

Oracle® Workflow

開発者ガイド

リリース 2.6.3

部品番号 : B12364-01

2004 年 3 月

Oracle Workflow 開発者ガイド, リリース 2.6.3

部品番号 : B12364-01

原本名 : Oracle Workflow Developer's Guide, Release 2.6.3

原本部品番号 : B10284-02

原著者 : Siu Chang, Clara Jaeckel

原協力者 : George Buzsaki, John Cordes, Mark Craig, Mark Fisher, Kevin Hudson, George Kellner, Angela Kung, David Lam, Jin Liu, Kenneth Ma, Steve Mayze, Santhana Natarajan, Tim Roveda, Robin Seiden, Sachin Sharma, Sheryl Sheh, Susan Stratton

Copyright © 2003 Oracle Corporation. All rights reserved.

制限付権利の説明

このプログラム（ソフトウェアおよびドキュメントを含む）には、オラクル社およびその関連会社に所有権のある情報が含まれています。このプログラムの使用または開示は、オラクル社およびその関連会社との契約に記された制約条件に従うものとします。著作権、特許権およびその他の知的財産権と工業所有権に関する法律により保護されています。

独立して作成された他のソフトウェアとの互換性を得るために必要な場合、もしくは法律によって規定される場合を除き、このプログラムのリバース・エンジニアリング、逆アセンブル、逆コンパイル等は禁止されています。

このドキュメントの情報は、予告なしに変更される場合があります。オラクル社およびその関連会社は、このドキュメントに誤りが無いことの保証は致し兼ねます。これらのプログラムのライセンス契約で許諾されている場合を除き、プログラムを形式、手段（電子的または機械的）、目的に関係なく、複製または転用することはできません。

このプログラムが米国政府機関、もしくは米国政府機関に代わってこのプログラムをライセンスまたは使用する者に提供される場合は、次の注意が適用されます。

U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation, and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

このプログラムは、核、航空産業、大量輸送、医療あるいはその他の危険が伴うアプリケーションへの用途を目的としておりません。このプログラムをかかるとして使用する際、上述のアプリケーションを安全に使用するために、適切な安全装置、バックアップ、冗長性 (redundancy)、その他の対策を講じることは使用者の責任となります。万一かかるプログラムの使用に起因して損害が発生いたしましても、オラクル社およびその関連会社は一切責任を負いかねます。

Oracle は Oracle Corporation およびその関連会社の登録商標です。その他の名称は、Oracle Corporation または各社が所有する商標または登録商標です。

目次

はじめに	ix
対象読者	x
このマニュアルの構成	x
その他の情報	xi
Oracle Applications データの変更を目的としたデータベース・ツール使用の禁止	xvi
オラクル社について	xvii
1 Oracle Workflow の概要	
Oracle Workflow の概要（開発者向け）	1-2
主な機能と定義	1-3
ワークフロー・プロセス	1-5
2 ワークフロー・プロセス定義	
Oracle Workflow Builder の概要	2-2
ナビゲータ・ツリーの構造	2-4
ナビゲータ・ツリーの表示	2-5
Oracle Workflow Builder によるプロセス定義の作成	2-8
項目タイプのオープンと保存	2-13
クイック・スタート・ウィザードの概要	2-19
各種リリースのサーバーでの Oracle Workflow Builder の使用	2-22
「項目タイプの定義」 Web ページ	2-25
3 ワークフロー・プロセスのコンポーネントの定義	
ワークフロー・プロセスのコンポーネント	3-2

項目タイプ	3-2
オブジェクトへのアクセス許可の設定	3-18
選択肢タイプ	3-20
メッセージ	3-24
アクティビティ	3-53
投票アクティビティ	3-73
Oracle Workflow Builder のオブジェクトの削除	3-79
Oracle Workflow Builder のオブジェクトの変更	3-80
バージョン管理をサポートするワークフロー・オブジェクト	3-81
バージョン管理をサポートしないワークフロー・オブジェクト	3-82

4 ワークフロー・プロセス・ダイアグラムの定義

「プロセス」ウィンドウ	4-2
Oracle Workflow Builder のフォントの変更	4-24
ワークフロー・プロセスのショートカット・アイコンの作成	4-25
ルール	4-26

5 Oracle Workflow の事前に定義されたアクティビティ

標準アクティビティ	5-2
And/Or アクティビティ	5-3
「比較」アクティビティ	5-3
「実行時間の比較」アクティビティ	5-4
「待機」アクティビティ	5-4
「ブロック」アクティビティ	5-6
「スレッドの遅延」アクティビティ	5-6
「プロセスの開始」アクティビティ	5-6
「Noop」アクティビティ	5-7
「ループ・カウンタ」アクティビティ	5-7
「開始」アクティビティ	5-9
「終了」アクティビティ	5-9
「ロール解決」アクティビティ	5-9
「通知」アクティビティ	5-10
「はい / いいえ投票」アクティビティ	5-11
「マスター / デテール連携」アクティビティ	5-12

「フロー待ち」アクティビティ	5-13
「継続フロー」アクティビティ	5-13
「割当」アクティビティ	5-15
「モニター URL」アクティビティ	5-15
「イベント・プロパティの取得」アクティビティ	5-16
「イベント・プロパティの設定」アクティビティ	5-17
「イベント・プロパティの比較」アクティビティ	5-18
「XML タグ値の取得」アクティビティ	5-19
「XML タグ値の比較」アクティビティ	5-20
「XML 変換」アクティビティ	5-22
コンカレント・マネージャの標準アクティビティ	5-23
「コンカレント・プログラムの実行」アクティビティ	5-23
「コンカレント・プログラムの発行」アクティビティ	5-24
「コンカレント・プログラム待ち」アクティビティ	5-25

6 Oracle Workflow のプロシージャおよび関数の定義

Oracle Workflow のプロシージャおよび関数の定義	6-2
関数アクティビティがコールする PL/SQL プロシージャの標準 API	6-3
関数アクティビティがコールする Java プロシージャの標準 API	6-8
項目タイプのセレクト関数またはコールバック関数の標準 API	6-12
PL/SQL 文書の標準 API	6-15
PL/SQL 文書	6-15
PL/SQL CLOB 文書	6-17
PL/SQL BLOB 文書	6-22
イベント・データ・ジェネレート関数の標準 API	6-24
キュー・ハンドラの標準 API	6-26
エンキュー	6-27
デキュー	6-28
イベント・サブスクリプションのルール関数の標準 API	6-29

7 ワークフロー定義のテスト

ワークフロー定義のテスト	7-2
開発者スタジオによるワークフロー定義のテスト	7-6

8 ビジネス・イベントの管理

ビジネス・イベントの管理	8-2
イベント・マネージャ	8-4
イベント	8-5
システム	8-19
エージェント	8-25
イベント・サブスクリプション	8-41
イベントの呼出し	8-60
システムのサインアップ	8-62
システムの同期	8-65
ローカル・キューの確認	8-68
Oracle Applications のためのイベント・マネージャ	8-72
イベント	8-73
イベント・サブスクリプション	8-86
エージェント	8-105
システム	8-125
ワークフロー・エージェントの Ping/ 確認	8-137
「ワークフロー・エージェントの Ping/ 確認」項目タイプ	8-139
「マスター Ping」プロセスの概要	8-140
「マスター Ping」プロセス・アクティビティ	8-141
「詳細 Ping」プロセスの概要	8-142
「詳細 Ping」プロセス・アクティビティ	8-143

