflow	713
25.5.6	Configuring Approval Processes	714
25.5.7	Agent Determination	716
25.5.8	Ad-hoc Agent Assignment	717
25.5.9	Implementing Dynamic Processes	719
25.5.10	Tracking Workflow Instances	724
25.6	Real-World Example of Application-Controlled Workflow	724
25.6.1	Without Approval	724
25.6.2	Two-Step Approval	725
25.7	Inbox	726
25.8	Offline Functionality	727
25.8.1	Outbound Processing	727
25.8.2	Offline Approval by the Recipient	728
25.8.3	Inbound Processing	728
25.9	Deadline Monitoring	728
25.10	Recommendations When Getting Started	729
25.10.1	Upgrading a System	730
25.10.2	Archiving	732
25.10.3	New Installation	732
26 SAP Customer Relationship Management		735
26.1	Introduction	736
26.1.1	One Order Concept	737

26.1.2	Business Transaction and SAP CRM Business Objects	739
26.1.3	Transaction History	741
26.1.4	SAP CRM UI Framework	742
26.1.5	Portal Integration	745
26.2	Customizing and Workflow Administration	745
26.2.1	Workflow Customizing	745
26.2.2	Workflow Administration	745
26.3	Integrating Your Own Workflows	746
26.3.1	Status Management	746
26.3.2	Actions	746
26.3.3	Agent Determination	747
26.4	SAP CRM Worklist	749
26.4.1	Basics About the Workflow Inbox	751
26.4.2	Dynamic Navigation and the Workflow Inbox	751
26.4.3	Navigational Customizing of the Workflow Inbox	753
26.4.4	Specifics of Dialog Tasks Based on Workflow ABAP Classes	755
26.4.5	Additional Customizing and Personalization	756
26.4.6	SAP ERP Integration	757
26.4.7	Universal Worklist Integration	759
26.4.8	Administrative Standard Workflows	760
26.5	SAP CRM Standard Workflows	760
26.5.1	Marketing: Campaign Automation	761
26.5.2	Sales — Lead to Opportunity	762
26.6	Sample Implementation of a Customer Workflow in SAP CRM	764
26.6.1	Scenario and Requirements for the Custom Workflow	765
26.6.2	Implementing the Workflow	765
26.6.3	Creating a BOL Model for Your SAP CRM Workflows	768
26.6.4	Implementation of the Workflow Application	771
26.6.5	Setting Up Dynamic Navigation	779
26.6.6	Action to Automatically Set the Status "Accepted by Sales"	782
26.6.7	General Settings for Follow-Up Process	786
26.6.8	Executing the Workflow	788
26.6.9	Optional Enhancement	790

27 SAP ERP Human Capital Management —Processes and Forms 793

27.1	Business Overview	794
27.1.1	SAP ERP HCM Processes	794
27.1.2	Form Submission	796
27.1.3	Form Approval	798
27.1.4	Process Form (Revision or Completion)	799
27.1.5	Tracking Forms	801
27.1.6	Other Features	801
27.1.7	Limitations	802
27.2	Technical Overview of HCM Processes and Forms	802
27.2.1	Adobe Document Services	803
27.2.2	Business Function Set	804
27.2.3	SAP ERP HCM Processes and Forms Design Time	804
27.2.4	Adobe LiveCycle Designer	805
27.2.5	Adobe Reader	807
27.2.6	Workflow Template	807
27.2.7	SAP Case Management	808
27.2.8	Web Dynpro ABAP Applications	809
27.3	Standard Workflows in HCM Processes and Forms	809
27.4	Workflow Techniques	811
27.4.1	Calling Web-Based SAP Applications Using Transaction SWFVISU	811
27.4.2	Program Exits	814
27.4.3	XML File for Universal Worklist Configuration	815
27.5	Creating Your Own Workflows	816
27.5.1	Interactive Components	817
27.5.2	Background Components	819
27.5.3	Troubleshooting	820
27.6	Universal Worklist Configuration	820
27.6.1	Subviews	821
27.6.2	Buttons (Actions)	823

28 Setting Up an SAP-Provided SAP ERP Workflow 825

28.1	How Can You Find Workflows Delivered by SAP	826
28.1.1	Using SAP Help Portal to Find SAP Documented Workflows	826
28.1.2	Finding Workflows Within Workflow Builder	827

28.1.3	Technical Workflows	828
28.2	What Is the Business Scenario	828
28.3	Which SAP Workflows Are Involved	829
28.4	How Is the Workflow Started	830
28.5	How Do I Activate This Scenario	831
28.6	Commonly Used SAP ERP Workflows	833
29 ArchiveLink		835
29.1	What Is ArchiveLink?	836
29.1.1	The Technical View of ArchiveLink	836
29.1.2	The Business View of ArchiveLink	837
29.2	ArchiveLink Standard Scenarios	838
29.2.1	How SAP Business Workflow Integrates ArchiveLink	839
29.3	Business Object Types	842
Appendices		843
A	Tips and Tricks	845
A.1	Working with Wizards	845
A.2	Working with Email	847
A.3	Showing the Decision Maker in a Follow-On Step	853
A.4	Creating Your Own User Decision Template	854
A.5	Using Secondary, Before, and After Methods	855
A.6	Looping Through a Multiline List	857
A.7	Creating Object References Dynamically	858
A.8	Deadlines Based on the Factory Calendar	859
A.9	Making the Most of Modeled Deadlines	863
A.10	Ad-Hoc Anchor Step	864
A.11	Review Workflows	865
B	Step-by-Step Troubleshooting Guide	867
B.1	A Workflow That Does Not Start	867
B.2	A Workflow That Stops in Mid-Track	872
B.3	The Most Likely Causes (and How to Avoid Them)	873
B.4	Binding Problems with the Workflow Trace	877
B.5	Why Duplicate or Multiple Workflows Are Triggered	878
B.6	Why an Agent Does Not Receive a Work Item	879
B.7	Why the Wrong Agent Receives the Work Item	881
B.8	Why the Work Item Follows the Wrong Route	882

C	The Administrator's First Aid Guide	883
C.1	Resolving Work Item Errors	883
C.2	Resolving Workflow Instance Errors	888
C.3	Finding Workflows That Appear to Have Disappeared	891
C.4	Resolving Event Linkage Errors	892
D	Workflow Macros	897
D.1	Macros Specific to Object Type Programs	897
D.2	Macros for General Workflow Programming	898
E	Preparation for Java Development	901
F	Additional Information for SAP SRM	907
F.1	Available Business Configuration Sets	907
F.2	BAdI Implementations for Agent Determination	908
F.3	Process-Controlled Workflow Advanced Troubleshooting	920
F.4	Application-Controlled Workflow Advanced Troubleshooting	924
G	The Authors	927
	Index	933

Whether you are building a new workflow from scratch or using an SAP supplied workflow, it is important that you understand the Workflow Builder tool. This chapter gets you started by enabling you to create your first workflow, enhance your workflow, understand data flow, understand types of steps you can use when building a workflow, and get a workflow to production.

8 Creating a Workflow

The central tool for creating, displaying, and processing a workflow is the Workflow Builder. The Workflow Builder is accessed through Transaction SWDD. Alternatively, you can use the menu path TOOLS • BUSINESS WORKFLOW • DEVELOPMENT • DEFINITION TOOLS • WORKFLOW BUILDER • WORKFLOW BUILDER. Within the Workflow Builder, you can create all components of a workflow, including the process flow, the details for each step, and the data flow between steps.

This chapter describes the main features of the Workflow Builder, giving a good foundation to start your own development. To introduce the Workflow Builder, we will explain how to create a simple workflow. We will build upon the simple workflow, enhancing it to show additional features. However, this is not an exhaustive description of all workflow features. When you need more information, the SAP Help Portal (<http://help.sap.com>) describes all of the Workflow Builder features in detail.

Most workflows are started by an *event* (e.g., when a sales order is created, when a quote is entered, when an email arrives, when an error occurs, when a document is printed) that happens in a business application. You define which data from this event needs to be passed to the workflow via *binding*. Events are described in Chapter 13, Using Events and Other Business Interfaces, and Chapter 14, Custom Programs. You can also start any workflow directly, for example, through a transaction code, user interface, or a concept called generic object services (refer to Chapter 13 as well). Because events are a major topic on their own and to keep the focus on the basics of the Workflow Builder, this chapter starts the workflow directly using test tools.

The *workflow container* is used to hold all of the data needed by the workflow. Each workflow has a number of *steps* that execute activities or control the workflow. Data may be passed from one step to another. The activities are handled within *tasks*. You can use the same task in several steps of a workflow (or even in the steps of several different workflows) if you wish. A task has a *task container* that holds all of the data necessary for that task. As described previously, *binding* is the term used to pass data from the workflow container to the task container or from the task container back to the workflow container. Every step has one or more possible *outcomes* depending on the step type, the task, and what the step is doing. For example, for an approval step, possible outcomes might be APPROVE or REJECT.

Expressions are variables used in the workflow to control the workflow (e.g., branches) or to deliver a result (e.g., the agent ID for executing a step). Examples of expressions are simple container elements or the attributes of objects (objects are discussed in Chapter 10, Business Objects, and Chapter 11, ABAP Classes). *Basic data* controls global aspects of the workflow, such as constructor and destructor methods and defaults for the workflow steps. One part of this basic data is version-dependent; the other part applies to all versions. Lastly, the workflow will have one *end point*. There are no hidden exit points.

8.1 Workflow Builder Basics

This section helps you get familiar with the Workflow Builder tool.

8.1.1 Look and Feel of the Workflow Builder

The Workflow Builder provides a graphical view of the workflow definition. The Workflow Builder screen is divided into the following frames (see Figure 8.1 as well); each of the frames can be resized:

► **Workflow**

Here you can insert new steps into the workflow definition and process existing ones. Double-clicking on a step displays the associated step definition.

► **Overview**

The overview graphic shows all steps in a workflow. The part of the workflow graphic displayed in the Workflow frame is marked with a green rectangle.

Changing the size or position of the rectangle changes the display in the Workflow frame.

► Step types

STEP TYPE is the default view when you enter a workflow. It lists all of the types of steps you can insert into your process. To insert a new step into the workflow, drag the desired step to the workflow panel, and drop it on the location where you want the step. When dragging in new step types, you will see a plus icon (+) in the appropriate locations to add steps. In Figure 8.1, you can also see a limited list of step options. By resizing the frame, you can see more step options as shown in Figure 8.2.

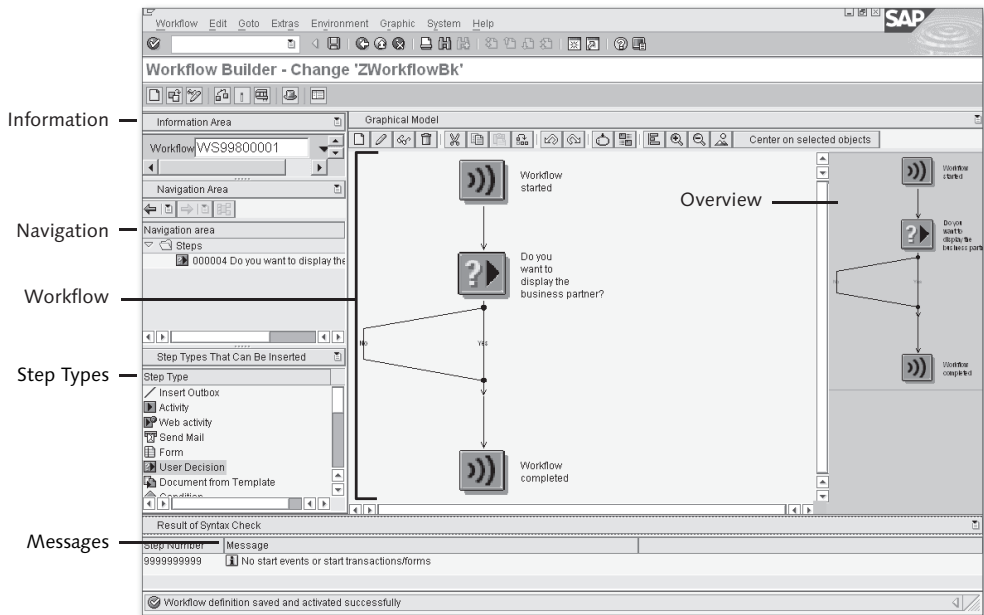


Figure 8.1 Major Elements of the Workflow Builder

► Information

The information frame (see Figure 8.3) displays which workflow is loaded, the status of the workflow, and the version number of the workflow in the original system. To load a different version, select the version. To load a different workflow, enter the workflow number in the format WS<number> and press **↵**. If you do not know the workflow number, select the arrow and you can search for the workflow.

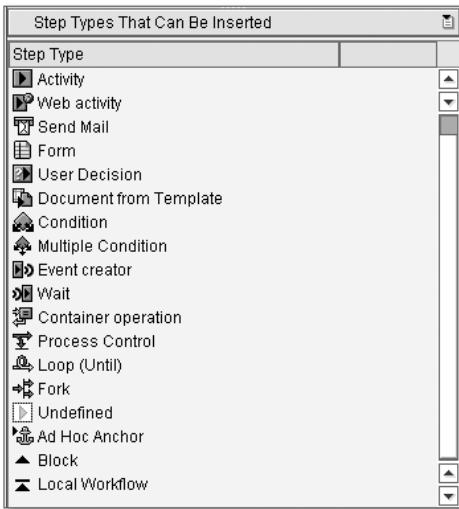


Figure 8.2 More Step Options

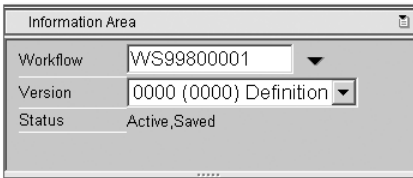


Figure 8.3 Information Area

► Navigation

The navigation frame contains a list of all of the steps in the workflow. You can jump directly to the relevant step definition from the list. As with all of the frames in the Workflow Builder, you can resize this frame to display the amount of information that you require. The step number corresponds to the number in the workflow technical log (logs are discussed in Chapter 6, Workflow Administration).

► Messages

This area contains messages, including general messages and results from where-used lists, syntax checks, and searches.

► Optional information to display

In addition to the frames you see in Figure 8.1, you can optionally switch the STEP TYPES to one of the items in Figure 8.4.

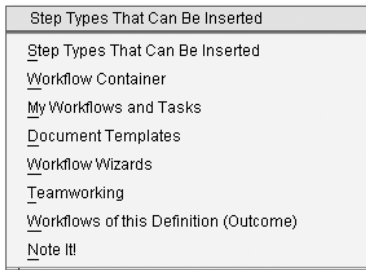


Figure 8.4 Optional Information to Display

- ▶ **Workflow Container**
The workflow container holds all elements required during the life of the workflow. Container elements will be created automatically, but you can also create your own workflow container elements.
- ▶ **My Workflows and Tasks**
This frame enables you to quickly see all workflows you have edited. Additionally, you can search for tasks to add to the list. Your choice also determines what is displayed in the Workflow Explorer, which is a separate transaction (SWDM).
- ▶ **Document Templates**
Document templates add digital documents to a workflow. All of the workflow's document templates that can be used in the step type DOCUMENT FROM TEMPLATE are displayed here. You can generate a where-used list to find the steps in which a document template is used.
- ▶ **Workflow Wizards**
All Workflow wizards available for the definition of your workflow are displayed here.
- ▶ **Teamworking**
Here you can search for steps by selected criteria such as who last edited the step in the definition or which steps are grouped together. The result is displayed graphically in the Workflow frame.
- ▶ **Workflows of this Definition (Outcome)**
Your workflow outbox is displayed here, which displays all currently running workflows for this definition.
- ▶ **Note It!**
You can create notes and documentation about the workflow in this space.