9 事前定義済ワークフロー・イベント

事前定義済ワークフロー・イベント	9-2
イベント定義イベント	9-3
イベント・グループ定義イベント	9-4
システム定義イベント	9-5
エージェント定義イベント	9-6
エージェント・グループ定義イベント	9-7
イベント・サブスクリプション定義イベント	9-8
イベント・システムの同期イベント	9-9
シード・イベント・グループ	9-10
エージェントの Ping イベント	9-12

システムのサインアップ・イベント	9-14
Any イベント	9-15
Unexpected イベント	9-18
User Entry Has Changed イベント	9-20
通知イベント	9-23
通知メーラー・イベント	9-31
ビジネス・イベント・システム制御イベント	9-32
汎用サービス・コンポーネント・フレームワーク制御イベント	9-35
ワークフロー送信プロトコル	9-38
「ワークフロー送信プロトコル」項目タイプ	9-39
ワークフロー・イベント・プロトコル・プロセスの概要	9-40
ワークフロー・イベント・プロトコル・プロセスのアクティビティ	9-42
ワークフロー送信プロトコル・イベント	9-45

10 デモンストレーション用ワークフロー・プロセス

サンプル・ワークフロー・プロセス	10-2
サンプル・ワークフローのプロセス・ダイアグラムの表示	10-3
購買申請プロセス	10-5
購買申請データ・モデルのインストール	10-6
購買申請ワークフローの開始	10-9
「購買申請」項目タイプ	10-13
購買承認申請プロセスの概要	10-14
購買申請プロセス・アクティビティ	10-15
「承認者に通知」サブプロセスの概要	10-21
「承認者に通知」サブプロセスのアクティビティ	10-22
StartProcess 関数の例	10-25
関数アクティビティの例	10-28
例：承認者の選択	10-28
例：承認権限の検証	10-31
通知アクティビティの例	10-33
例：購買申請承認が必要なことを通知	10-33
ドキュメント・レビュー・プロセス	10-36
「文書管理」項目タイプ	10-37
ドキュメント・レビュー・プロセスの概要	10-38
ドキュメント・レビュー・プロセスのアクティビティ	10-39

エラー・チェック・プロセス	10-42
「定期アラート」項目タイプ	10-43
エラー・チェック・プロセスの概要	10-44
エラー・チェック・プロセスのアクティビティ	10-45
ユーザー定義の警告処理プロセスの概要	10-50
ユーザー定義の警告処理プロセスのアクティビティ	10-51
イベント・システム・デモンストレーション	10-52
イベント・システム・デモンストレーション・データ・モデルのインストール	10-54
イベント・システム・デモンストレーション・ワークフローの開始	10-55
「イベント・システムのデモ」項目タイプ	10-60
バイヤー：最上位の発注プロセスの概要	10-62
バイヤー：最上位の発注プロセスのアクティビティ	10-64
バイヤー：サプライヤへの発注送信サブプロセスの概要	10-67
バイヤー：サプライヤへの発注送信サブプロセスのアクティビティ	10-68
バイヤー：サプライヤの発注承認の受信サブプロセスの概要	10-69
バイヤー：サプライヤの発注承認の受信サブプロセスのアクティビティ	10-71
バイヤー：アドバンスト出荷通知サブプロセスの概要	10-73
バイヤー：アドバンスト出荷通知サブプロセスのアクティビティ	10-74
バイヤー：サプライヤの請求の受信サブプロセスの概要	10-75
バイヤー：サプライヤの請求の受信サブプロセスのアクティビティ	10-76
サプライヤ：最上位のオーダー・プロセスの概要	10-77
サプライヤ：最上位のオーダー・プロセスのアクティビティ	10-78
サプライヤ：オーダー詳細の取得サブプロセスの概要	10-81
サプライヤ：オーダー詳細の取得サブプロセスのアクティビティ	10-82
サプライヤ：クレジット・チェック・サブプロセスの概要	10-85
サプライヤ：クレジット・チェック・サブプロセスのアクティビティ	10-86
サプライヤ：在庫チェック・サブプロセスの概要	10-87
サプライヤ：在庫チェック・サブプロセスのアクティビティ	10-88
サプライヤ：アドバンスト出荷通知サブプロセスの概要	10-89
サプライヤ：アドバンスト出荷通知サブプロセスのアクティビティ	10-90
サプライヤ：サプライヤの請求書の送信サブプロセスの概要	10-92
サプライヤ：サプライヤの請求書の送信サブプロセスのアクティビティ	10-93
発注イベント	10-94
発注承認イベント	10-97
アドバンスト出荷通知イベント	10-99
請求書イベント	10-100

11 エラー処理

エラー処理	11-2
ワークフロー・プロセスのエラー処理	11-2
イベント・サブスクリプション処理のエラー処理	11-5
「システム:エラー」項目タイプと項目属性	11-7
デフォルト・エラー・プロセス	11-8
「再試行のみ」プロセス	11-12
デフォルト・イベント・エラー・プロセス	11-15

A Oracle Workflow 開発者のナビゲーション・パス

Oracle Workflow 開発者のナビゲーション・パス	A-2
--------------------------------------	-----

B Oracle Workflow Builder のメニューとツールバー

Oracle Workflow Builder のメニュー	B-2
Oracle Workflow Builder のツールバー	B-8

C 他の Oracle 製品への Oracle Workflow の実装

Oracle E-Business Suite 埋込みの事前定義済のワークフロー	C-2
適用されるテクノロジー	C-2
ビジネス・インテリジェンス	C-5
契約	C-5
会計関係のアプリケーション	C-6
HRMS アプリケーション	C-16
賃貸契約	C-20
メンテナンス・アプリケーション	C-22
製造業関連のアプリケーション	C-22
オーダーの履行	C-25
調達	C-26
プロジェクト	C-27
販売、マーケティングおよび電子取引	C-28
サービス	C-30
サプライ・チェーン計画	C-30

Oracle E-Business Suite への Oracle Workflow ビジネス・イベント・システムの実装	C-32
適用されるテクノロジー	C-32
会計関係のアプリケーション	C-33
製造業関連のアプリケーション	C-34
Oracle テクノロジ・スタックへの Oracle Workflow の実装	C-37
Oracle Workflow のサポート・ポリシー	C-39
カスタマイズのガイドライン	C-39
カスタマイズに関する問題点の解決	C-40
サポート対象外のカスタマイズ	C-40
サポート対象のカスタマイズ	C-41

用語集

索引

はじめに

対象読者

このマニュアルは、次の作業上の知識を前提としています。

- ビジネスエリアの基礎と基本操作
- Oracle Workflow

Oracle Applications 製品情報の詳細は、「その他の情報」を参照してください。

また、このマニュアルは、オペレーティング・システムの概念を基本的に理解し、Oracle Database、PL/SQL および Oracle Application Server のテクノロジーに精通している読者を想定しています。