8.1.2 Building Your First Workflow

In this section, you will build a simple workflow, adding to it step by step. In the end, you will have a workflow with different types of workflow steps:

- ▶ The first user interaction step asks the user to make a decision: "Do you want to display the business partner?"
- ▶ If the answer is yes, the business partner is displayed.
- ▶ If the answer is no, an email is sent to the user, telling the user that he did not choose to display a business partner.




This is a simple scenario that should demonstrate how easy it is to build and execute your first workflow process. Eventually, we will link this workflow to a business partner creation in the application that triggers the workflow, so someone will create a business partner, the workflow will notify that a new business partner has been created, and give the user the option to review the business partner. In this chapter, we focus on creating the basic workflow.

We start by creating the workflow and creating a decision step. To keep it simple, you will be the agent. If you have a test system, you may want to build this process yourself. By following a simple example that becomes more sophisticated as the chapter progresses, you will get a good idea of what workflow can achieve.

Starting the Workflow Builder

When the Workflow Builder is called for the first time or you opt to create a new workflow, a newly created initial workflow definition appears (see Figure 8.5).

This initial workflow has the following parts:

- ▶ The start of the workflow definition is indicated by `WORKFLOW STARTED` .
- ▶ The end of the workflow definition is indicated by `WORKFLOW COMPLETED` .
- ▶ The area in which you insert the new workflow definition is indicated by an undefined step with one outcome . Steps are represented by symbols. The name of the outcome is displayed on the arrow leading to the next step in the standard view.

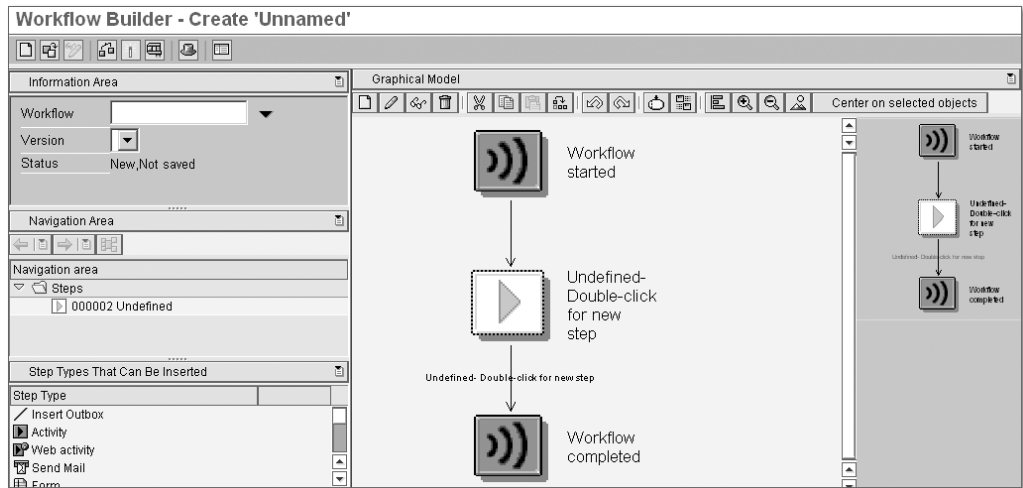


Figure 8.5 Initial Workflow

Create and Save Your First Workflow

Now you are ready to create your first workflow:

1. If you haven't done so yet, navigate to the Workflow Builder by entering Transaction SWDD or using the menu path **TOOLS • BUSINESS WORKFLOW • DEVELOPMENT • DEFINITION TOOLS • WORKFLOW BUILDER • WORKFLOW BUILDER**.
2. If you do not see an initial workflow similar to Figure 8.5, select **WORKFLOW • NEW** or click on the **CREATE NEW WORKFLOW** icon (📄).
3. The workflow now has the title "Unnamed" and has a status, **NEW, NOT SAVED**.
4. Save your workflow by providing an abbreviation and a name. This is discussed in more detail in Section 8.1.3, **Saving, Activating, and Testing**. For now, enter a name you will remember later, such as "zFirstWorkfl". You will also need to provide transport information. Assuming this work is being done on a sandbox, select **LOCAL OBJECT**.
5. Congratulations, you have just created and saved your first workflow.


Your first step illustrates how a user decision works. *User decisions* have their own step type and symbol that looks like this: 🗑️. When a user decision executes, a question together with a predefined list of answers (the possible outcomes) is displayed to the recipients. User decisions are useful in the following situations:

- ▶ Only *one of several* possible alternatives should be processed in the workflow at execution time. An agent needs to make a business decision about which alternative is appropriate.
- ▶ An *instruction* (e.g., a user decision with only one outcome) is required to continue the workflow.
- ▶ For *approval*, *release*, or *status change* steps, the recipient needs to choose whether to approve or reject.


On the DECISION tab, you can make all of the entries required to define an executable user decision. At runtime, the user will see the DECISION TITLE as the SUBJECT text of the work item in his inbox.

Add a User Decision Step to Your Workflow

In the next step of our example, you create a workflow that request a decision about displaying a business partner. The decision will let the user respond "Yes" or "No."


1. In the Workflow Builder, locate the USER DECISION step type in the STEPS THAT CAN BE INSERTED frame.
2. Drag the USER DECISION icon () to the UNDEFINED step. The step definition of the user decision is now displayed.
3. Enter the title "Do you want to display the business partner?"
4. Enter the decision texts "Yes" and "No." The outcome names default to the decision texts, but you can specify your own names if you prefer.
5. Select EXPRESSION in the AGENTS area of the screen. You will see a dropdown box. Set the agent to the expression to WORKFLOW INITIATOR.



The workflow initiator is always the person who started the workflow. There are many other options when determining an agent, as discussed in Chapter 5, Agents. However, to keep the first workflow simple, we will route all steps to the workflow initiator.


6. Complete the step by choosing the TRANSFER TO GRAPHIC button ()

You have now defined a workflow with a user decision step. However, it must be saved and activated before you can use it.

8.1.3 Saving, Activating, and Testing

When you choose **SAVE** () for a new workflow, you must enter an abbreviation and a name for the workflow. You can change both at any time in the basic data of the workflow. You also have to choose a package for transporting the workflow to other systems. If you are in your sandbox environment, then you can select **LOCAL OBJECT**, which denotes it will not be moved to any other system. The status in the title bar of the Workflow Builder is always visible. After you save your workflow, notice your workflow has a **WS** name and a number. Workflow templates are saved on the database with a **WS** and a number. The number range is from the settings you made in Chapter 3, Configuring the System.

To execute your workflow, activate it by choosing **ACTIVATE** () , which compiles the definition so that the SAP Business Workflow Engine can interpret the workflow for execution. Before activating, the workflow definition is subjected to a syntax check. If you only want to check the syntax of the workflow definition, you can choose **SYNTAX CHECK** (). All recognized problems are classified as errors or warnings and are output in the message area, together with other useful information. You can process the step in which the error occurs by clicking on the message.



The workflow will only be activated if no syntax errors are found. The status of the workflow is now **ACTIVE, SAVED**. You can now test your workflow by choosing **TEST** ().

Tip

When you choose **TEST**, the workflow is automatically saved, checked, and activated if it is in the *inactive* state and you are in change mode of the Workflow Builder. There is no need to check and activate separately.

Test One

In this example, you will save, activate, and test your workflow:

1. **SAVE** your new workflow.
2. **ACTIVATE** the workflow.
3. Test the workflow by choosing **TEST** ().
4. In the test environment, choose **EXECUTE** () to start the workflow.

5. Execute the user decision. Notice the text "Do you want to display the business partner?" and the choices available. Choose YES or NO.
6. Now return to the Workflow Builder (by using the BACK arrow in the TEST screen). Toggle the STEPS THAT CAN BE INSERTED to the WORKFLOWS OF THIS DEFINITION frame.
7. You can double-click on the new entry to see the matching workflow log. Notice the status of the workflow is COMPLETED, and the result of the decision step is displayed in the Result column.

Test Two

In the previous test, you executed the workflow immediately, without navigating to the inbox. In this example, you will test the workflow again but also use the inbox:

1. Test the workflow again, selecting TEST and then EXECUTE as you did in the previous test. The workflow executes immediately.
2. When the decision appears, notice you have three options: YES, NO, and CANCEL AND KEEP WORK ITEM IN INBOX.
3. Select the option CANCEL AND KEEP WORK ITEM IN INBOX.
4. Select BUSINESS WORKPLACE.
5. Select INBOX • WORKFLOW. You will see your work item. Double-click the work item, and select either YES or NO.
6. Use the BACK arrow to return from the inbox to the workflow definition.
7. Notice both times when you execute the workflow, there is only one step, the step to make a decision. Later in this chapter, you will add the step to display a business partner if the choice is YES and to send an email if the choice is NO.

If you tried this example in a test system, you may be surprised that when testing the workflow, you are presented with the decision straight away without having to look in your workflow inbox first. This is because the step is configured as part of the synchronous dialog chain by default (described in Chapter 4, Work Item Delivery). Because the person starting the workflow (you) is identical to the person assigned to perform the first step in the workflow (you), you are presented with the task straight away. To change this behavior, follow these steps:

1. Double-click on the USER DECISION step in your workflow.
2. Select the DETAILS tab.
3. Deselect the checkbox ADVANCE WITH DIALOG.
4. Return to your workflow, and ACTIVATE and TEST it.
5. This time when you test, you receive a message at the bottom of your screen that says *Task started under work item ID ##### (current status: In Process)*. The work item number you receive is the process id. Your workflow has the status IN PROCESS and is in the inbox.
6. To execute the work item, select BUSINESS WORKPLACE.

Congratulations on executing your very first workflow!

8.2 Enhancing Your Workflow

Now that you have created your first workflow, it is time to enhance it. This section covers many topics needed to create workflows. You will add a simple deadline to ensure work is performed on time according to process regulations. You will also add the step to display a business partner, which requires you to create new tasks and understand how objects are used in workflow, including how data is accessed and managed. You will also learn about ad-hoc activities for workflows and the use of review workflows so key or sensitive processes can be closely monitored.

8.2.1 Deadline Monitoring

A major advantage of workflow is the ability to monitor workflow steps according to a predefined schedule. This can be very useful if you want to monitor service level agreements or other process controls that ensure timeframes are enforced in the process. You can monitor a number of different date/time deadlines against each workflow step: requested start, latest start, requested end, and latest end:

- ▶ If a *requested start* deadline is active for a work item, then the work item only becomes visible to the recipients after the date/time specified. Background work items are started (executed) when the start deadline is reached.
- ▶ If a *latest start*, *requested end*, or *latest end* deadline is active, then the workflow reacts to the deadline when the specified date/time is reached.

The standard reaction of the workflow system is to send an escalation email. However, you can perform more complex escalation procedures by specifying a deadline outcome name. This lets you add steps to your workflow, which are executed after the deadline fails. This is called a *modeled deadline*.

You define deadlines with respect to a *reference date/time*. The system offers the following reference date/times:

► **The creation date/time of the work item**


For example, assume a workflow has 10 steps. Step 6 must be executed within three hours of its start time. The 3-hour clock starts when Step 6 is initiated.


► **The creation date/time of the workflow to which the monitored work item belongs**

In this example, assume Step 6 of the 10-step workflow must be completed within 2 days of the workflow starting. The clock for the deadline starts from the moment the workflow was initiated, not from when Step 6 was initiated.

► **A date in the form of an expression, which is derived from the context of the application during execution of the workflow**

In this example, assume the step must be completed according to a specific business guideline. Perhaps you have 2 days for a priority B service complaint but only 1 day for a priority A service complaint. Another example would be within 3 days of an invoice posting date. The work item must read the invoice posting date and start the deadline based on the date.

To see the deadline options, double-click on the USER DECISION task in your workflow, and notice the tabs: LATEST END, REQUESTED START, LATEST START, and REQUESTED END. You can activate monitoring of the relevant deadline by selecting one of the deadline tabs, selecting the reference date/time for the deadline, and providing the time details. Activated deadlines are marked with a ringing bell icon () in the tab index.

If you choose EXPRESSION, you must define the reference date/time by specifying expressions for the date or time. Use the value help () for entering expressions. In the example mentioned previously, of a deadline within three days of a posting date, you need to have the posting date in the workflow container. You then use EXPRESSION to select the posting date variable from the container, and select three days for the time. We will discuss more about how to get the posting date (and other fields) in the container in Section 8.3, Basics of Containers and Bindings.

Tip

The value referenced using the expression must be of data type `D` for the date and data type `T` for the time. If you specify a date but *no time*, the system sets the time to `00:00:01` (requested and latest start) or `23:59:59` (requested and latest end).

Specify the deadline by entering duration and an appropriate time unit (e.g., minutes, hours, days). Negative durations can only be used if you define the reference date/time via an expression.

When specifying the type of deadline, the date/time threshold, you can also specify who to notify and what text to send. The text is stored in the details of the task being monitored. For example, if the deadline is on a step to approve purchase requisitions, the task to approve the purchase requisitions holds the text that will be used in case of a deadline. Each task can have its own deadline text.

Tip

With the standard deadline reaction, the status of the monitored work item is unchanged. The work item still has to be executed by one of its recipients before the workflow can continue. If the monitored work item is to be aborted when the deadline is exceeded, you need to use the modeled deadline reaction. Refer to Chapter 9, Advanced Workflow Design Techniques, and to Appendix A, Tips and Tricks, for more details.

Add a Deadline to Your Process

In the following example, you add a deadline to your user decision step and test the deadline:


1. Return to your workflow definition in the Workflow Builder.
2. Double-click on the USER DECISION step. In your user decision step, choose the LATEST END tab.
3. For the reference date and time, select WORK ITEM CREATION.
4. For the TIME field, select MINUTES and enter "2". This means the user will have two minutes from the moment the work item is created to complete the work item.
5. For the RECIPIENT OF MESSAGE WHEN LATEST END MISSED, select EXPRESSION and then select WORKFLOW INITIATOR from the dropdown menu.

6. Test your changed workflow (remember, saving and activating is performed automatically when you choose the TEST option from the Workflow Builder). This time, do not execute the decision step (cancel out of it if you have not removed the ADVANCE WITH DIALOG checkbox).
7. Navigate to the Business Workplace.
8. Wait for the deadline to be exceeded, and you will receive a deadline message in the Business Workplace: The DEADLINE MESSAGES folder contains a message that the deadline was missed. The OVERDUE ENTRIES folder displays all work items that have an overdue deadline.

Tip

The background job for deadline monitoring must be scheduled so that the Workflow Engine can monitor and escalate deadlines. This job is explained in Chapter 3, Configuring the System. When the deadline job runs, all exceeded deadlines are escalated. If you are running this job periodically, then the actual time of escalation is delayed until the job next executes. Use Transaction SWWB to have the job run immediately.

8.2.2 How to Create and Use Tasks

In this section, you add more tasks to our workflow. You will learn how to create a task to display a business partner and how to create a task to send an email. Most steps in your workflow will be tied to business functions: updating a business partner, posting an invoice, approving a purchasing document, updating employee data, and so on. To execute business functions, you use the ACTIVITY steps .

Activity steps are related to tasks, which start with TS. Workflows are created with WS and a number. The number range is determined from the settings you made in Chapter 3, Configuring the System. Those same number ranges are used for TS steps within your workflow.

The user decision step you used earlier is based on a generic decision task (TS00008267) as standard. If you double-click on the user decision step in your workflow, select the CONTROL tab, and you will see the task number. After a TS task is created, it can be reused in multiple workflows.

In this section, you create tasks from within the workflow. However, you can also create tasks independent of the Workflow Builder using Transaction PFTC.

Regardless of how you call the task definition, the same screen for editing the task definition is displayed.

Explanation of Standard Task (TS): Create Screen

Before creating a task to display a business partner, a discussion of the options available when creating a task is needed. Figure 8.6 shows the fields available when creating a new task.

Figure 8.6 Create a New Standard Task (TS Task)

Every task must answer two major questions:

- ▶ *What* should the task do (*display* a business partner, *update* a business partner, *approve* employee leave)?
- ▶ *Who* can do the task?

As the options in Figure 8.6 are discussed, keep in mind that the task must always be able to address the “what” and “who” questions:

- ▶ **ABBR.** is the abbreviated name of the task. You use this name when searching for the task. You should have a naming convention for creating both **TS** tasks and **WS** tasks.

Tip

It is a good idea to decide on a naming convention for task abbreviations to make tasks easier to find, such as a specific prefix for all tasks in the project. Customers normally define their own naming conventions. Some customers may start all of their workflows with **Z**. Others may use the first two letters as functional area, then an underscore, and nine characters of text. An example might be `HR_DisplPern` for display personnel workflow that is part of the HR area, or `ZHR_DisplPern` if you want to start all workflows with a **Z** to indicate they are custom built.

- ▶ **NAME** is the full name of the task.
- ▶ **WORK ITEM TEXT** is the description that will appear in the inbox at runtime. The work item text is very important because it is the first thing the user will see, and it should describe the task for the user. You can pass variables (such as business partner number, invoice amount, etc.) into the work item text to give the item more meaning. Keep in mind that users may have hundreds of work items in their inbox, so the text should be meaningful. During your design phase, you should work with the business users to determine brief but meaningful text to use.
- ▶ **OBJECT CATEGORY** describes how you link this task to actual business data. The options available are Business Object Repository (BOR) objects and ABAP classes. BOR objects are discussed in detail in Chapter 10, Business Objects; ABAP Classes are discussed in Chapter 11.
- ▶ **OBJECT TYPE** is where you enter the actual object name. The trick to this field is you need to know which object to use. Over time, you will become familiar with the objects provided by SAP, and you will become very familiar with the ones you create yourself. Common BOR objects include: `BUS1006` (Business Partner), `BUS2032` (Sales Order), and `PERSDATA` (Employee Personal Data).
- ▶ **METHOD** is the action you want to execute for the task. Examples of methods include create, display, update, block, remove block, approve, and release.

The combination of the OBJECT and METHOD fields answers the “what” question. You need both an object type and method to know what the task can do. When you insert the method, the system takes the following from the definition of the object method, as applicable:

- ▶ Synchronous or asynchronous object method
- ▶ Object method with or without dialog

You cannot change these. If method parameters are defined for the object method, the system gives you the option of creating matching container elements automatically in the task container. The names of these container elements are then identical in the task container and the method container.

Tip

Methods that you execute with a task may be synchronous or asynchronous methods. Section 8.2.3, *Using Asynchronous Tasks*, explains the differences and their effect on modeling.

- ▶ To answer the “who” question, from Figure 8.6, follow the menu path ADDITIONAL DATA • AGENT ASSIGNMENT • MAINTAIN. You are assigning all of the agents who could ever possibly do this task. (Agents are discussed in detail in Chapter 5, *Agents*, and Chapter 12, *Agent Determination Rules*.) For example, if the task is displaying a business partner, who are all of the people who would ever need to display a business partner, or approve a purchase requisition, or enter an expense report. The “who” assigned here is who in the broadest sense of the term. In our examples, we will normally make the task a GENERAL task, which means everyone is a possible agent. To make a task a general task, select ATTRIBUTES • GENERAL TASK.

In addition to the fields in Figure 8.6, also notice the following tabs:

- ▶ The DESCRIPTION tab enables you to add a longer task description. This *task description* appears in the users' inbox at runtime. The *work item text* is the one liner that appears in the inbox, and the task description is the long description the user will see after selecting the work item. The *task description* can also have variables to better describe to the user what the task is and what is required for the task.

The DESCRIPTION tab also enables you to add texts for the deadlines. This is the text the user will see when a deadline has passed. For example, if a user has


two days to update a business partner, after the deadline has passed, a note is sent to the manager that the deadline has passed. The note sent to the manager contains the text entered in the deadline task description. There is text for each type of deadline: `LATEST END TEXT`, `REQUESTED END TEXT`, and `LATEST START TEXT`. Additionally, there is also `COMPLETION TEXT`, which is used for notifications (discussed in Section 8.2.5, Notifications).

- ▶ The `CONTAINER` tab contains data in the task container. The task container holds all required runtime data. The container always contains what object is used and who is executing the task.
- ▶ The `TRIGGERING EVENTS` and `TERMINATING EVENTS` tabs contain events that can be used to stop and start this specific task. This topic is a bit more advanced and is discussed in Chapter 13, Using Events and Other Business Interfaces.
- ▶ The `DEFAULT RULES` tab is used when the task will execute outside of the workflow template (WS task).

An example is a task that starts due to an inbound IDoc (Intermediate Document). Inbound IDocs normally execute a single `TS` task, not a full `WS` workflow. In that case, you must know who should get this task at runtime. For example, if an inbound sales order IDoc fails, you may want to route it to the sales area manager for the sales organization. You would use a rule to find the correct sales area manager as discussed in Chapter 12, Agent Determination Rules.

Creating a Task to Display the Business Partner

Now you will create the task to display the business partner. In this example, you add a step to display a business partner if the result from the user decision step is `YES`.

1. Return to your workflow in the Workflow Builder.
2. Drag an `ACTIVITY` step () to the `YES` branch of your user decision.
3. Select the `CREATE TASK` option from the button list on the button next to `TASK` as shown in Figure 8.7.
4. Enter appropriate texts for the abbreviation and name. For example:
 - ▶ `ABBR`: `zbp_display`
 - ▶ `NAME`: Display business partner
5. Enter the following for the business object fields:

- ▶ OBJECT CATEGORY: BOR Object type
- ▶ OBJECT TYPE: BUS1006
- ▶ METHOD: Display

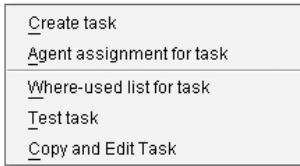





Figure 8.7 Dropdown to Create a Task from the Workflow Builder

Tip

When entering the method, you can see a list of all methods for the object by selecting the dropdown.

6. If you see a message *Transfer missing elements from the object method?*, answer YES (the system is helping you set up the task container; more on this later in Section 8.3, Basics of Containers and Bindings).
7. Provide a work item text. Remember, this is the text the user will see at runtime. To give the text more meaning, we will pass in variables from the business object. Enter the text "Review business partner".
8. Select the INSERT VARIABLES icon (). Select BUS1006 • BUSINESS PARTNER NUMBER. Notice your work item text now has the following variable: `&_WI_OBJECT_ID.BUSINESSPARTNER&`. Place your cursor at the end of the text, and select the description for the business partner by inserting the variable `BUS1006 • DESCRIPTION`.
9. Your work item text should now read *Review business partner &_WI_OBJECT_ID.BUSINESSPARTNER& &_WI_OBJECT_ID.DESCRPTION&*.
10. SAVE () the task. (You need to select LOCAL OBJECT or a development class.)
11. Set up the security for this task by selecting ADDITIONAL DATA • AGENT ASSIGNMENT • MAINTAIN. Normally you would assign the appropriate security role or organizational objects here, but for simplicity, make this a general task by selecting ATTRIBUTES • GENERAL TASK.
12. Return to the workflow. The workflow suggests a binding between the workflow and the task. Accept the binding by choosing ENTER (). Set the agent of

the activity step to the workflow initiator expression using the dropdown help.

13. When you return to your workflow, you should see your new task as part of the YES branch from the user decision task. Now when the workflow executes, you can display a business partner.
14. For testing purposes, you need to provide this business partner number when the workflow starts. To do this, toggle the STEPS THAT CAN BE INSERTED frame to WORKFLOW CONTAINER, and double-click on the workflow container element BUS1006.
15. Select the PROPERTIES tab, and select the IMPORT parameter setting. This means the business partner can be imported when the workflow starts. Normally, this is set so that the application can pass the business partner to the workflow container. If the flag is not set, then the business partner is solely contained in the context of the workflow.
16. Test your workflow. To do so, you must enter an object instance of your business partner object type. Select BUS1006. At the bottom of the screen, you see the OBJECT TYPE and a KEY. Select the dropdown in the KEY field, and enter a valid business partner number. (If you do not have a business partner and are on a sandbox system, you can use Transaction BP to create a business partner. When creating a business partner, it is easiest if you create a person and just provide first name and last name.)
17. After you enter a valid business partner number, select SAVE in the TEST DATA area of the screen. This enables you to select LOAD in subsequent tests to use the same business partner number, instead of entering the business partner number for each test.
18. After you start your workflow, you should execute two tests. In the first test, select YES to the user decision. When you select YES, a business partner should display. In the other test, select NO to the user decision, and the workflow should end.
19. Remember, you can see each execution in the WORKFLOWS OF THIS DEFINITION (OUTCOME) frame.

Hint

When testing the workflow, the first step may or may not start automatically depending on the setting of the ADVANCE WITH DIALOG flag in the DETAILS tab of the task.

Tip


The basic data is used to identify a task. Each task is identified by the object identifier TS (for standard task) and an eight-digit task number automatically allocated by the system during creation.

You must specify a package when saving the task in the same way that you specified a package when saving the workflow definition. The transport object in this case is PDTS. If you use LOCAL OBJECT, this means the workflow cannot be transported to another system.


Tip

If you also assign your tasks and workflows to a task group, you can use the task group as a logical package for the complete scenario. This is not only useful for documenting which workflows are the active valid workflows (as opposed to test workflows or scrapped versions) but also makes navigation in the Workflow Builder and Workflow Explorer easier.

Remember the importance of the work item text for your task. The work item text is displayed in the inbox, in the work item selection reports, and in the workflow log. If a work item text is not specified, the name of the task is used instead. You can use expressions within the work item text. They are replaced at execution time with the current values from the task container.

Note that container elements used in the work item text must be filled using a binding. To use expressions in the work item text, position the cursor at the relevant point in the text, and choose EXPRESSION ()

Tip

If you are looking at an SAP task or a task imported from another system in display mode, you can redefine the task description (which appears in the inbox as the work item text). Choose the REFINISH WORK ITEM TEXT icon () and enter a new text to replace the original text. Using the REDEFINE option overlays your version over the original task description as an extension rather than as a modification.

8.2.3 Using Asynchronous Tasks

A full description of the reasons for using asynchronous methods and the consequences is given in Chapter 10, Business Objects, but for the time being, just

think of them as methods that are not completed until the application itself confirms completion by sending an event to the workflow. For an example, think of a step that deletes something: An employee requests vacation and then later decides he does not want the vacation. The original request is deleted.

Deleting is a natively asynchronous activity. You do not want the task completed until the deletion is completed, so a delete task is normally an asynchronous task. From the Workflow Builder point of view, an asynchronous task requires the terminating events to be activated as outcomes of your step. When modeling an activity step, you can bind the results of a synchronous method to a workflow when the method is completed. However, the asynchronous method cannot return information to the workflow via the bindings from method to task, task to workflow. Instead, you define at least one terminating event. You can bind data from the terminating event to the task.

The *terminating events* of a task express the possible end statuses of the task. A terminating event always refers to the specific object for which the event must be triggered. For example, if the work item deletes document 123456, then the work item should only be completed when the event "document 123456 is deleted" occurs and not when any other document is deleted.

When the work item is created, the Workflow Engine automatically registers that this specific work item is waiting for a terminating event for the specific object involved. This is called an *event instance linkage* (refer to Chapter 13, Using Events and Other Business Interfaces, for more details). You define the object for which the event must be triggered in the terminating event definition using a container element, referenced to the event's object type, in the task container.


Example

For example, the event `DELETED` of object type `BUS2032` (Sales Order) is defined as a terminating event for a task. At runtime, the sales order 123456 is passed via the task container. The work item is now only terminated if the sales order 123456 is deleted and the event `BUS2032.DELETED` is triggered for that sales order. If another sales order is deleted, the work item is not terminated.

If a work item is terminated by an event, the execution of the workflow is continued by a matching outcome. You must activate this outcome, or the terminating event will be ignored. The method does not necessarily have to have been executed before the event occurs. If a work item is aborted, or undefined processing

statuses arise, the work item is not terminated because no event is created. Only when the terminating event is received will the work item finish and the workflow continue.

Within a TS task, you maintain the terminating events on the TERMINATING EVENTS tab. You specify a container element of the task container that at runtime will contain a reference to the relevant object. This is generally the task container element `_WI_Object_ID`. The fields OBJECT TYPE CATEGORY and OBJECT TYPE are filled automatically.

You identify the event by specifying its event name. The event must be defined for this object type. The workflow system creates the event instance linkage required at runtime and activates it. To check the properties of the terminating event, choose the SETTINGS icon (). The properties of the instance linkage are displayed and can be changed.

8.2.4 How to Access Data and Activities

To access data, activities, and events within your workflows, they need to be defined as parts of an object. The object could be part of the Business Object Repository (BOR) (discussed in Chapter 10, Business Objects) or an ABAP class (discussed in Chapter 11, ABAP Classes). Objects can be used in many different workflows, tasks, and rules. SAP components contain many predefined business object types and ABAP classes. These predefined data, activities, or events can be used as is, or you can create your own:

- ▶ Objects describe, for a particular business entity, the data (attributes), functions (methods), and events used in a workflow. In our example, we used business object type `BUS1006`, representing a business partner.
- ▶ Data relating to a business entity needs to be defined as an *attribute* of a business object type before it can be used in a workflow. Attributes are defined as part of the object type to which they are most closely related. For example, the attributes `Material name` and `Material number` are defined within the `Material` object type, but `Order number` and `Order value` are defined as part of the `Order` object type. In other words, the attributes describe data related to the business object.
- ▶ Activities to be performed on or by using a business entity are defined as *methods* of a business object type before they can be used in a workflow. Every

object has methods that define activities that can be executed on that object or its data (e.g., “create business partner” or “update business partner”). Within the method, you can call SAP functions, your own functions, or other external applications.

- ▶ Events are another important component of an object. The events describe the status changes that an object can undergo (e.g., “business partner deleted” or “business partner changed”). A workflow can be started, canceled, or resumed (wait step) when an event of this kind is raised. Just like attributes and methods, events need to be defined as part of a business object before they can be used in a workflow. Events are discussed in Chapter 13, Using Events and Other Business Interfaces.

If you want to use workflow to implement a business process, this is a rough picture of what is involved in accessing the relevant data and functions:

1. Identify all business entities involved in your business process. You sort out which business functions and events you want to map in your scenario and which data you want to access.
2. Check whether the relevant business object types with their methods, attributes, and events are defined in the BOR or as an ABAP class. The grouping of object types in the application component hierarchy, and the option of searching generically for parts of a name, help when looking for existing object types:
 - ▶ If you find an object type whose definition meets your requirements, you can use it without making any modifications.
 - ▶ If you find an object type whose definition does not quite meet your requirements, you can extend its definition.
 - ▶ If you do not find a suitable object type, you can define your own object type.
3. Use the methods, attributes, and events of the object type in the relevant parts of your workflow.

Further information on how to create your own object types or extend existing object types can be found in Chapter 10, Business Objects, and Chapter 11, ABAP Classes.

Adding an Attribute of a Business Partner Object to a User Decision Step

In this example, you add an attribute of the business partner object `BUS1006` to your user decision step:

1. Return to your workflow, and double-click on the user decision step.
2. Add a variable in your user decision title, for example, *Do you want to display business partner &1?*, and set `PARAMETER 1` to the description of your business partner (`&BUS1006.Description&`) using the input help.
3. TEST your workflow. If you saved the data from a previous test, select `LOAD`. Otherwise, you must enter an object key of your business object type before executing the workflow test. You can reuse your test data for each execution by selecting `SAVE` in the `TEST DATA` area after you select a business partner from the dropdown. On subsequent tests, select `LOAD` from the `TEST DATA` area to load your business partner number.

Now that you have tested the workflow a couple of times, you know that after you start the test, you can go to the Business Workplace to see the work item. This time, notice the parameters in the work item text for the user decision task.

8.2.5 Notifications

You have the option to notify someone when a step is completed. It is a simple notification that sends a text note (not a work item) to a specified user when a step completes. Normally it is used to inform someone when a critical step has completed.