このマニュアルの構成

このマニュアルには、Oracle Workflow を理解して様々な開発を行うために必要な情報が記載されています。

- 第1章では、Oracle Workflow の概要について説明します。
- 第2章では、ワークフロー・プロセスを定義する方法について説明します。
- 第3章では、ワークフロー・プロセスの作成に必要なコンポーネントを定義する方法について説明します。
- 第4章では、ワークフロー・プロセスのダイアグラムを作図および定義する方法について説明します。
- 第5章では、Oracle Workflow に用意されている標準アクティビティについて説明します。
- 第6章では、Oracle Workflow でコールできる PL/SQL ファンクションと Java 関数の標準 API について説明します。
- 第7章では、テストのためにワークフロー・プロセスを起動する方法について説明します。
- 第8章では、ビジネス・イベントの管理方法について説明します。
- 第9章では、Oracle Workflow に用意されている標準イベントについて説明します。
- 第10章では、Oracle Workflow に含まれているデモンストレーション用のワークフロー・プロセスについて説明します。
- 第11章では、ワークフロー・プロセスのエラー処理およびイベント・サブスクリプション処理について説明します。
- 付録 A では、Oracle Applications に対してシードされた Oracle Workflow 職責での、Oracle Workflow 開発者 Web ページへのナビゲーション・パスについて説明します。
- 付録 B では、Oracle Workflow Builder のメニューとツールバーについて説明します。

- 付録 C では、Oracle Applications に組み込まれている Workflow に含まれている事前定義済みのワークフロー・プロセス、ビジネス・イベント・システムを利用した Oracle Applications 機能、および Oracle Workflow を利用した Oracle テクノロジ・スタックの機能について説明します。Oracle Workflow のサポート・ポリシーも記載されています。

このマニュアルの最後には、Oracle Workflow の用語集が記載されています。

その他の情報

オンライン・マニュアルやサポート・サービスなど、複数の情報源から、Oracle Workflow に関する知識と理解を深めることができます。

このマニュアルで他の Oracle Applications マニュアルに言及している場合は、リリース 11i のマニュアルを指しています。

オンライン・マニュアル

Oracle Applications に組み込まれている Workflow を使用している場合は、すべての Oracle Applications マニュアルをオンライン (HTML または PDF) で使用可能です。

- **オンライン・ヘルプ:** HTML ヘルプの新機能の項に、リリース 11i の新機能に関する説明があります。この情報は、Oracle Workflow のリリースごとに更新されます。新機能の項には、このマニュアルの印刷時には使用可能になっていなかった機能に関する情報も記載されています。たとえば、管理者がミニ・パックまたはアップグレードからソフトウェアをインストールすると、このヘルプに新機能の説明が記載されるようになります。
- **11i Features Matrix:** このドキュメントには、パッチによって使用可能になる新機能と、新機能に関連する新しいドキュメントの一覧が記載されています。
- **README ファイル:** ダウンロード可能な新しいドキュメントまたはドキュメント・パッチについては、インストールしたパッチの README ファイルを参照してください。

このマニュアルの一部は、Windows ヘルプ形式のオンライン・ヘルプとしても使用可能です。Windows ヘルプには、Oracle Workflow Builder の「ヘルプ」メニューからアクセスできます。

関連するユーザーズ・ガイド

Oracle Workflow は、他の Oracle Applications 製品で使用されて、埋込みのワークフローを提供します。Oracle Applications に組み込まれている Workflow を使用している場合は、Oracle Workflow を設定および使用しているときに他のユーザーズ・ガイドを参照すれば、埋込みのワークフローの詳細を理解することができます。

オンライン・マニュアルを参照するには、HTML ヘルプ・ウィンドウの拡張可能メニューから「Library」を選択する方法、メディア・パックに同梱されている Oracle Applications ド

キュメント・ライブラリ CD から参照する方法、またはシステム管理者から提供される URL を使用して Web ブラウザから参照する方法があります。

全製品に関連するガイド

『Oracle Applications ユーザーズ・ガイド』

このマニュアルには、このリリースの Oracle Workflow で利用可能な Graphical User Interface (GUI) を使用して、データの入力、問合せ、レポートの実行およびナビゲートを行う方法について記載されています。また、ユーザー・プロファイルの設定や、レポートと同時プロセスの実行および確認についても記載されています。

このユーザーズ・ガイドには、Oracle Applications ヘルプ・ファイルから「Getting Started with Oracle Applications」を選択してオンラインでアクセスすることもできます。

この製品に関連するマニュアル

『Oracle Workflow 管理者ガイド』

このマニュアルには、ワークフロー対応プロセスも含め、Oracle Applications 製品に必要な設定手順を完了する方法と、実行時のワークフロー・プロセスの進行状況を監視する方法について記載されています。

『Oracle Workflow ユーザーズ・ガイド』

このマニュアルには、Oracle Applications のユーザーがワークフローの通知を表示して応答する方法と、ワークフロー・プロセスの進行状況を監視する方法について記載されています。

『Oracle Workflow API リファレンス』

このマニュアルには、Oracle Workflow にアクセスする開発者と管理者に提供される API について記載されています。

『Oracle General Ledger ユーザーズ・ガイド』

このマニュアルには、仕訳入力、予算処理、複数会社会計および連結に関する情報が記載されています。

『Oracle Purchasing ユーザーズ・ガイド』

このマニュアルには、発注と購買申請を入力して管理する方法が記載されています。

『Oracle Self-Service Human Resources (SSHR) インプリメンテーション・ガイド』

このマニュアルには、管理者と従業員を対象として、セルフサービス人事管理機能を設定する方法が記載されています。管理者と従業員は、イントラネットと Web ブラウザを使用して、個人情報およびキャリア管理機能に簡単にアクセスできます。

『Oracle Payables ユーザーズ・ガイド』

このマニュアルには、仕入先、請求書および支払を入力して管理する方法が記載されています。

『Oracle Projects ユーザーズ・ガイド』

このマニュアルには、プロジェクト、予算、費用、原価および請求の入力および管理方法が記載されています。

『Oracle Receivables ユーザーズ・ガイド』

このマニュアルには、顧客、入金、回収および取引の入力および管理方法が記載されています。

『Oracle Business Intelligence System インプリメンテーション・ガイド』

このマニュアルには、環境に Oracle Business Intelligence (BIS) をインプリメントする方法が記載されています。

『BIS 11i User Guide Online Help』

このマニュアルは、BIS アプリケーション専用のオンライン・ヘルプとして提供されるもので、インテリジェンス・レポート、Discoverer ワークブックおよび Performance Management Framework に関する情報が含まれています。

『Oracle Financials Open Interface Reference』

このマニュアルは、Oracle Financial Applications のすべてのユーザーズ・ガイドに記載されているオープン・インタフェースの説明を集約したものです。

『Oracle XML Gateway ユーザーズ・ガイド』

このマニュアルには、適切に構成された有効な XML メッセージを Oracle Applications と取引パートナー間で作成および取り込む方法について記載されています。

インストールとシステム管理

『Oracle Applications 概要』

このマニュアルでは、Oracle Applications リリース 11i の概念、機能、テクノロジー・スタック、アーキテクチャおよび用語が紹介されています。Oracle Applications のインストール前に参照すると役立つ入門書です。また、Business Intelligence (BIS)、言語セットとキャラクタ・セット、Self-Service Web Applications など、Oracle Applications 全体の機能の基盤となる概念も紹介しています。

『Oracle Applications のインストール』

このマニュアルには、Oracle Applications 製品のインストレーションを管理する手順が記載されています。リリース 11i の場合、大部分のインストール処理は Oracle Rapid Install を使

用して処理されます。これにより多数の必須手順が自動化され、Oracle Applications および Oracle Database のテクノロジ・スタックを最短時間でインストールできます。このマニュアルには、Oracle Rapid Install の使用方法と、インストール完了までに必要なタスクの一覧が記載されています。このマニュアルは、個々の製品のユーザーズ・ガイドおよび実装ガイドと併用する必要があります。