Adding a Notification to the Business Partner Display Step

Follow these steps to add a notification so someone will be notified when the business partner display step is completed:

1. Return to your workflow in the Workflow Builder.
2. Double-click on your step to display the business partner (in the `YES` branch of the user decision step).
3. Select the `NOTIFICATION` tab. Notice in this tab you update *who* to send the notification to and *what* the notification text should say.
4. Currently there is no specific `NOTIFICATION (COMPLETION)` text. Double-click on the link to add a notification text.

5. Ensure you are in change mode for the task (using the DISPLAY – CHANGE icon to toggle between change and display mode).
6. Select the DESCRIPTION tab. For the TEXT TYPE, select COMPLETION TEXT. Select the CHANGE TEXT icon so you can update the text.
7. After you have added a COMPLETION TEXT, save your task, and use the BACK arrow to return to the workflow.
8. Update the MESSAGE RECIPIENT FOR COMPLETION to be the WORKFLOW INITIATOR.
9. TEST the workflow again. Be sure to follow the path to display the business partner. After the workflow completes, go to the Business Workplace. You will see the notification text in the DOCUMENTS folder of the inbox.

8.3 Basics of Containers and Bindings

Containers and bindings are a bit tricky when first learning workflow, but as you understand the stability, flexibility, and scalability they provide, you will soon come to appreciate the powerful use of binding between containers. Here are a few of the advantages:

- ▶ You can reuse elements in your workflow.
- ▶ You can make major changes to activities within the process without jeopardizing the process as a whole (or vice versa).
- ▶ Even when the applications that trigger the workflows are changed from release to release, your workflow is sheltered from these changes.
- ▶ You can use parallel activities within the workflow without worrying about data reconciliation problems or interference between the activities.

Containers and bindings are explained in more detail in Chapter 9, Advanced Workflow Design Techniques. This section provides an introduction to how they are used by first focusing on the task container and then focusing on the workflow container.

All of the data needed to execute the method or to display in the task text must be available in the task container. Container elements for the task container are generated automatically when you enter a method in the task. The container elements needed for the execution are recognized by the workflow system, and the workflow system prompts you to automatically insert these container elements in

the task container. In addition to what is automatically provided in the containers, you may want to create your own container elements in the task container and define a binding between the task and the workflow so that these container elements are filled at runtime.

8.3.1 How to Create Containers and Bindings for Tasks

The task container is edited on the CONTAINER tab page (refer to Figure 8.6 as well). To enable the method to process the data, you may (optionally) define a binding from the task container to the method container. The task object itself is automatically bound to the method container. For other method parameters, the system makes a proposal for the binding that can be reviewed on the BASIC DATA tab of the task by selecting the BINDING OBJECT METHOD icon.


However, not defining any binding between task and method is simpler and offers a performance gain, provided the container element names are the same in both containers. In this case, the contents of the task container are matched by element name and automatically copied to the method container (for all elements defined in the method). The same applies to the reverse binding.

Variables used in your work item texts and descriptions are also bound from the task container. In the “Creating a Task to Display the Business Partner” example in Section 8.2.2, How to Create and Use Tasks, you added a task to your workflow and bound the business partner number and description to the work item text. As a reminder, to add variables while editing your description, choose INSERT EXPRESSION to choose a variable from the task container. You can add as many variables to the text as you want (up to the limit of the text field).

8.3.2 Creating Container Elements in the Workflow Container

The work item text of the user decision can display current runtime values from the workflow. You can integrate these values by including variables relevant to the decision directly in the work item text. The variables are replaced at runtime with values from the matching workflow container elements.

Of course, this is just one example of how container elements are used in the workflow, but it is very easy for you to try yourself. (Using the workflow container to link variables for the user decision work item text was done in an example provided in Section 8.2.4, How to Access Data and Activities.)

1. Create container elements by selecting the WORKFLOW CONTAINER frame and double-clicking on the <DOUBLE-CLICK TO CREATE> line in the workflow container tray.
2. Enter the technical name of the container element in the ELEMENT field.
3. Give each container element a technical name (minimum of two characters) that can be used to identify it uniquely. The technical name is not case sensitive and must begin with a letter, but it can be followed by letters, underscores, or digits. Because the technical name is not translated, it is conventional to use English words in multilanguage environments.
4. Under TEXTS, maintain the NAME and the DESCRIPTION (OPTIONAL). Both of these can be translated in multilanguage environments.
5. According to the data type reference of the container element, make the following entries on the DATA TYPE tab: First check whether your container element is modeled on one of the predefined types. Choose the SELECTION OF PRE-DEFINED TYPES icon () and double-click to choose the predefined type. The system carries out the necessary entries for the data type. If you want to create a container element that is not predefined, make the following entries, depending on the data type:
 - ▶ **Object type**
Choose OBJECT TYPE, select an object type category, and enter the name of the object type. Examples include a specific BOR object (such as BUS1006) or a specific ABAP class.
 - ▶ **ABAP Dictionary Reference**
Choose STRUCTURE and FIELD. In this case, you enter a table/structure and field that the container data should be based on. This reserves space in the container equivalent to the field you enter. It works as a "like" statement.
 - ▶ **ABAP Dictionary Data Type**
Choose ABAP DICTIONARY REFERENCE, and enter the table or structure in the field TYPE NAME. Use this to provide a data type to describe the field in the workflow container.

Tip

The specification of an object type is not mandatory. If no object type is specified, the container element can be assigned a reference to any object type at runtime. However, binding restrictions may limit its use later in the workflow.

Tip

A common misconception of workflow is that only one business object can be used per workflow. This is *not* the case. Often cross-application workflows use several different business objects, and the flow itself forms the link between them. A simple example of such a scenario is the link between a scanned document (e.g., the object type `IMAGE`) and the invoice record that is posted to the database (e.g., the object type `BUS2081`).

On the `PROPERTIES` tab, select whether the new element is to be an `IMPORT` and/or an `EXPORT` element. Mark an import element as `MANDATORY` if applicable. Import means this field will be passed from the application to the workflow. For example, a document is created, triggering an event that starts a workflow. For workflow to receive the document information from the application, the receiving element in the workflow container must be marked as `IMPORT`.

8.3.3 Changing Container Elements

For any workflow container element you add, it is your responsibility to bind data to the workflow element; otherwise, the element will be empty. There are several ways to get data into your custom workflow container elements:

- ▶ **By initial values**

You can assign an initial constant value to a container element. When the workflow is executed, the container element is initially filled with this value. Any changes made to the contents of the container element will overwrite this value.

- ▶ **By a container operation step**

A container operation step lets you fill a container element with a constant or another container element.

- ▶ **By bindings in a workflow step**

From any workflow step that can output data to the workflow (such as activity steps, user decision steps, document from template steps, etc.), you can transfer data from the task container of the workflow step to the workflow container (or vice versa) via container bindings. Think of bindings as the rules for parameter passing within your workflow.

- ▶ **By bindings from an event**

Whenever your workflow responds to an event — for example when it is started by a triggering event — data can be passed from the event container to the workflow container. If you want to pass data from a triggering event to

start your workflow, the workflow container elements to be filled from the event container need the `IMPORT` flag turned on before the bindings can be defined. Refer to Chapter 13, *Using Events and Other Business Interfaces*, for more details.

Adding a Custom Workflow Container Element and Binding Data to the Element

In this example, you experiment with adding workflow container elements and manipulating them. This example shows how to add and use your own container elements. In this example, you add a container element to represent a date, add another date to this date, and use the new date in the user decision. (Normally you do not add two dates together, so this is not something you would probably use in production, but it will get you familiar with working with container elements.)

1. Toggle to the `WORKFLOW CONTAINER` frame, and create a new container element by double-clicking on the `<DOUBLE-CLICK TO CREATE>` line. Provide the following information:

- ▶ `ELEMENT`: `NewDate`
- ▶ `NAME`: `MyNewDate`
- ▶ `SHORT DESCRIPTION`: `My first try with containers`

2. Select the `ABAP DICTIONARY REFERENCE`:

- ▶ `STRUCTURE`: `SYST`
- ▶ `FIELD`: `DATLO`

You can now see the new field in your workflow container.

3. Switch from the `WORKFLOW CONTAINER` frame to the `STEPS THAT CAN BE INSERTED FRAME`. Drag the step type `CONTAINER OPERATION` to the line before the `USER DECISION` step, and enter the following information:

- ▶ `STEP NAME`: `AddDates`
- ▶ `OUTCOME NAME`: `two dates added`
- ▶ `RESULT ELEMENT`: Use the dropdown to select `NEWDATE`.
- ▶ `EXPRESSION`: Use the dropdown to see all of the options in the workflow container. Select the object `BUS1006`, and select `CREATEDON`.
- ▶ `OPERATOR`: Use the dropdown to select `ADD`.
- ▶ `EXPRESSION`: Use the dropdown to select `SYSTEM FIELDS`. Then select `TIMLO`.

4. In your Workflow Builder, you now have an ADDDATES step before the user decision step. You will display the result of the container operation step in the user decision step.
5. Double-click on the user decision step. Set PARAMETER 2 to your NEWDATE container element. Use the dropdown help to do this.
6. Use the variable in your work item text by writing &1 in the text where you want the value to appear, for example, *Do you want to display the business partner &1 &2?*

Tip

The &2 declares this to be a reference to parameter 2 of this step definition. It is optional — you can just use &. However, in a multilingual environment, specifying the number is very useful because the variables often appear in a different order in the translation.

7. SAVE, ACTIVATE, and TEST your workflow. Notice the date you see in the work item text. It should be a date far in the future. (Keep in mind the only point for this example was to show how to add a workflow container element and use the container operation step type.)

A Word About the Task Description on the User Decision Step

Although the short text in the generic decision is part of the step definition (to make things simpler), the long text is part of the task. To create your own long text, you can copy task TS00008267 to a new task and write a suitable task description for this new task. You may select your own variables and add these to the task container. After you have created your task, substitute it into the step's CONTROL tab in place of task TS00008267. Do not forget that you need to assign possible agents to your new task.

8.4 Steps

As well as the step types USER DECISION, CONTAINER OPERATION, and ACTIVITY shown earlier, there are other step types available for modeling a workflow. Although ACTIVITY is the main step type to link the workflow to the application, there are many other step types needed to control the workflow process.

8.4.1 What Other Step Types Exist

Table 8.1 shows all step types available in SAP NetWeaver 7.0. The step types cover all of the functions you need to control the workflow process, from what business functions to call, to looping, conditions, container manipulation, using multiple branches, and many other functions.








Step Type	Icon	Runtime Function
ACTIVITY		Execution of a task or subworkflow. At runtime, data is passed from the task or subworkflow to the workflow container on creation of the matching work item, and vice versa on work item completion.
AD HOC ANCHOR		In the definition, you specify workflows that can replace this step. At runtime, an authorized user can select one of the specified workflows. The steps of this workflow then dynamically replace the AD HOC ANCHOR.
CONDITION		Depending on the result of the CONDITION, either the true or the false path is followed. In the condition editor, you can simulate the results of the condition to make the testing of complex conditions easier.
CONTAINER OPERATION		The CONTAINER OPERATION is used to perform arithmetic operations or value assignments to workflow container elements using constants and data in the workflow container. This includes operations on multiline container elements, for example, appending to a list.
DOCUMENT FROM TEMPLATE		A digital document is created from a document template using variables in the text that are filled during workflow execution using the workflow container elements. The workflow container receives a new container element that contains the document ID.
EVENT CREATOR		An event is raised. You fill the event container from the workflow container.
FORK		A FORK is used for parallel processing. You can define how many parallel branches exist and how many branches must be completed for the fork to terminate and the workflow to continue. Alternatively, simply define an end condition.

Table 8.1 Step Types










Step Type	Icon	Runtime Function
FORM		A structure-based container element can be displayed, processed, or approved as a FORM. The data is transferred directly from the workflow container and back again.
LOOP (UNTIL)		A sequence of steps is processed at least once and then repeatedly until the defined termination condition occurs.
MULTIPLE CONDITION		Based on the value of a workflow container element, one of several branches defined in the workflow definition is processed. Any value not specifically assigned to a branch can be processed in an OTHER VALUES branch.
PROCESS CONTROL		This can be used to cancel the execution of a work item or workflow or set a work item to obsolete, so that alternative steps can be taken in the PROCESSING OBSOLETE branch.
SEND MAIL		The text entered in this step type is sent as an email. The task required and the necessary bindings are automatically created by the workflow system.
BLOCK		A BLOCK is a modeling construct that enables you to model a group of steps together. The block has a data interface. Additionally, you can add deadlines to a block, ensuring the entire block must be completed in a certain timeframe. Blocks are discussed in more detail in Chapter 9, Advanced Workflow Design Techniques.
LOCAL WORKFLOW		A LOCAL WORKFLOW is a "free-floating" block that is not connected to the main workflow. Local workflows are triggered by events and enable a design element to be incorporated into the workflow. They may need to execute multiple times during a workflow execution, or may not be executed at all during a workflow execution. Local workflows are discussed in more detail in Chapter 9, Advanced Workflow Design Techniques.
UNDEFINED STEP		An UNDEFINED STEP can be used as a placeholder during development. These steps are ignored at runtime.
USER DECISION		The agent is asked a question and given a predefined list of answers. Each predefined answer is a separate branch in the workflow.

Table 8.1 Step Types (cont.)



Step Type	Icon	Runtime Function
WAIT FOR EVENT		The system waits for a specific event. The work item is only completed if the expected event occurs. Data from the event container can be sent to the workflow container using a binding.
WEB ACTIVITY		The selected container elements are posted using HTTP in an XML or SOAP message. This step can also wait for a message reply.

Table 8.1 Step Types (cont.)

8.4.2 How to Insert New Steps

When inserting new steps, you drag and drop the step type to the location where you want the step. You can insert steps before or after an existing step. Table 8.2 provides an overview of how to insert steps into a workflow.


Where Do You Want to Insert the Step?	What Do You Have to Select?
After a step	Drag and drop on the outcome of the preceding step.
Before a step	Drag and drop on a step to insert before the step.
As a new branch of a fork	Drag and drop on the FORK symbol () at the start of the fork.

Table 8.2 Insert Steps Into a Workflow

The MY WORKFLOWS AND TASKS frame provides an efficient way of inserting tasks as activities in your workflow. MY WORKFLOWS AND TASKS displays tasks and workflows that you have selected or previously edited. The selection is made using a search area that provides diverse selection criteria. If you frequently need a group of tasks to define your workflows, you can put these tasks together in a task group and insert the group into your search area.

Display the contents of the task group in the tray, select the position in your workflow where you want to insert the task, and choose the task by double-clicking on it. An activity step is then automatically created in your workflow that refers to this task.

Insert a Send Mail Step

In this example, you insert a send mail step if the user decides not to display the material:

1. Return to your workflow, and drag the SEND MAIL step to the No branch of the USER DECISION step.
2. Enter the following information:
 - ▶ SEND EXPRESS: Select the checkbox
 - ▶ SUBJECT:
 - Part 1: You chose not to display business partner.
 - Part 2: Select the INSERT EXPRESSION icon, and select BUS1006 • BUSINESS PARTNER.
3. In the large text box, enter text for the email.
4. Select the green checkmark (TRANSFER AND TO GRAPHIC). You are asked for an abbreviation and name, which creates a new task. Enter appropriate values for the name and the description. You also need to provide a development class or have the task be a LOCAL OBJECT.

At this point, your process should look like Figure 8.8. The user decision step is followed by two steps: review of the business partner and send email.