『Oracle Applications のアップグレード』

Oracle Applications リリース 10.7 またはリリース 11.0 製品からリリース 11i にアップグレードする場合は、このマニュアルを参照してください。このマニュアルには、アップグレード・プロセスの説明と、データベース固有のアップグレード・タスクおよび製品固有のアップグレード・タスクの一覧が記載されています。リリース 11i には、リリース 10.7 (NCA、SmartClient またはキャラクタ・モード) またはリリース 11.0 からアップグレードする必要があります。リリース 10.7 以前のリリースからリリース 11i に直接アップグレードすることはできません。

Oracle Applications の保守に関するドキュメント・セット

このドキュメント・セットを参考にして、AutoUpgrade、AutoPatch、AD 管理、AD コントローラ、AD 再リンク、ライセンス・マネージャなどの各種 AD ユーティリティを実行します。このマニュアルには、操作手順、スクリーンショットおよび AD ユーティリティの実行に必要な他の情報が含まれています。また、Oracle Applications のファイル・システムおよびデータベースの管理についても記載されています。

『Oracle Applications システム管理者ガイド』

このマニュアルには、Oracle Applications のシステム管理者向けの計画およびリファレンス情報が記載されています。セキュリティの定義方法、メニューとオンライン・ヘルプのカスタマイズ方法および同時処理の管理方法が含まれています。

『Oracle Alert ユーザーズ・ガイド』

このマニュアルには、Oracle Applications データの状態を監視するために、定期警告とイベント警告を定義する方法について記載されています。

『Oracle Applications 開発者ガイド』

このマニュアルには、Oracle Applications 開発スタッフが従う必要のあるコーディングの標準が含まれています。『Oracle Applications User Interface Standards for Forms-Based Products』に記載されている Oracle Applications ユーザー・インタフェースの実装に必要な、Oracle Applications Object Library コンポーネントについて説明しています。また、カスタムの Oracle Forms Developer 6i フォームを作成して Oracle Applications に統合する場合に役立つ情報も記載されています。

その他の実装のマニュアル

『Oracle Applications Product Update Notes』

このマニュアルは、Oracle Applications のアップグレードに関するリファレンスです。リリース 11.0 からリリース 11i への個々の Oracle Applications 製品の変更履歴が記載されています。また、この 2 つのリリース間でデータベース・オブジェクト、プロファイル・オブションおよびシード・データに加えられた新機能、機能拡張および変更も含まれています。

『Oracle Applications における複数報告通貨』

Multiple Reporting Currencies 機能を使用して複数の通貨で取引を記録する場合は、Oracle Workflow を実装する前にこのマニュアルを参照してください。このマニュアルには、Oracle Workflow にこの機能を実装するための追加手順と設定に関する注意事項が詳しく記載されています。

『Oracle Applications における複数組織』

このマニュアルには、Oracle Workflow の単一インストールを実行するときに複数の組織構造を定義およびサポートできるように、Oracle Applications の Multiple Organization サポート機能を Oracle Workflow に設定して使用方法が記載されています。

『Oracle Applications フレックスフィールド・ガイド』

このマニュアルには、Oracle Workflow 導入チームおよび Oracle Applications 製品データの進行中のメンテナンスの担当者を対象として、フレックスフィールドの計画作成、設定および参照情報が記載されています。また、フレックスフィールド・データのカスタム・レポートを作成する方法も記載されています。

『Oracle eTechnical Reference Manuals』

各『eTechnical Reference Manual (eTRM)』には、特定の Oracle Applications 製品のデータベース・ダイアグラムと、データベース表、フォーム、レポートおよびプログラムに関する詳細な説明が記載されています。この情報を参考にして、既存アプリケーションからのデータ変換、Oracle Applications データと Oracle 以外のアプリケーションの統合および Oracle Applications 向けカスタム・レポートの記述を行うことができます。

『Oracle Applications フォーム・ベース製品のユーザー・インタフェース標準』

このマニュアルには、Oracle Applications 開発スタッフが従う必要のあるユーザー・インタフェース (UI) の標準が含まれています。Oracle Applications 製品の UI と、この UI を Oracle Forms で作成するアプリケーションの設計に適用する方法を説明しています。

『Oracle Manufacturing APIs and Open Interfaces Manual』

このマニュアルには、他の Oracle Manufacturing アプリケーションおよび他のシステムとの統合に関する最新情報が記載されています。また、Oracle Manufacturing の API とオープン・インタフェースが記載されています。

『Oracle Order Management Suite APIs およびオープン・インタフェース・マニュアル』

このマニュアルには、他の Oracle Manufacturing アプリケーションおよび他のシステムとの統合に関する最新情報が記載されています。また、Oracle Order Management Suite の API とオープン・インタフェースに関する説明もあります。

『Oracle Applications Message Reference Manual』

このマニュアルには、すべての Oracle Applications メッセージが記載されています。このマニュアルは、リリース 11i のドキュメント CD-ROM から HTML 形式で利用できます。

サポート

オンサイト・サポートからセンターでのサポートまで、経験豊富な専門家チームが、Oracle Workflow の継続的運用に必要な支援や情報を提供します。このチームには、サービス技術員と会計管理担当の他、顧客の業務部門、Oracle Database、ハードウェアおよびソフトウェア環境に精通した、オラクル社のコンサルタントとサポートの専門家からなる大勢のスタッフが含まれます。

Oracle Applications データの変更を目的としたデータベース・ツール使用の禁止

特に指示がないかぎり、Oracle Applications データの変更には、SQL*Plus、Oracle Data Browser、データベース・トリガーまたは他のツールを使用しないでください。

オラクル社は、Oracle Database 内の情報の作成、格納、変更、取出しおよび保守に使用できるように、強力なツールを提供しています。ただし、SQL*Plus などの Oracle のツール製品を使用して Oracle Applications データを変更すると、データの整合性が破壊され、データ変更を監査できなくなる恐れがあります。

Oracle Applications 表は相関関係を持っているため、Oracle Applications を使用して変更を加えると、一度に多数の表が更新される可能性があります。ただし、Oracle Applications 以外のツールを使用して Oracle Applications を変更すると、ある表の行を変更しても、関連する表にはそれに対応する変更が加えられない場合があります。表が相互に同期なくなると、誤った情報を取り出したり、Oracle Applications 全体で予測できない結果を生じる恐れがあります。

Oracle Applications を使用してデータを変更すると、変更が有効かどうか Oracle Applications によって自動的にチェックされます。Oracle Applications では、情報を変更したユーザーが管理されます。データベース・ツールを使用してデータベース表に情報を入力すると、無効な情報を格納する可能性があります。また、SQL*Plus およびその他のデータベース・ツールでは変更の記録が保持されないため、情報を変更したユーザーを追跡できなくなります。

オラクル社について

オラクル社は、Oracle Applications のみでなく、データベース管理、アプリケーション開発、意思決定支援およびオフィス・オートメーション用の統合ソフトウェア製品ラインの開発および販売を行っています。Oracle Applications は、財務管理、サプライ・チェーン管理、製造、プロジェクト・システム、人事およびカスタマ・リレーションシップ・マネジメント用の 160 以上のソフトウェア・モジュールが完全に統合された製品です。

Oracle 製品は、メインフレーム、ミニコンピュータ、パーソナル・コンピュータ、ネットワーク・コンピュータおよびパーソナル・デジタル・アシスタントで使用できます。そのため、組織内の異なるコンピュータ、オペレーティング・システム、ネットワークおよびデータベース管理システムを単一の統一コンピューティングおよび情報リソースに統合できます。

オラクル社は情報管理ソフトウェアのリーディング・サプライヤであり、世界第 2 位の規模を誇るソフトウェア会社です。データベース、ツールおよびアプリケーション製品のみでなく、関連コンサルティングおよびサポート・サービスを世界 145ヶ国以上で提供しています。

1

Oracle Workflow の概要

この章では、ワークフロー・プロセスの概念と、Oracle Workflow の主要な機能について説明します。

Oracle Workflow の概要（開発者向け）

Oracle Workflow は、ビジネス・プロセスに基づく統合をサポートする完全なワークフロー管理システムを備えています。Oracle Workflow のテクノロジーを利用すると、ユーザー定義のビジネス・ルールに基づいてあらゆる種類の情報をルーティングしながら、ビジネス・プロセスをモデル化および自動化し、継続的に改善できます。

E-Business により、社内システムへの取引パートナーおよび企業間取引の統合のみならず、企業内のアプリケーションの統合に対する需要が加速しています。Oracle Workflow を導入すると、従来のアプリケーションに基づくワークフローおよび E-Business の統合ワークフローをサポートしながら、企業内および企業間の両方でビジネス・プロセスを自動化して効率よく作業できます。Oracle Workflow は、アプリケーション間の内部プロセスとビジネス・プロセスの連携をもたらす、他に例を見ないワークフロー・ソリューションです。

情報のルーティング

現代のビジネス・プロセスでは、常に変化するルールに応じて、複数の人に多くの種類の情報を伝達する必要があります。使用できる情報が多く、その形式が多様な場合、どのようにして適切な情報を適切な人に伝達しますか。Oracle Workflow を使用すると、それぞれのの人に、その人が必要としているすべての情報を提供できます。Oracle Workflow では、企業内外のビジネス・プロセスの各意思決定者に支援情報を提供することができます。

ビジネス・ルールの定義と変更

Oracle Workflow では、ドラッグ・アンド・ドロップで操作できるプロセス・デザイナを使用してビジネス・プロセスを定義し、継続的に改善していくことができます。

単に文書のあるユーザーから他のユーザーへと、いくつかの承認手順を経てルーティングしていくワークフロー・システムとは異なり、Oracle Workflow では洗練されたビジネス・プロセス・モデルを作成できます。ループするプロセス、並列フローに分岐して再び合流するプロセス、サブフローやさらにそこから細かい流れに分割されるプロセスなどを定義できます。Oracle Workflow では、ストアド・プロシージャの結果に基づいて選択するパスを判別できるため、Oracle Database の言語である Java および PL/SQL の機能を活用して、ワークフロー・プロセスに影響するあらゆるビジネス・ルールを定義できます。1-5 ページの「[ワークフロー・プロセス](#)」を参照してください。

電子通知の配信

Oracle Workflow では、ビジネス・プロセスの自動化の範囲を企業全体、さらに電子メールやインターネット・ユーザーを含む企業外のユーザーにまで拡張できます。Oracle Workflow では、人々に彼らの対応が待たれる項目の通知を電子メールにより発信し、電子メールの応答に基づいて処理します。また、必要な支援情報が含まれた作業リストを表示し、標準の Web ブラウザを使用して処理を実行することもできます。

システムの統合

Oracle Workflow では、ワークフローを開始できるビジネス・イベントへのサブスクリプションを設定したり、ビジネス・イベントが発生したときにシステム間でメッセージを伝播できます。社内のシステム間や外部システムとの間でイベントを伝達することもできます。この方法により、2 地点間のメッセージ統合を実現したり、より複雑なシステム統合のメッセージ伝達を Oracle Workflow を使用して集中管理できます。ルーティングおよび処理ルールが複雑なビジネス・プロセスをモデル化すれば、イベントを強力かつ柔軟に処理することができます。

主な機能と定義

Oracle Workflow Builder

Oracle Workflow Builder は、簡単なドラッグ・アンド・ドロップ操作で、ビジネス・プロセスを作成、表示または変更できるグラフィカル・ツールです。Oracle Workflow Builder を使用して、アクティビティ、項目タイプおよびメッセージなど、あらゆるワークフロー・オブジェクトを作成、変更できます。1-5 ページの「[ワークフロー・プロセス](#)」を参照してください。

いつでもワークフロー・アクティビティを追加、削除または変更でき、アクティビティ間に必要な新しい関連を設定できます。作業はワークフローの要約レベルのモデルで簡単に行うことができます。必要に応じて、ワークフロー内のアクティビティを展開して、より詳細なレベルで表示することもできます。Oracle Workflow Builder は、デスクトップ PC からでも、ネットワークに接続していないラップトップ PC からでも操作できます。

ワークフロー・エンジン

Oracle Database に組み込まれたワークフロー・エンジンでは、実行時にプロセス定義を実装します。ワークフロー・エンジンは、ワークフローの状態を監視し、プロセスのアクティビティのルーティングを調整します。たとえば、ワークフロー・アクティビティの完了など、ワークフローの状態が変化すると、PL/SQL API または Java API を介してエンジンにシグナルが送られます。エンジンは、柔軟に定義されているワークフロー・ルールに基づいて実行対象のアクティビティを判別し、そのアクティビティを実行します。ワークフロー・エンジンでは、ループ、分岐、並列フロー、サブフローなど、高度なワークフロー・ルールをサポートしています。

ビジネス・イベント・システム

ビジネス・イベント・システムは、Oracle Advanced Queuing (AQ) インフラストラクチャを使用してシステム間でビジネス・イベントを伝達するためのアプリケーション・サービスの 1 つです。ビジネス・イベント・システムは、重要なイベントへのサブスクリプションを登録するイベント・マネージャおよびワークフロー・プロセス内のビジネス・イベントをモデル化するイベント・アクティビティから構成されます。

ローカル・イベントが発生すると、イベントを呼び出したコードと同じトランザクション内でサブスクリバ・コードが実行されます。サブスクリプション処理では、イベント情報に関するカスタム・コードの実行、ワークフロー・プロセスに対するイベント情報の送信、他のキューまたはシステムに対するイベント情報の送信などが行われます。

ワークフロー定義ローダー

ワークフロー定義ローダーは、ワークフロー定義をデータベースとそれに対応するフラット・ファイル間で移動するユーティリティ・プログラムです。このプログラムを使用して、ワークフロー定義を開発用データベースから本番データベースに移行したり、既存の定義をアップグレードできます。ワークフロー定義ローダーはスタンドアロン・サーバー・プログラムですが、Oracle Workflow Builder にも統合されており、データベースやファイル内のワークフロー定義のオープンや保存が可能です。

完全なプログラム拡張性

Oracle Workflow では、独自の PL/SQL プロシージャや外部関数をアクティビティとしてワークフローに入れることができます。アプリケーション・コードを変更しなくても、ワークフロー・エンジンによりプログラムの前提条件が満たされていることが確認されれば、いつでも独自のプログラムを実行できます。