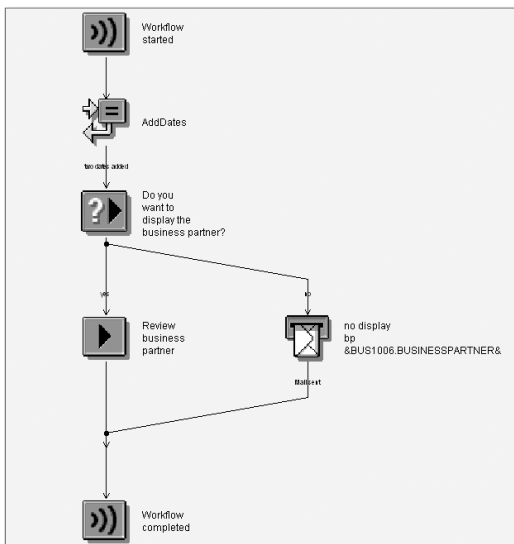


Figure 8.8 Example Process Built in This Section

8.4.3 What Kinds of Outcomes Exist

Outcomes are what the calling step/function can return. Certain outcomes appear in the workflow modeler by default. Outcomes are important because the workflow process branches are based on outcomes. There are different outcomes available according to the step type chosen. To see the possible outcomes of a step, follow these steps:

1. Double-click on a step, and select the **OUTCOMES** tab.
2. In the workflow from Figure 8.8, if you double-click on the **USER DECISION** step (*Do you want to display the business partner?*) and select the **OUTCOMES** tab, you will see three outcomes: **YES**, **NO**, and **PROCESSING OBSOLETE**.
3. If you double-click on the **ACTIVITY** step (*Review business partner*) and select the **OUTCOMES** tab, you see two outcomes: **STEP EXECUTED** and **PROCESSING OBSOLETE**.

Notice in the **USER DECISION** step, the outcome **PROCESSING OBSOLETE** does not appear in the workflow modeler. Normally, the only outcomes that are displayed are ones that require the workflow to react. Some outcomes are optional, and others are only displayed by the system if they are necessary as a result of specific settings. Table 8.3 shows all possible outcomes.

The Outcome Is ...	The Outcome Exists If ...	Notes and Comments
Event name (terminating event of task)	The task was defined with terminating events.	If the underlying method is an asynchronous method, you must activate at least one event as an outcome.
Value name (possible value of method result)	The synchronous object method is defined with a result for which fixed values are maintained in the ABAP Dictionary.	If you deactivate all values of the results, the system activates the STEP EXECUTED outcome instead.
Exception name (method exception)	The object method is defined with exceptions.	Refer to Chapter 10, Business Objects.
System outcome: DOCUMENT COULD NOT BE CREATED	The step is a document from template step.	This outcome is triggered if document creation fails.

Table 8.3 Step Outcomes

The Outcome Is ...	The Outcome Exists If ...	Notes and Comments
System outcome: TASK EXECUTED SYNCHRONOUSLY	The step is a document from template step.	Normal completion of a document from template step.
System outcome: STEP EXECUTED	<ul style="list-style-type: none"> ▶ The activity refers to a synchronous object method without result. ▶ The activity refers to a synchronous object method with result, but no result is selected. 	Normal completion of a step.
System outcome: PROCESSING REJECTED	The indicator PROCESSING CAN BE REJECTED is set.	If processing of the relevant work item is rejected at runtime (e.g., using REJECT EXECUTION in the Business Workplace), the steps defined after this outcome are executed.
System outcome: PROCESSING OBSOLETE	The work item can be set to obsolete using a PROCESS CONTROL step.	<ul style="list-style-type: none"> ▶ The steps defined after this outcome are executed. ▶ This outcome is used to skip steps when modeled deadlines are missed.
System outcome: REQUESTED END, LATEST END, LATEST START	The relevant deadline monitoring is activated and a modeled reaction required. This applies to the workflow wizard MODEL DEADLINE MONITORING.	<ul style="list-style-type: none"> ▶ Within these branches, you model steps to be executed when the deadline is missed. For example, you can model a PROCESS CONTROL step that sets the work item of this step to obsolete. ▶ You cannot deactivate these outcomes.

Table 8.3 Step Outcomes (cont.)

8.4.4 Which Task and Step Attributes Impact Work Item Execution

When setting up a task, there are certain options that impact how the task behaves at runtime. Attributes that influence the execution of work items can be found in both the task definition (when creating the `TS` task) and the step definition (when inserting the task into a workflow). The following settings reside in the task definition (refer to Figure 8.6):

► **Background processing**

Set this checkbox if you want the workflow system (i.e., user `WF-BATCH`) to execute the work item automatically in the background without user involvement. This flag is only available if the underlying method is non-dialog, meaning that it does not require user involvement. The work item will not appear in any inbox, but you can view it via the work item reports or workflow logs.

► **Confirm end of processing**

Set this checkbox if you want the user to decide when the work item is complete. As long as this confirmation has not taken place, the relevant work item remains in the inbox of the agent even if the work item has already been executed. The agent can execute the work item again or forward it. You *cannot* assign this indicator for tasks that are to be executed in the background.

Tip

`CONFIRM END OF PROCESSING` forces the user to indicate that he is done with the task, in addition to completing the work for the task. Only use this flag if you want additional confirmation before completing the work item. This setting also enables the user to add an attachment with the confirmation completion.

The following settings are allowed in the step definition (in the Workflow Builder, double-click on a task, and select the `DETAILS` tab):

► **Processing can be rejected**

Set this checkbox if the user can opt to skip this step. You can model alternative steps to be taken against the matching `REJECTED` outcome.

► **Step not in workflow log**

Work items for this step do not appear in the standard logs, but they are always displayed in the technical workflow log. The graphical log filters out not only these steps but also the outcomes. If a step with several outcomes is filtered out, all of the outcome branches and the steps included in these branches are

filtered out of the graphical log, until the point is reached where the paths merge together again.

► **Advance with dialog (Synchronous dialog chain)**

If the agent of the previous step is also an agent of this step, this step is executed immediately on completion of the previous step as described in Chapter 4, Work Item Delivery.

8.5 Documenting, Translating, Transporting, and Team Development

This section covers topics that should be included in your workflow design, including documenting your workflows, translation, moving your workflow from development to production, and options for collaborative team workflow development.

8.5.1 Documenting Workflow Definitions

Your project should have certain guidelines and expectations around workflow documentation. A workflow definition can be documented in several parts. First of all, you can describe the purpose of the workflow, how it is started, and which subworkflows it calls in the description of the workflow definition. This documentation is found in the Workflow Builder by selecting GOTO • BASIC DATA. Then select the DESCRIPTION tab. You can document using the NOTE IT! link. Additionally, at the task level, you can provide detailed task descriptions that will be available at runtime. Objects can be documented at the object level.

Note

The task description for dialog steps is displayed in the work item preview, so for dialog steps, only instructions to the user should be entered here. Only add technical documentation to the task description if the step is to be executed in the background.

Obviously, there is a lot more needed to document a workflow project than simply the workflow definition. Checklists can be downloaded from <http://www.sap-press.com> to help you with this.

8.5.2 Translating a Workflow into Other Languages

You can translate all language-dependent texts that appear in the workflow definition. This applies to:

- ▶ The names of steps and outcomes
- ▶ Decision texts and titles
- ▶ Container element names

A user can use personal settings to specify whether work item texts and the work item preview are to be displayed in the original language of the workflow or in the logon language (if the text is available in this language).

To get a complete translation, you have to translate the steps in the Workflow Builder and also translate all of the tasks used in the workflow in the translation transaction (SE63). Finally use the compare translation function in the Workflow Builder to import the changes into the current workflow version; otherwise, the changes will only be visible when the workflow is imported into the next system downstream.

8.5.3 Transporting New Versions of a Workflow

A new workflow always has one version with the number 0000. This version is overwritten by default every time you save your workflow. If you do not want the system to do this, you can generate a new version by choosing `WORKFLOW • GENERATE VERSION`. The workflow definition is set to status `NEW, SAVED`.

To avoid increasing the disk space requirements for workflow definitions excessively, only generate a new version under the following circumstances:

- ▶ If you have made incompatible changes
- ▶ If there are production workflows running that refer to the current version

The system manages several versions of a workflow definition. Only one of the versions of a workflow is the active version. Select the active version by activating the appropriate version.

Tip

The import and export parameters of the workflow container are not subject to any versioning.

The information tray displays the version you are processing and whether an active version is involved. To display an overview of all versions, choose the BASIC DATA symbol (📄) in the Workflow Builder. The version number is displayed on the VERSION OVERVIEW tab in the version-independent basic data. An overview of all versions of the workflow definition can be found on the VERSIONS tab in the version-dependent basic data.

Tip

A running workflow always refers to the version of the workflow active at the time it started. Even if subsequently a new version of the workflow becomes the active version, workflows still running continue to refer to the version active when they were started. If you overwrite this version while there are still active workflows, for example, by making changes directly in the production system, unexpected errors can occur.

If a workflow definition is transported into another system, only the active version is transported. If the workflow definition exists in the target system with the same version number, it is overwritten by the transported version if it has no workflows running. Otherwise, the transported workflow definition is saved with a new, free version number. The transported workflow definition becomes the active workflow definition in the target system.

8.5.4 How Do You Share Workflow Development Within a Team

The Workflow Builder offers teamworking functions that support workflow development by a team of developers. It also offers the option to assign a self-defined grouping characteristic to each step. In the optionally displayable TEAMWORKING frame, you can search for steps that have a particular grouping characteristic or particular change data. The Workflow Builder options can also graphically highlight the steps according to the grouping characteristics or according to the last user to make a change. The last user can be seen on the CHANGE DATA tab of the step definition. You define the grouping characteristic of each step on the CHANGE DATA tab page in the step definition.

A grouping name can be assigned to every step. Using this grouping name, you can structure the graphical representation of the workflow. The selection field displays all of the group descriptions previously defined. One of the group descriptions just entered is automatically entered in the selection field and is also

available as a selection for all other steps. You can use this to mark steps that need rework or that are to be transferred to a subworkflow, or you can use this to demarcate the different logical parts of a large workflow.

Index

_EVT_CREATOR 447, 459, 461
_EVT_OBJECT 447, 459, 461
_WI_Group_ID 177
_WI_Object_ID 177, 280

A

ABAP class 236, 290, 353, 444, 451, 679,
741, 743, 751, 754
BI_PERSISTENT~LPOR 377
Business class 374
CL_CRM_MKTP_LAPPL_BASE 747
CL_HRASR00_PROCESS_OBJECT 807
CL_SWF_FORMABSENC 354, 404
Constant 360
Exception class 387
FIND_BY_LPOR 379
Functional method 371
GUID 379
Instance management 382, 406
Private 364
Protected 364
Public 364
Public static constant 403
SAPSRM/CL_WF_PDO_PO 680
Static 360
Structured key 378
Utility class 361, 366, 371
ABAP Debugger 543
ABAP Dictionary
Data type 248
Reference 248
ABAP expression 371
AC_CONTAINER 430
Access sequence 454
Action 823
Action handler 609
Action pane 661
Activation 168
Event 831
Linkages 86
SAP-provided workflow 825
Active version 87, 260

Activity 252
Step 234
ACTOR_TAB 430
Actual agent 141
Adaptive RFC model 901
Ad-hoc
Agent 421, 703, 717
Agent assignment 286, 415
Anchor 252, 718, 864
Anchor step 287
Approver 703
Features 286
Process 681
Process extension 685
Workflow 53, 109, 184, 286, 471
Workflow extension 287
Administration 883
Administrative standard workflow 760
Administrator 74, 152, 161, 472
Default 89
Skills 199
Adobe 798, 802, 806
Designer 642
Document Services 803, 804
LiveCycle Designer 803, 805
Reader 803, 806, 807
Advance
With dialog 231, 240, 259
After method 856
Agent 56, 57, 65, 68, 69, 72, 73, 164, 167,
172, 197, 207, 268, 536, 879, 880
Choose 846
Excluded 879, 881, 925
Possible 92, 93, 471, 485
Responsible 699, 725
Selected 682
Agent assignment 92, 142, 766
Agent determination 64, 72, 74, 141, 178,
188, 550, 698, 716, 726, 745, 747, 816,
878, 879, 881
Error 169
Error handling 188

- Agent determination rule 291, 413, 414, 415, 747
 - Container* 430, 436
 - Defining* 415
 - Exception* 431
 - Expression* 437
 - Function module* 429
 - Manual selection* 421
 - Result* 421, 422
 - Using* 420
 - Alert 112, 215
 - Alloy 97, 568, 645, 652, 655
 - Decision body* 660
 - Sidebar* 646, 659
 - Anchor
 - Ad-hoc* 252
 - Step* 865
 - Annotate PDF document 803
 - API 484
 - Application Programming Interface → API
 - Application-controlled workflow 688, 710
 - Approval framework 681
 - Approval level 685
 - Approval Procedure wizard 845
 - Approval process 40
 - Overview* 923
 - Approver
 - Multiple* 668
 - Archive Development Kit 838
 - ArchiveLink 835, 838, 839, 857
 - Archiving 81, 208, 838
 - Workflow* 556
 - Association 295, 296
 - Asynchronous 303, 320, 464
 - Method* 237, 303, 320, 323, 324, 340, 466
 - Task* 105, 241, 875
 - Attachment 69, 98, 101, 102, 103, 115, 795, 804, 850
 - SAP Business Object* 101
 - Attribute 178, 292, 293, 301, 310, 315, 360, 368, 385
 - Buffer* 333
 - Definition* 301
 - Audit trail 108
 - Auditor 75
 - Authorization 88, 90, 136, 188, 189, 553
 - HR structural* 874, 880
 - Problem* 207
 - Role* 90
 - Structural* 91, 149
 - Automate task 662
 - Automatic forwarding 171
 - Automatic workflow customizing 83, 87, 88, 539
- ## B
-
- Backend service
 - SAP_PA* 805
 - SAP_PD* 805
 - SAP_PT* 805
 - Background
 - Method* 327, 340, 872, 873
 - Processing* 258, 551, 883
 - Step* 536, 819
 - Task* 89, 174, 476, 544, 847
 - Work item* 175, 478, 884
 - Background job
 - Missed deadline* 89
 - SWWDHEX* 885
 - BAdI 448, 490, 492, 692, 699, 700, 712, 715, 720, 726, 747, 783, 884, 908, 925
 - BBP_WFL_APPROVE_BADI* 712, 717, 720, 723, 724
 - BBP_WFL_SECURE_BADI* 712
 - SAPSRM/BD_WF_DEADLINES* 729
 - SAPSRM/BD_WF_RESP_RESOLVER* 908
 - SAPSRM/BD_WF_REVIEWER_F4* 704
 - SAPSRM/BD_WF_REVIEWER_RULE* 704
 - WF_CORE_RESP_RESOLVER* 700
 - BAPI 292, 319, 321, 515, 654, 673
 - Basic data 222
 - Batch job 489
 - Cleaning up* 208
 - Customizing* 94
 - Monitoring deadline* 89
 - BC Set 705, 706, 733, 907
 - Before method 856
 - Benefit 95
 - BI → SAP NetWeaver Business Warehouse