電子通知

Oracle Workflow では、ユーザーをワークフローに組み込んで、購買申請や受注の承認など、自動化できないアクティビティを処理できます。通知システムでは、ワークフロー内のユーザーに通知を送信し、ユーザーからの応答を処理します。電子通知はロールにルーティングされます。ロールは、個々のユーザーまたはユーザー・グループです。そのロールに関連付けられているユーザーなら誰でも、その通知に基づいて作業を実行できます。

各通知には意思決定に必要なすべての情報を含むメッセージが入っています。情報は、メッセージの本文に埋め込まれているか、別の文書として添付されています。Oracle Workflow ではそれぞれの通知アクティビティの応答を理解し、次のワークフロー・アクティビティに移動する方法を判断します。

電子メールの統合

電子メール (E-mail) ・ユーザーは、未処理の作業項目に関する通知を受信し、選択した電子メール・アプリケーションを使用してその通知に応答できます。電子メール通知には、通知に対する応答の別手段となる添付ファイルを添付できます。

インターネット対応ワークフロー

標準の Web ブラウザにアクセスできるユーザーは、誰でもワークフローに組み込むことができます。Web ユーザーは、「通知」Web ページにアクセスして未処理の作業項目を表示し、別のページに移動して、より詳細な内容の表示や応答が可能です。

監視と管理

ワークフローの管理者とユーザーは、Java サポート機能を持つ標準 Web ブラウザを使用して、ワークフロー・モニターに接続し、ワークフロー・プロセスの作業項目の進捗を表示できます。ワークフロー・モニターでは、ワークフロー・プロセスの特定インスタンスのプロセス・ダイアグラムが説明付きのビューで表示されるため、ユーザーは作業項目のステータスをグラフィカルに見ることができます。また、作業項目、プロセスおよびプロセス内の各アクティビティについて、個別のステータス要約も表示されます。

Oracle Applications に組み込まれた Oracle Workflow を使用し、Oracle Applications Manager を実装している場合は、Oracle Applications Manager の Oracle Workflow Manager コンポーネントを Oracle Workflow の追加管理ツールとして使用することもできます。Oracle Applications Manager は、Oracle Applications のコンカレント処理、Oracle Workflow、その他の機能の管理および診断機能を提供するツールの 1 つです。詳細は、Oracle Applications Manager のオンライン・ヘルプを参照してください。

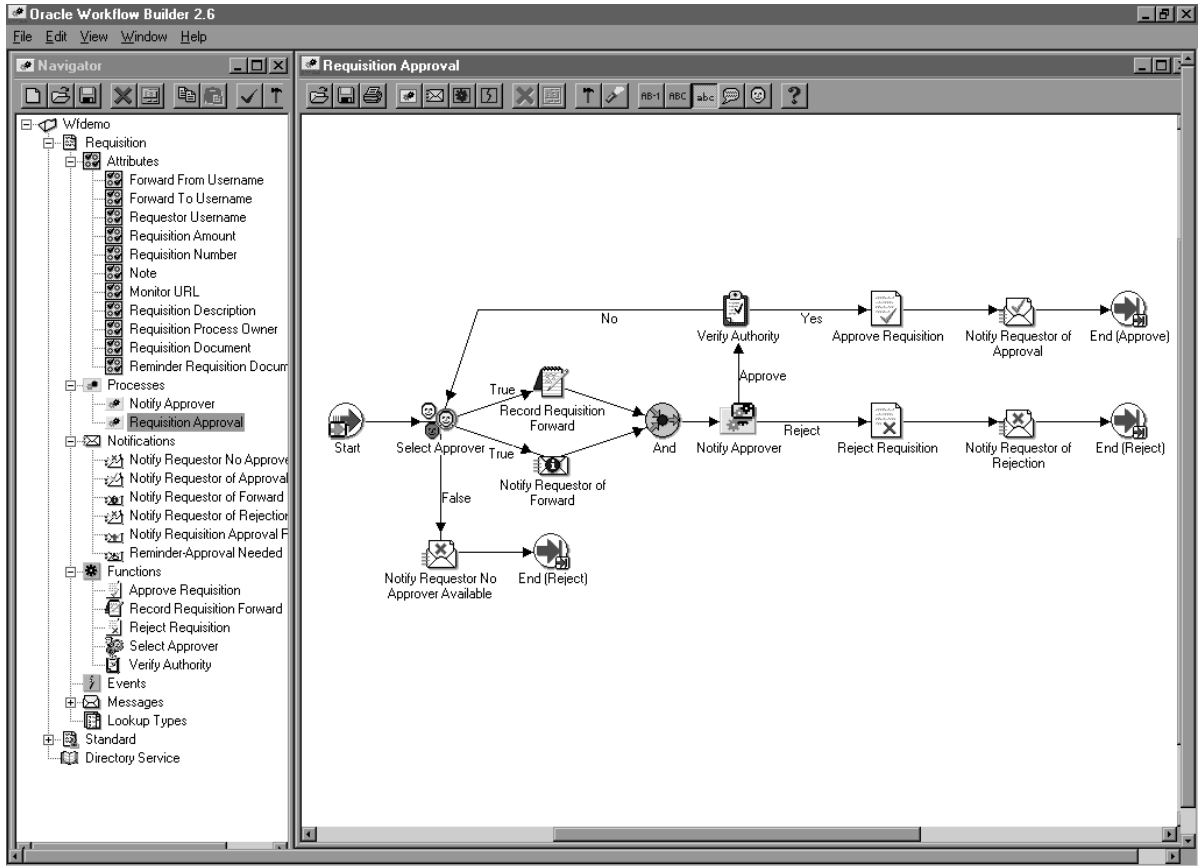
また、スタンドアロン版の Oracle Workflow を使用している場合は、Oracle Enterprise Manager から利用可能なスタンドアロン Oracle Workflow Manager コンポーネントを、Oracle Workflow の追加管理ツールとして使用できます。詳細は、Oracle Workflow Manager のオンライン・ヘルプを参照してください。

ワークフロー・プロセス

Oracle Workflow は、ユーザーが定義したルールに従ってビジネス・プロセスを管理します。ルールはワークフロー・プロセス定義と呼ばれ、プロセス内で発生するアクティビティと各アクティビティ間の関連を含みます。プロセス定義に含まれるアクティビティには、PL/SQL のストアード・プロシージャや外部関数によって定義される自動化関数、任意に応答を要求する場合があるユーザーやロールへの通知、ビジネス・イベント、より細かいアクティビティの集合で構成されるサブフローなどがあります。

ワークフロー・プロセスは、アプリケーションで一連の Oracle Workflow Engine API をコールしたときに起動されます。ワークフロー・エンジンには、アプリケーションで定義されている関連作業項目を、特定のワークフロー・プロセス定義を介して実行する機能があります。ワークフロー・エンジンでは、ワークフロー・プロセス定義に従って自動化の手順を実行し、外部処理が必要な場合は該当するエージェントを起動します。

次のダイアグラムは、購買申請を管理者や複数の管理者承認をもらうためにルーティングする、簡単なワークフロー・プロセス定義を表しています。



この全体図を、プロセスまたはプロセス・ダイアグラムと呼びます。アイコンはアクティビティを表し、矢印はアクティビティ間の遷移を表しています。この表の例では、ユーザーが該当するアプリケーションで購買申請を作成して送信すると、プロセスに新規項目が作成されます。

このプロセスには、PL/SQL のストアド・プロシージャとして実装されている次のような複数のワークフロー・アクティビティが含まれています。

- 承認者の選択： ビジネス・ルールに従って、購買申請を承認するユーザーを選択します。
- 承認権限の検証： 選択された承認者に、購買申請の承認権限があるかどうかを検証します。

2

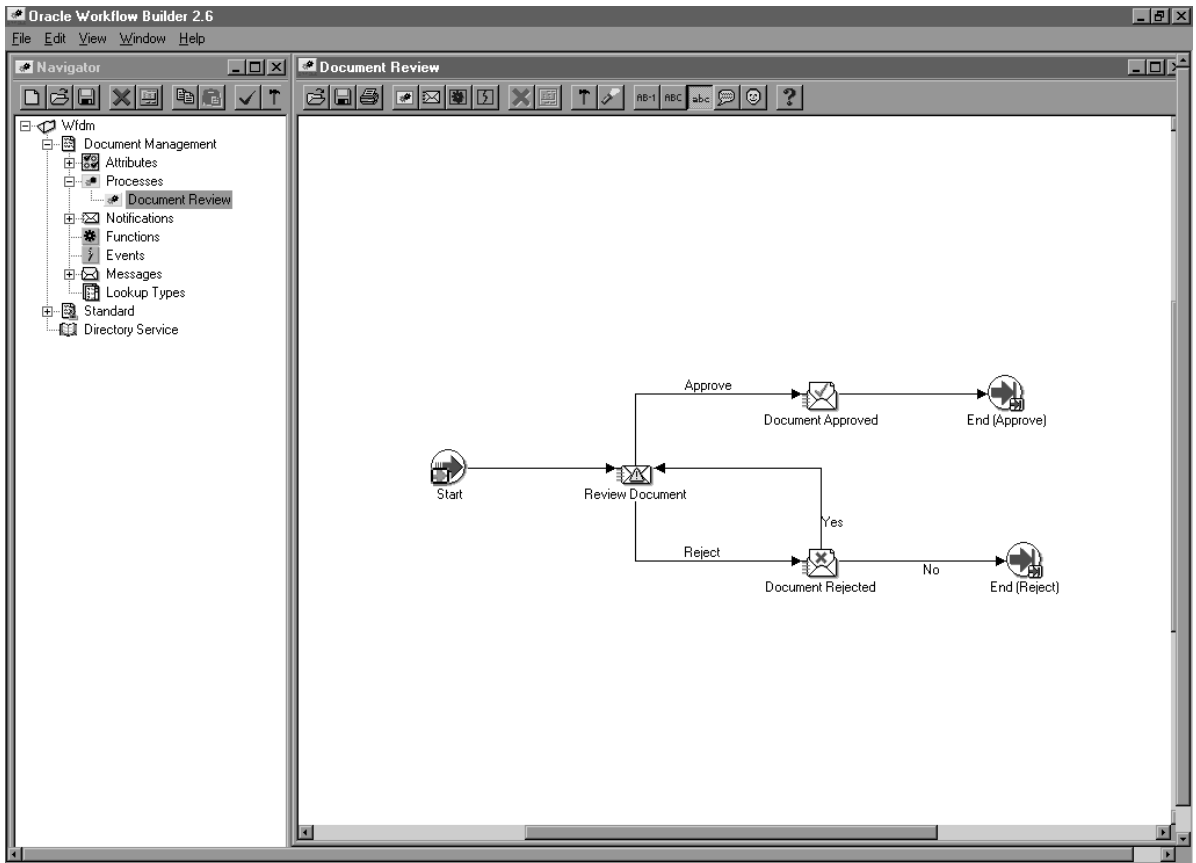
ワークフロー・プロセス定義

この章では、Oracle Workflow Builder を使用してワークフロー・プロセスを定義する方法について説明します。

Oracle Workflow Builder の概要

Oracle Workflow Builder は、ワークフロー・プロセス定義の作成、表示および変更に使用するグラフィカル・ツールです。Oracle Workflow Builder には「ナビゲータ」ウィンドウがあり、このウィンドウでビジネス・プロセスのアクティビティとコンポーネントを定義します。次に、プロセス・ウィンドウでアクティビティを組み合わせて、プロセス・ダイアグラムを作成します。2-8 ページの「[Oracle Workflow Builder によるプロセス定義の作成](#)」を参照してください。

注意： ワークフロー・プロセス定義は、テキスト・エディタで開いて編集できるフラット・ファイルとしても保存できます。このため、プロセス定義をスクリーン・リーダーが読み上げることができ、ユーザー補助機能を向上させることができます。



注意：「ナビゲータ」ウィンドウや Oracle Workflow Builder のプロセス・ウィンドウを最大表示にすると、[Alt] キーを使用してキーボードからメニューにアクセスできなくなります。

ナビゲータ・ツリーの構造

「ナビゲータ」ウィンドウには、Oracle Workflow Builder で開いたり、ロードした各データ・ストアのナビゲータ・ツリー階層が表示されます。データ・ストア（最上位のブランチ）は、ワークフロー・プロセス定義を保存するデータベース接続またはフラット・ファイルです。各データ・ストア内に 1 つ以上の項目タイプのヘッダー（2 次ブランチ）があり、プロセスとそのコンポーネント・オブジェクトの特定セットのグループを表します。各項目タイプのブランチの下に次の 7 つの 3 次ブランチが表示されます。

- **属性：** 現行の項目タイプの属性リスト。項目タイプ属性は、項目タイプの特徴を示します。たとえば、ある項目タイプが購買申請の場合、購買申請金額や購買申請 ID などが項目タイプ属性になります。3-2 ページの「[項目タイプ属性](#)」を参照してください。
- **プロセス：** 現行の項目タイプのプロセス・アクティビティまたはワークフロー・プロセス定義のリスト。4-2 ページの「[プロセス](#)」ウィンドウ」および 3-53 ページの「[アクティビティ](#)」を参照してください。
- **通知：** 現行の項目タイプに関連した通知アクティビティのリスト。通知アクティビティは、ユーザーやロールにメッセージを送信します。メッセージには応答を求めるプロンプトが表示されることもあれば、単に情報が表示されることもあります。3-53 ページの「[アクティビティ](#)」を参照してください。
- **関数：** 現行の項目タイプに関連した関数アクティビティのリスト。関数アクティビティは、ワークフロー・エンジンで自動的に実行される PL/SQL のストアド・プロシージャを示します。関数アクティビティには、アクティビティ属性が関連付けられることもあります。3-53 ページの「[アクティビティ](#)」を参照してください。
- **イベント：** 現行の項目タイプに関連したイベント・アクティビティのリスト。イベント・アクティビティは、プロセスによって受信、発生または送信されるビジネス・イベントを表します。3-53 ページの「[アクティビティ](#)」を参照してください。
- **メッセージ：** 現行の項目タイプに関連した通知アクティビティがユーザーやロールに送信できるメッセージのリスト。メッセージにはメッセージ属性が関連付けられることもあります。3-24 ページの「[メッセージ](#)」を参照してください。
- **選択肢タイプ：** 現行の項目タイプに関連した選択肢タイプのリスト。選択肢タイプには、選択肢コードと呼ばれる 1 つ以上の値が関連付けられています。選択肢タイプは、メッセージで参照されたり、考えられる結果タイプとして通知、関数またはプロセスで参照できる値のリストです。3-20 ページの「[選択肢タイプ](#)」を参照してください。