- Binding 56, 173, 221, 222, 246, 249, 263, 277, 279, 315, 420, 459, 460, 479, 480, 545, 546, 549, 556, 560, 570, 602, 607, 634, 807, 850, 874, 877, 884, 886, 888
 - Definition* 246
 - Detailed explanation* 276
 - Error* 889
 - Problem* 895
- BOL 737, 742
 - Framework* 751, 752
- BOR 178, 236, 289, 353, 354, 358, 386, 444, 450, 451, 544, 717, 739, 741, 743, 751, 842
 - Browser* 297
 - BUS1006* 236, 239, 444
 - BUS2032* 236
 - IFARCH21* 842
 - Object method* 753
 - SOFM* 643
 - WF_DOC* 642
- BOR object 123
 - BUS2000108* 762
 - BUS2000111* 762
 - BUS2000116* 755
 - BUS2012* 123
- BPEL 507
- BPM → SAP NetWeaver Business Process Management
- Branch 252
- Breakpoint
 - External* 545
- BRF 689, 690, 697
 - Evaluation ID* 708, 710
 - Event* 698
 - Expression* 698
- BSP 568, 582, 613, 614, 625, 626, 628, 633, 636
 - Application* 613, 617, 619, 622, 625, 629, 632, 634
- Buffering 192, 194, 436
 - Error* 169, 192
 - Problem* 427
- Business Object
 - SYSTEM* 846
- Business Add-In → BAdI
- Business Application Programming Interface → BAPI
- Business Communications Services 500, 727
- Business Configuration Set → BC Set
- Business driver 59
- Business function set 803, 804
- Business Intelligence → SAP NetWeaver Business Warehouse
- Business Object 66, 69, 70, 222, 543, 659
 - ASBPURLIST* 725
 - Attribute* 293
 - BUS1001* 310
 - BUS20001* 738, 739
 - BUS2010022* 761
 - BUS2012* 405
 - BUS2032* 271
 - BUS2121* 347, 348, 349, 679, 707, 714, 715, 910
 - BUS2200* 919
 - BUS2201* 911
 - BUS2202* 918
 - BUS2203* 914
 - BUS2205* 916
 - BUS4101* 712
 - Encapsulation* 292
 - Event* 294
 - FORMABSENC* 346, 864
 - Inheritance* 292
 - Interface* 296, 842
 - Key* 293
 - LFA1* 310
 - Method* 294
 - Overview* 291
 - Polymorphism* 292
 - SelfItem* 847, 848
 - SOFM* 850
 - Status* 310
 - SYSTEM* 346, 767, 857, 858
 - SYSTEM.GenericInstantiate* 858
 - USR01* 346, 354, 374, 401, 853
 - WEBSERVICE* 629, 631
- Business Object Browser 298
- Business Object Builder 297, 299, 313, 329, 339, 450
- Business Object Layer → BOL
- Business Object Repository → BOR
- Business object type 294, 295, 329, 459, 466
- Business process 59, 66, 201, 441, 667

- Business Process Execution Language → BPEL
 - Business Process Management → SAP NetWeaver Business Process Management
 - Business process owner 201
 - Business Rules Framework 923
 - Business Server Pages → BSP
 - Business transaction 737
 - Business transaction event 393, 448, 455
 - Business user 236
 - Business Workplace 86, 96, 97, 98, 100, 128, 230, 234, 592, 625, 656, 670, 728, 880, 922
 - BW → SAP NetWeaver Business Warehouse
- C**
-
- Calendar 647
 - Entry* 651
 - Item* 650
 - Callback
 - Dialog* 621
 - Handler* 621
 - URL* 618, 625, 629, 630, 631, 632
 - Canceled status 482
 - Case management 808
 - ccBPM 215
 - CCTS 507
 - Change
 - Management* 82
 - Salary* 801
 - Working time* 801
 - Change document 393, 450
 - CDHDR* 451
 - Characteristics 216
 - Check function module 291, 462, 492, 493, 494, 495, 542, 553, 868, 879, 894, 895
 - Choose agent 846
 - Cinderella effect 193
 - Circular Distribution wizard 846
 - CLASS_CONSTRUCTOR 369
 - Clearing task 208
 - Client copy 92
 - Commit 490
 - COMMIT WORK 485, 868
 - Complete manually 175
 - Complete workflow 488
 - Completion text 238
 - Complex data type 279
 - Composition 295, 296
 - Condition 252, 454, 464, 493, 547, 552, 888
 - Multiple* 264
 - Record* 454
 - Simulation* 552
 - Start* 541
 - Start condition* 462
 - Confirm
 - End of processing* 258, 363, 367, 754
 - Consistency check 554, 870
 - Constant attribute 369
 - CONSTRUCTOR 369, 375
 - Container 56, 222, 276, 277, 479, 480, 484, 487, 495, 556, 570, 886
 - _ATTACH_OBJECTS* 850
 - Agent determination rule* 416
 - Data* 673
 - Definition* 246, 247
 - Detailed explanation* 276
 - Displaying* 181
 - Element* 249, 279, 424, 430
 - Event* 277
 - Method* 277, 856
 - Modifying* 182
 - Operation* 252, 476, 559
 - Operation step* 249
 - Persistent* 899
 - Purpose* 246
 - Rule* 277, 278
 - Runtime* 899
 - Task* 277, 278
 - Type* 277
 - Workflow* 277
 - Container element 185, 466, 633, 848, 851
 - _RULE_RESULT* 851
 - _WI_Group_ID* 177
 - _WI_Object_ID* 177, 280
 - Change* 249
 - Multiline* 268, 269, 271
 - Content server 839
 - Contract 913
 - Controller layer 603
 - Core Component Technical Specification
 - CCTS
 - Cost saving 47

Create event 553
 Create task 235, 238
 Crisis response 203
 CRM → SAP CRM
 cross-component Business Process Management → ccBPM
 Custom code 561
 Custom relationship 435
 Customer handler
 Inbound 676
 Customer Relationship Management → SAP CRM
 Customizing 83, 539, 559
 Activities 94
 Automatic 83, 85, 539, 559
 SAP-provided workflow 825
 Task-specific 91

D

Data Dictionary 560
 Data quality 64
 Data source
 Model 674
 Data Store Object (DSO) 213, 215
 Data type 521
 Complex 279
 Database
 Attribute 317, 335, 337
 Reorganization 208
 Deadline 53, 61, 165, 180, 233, 554, 563, 647, 885
 Agent 143
 Customizing 89
 Data 215
 Defining 231
 Job 886
 Latest end 231
 Latest start 231
 Message 234
 Modeled 232, 274, 863
 Monitoring 165, 166, 231, 728
 Monitoring for work item 195
 Notification 207
 Object 859
 Reporting 166

Deadline (cont.)
 Requested end 231
 Requested start 180, 231
 Work item 143, 174, 478
 Debugger
 ABAP 543
 Debugging 550
 Decision 648
 By type view 649
 Decision set 688, 917
 Default agent-determination rule 138
 Default rule 238
 Default workflow administrator 89
 Defining event 447
 Delegation 295, 312, 583
 Delete
 Work item history 209
 Workflow 209
 Delta pull 114
 Design
 Phase 236
 Process-oriented 508
 Sanity check 78
 Destination
 Logical 87
 Diagnosis
 Buffer synchronization 192
 Transaction 538
 Utility 151
 Diagnostics 535
 Log 537
 Tool 535
 Dialog
 Decision 654
 Work item 174, 479, 481, 482, 884
 Dialog chain
 Synchronous 105, 471, 489
 Dictionary type
 WFSYST-AGENT 853
 Digital personnel file 809
 Display work item 187
 DMR 665
 Document 835, 838
 Finder 836
 From template 252, 638
 Scanning 836

- Document (cont.)
 - Storage* 836
 - Template* 225, 641
 - Viewer* 836
 - Documentation 259, 303, 536, 830
 - Workflow definition* 259
 - Document-based workflow 840
 - Domino Designer 654, 659
 - Duet 97, 568, 661, 667
 - Action* 663
 - Action Pane* 664
 - Approval workflow* 670, 671
 - Client* 666, 671
 - Client add-on* 669
 - Server* 669
 - Template* 674
 - Toolbar* 663
 - Workflow approval* 661, 663, 667, 669
 - Dynamic contextual message 666
 - Dynamic link 663
 - Dynamic loop 268
 - Dynamic Messages Region → DMR
 - Dynamic parallel processing 143, 267, 268, 857
 - Dynamic parameter 609
 - Dynamic validation 801
- E**
-
- ECL 836
 - Edit form 800
 - Electronic image 838
 - Email 98, 275, 473, 499, 666, 847, 884
 - Header* 663
 - Notification* 663
 - Employee
 - Hiring* 804
 - Employee Self-Service → ESS
 - Encapsulation 292
 - End date/time
 - Latest* 729
 - Requested* 729
 - End of processing
 - Confirm* 258
 - Engineering Client Viewer → ECL
 - Enhancement Package 804
 - Enhancement Spot
 - HRASR00GENERIC_SERVICES* 805
 - Enterprise JavaBeans 804
 - Enterprise Service
 - Bundle* 510
 - Bus* 509
 - Enterprise Services Builder 523
 - Enterprise Services Repository 511, 520, 529
 - Enterprise SOA → SOA
 - Error
 - Administration* 168, 171
 - Agent determination* 169
 - Buffering* 169
 - Event linkage* 170
 - Monitoring* 171
 - Resolution* 168
 - Status* 169
 - Work item* 170
 - Escalation 53, 67, 69, 81, 232
 - ESS 796
 - Evaluation path 415, 418, 420, 434
 - Event 92, 221, 294, 328, 340, 361, 441, 442, 481, 488, 632, 634, 635
 - Activation* 92
 - Binding* 870, 894
 - Business partner workflow* 460
 - Business transaction event* 455
 - Change document* 449
 - Container* 277, 450
 - Container element* 450
 - Create* 553
 - Creation* 448
 - Creator* 252, 323, 442, 457
 - Definition* 303, 328, 442, 447
 - Event container element* 459
 - Handler* 394
 - Initiator* 442, 459, 894
 - Local* 468
 - Manager* 495
 - Master data change* 454
 - Material change* 450
 - Message control* 453
 - Name* 442, 466
 - Parameter* 442, 450
 - READY_FOR_RELEASE* 705
 - READY_FOR_WORKFLOW* 693

- Event (cont.)
 - Receiver* 394, 443, 459, 465
 - Simulation* 552, 870, 878
 - Start* 460
 - Start condition* 458, 462
 - Status change* 451
 - Terminating* 457, 464, 466, 492
 - Trace* 446, 451, 470, 539, 540, 541, 553, 869
 - Triggering* 268, 457, 458, 828
 - Triggering step* 252
 - Type linkage* 466
 - User event* 801
 - Waiting for* 254
 - Work item* 478
 - Workflow API* 457
 - Event linkage 86, 291, 443, 459, 460, 465, 466, 467, 469, 494, 745, 760, 804, 807, 831, 868, 870, 878, 879, 892, 893, 894, 895, 920
 - Error* 170
 - Table* 444, 559
 - Event queue 467, 469, 470, 488, 539, 540, 870, 875, 892
 - Event raising 447
 - Business transaction event* 455
 - Change document* 448, 449
 - Financials* 455
 - Human Resources* 454
 - Message control* 448, 453
 - Status change* 451
 - Status management* 448
 - Exception 324, 325, 342, 716
 - Exception class
 - CX_BO_ABORT* 388
 - CX_BO_ERROR* 388
 - CX_BO_TEMPORARY* 388
 - Exception handling 273
 - Exchange Infrastructure → SAP NetWeaver
 - Process Integration
 - Excluded agent 134, 139, 879, 881, 925
 - Execute
 - Rule for work item* 191
 - Without agent check* 136
 - Work item without agent check* 192
 - Expense approval 653
 - Export parameter 325
 - Expression 100, 222, 232
 - Agent determination* 139
 - Extended notification 97, 125, 127, 128, 671
 - Extended organizational plan 148
 - External breakpoint 545
 - External user 130
- ## F
-
- Factory calendar 373, 859, 860
 - Fax 841
 - Flowchart 78
 - Fork 252, 267, 272
 - Form 253, 640
 - Approval* 798, 799
 - Edit* 800
 - Electronic* 637
 - PC document* 641
 - Simple* 638
 - Submission process* 798
 - Tracking* 801
 - Transfer* 800
 - Forum 82
 - Forward task 756
 - Forwarding 98, 106, 134
 - Work item* 106
 - Function module 320, 343, 415, 675
 - Check* 462, 492, 493, 494, 495, 542, 553, 868, 879
 - Receiver type* 462, 494, 495, 496, 497
 - RH_GET_ACTORS* 851
 - Rule* 429
 - SAP_WAPI_READ_CONTAINER* 480
 - SWE_CREATE_EVENT* 869
 - SWE_EVENT_CREATE_FOR_UPD_TASK* 495
 - SWE_EVENT_CREATE_IN_UPD_TASK* 491
 - SWE_EVENT_MAIL* 871
 - Functional method 370, 371, 385, 615
- ## G
-
- General task 136, 141, 150, 237, 239, 624, 767, 831, 880, 922
 - Generic decision task 672

Index

Generic Interaction Layer → GenIL
Generic Object Services → GOS
GenIL 742, 768
GET_PROPERTY 897
Globally Unique ID → GUID
Going live 80
GOS 221, 405, 441, 470, 471, 497
Graphical workflow log 109, 184
Groupware 54
GUID 359, 796, 814
 Format 163

H

Help desk 196, 200
Hide task 162
Hiring
 Employee 804
 Process 798
History region 666
HR administrative service 793, 804
HR infotype 454
HR structural authorization 874, 880
HRASR00GENERIC_SERVICES 805
Human factor 40
Human involvement 44
Human resources
 Event 394
Human role 206
Human-centric process 40

I

IFDISPLAY 314
IMG 84, 90, 91, 171, 656, 658, 689, 694,
 700, 730, 738, 742, 752, 753, 754, 757,
 760, 762, 764, 780, 825, 921
Implementation guide → IMG
Import
 Adaptive RFC model 605
 Parameter 324
In process 482
Inbound customer handler 676
Inbox 98, 236, 648, 726
 Log 108

Include
 330, 330, 330
InfoCube 213, 215, 216
InfoObject 212
 Characteristics 213, 216
 Key figure 213, 216
InfoProvider 213
InfoSet 213
Infotype 149, 455, 805, 819
 105 149
Inheritance 292, 295, 312
Instance linkage 180
Instance-dependent 320
Instance-independent 316, 320, 360
Instantiate 382
 Object 383
Interface 294, 295, 296, 313, 358, 428, 614
 BI_OBJECT 362
 BI_PERSISTENT 359, 362, 363, 376
 IF_WORKFLOW 362, 365, 374
 IFAPPROVE 359
 IFDISPLAY 314
 IFOBJECT 358
Invoice 908, 916
Item type 823
iView 603

J

Java class 605
Java EE 5 595
JCo destination 605, 901, 902
Job 144, 147
 SWWCOND 265
 SWWDHEX 165
 SWWERRR 873
Job role 684

K

Key 293
Key field 300, 314
Key figure 216

L

Latest end 231, 232, 233, 373
 Latest end date/time 729
 Latest end text 238
 Latest start 231
 Latest start date/time 729
 Latest start text 238
 Launch handler 621
 Leave request 663
 Link
 Dynamic 663
 Local event 468
 Local object 229
 Local Persistent Object Reference → LPOR
 Locking 561, 562
 Log 85
 Technical 224, 537
 Logical destination 87
 Logical expression 462
 Logical unit of work 490
 Logically deleted 272
 Loop
 Until 253
 Lotus 647, 648, 650, 651
 Client 656
 Lotus Notes 53, 54, 97, 645, 646, 654
 LPOR 359, 376, 380, 401
 Persistent 376
 Transient 376

M

Macro 329
 898
 BEGIN_METHOD 897
 SWC_CALL_METHOD 344, 405, 900
 SWC_CONTAINER 898
 SWC_CREATE_CONTAINER 898
 SWC_CREATE_OBJECT 332, 336, 899
 SWC_GET_ELEMENT 340, 430, 899
 SWC_GET_PROPERTY 331, 333, 339, 405, 900
 SWC_GET_TABLE 340, 430
 SWC_GET_TABLE_PROPERTY 331
 SWC_REFRESH_OBJECT 900

Macro (cont.)
 SWC_SET_ELEMENT 332, 344
 SWC_SET_TABLE 334, 899
 Mail
 inbox 646, 647
 Sending 276
 Manager Self-Service → MSS
 Mass deletion 192
 Message
 Control 453
 Dynamic contextual 666
 Type 521, 526
 Method 236, 292, 294, 303, 310, 319, 321, 339, 341, 365, 544, 550
 After 856
 Asynchronous 237, 320, 323, 466
 Attribute 332
 Before 856
 Container 277, 278, 856, 878
 Creation 319
 Default 178, 321
 Exception 303, 324
 Instance-independent 320
 Key 330
 Parameter 303, 320, 324
 Result 320
 Secondary 855, 856
 Synchronous 237, 322
 Testing 550
 Virtual 327
 WF_DOC_CREATE_SOFM 642
 Without dialog 237
 Metrics 62
 Microsoft Office 54, 642, 661
 Outlook 53, 97, 661, 663, 667, 673
 Suite 669
 Visual Studio 669, 671, 672, 674
 Midnight magic 193
 Model data source 674
 Model deadline monitoring 846
 Model layer 603
 Modeled deadline 232, 274, 863
 Monitoring 257
 Modeling Support wizard 845
 MSS 796

Index

Multiline 71, 301
 Attribute 334, 337
 Container element 268, 269, 271
 List 857
Multilingual environment 260
Multiple approver 668
Multiple condition 264
Multiple line items 267
MVC 613
My Workflows and Tasks 225

N

Naming convention 236
NOBODY_FOUND 430
Notification 112, 131, 238, 245, 864
 Agent 143
 Extended 671
NWDI 120, 903
NWDS 120, 901, 905, 906

O

OAGI 506
OASIS 506
Object 58
 Category 236
 Identification region 665
 Instance 293, 319, 618
 Key 170
 Local 229
 Organizational 139, 715
 Reference 182, 293, 858
 SBOOK 299
 Selection 796
 Visualization 813
 WI_OBJECT_ID 278
Object type 236, 248, 293, 314, 442
 Attribute 315
 Example 346
 Key 314
 Method 319
 SELFITEM 345
 Status 312
 User 346
 WF_TASK 346

Object-based navigation 753, 759
Object-oriented 55, 354, 375, 394
 Data model 69
Obsolete processing 276, 863
OCR 839
Offline 672
Offline scenario 803
Open Applications Group → OAGI
Open process 801
Opportunity and lead management 735
Optical Character Recognition → OCR
Organization for the Advancement of Structured Information Standards → OASIS
Organizational data 416, 428
Organizational Management 52, 144, 745, 796, 804, 805, 809, 880
Organizational model 198, 416
Organizational object 139, 415, 419, 427, 431, 436, 715
Organizational plan 91, 144
 Avoiding 150
 Buffer 157
 Extended 148
 Maintenance strategy 151
 Relationship 146
 Transport 153
 Validity period 145
Organizational unit 89, 144, 146
Outbox 96, 98, 110, 891
Outcome 222
Overdue entry 234

P

Parallel 667
Parallel processing 267
 Dynamic 143, 857
 Table-driven dynamic 267
Parameter 324
 Export 325
 FORM_SCENARIO_STAGE 800
 Import 324
 INITIATOR_ROLE 797
 PROCESS_GROUP 797
 Result 303, 320, 324
 WI_ID 812

- Parameter (cont.)
 - WLC* 177
 - Parked event 468
 - Participant 56
 - Partner channel management 756
 - Partner determination procedure 747
 - Partner processing 747
 - Pattern 81
 - PC document 641
 - PDF 803
 - Pending view 649
 - Perform auto-customizing 83
 - Performance 164, 167
 - Persistent container 899
 - Person 149
 - Personnel administration 795, 805
 - Personnel number 149
 - PI → SAP NetWeaver Process Integration
 - Plan version 87, 148
 - Polymorphism 292
 - Position 145, 147
 - Possible agent 92, 93, 106, 134, 135, 413, 471, 485, 831, 879, 880
 - Assignment* 137
 - Post processing framework 746
 - Prefix number 90
 - Presentation layer 603
 - Print list 838
 - Priority
 - Work item* 107
 - Private
 - Attribute* 385
 - Method* 365
 - Procedures manual 63, 66
 - Process
 - Consistency* 61
 - Control* 863
 - Control step* 253
 - Form* 800
 - Human-centric* 40
 - Level* 917
 - Object* 796, 798, 808, 820
 - Quality* 61
 - Reference number* 798
 - Schema* 695, 733
 - Selection* 797
 - Process (cont.)
 - Speed* 61
 - Transfer* 801
 - Process form 799
 - Process-controlled workflow 688, 689, 729
 - Processing
 - Can be rejected* 258
 - Duration* 165
 - Obsolete* 276, 863
 - Parallel* 267
 - Time* 166
 - Process-oriented design 508
 - Project management 59
 - Proof of concept 556
 - Proxy 520, 528
 - Implementation* 530
 - Publish and subscribe 443
 - Purchase order 907, 911
 - Purchasing specialist 682
- ## Q
-
- Query 214
 - Queue
 - WORKFLOW_LOCAL_xxx* 542
 - Quote 908, 917
- ## R
-
- Reader rights 803
 - Ready status 481
 - Receiver
 - Function* 459
 - Function module* 291, 460, 462, 467, 494, 495, 496, 497, 553
 - Recipient 134, 140
 - Reduce costs 61
 - Refresh
 - Buffer* 193
 - Organizational environment* 194
 - Rejecting
 - Work item* 110
 - Repetition 44
 - Replace 98, 481
 - Manually* 175

- Report
 - Standard* 162
 - Reporting 54, 82, 162, 211, 487
 - Agent workload* 166
 - Deadline target* 166
 - Diagnosis of workflow with error* 164, 172
 - Orphaned work item* 422
 - Processing duration* 165
 - Queued workload* 167
 - Work Item by processing duration* 165
 - Work item by task* 165, 175
 - Work item selection* 173
 - Work item with monitored deadline* 165, 166
 - Workflow for object* 163
 - Workflow performance* 164
 - Workload analysis* 166
 - Request view 648
 - Requested end 231
 - Requested end date/time 729
 - Requested end text 238
 - Requested start 231, 232, 481
 - Requested start deadline 362, 876
 - Requirements gathering 44, 57
 - Agent determination* 72, 74
 - Data* 66
 - Future enhancement* 81
 - Suitability* 60
 - Reserve 98, 482
 - Work item* 102, 113
 - Responsibility 416, 424, 425, 686
 - Responsibility rule 76, 191, 415, 423, 427
 - Priority* 425
 - Responsible agent 134, 138, 699, 725, 910, 919
 - Restart
 - After error* 175
 - Workflow* 488
 - Resubmission 103, 107
 - Result parameter 303, 320, 324
 - Return on investment 46, 62, 289
 - Reusability 291
 - Review workflow 287, 719, 865
 - Reviewer 287, 684, 703, 719
 - RFC 488, 541, 542
 - Error* 549
 - Model* 605, 607
 - RFC (cont.)
 - Monitor* 893
 - Queue* 488, 541, 868, 877
 - RFC workflow destination
 - Configuring* 87
 - Workflow_Local* 87, 542
 - RFx 919
 - RH_GET_ACTORS 421, 851
 - RH_SAP_ORG_OBJEC_RELATE 428
 - Role 90, 795
 - SAP_BC_BMT_WFM_ADMIN* 91
 - SAP_BC_BMT_WFM_CONTROLLER* 91
 - SAP_BC_BMT_WFM_DEVELOPER* 91
 - SAP_BC_BMT_WFM_PROCESS* 91
 - SAP_BC_BMT_WFM_SERV_USER* 89
 - SAP_BC_BMT_WFM_UWL_ADMIN* 91
 - SAP_BC_BMT_WFM_UWL_END_USER* 91
 - SAP_BC_SRV_USER* 91
 - Role resolution → Rule resolution
 - Roll-out 80, 158
 - Routing 52
 - RSS feed 646
 - RSWELOGD 540
 - RSWUWFML2 171
 - RSWWWIDE 558
 - RSWWWIDE_DEP 558
 - Rule 413, 816
 - AC00000168* 437
 - AC30000012* 437
 - Container* 419
 - Resolution failure* 421
 - Rule container 277, 278
 - Rule determination 191
 - Rule resolution 414, 420, 544, 879, 881
 - Runtime buffer 194
 - Runtime container 899
 - Runtime table 558
 - RVNSWE01 454
- ## S
-
- SAP Application 647
 - SAP Business Explorer 212
 - SAP Business Suite 43, 654, 669, 673
 - SAP Business Workflow 39, 42, 45, 53, 74, 645, 653, 660, 661, 839

- SAP Case Management 803
- SAP Content Server 836
- SAP CRM 735
 - Action 746, 775
 - Action condition 786
 - Alert inbox 749
 - Assignment block 741, 772, 789
 - BOL 754
 - BOL entity 742, 773, 778
 - BOL framework 742
 - BOL model 768
 - BOL object 743
 - BOL root object 755
 - BTOOrder 770
 - Business document 736
 - Business object type 740
 - Business transaction 739
 - Campaign automation 761
 - Change history 741
 - Claims and funds management 736
 - Component controller 771
 - Component Workbench 743
 - CRM inbox 749, 781, 788
 - CRM_ORDER_MAINTAIN 738
 - CRM_ORDER_READ 738, 747
 - CRM_ORDER_SAVE 738
 - Customer interaction center 736
 - Dynamic mapping 751
 - Dynamic navigation 751, 769, 779
 - Enhancement set 744
 - Event callback 738
 - Event handler table 738
 - Follow-up document 742
 - GenIL 742, 751, 768
 - GenIL component 742, 768, 770, 790
 - GenIL framework 736
 - GenIL implementation class 769, 770
 - Inbound plug 773
 - Lead 762
 - Mapping class 752
 - Marketing 735
 - Middleware 736
 - Object mapping class 780
 - One Order 736, 737, 738
 - Opportunity 762
 - Outbound plug 773
- SAP CRM (cont.)
 - Partner channel management 756
 - Partner determination 763
 - Partner function 749
 - Partner function category 749
 - Partner processing 747
 - Runtime repository editor 772
 - SAP ERP integration 757
 - Standard workflow 760, 761, 762
 - Target ID 779
 - Task-specific navigation 755
 - Transaction history 741, 763, 789
 - Transaction Launcher 757, 758
 - UI component 751, 752, 767, 771, 779
 - UI framework 742, 754, 764, 771
 - UI layer 743
 - UI object 755
 - UI object type 752, 779, 780
 - Web Client UI 737, 753, 758
 - Workflow 756
 - Workflow Inbox 97, 749, 751, 753, 759
 - Worklist 741, 749
 - XML Runtime Repository 771
- SAP ERP HCM
 - Process 795, 796, 799
 - Process and Form 793, 801, 802, 803
- SAP GUI 563, 592, 626, 642, 805
- SAP Interactive Forms by Adobe 568, 569, 637, 793, 802
- SAP Knowledge Provider 642
- SAP NetWeaver 43, 357, 561, 670, 671
- SAP NetWeaver Application Server 595, 884
- SAP NetWeaver Business Client 97, 128, 569
- SAP NetWeaver Business Process Management 39, 42, 43, 511
- SAP NetWeaver Business Warehouse 54, 211, 213, 804
 - BI content 214
- SAP NetWeaver Composition Environment 522
- SAP NetWeaver Developer Studio → NWDS
- SAP NetWeaver Development Infrastructure → NWDI
- SAP NetWeaver Portal 53, 97, 114, 610, 651, 745, 793, 796, 806, 820
 - Business package 804, 815

Index

- SAP NetWeaver Portal (cont.)
 - iView* 813
- SAP NetWeaver Process Integration 215, 502, 513, 522, 804
- SAP Records Management 814
- SAP Smart Forms 747
- SAP SRM 545, 558, 679
 - Add Approver* 718
 - Ad-hoc process level* 698
 - Application-controlled workflow* 713, 724, 729, 731
 - Approval depth* 683
 - Approval framework* 686, 688
 - Approval process overview* 681, 685, 701
 - Approval with completion* 690, 695, 697
 - Approver* 684
 - Combined completion and approval* 683
 - Complex approval process* 682
 - Decision set* 699
 - Dynamic process* 719
 - Inbox* 97
 - Line item-based BAdI* 713
 - n-step* 711, 719
 - Offline functionality* 727
 - One step approval* 710
 - Process level* 690, 695
 - Process schema* 690, 693, 698, 705, 708
 - Process-controlled workflow* 704, 730
 - Requester* 684
 - Responsibility determination strategy* 692
 - Responsible resolver name* 696
 - Rule-based reviewer* 704
 - Shared responsibility* 687
 - Shopping cart* 701, 714, 723
 - Single responsibility* 687
 - Specialist* 684
 - Split responsibility* 687, 722
 - Two-step approval* 710
- SAP Supplier Relationship Management →
 - SAP SRM
- SAP_ALL 88
- SAP_BC_BMT_WFM_ADMIN 91
- SAP_BC_BMT_WFM_CONTROLLER 91
- SAP_BC_BMT_WFM_DEVELOPER 91
- SAP_BC_BMT_WFM_PROCESS 91
- SAP_BC_BMT_WFM_SERV_USER 89
- SAP_BC_BMT_WFM_UWL_ADMIN 91
- SAP_BC_BMT_WFM_UWL_END_USER 91
- SAP_BC_SRV_USER 91
- SAP_PA 805
- SAP_PD 805
- SAP_PT 805
- SAP_WAPI_CREATE_EVENT 485, 490
- SAP_WAPI_DECISION_READ 579, 588
- SAP_WAPI_GET_HEADER 572, 583, 584
- SAP_WAPI_LAUNCH_URL_GET 626
- SAP_WAPI_READ_CONTAINER 480, 487, 572, 583, 584, 595, 598
- SAP_WAPI_RESERVE_WORK_ITEM 572, 583, 584
- SAP_WAPI_START_WORKFLOW 485, 488, 512, 595, 598
- SAP_WAPI_WORKITEM_COMPLETE 573, 583, 595, 600, 632
- SAP_WAPI_WORKITEM_RECIPIENTS 630
- SAP_WAPI_WORKITEMS_TO_OBJECT 487
- SAPconnect 728
- SAPoffice 848, 871
- SAPoffice Notification Connector → SONiC
- SAPscript 386
- SAPWebDynproLauncher 608
- SBOOK 299
- Scanning 839
- Scenario
 - Offline* 803
- Secondary method 855, 856
- Secondary priority 423
- Security 189
 - Role* 135, 150
- Segregation of duties 60, 61
- Selected agent 134, 140, 413, 682
- SELFITEM 345
- Send mail 234, 276, 864
- Send mail step 847
- Service
 - ID* 622
 - Interface* 520, 526
 - Registry* 518
 - WSHANDLER* 625, 626, 628, 630
- Service-Oriented Architecture → SOA
- Shopping cart 907
- Sidebar 648, 673

- Simple form 638
- Simplified workflow definition 846
- Simulate event 552, 870
- Simulate rule resolution 417
- SLD 523, 902
- SO_DOCUMENT_READ_API1 500
- SO_NEW_DOCUMENT_ATT_SEND_API1 500
- SO_NEW_DOCUMENT_SEND_API1 500
- SO_OLD_DOCUMENT_SEND_API1 500
- SOA 502, 503, 507
 - Middleware* 502, 509
- SOAP 505, 526
- Software component 904
- SONiC 121
 - Connector* 112, 728
- Source system 213
- Standard BOR object 753
- Standard report 162
- Standard task 93, 174, 234, 236, 241, 700, 760, 766
- Start
 - Form* 641
 - Latest* 729
 - Requested* 481
 - Task* 551
- Start condition 458, 462, 541, 716, 745, 746, 879, 893
 - Workflow* 463
- Static attribute 369
- Status
 - Canceled* 482
 - Ready* 481
- Status management 394, 451, 746
- Status notification 648
- Step 222
 - Anchor* 865
 - Background* 536
 - Outcome* 256
 - Send mail* 847
 - Type* 223
 - Workflow log* 258
- Step condition 263
 - Complete work item* 263, 265
 - Completion execution* 263, 266
 - Create work item* 263, 265
- Step condition (cont.)
 - Evaluate preconditions and postconditions* 266
- Storage
 - Document* 838
- Structural authorization 91
 - Profile* 90
- Structure
 - WFSYST* 68
- Substitute 155, 168, 188, 726, 756
 - Active* 156
 - HR-based* 155
 - Passive* 156
 - Task classification* 155
 - User-based* 156
- Substitution 111, 113, 668
- Subtype 295
- Subworkflow 273, 287, 476, 479, 857
- Supertype 295, 300, 318
- Support package 559
- Support tool
 - Finding erroneous workflow* 172
 - Restarting a workflow* 195
 - Rule monitoring* 195
- SWC_CALL_METHOD 344, 405
- SWC_CREATE_OBJECT 332, 336
- SWC_GET_ELEMENT 340
- SWC_GET_PROPERTY 331, 333, 339, 405
- SWC_GET_TABLE 340
- SWC_GET_TABLE_PROPERTY 331
- SWC_OBJECT 332, 336
- SWC_SET_ELEMENT 332, 343, 344
- SWC_SET_OBJECTKEY 344
- SWC_SET_TABLE 334, 343
- SWE_EVENT_CREATE 869, 899
- SWE_EVENT_CREATE_FOR_UPD_TASK 492, 495
- SWE_EVENT_CREATE_IN_UPD_TASK 491
- SWE_EVENT_MAIL 871
- SWHACTOR 421
- SWU_OBUF 158
- SWU3 83, 88
- SWW_CONT 480, 560, 561
- SWW_CONTOB 480, 560, 561
- SWWCOND 265
- SWWDHEX 165, 885

SWWERRE 562, 873
 Synchronize 492
 Synchronous 303, 320
 Dialog chain 105, 230, 259, 471, 489
 Method 237, 322, 324, 339
 Syntax check 229
 System
 Workflow_Local 206
 System administrator 475
 System crash
 Recovery 195
 System Landscape Directory → SLD

T

Table

SWW_CONT 480, 560
 SWW_CONTOB 480, 560
 Table-driven dynamic parallel processing 267
 Task 56, 58, 64, 65, 90, 222
 Asynchronous 241
 Automate 662
 Background 544
 Container 222, 238, 246, 277, 278
 Create 235, 238
 Description 100, 237, 251
 Group 150, 241, 807, 816
 Hide 162
 ID 174
 Prefix number 90
 Start 551
 TS00008267 234, 578, 581
 Update 875
 Visualization 627, 811
 Task-specific customizing 91
 Teamworking 225, 261
 Technical workflow log 224, 537, 880
 Terminating event 180, 242, 243, 457, 459,
 464, 466, 467, 492, 633, 745
 Terminology 55
 Test 170, 550
 Authorization 553
 Business object 550
 Condition 552
 Environment 539
 Error scenario 553

Test (cont.)

Event linkage 552
 Method 550
 Triggering event 553
 Utility 550
 Text
 Variable 247
 Third-party software 837
 Time characteristics 217
 Time data 795
 Tracking
 Form 801
 Training 69, 158, 168
 Transaction 292
 BF01 456
 BF24 456
 BF34 456
 BP 446, 461, 624
 BSP_WD_CMPWB 743, 771
 BSVW 452, 746
 BSVX 452
 GENIL_MODEL_BROWSER 770
 HRASR_DT 804, 806, 807
 HRASR_TEST_PROCESS 820
 MM02 450
 NACE 453
 PFAC 415, 417, 425, 550, 552
 PFCG 713
 PFOM 427, 428
 PFOS 428
 PFTC 234, 366
 PFTC_COP 578
 RSA1 217
 RSOR 217
 RSWWCOND 888
 SARA 478
 SBWP 461, 563, 625, 626, 636, 728
 SCASE 809, 820
 SCPR20 705
 SE16 623
 SE24 363, 394, 408, 447, 531, 922
 SE37 924
 SE38 329
 SE63 260
 SE80 329, 513, 516, 590, 617
 SFP_ZCI_UPDATE 806

Transaction (cont.)

SLG1 707, 923
SM37 206
SM58 165, 206, 541
SO16 848
SO28 841
SOAMANAGER 516, 518, 533
SPRO 84
SPROXY 528
ST22 206, 921
SU01 177, 713, 852
SWB_COND 463, 714, 893, 894
SWDD 221, 227, 366, 460, 767, 827
SWDD_CONFIG 833
SWDM 225, 809
SWE2 444, 460, 473, 705, 706, 738, 892
SWEC 449, 451
SWEHR1 454
SWEHR2 454
SWEHR3 454
SWEINST 467
SWEL 446, 451, 461, 496, 715, 924
SWELS 446, 496, 924
SWEQADM 469, 488, 494, 892
SWEQBROWSER 470
SWETYPV 444, 469, 494, 871
SWF_ADM_SUSPEND 890
SWF_ADM_SWWWIDH 890
SWFVISU 118, 119, 569, 571, 573, 590,
 591, 627, 759, 793, 808, 811, 813, 817,
 821
SWFVMD1 569
SWI1 173, 174, 478, 867, 876, 878
SWI1_RULE 191
SWI14 163
SWI2_ADM1 192, 422
SWI2_DEAD 165, 166, 887
SWI2_DIAG 164, 172, 205, 206, 887
SWI2_DURA 165
SWI2_FREQ 165
SWI5 166
SWI6 163, 724, 921
SWIA 136, 192
SWN_SELEN 127
SWNADMIN 128
SWNCONFIG 127

Transaction (cont.)

SWO_ASYNC 563
SWO1 297, 299, 311, 312, 444, 447, 490,
 550, 563, 765, 874
SWO3 298
SWPA 184
SWPC 195, 890
SWPR 176, 195, 483, 887, 890
SWU_EWBTE 455
SWU_OBUF 158, 193, 194, 427, 880
SWU0 460, 714, 893
SWU2 895
SWU3 83, 88, 195, 831, 885
SWU9 545, 547
SWUD 538, 541, 542, 545, 551, 830, 867
SWUE 714, 889, 892, 893, 894, 924
SWUI 472
SWUI_DEMO 103
SWUI_VERIFY 87
SWUS 193, 375, 472, 544, 626, 636, 894
SWUY 474
SWWA 195
SWWB 234
SXMB_IFR 522
WF_EXTSRV 622, 625
WF_HANDCUST 621
WSADMIN 516
 Transactional RFC → tRFC
 Transfer
 Form 800
 Process 801
 Translation 260
 Transport 84, 92, 153, 194, 229, 260
 Log 889
 New workflow version 260
 Setting 93
 Travel approval 653
 tRFC 165, 895
 Triggering event 268, 458, 459, 828, 867,
 868
 Troubleshooting 537, 867
 TS → Standard task
 TS00008267 578, 581
 Type
 Sub 295
 Super 295, 300, 318

U

UDDI 504
 UN/CEFACT 506
 Union 66
 Unit
 Organizational 144, 146
 United Nations Centre for Trade Facilitation
 and Electronic Business → UN/CEFACT
 Universal Description, Discovery, and Integra-
 tion → UDDI
 Universal Worklist → UWL
 Update rule 215
 Update task 875
 Upgrade 82, 555, 730
 Archiving 556
 Planning 555
 User 84
 Decision step 602
 Event 801
 Exit 673, 675
 Relocation 154
 Substitute 155
 Type 96
 WF_INITIATOR 278
 WF-BATCH 88, 89, 94, 165, 206, 258, 478,
 852
 User decision 227, 228, 231, 232, 233, 234,
 575, 580, 582, 671, 672, 675, 865
 TS00008267 854
 UserDecisionMemo 579
 With note 577, 581
 User ID 145, 147
 WF-BATCH 133
 UWL 96, 111, 114, 120, 127, 130, 482, 489,
 569, 575, 576, 577, 580, 590, 592, 593,
 594, 598, 600, 603, 608, 610, 625, 626,
 627, 630, 631, 632, 656, 659, 670, 676,
 726, 728, 759, 793, 798, 799, 801, 806,
 808, 809, 811, 813, 817, 820, 821, 822,
 823, 922
 Action handler 117, 608
 Add memo 577
 Administrator 114
 API 111, 120
 Cache administration 579

UWL (cont.)
 Configuration wizard 118, 119, 573
 Custom column 117, 122
 Customizing 116
 Delta pull mechanism 114
 FunctionModuleActionHandler 118
 HRAdmin 815
 Integration 759
 Mandatory rejection memo 125
 Memo 117
 Personalization 115
 SAPBSPLauncher 118
 SAPWebDynproABAPLauncher 117
 SAPWebDynproLauncher 117
 SONiC connector 120
 Sorting work item 115
 System alias 114
 UpdatingContainerHandler 118
 URLLauncher 118
 UserDecisionHandler 118

V

Validation
 Dynamic 801
 Variable 100
 _Attach_Objects 102
 _WI_OBJECT_ID 280
 Verification workflow 85, 86
 Version
 Active 260
 Number 223
 Overview 261
 Versioning 260
 View
 Pending 649
 Request 648
 Virtual 316
 Attribute 301, 318, 327, 332, 335
 Method 327
 Visual Administrator 803

W

W3C 504
 Wait for event 274, 457, 466

- Wait status 481
- Wait time 166
- WAPI 899
 - SAP_WAPI_CREATE_EVENT* 485, 490, 747
 - SAP_WAPI_DECISION_READ* 579, 588
 - SAP_WAPI_GET_HEADER* 572, 583, 584
 - SAP_WAPI_LAUNCH_URL_GET* 626
 - SAP_WAPI_READ_CONTAINER* 487, 572, 583, 584, 595, 598
 - SAP_WAPI_RESERVE_WORKITEM* 572, 583, 584
 - SAP_WAPI_START_WORKFLOW* 485, 488, 512, 595, 598
 - SAP_WAPI_WORKITEM_COMPLETE* 573, 583, 595, 600, 632
 - SAP_WAPI_WORKITEM_RECIPIENTS* 630
 - SAP_WAPI_WORKITEMS_TO_OBJECT* 487
- Web Dynpro 117, 608, 803, 808, 813, 901, 902
 - Application* 609
 - Form* 638
 - Model* 901
 - Template wizard* 607
- Web Dynpro ABAP 129, 568, 575, 809, 817
 - ASR_FORM_DISPLAY* 809
 - ASR_PD_PROCESS_EXECUTE* 809
 - ASR_PROCESS_EXECUTE* 809, 814
 - ASR_PROCESSES_DISPLAY* 809
 - PAOC_ASR_WD* 809
- Web Dynpro Java 568, 593, 600, 603, 605, 607, 608, 610, 903, 904
- Web service 502, 503, 623, 624, 627
 - Handler* 621
 - Standards* 506
 - Task* 625
 - Work item* 625
- Web Service Navigator 517
- Web Services Description Language → WSDL
- WebDynproApplication 608
- WebDynproDeployableObject 608
- WF_DOC_CREATE_SOFM 642
- WF_INITIATOR 278
- WF_TASK 346
- WF-BATCH 88, 89, 94, 133, 165, 206, 258, 478, 852, 874, 920
- WFSYST 68
- WFSYST-AGENT 853
- WI_ID 572, 812
- WI_OBJECT_ID 278
- Widget framework 646
- Wizard 82, 225, 276, 346, 449, 623, 845
 - Agent selection* 421
 - Approval procedure* 845
 - Circular distribution* 846
 - Form creation* 641
 - Modeling support* 845
- WLC parameter 177
- Work item 56, 72, 77, 98, 177, 206, 208, 215, 266, 294, 315, 476, 479, 481, 498, 536, 537, 554, 614, 618, 626, 629, 635, 654, 672, 798, 814, 815, 839, 855, 873, 880, 881, 883, 886, 887, 888, 891, 922
 - Background* 478
 - Container* 181, 570, 674
 - Container element* 676
 - Creation* 233
 - Creation date* 180
 - Deadline* 174, 478
 - Deleting* 209
 - Delivery* 77, 86, 95
 - Description* 99, 207, 313
 - Dialog* 174, 479, 481, 482
 - Display* 176, 187, 891
 - End date/time* 180
 - Error* 170
 - Executing* 103
 - Execution with agent check* 136
 - Forwarding* 106, 141
 - Frequency* 175
 - Hierarchy* 477
 - History* 215, 476
 - ID* 179, 213, 479, 489, 561, 562, 598, 603, 609, 610, 618, 630, 812
 - Locating* 173
 - Processing* 133
 - Recipient* 134
 - Rejecting processing* 258
 - Rejection* 110
 - Replacing in queue* 102
 - Requested start* 180
 - Rule monitoring* 195
 - Short text* 99

Index

Work item (cont.)

Start date/time 180
Task description 100
Technical view 179
Text 236, 247, 817, 848
Timestamp 180
Type 174, 479
WI_ID 583
Without agent 192, 422

Work item status 162, 481, 483

Committed 163
Completed 163
Ready 163
Selected 163

Work queue 477

Workbook 214

Workers council 66, 80, 184

Workflow 40, 56, 59, 66, 67, 69, 72, 74, 78,

79, 81, 82, 535, 659, 798, 799, 800, 839
Ad-hoc 53, 109, 184
Administration 74, 80, 161, 169, 211, 745
Administrator 136, 168, 171, 175, 198,
483, 535, 922, 925

Agent 184

API 571

Application-controlled 688, 710

Archive 556, 558

Binding 542

Chronicle 183

Complete 488

Container 117, 123, 124, 222, 225, 232,
249, 277, 278, 459, 460, 570, 628, 657,
659, 672, 674, 819, 862, 876

Continue after system crash 195

Decision 672

Design 78, 170

Developer 98, 200

Document-based 840

Engine 475

Error 889

Header event 467

ID 476, 539

Inbox 755

Initiator 68, 73, 228, 437, 624

Instance 56, 164, 182, 476, 545, 557

Item 478

Workflow (cont.)

Macro 330, 897

Object 184

Outbox 110, 225

Prefix number 90

Process-controlled 688, 689, 729

Report 82, 211

Restart 488

Restart After Error 195

Review 287, 719

Role 91

Runtime system 87

Start condition 463, 745

Starting 441, 471, 474

Subworkflow 273, 287, 476, 479

Test plan 557

Toolbox 103, 498

Trace 545, 546, 547, 924

Training 167

Version number 82, 476

WF_VERIFY 828

Wizard 225, 276, 286

WS10001011 762

Workflow Builder 78, 221, 222, 226, 238, 245, 420, 560, 761, 827, 871, 877

Container definition 225

Frame 222

Workflow customizing 745

Automatic 87, 88

Workflow definition 56, 93, 131, 225, 240, 539, 870

Activating 229

Basic data 222

Binding 249

Branch 252

Conditional branch 252

Container 247

Deadline 231

Documenting 259

Form 253

Saving 229

Syntax check 229

Testing 229

Workflow log 98, 103, 108, 176, 182, 292, 471, 473, 537, 872, 880, 891, 925

Agent view 184

- Workflow log (cont.)
 - Chronicle view* 183
 - Filtering* 258
 - Graphical view* 184
 - Object view* 184
 - Technical view* 185
 - Workflow step 222
 - Activity* 252
 - Ad-hoc anchor* 252
 - Advance with dialog* 259
 - Branch* 252
 - Case* 253
 - Condition* 253
 - Conditional branch* 252
 - Container operation* 249, 252
 - Event creation* 252
 - Fork* 252
 - Loop* 253
 - Manual confirmation* 258
 - Outcome* 256
 - Process control* 253
 - Result* 256
 - Send mail* 253
 - Simple form* 253
 - Type* 252
 - User decision* 227, 253
 - Waiting for event* 254
 - Web activity* 254
 - Workflow step (cont.)
 - Wf-XML* 254
 - Workflow template 93, 174, 234, 236, 238, 803, 804, 807, 810, 821
 - WS10000031* 725
 - WS10000060* 710, 724
 - WS10000129* 710
 - WS14500015* 723
 - WS17900260* 818
 - Workflow_Local 206, 542
 - Working time
 - Change* 801
 - Workload analysis 155, 166
 - World Wide Web Consortium → W3C
 - WS → Workflow template
 - WS-BPEL 507
 - WSDL 503
 - WSHANDLER 625, 626, 628, 630
 - WS-Policy 507
 - WS-ReliableMessaging 507
 - WS-Security 506
- X**
-
- XI → SAP NetWeaver Process Integration
 - XML 123, 479, 480, 526, 547, 577, 593, 608, 609, 627, 657, 796, 806, 809, 813, 814, 821, 923
 - Persistence* 560