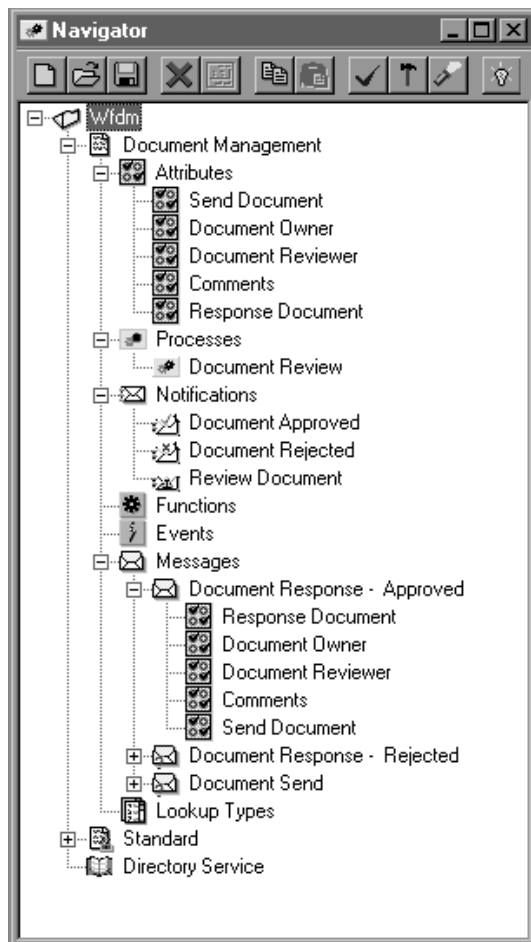
注意： それぞれのデータ・ストアには、「ディレクトリ・サービス」のブランチも含まれています。「ディレクトリ・サービス」のブランチには、Oracle Workflow データベースからロードしたディレクトリ・サービスのロールがすべて表示されます。4-26 ページの「[ロール](#)」を参照してください。

データ・ストアがデータベース接続されており、そのデータベースに、Oracle Workflow Builder にロードされていない他の項目タイプが含まれている場合は、「項目タイプの非表示」というブランチが表示されます。「項目タイプの非表示」をダブルクリックすると、「項目タイプの表示」ウィンドウが表示され、Oracle Workflow Builder に他の項目タイプをロードできます。

ナビゲータ・ツリーの表示

ナビゲータ・ツリーは、ファイル・システムの階層とよく似た形で編成され、対象コンポーネントが見つかるまで、先頭にプラス記号 (+) が付いているブランチをより詳細なサブブランチに展開していくことができます。サブブランチは、展開元ブランチの下にインデントされて表示されます。展開されたブランチの先頭にはマイナス記号 (-) が付きます。ブランチにプラスやマイナスの記号が付かなくなるまで、展開していくことができます。ナビゲータ・ツリーを展開または縮小するには、マウスを使用する方法と、キーボードの矢印キーを使用する方法があります。

「ナビゲータ」ウィンドウには、そのウィンドウ内でのアクションの実行に使用できるツールバーもあります。B-8 ページの「[ナビゲータのツールバー](#)」を参照してください。



▶ ナビゲータ・ツリーでのオブジェクトの検索

The screenshot shows a 'Search' dialog box with the following elements:

- Search Text:** A text input field.
- Display Name:** checkbox.
- Internal Name:** checkbox.
- Object Type:** A group box containing:
 - Item Type
 - Message
 - Function
 - Lookup Type
 - Notification
 - Lookup Code
 - Process
 - Attribute
 - Role
- All Objects:** checkbox.
- Search:** A button.
- Close:** A button.

1. 「編集」メニューから「検索」を選択して「検索」ウィンドウを表示します。このウィンドウで、ナビゲータ・ツリーでオブジェクトを検索する検索基準を指定します。
2. 「検索テキスト」フィールドに検索するテキストを入力します。検索時には大文字 / 小文字は区別されず、フィールドで指定したテキスト・パターンが検索されます。
3. このテキストをオブジェクトの「表示名」または「内部名」で検索するように指定します。
4. この検索範囲を限定する場合はオブジェクト・タイプを指定し、すべてのオブジェクトのプロパティ画面内でテキストを検索する場合は「すべてのオブジェクト」を選択します。
5. 「検索」ボタンをクリックします。
6. 「編集」メニューから「再検索」を選択すると、「検索」ウィンドウで直前に定義した検索基準を使用して検索を繰り返せます。

Oracle Workflow Builder によるプロセス定義の作成

Oracle Workflow Builder を使用する前に、プロセスで達成する内容を計画する必要があります。特に、必要なアクティビティ、各アクティビティの順序、プロセスのブランチを左右する結果、通知する相手や通知内容などを決定します。Oracle Workflow には、デモンストレーション・ワークフローのサンプルがいくつか用意されています。10-2 ページの「[サンプル・ワークフロー・プロセス](#)」を参照してください。

ワークフロー・プロセスの定義を作成するには、次の方法があります。

- **トップダウン・デザイン**： 上位レベルからプロセス定義を設計する場合は、最初にプロセス・ダイアグラムとアクティビティの概要を決定してから、各アクティビティでサポートしているオブジェクトを具体的に作成できます。2-11 ページの「[トップダウンによるプロセス定義の作成](#)」を参照してください。
- **ボトムアップ・デザイン**： 最初に詳細な部分を定義して設計する場合は、最初に下位のプロセスでサポートしているオブジェクトの定義を作成してから、上位レベルのプロセス・ダイアグラムを作成します。2-9 ページの「[ボトムアップによるプロセス定義の作成](#)」を参照してください。

クイック・スタート・ウィザード

クイック・スタート・ウィザードを使用すると、プロセス定義のテンプレートを使用して、最初からプロセス定義を簡単に作成できます。クイック・スタート・ウィザードでは、まずプロセスに関する新しい項目タイプを作成し、必要な最低限の情報を入力するように要求されます。次に、基本のプロセス・ダイアグラムを作成し、それぞれの詳細なアクティビティを具体的に定義します。クイック・スタート・ウィザードでテンプレートを設定した後は、トップダウンまたはボトムアップのアプローチを使用して設計を完成します。2-19 ページの「[クイック・スタート・ウィザードの使用](#)」を参照してください。

バージョンと有効日

新規のアクティビティを作成すると、Oracle Workflow Builder によりバージョン番号が割り当てられます。既存のアクティビティを変更するたびに、バージョン番号が更新されます。アクティビティの新バージョンはデータベースに保存され、旧バージョンが上書きされることはありません。Oracle Workflow では、アクティビティの有効日も設定され、常にアクティビティの 1 バージョンのみが有効になっています。Oracle Workflow では、プロセスの実行中は、そのプロセスの開始時に有効だったアクティビティのバージョンが使用されます。プロセスの途中でアクティビティのバージョンが切り替わることはありません。プロセス自体がアクティビティであるため、そのプロセスのインスタンスが完了するまで、プロセス定義は変化しないので注意してください。

また、Oracle Workflow Builder では、有効日に従ってプロセス定義を保存し、ロードすることもできます。たとえば、過去に有効になっていた定義を Oracle Workflow Builder にロードできます。また、定義をデータベースに保存し、将来の任意の時点で有効にすることもできます。

Oracle Workflow Builder では、項目タイプ、項目タイプ属性、メッセージおよび選択肢タイプなど、定数と見なされるオブジェクトのバージョン情報を管理しないので注意してください。これらのオブジェクトの場合は常に最新の定義が適用されるため、これらのオブジェクトを変更した場合、変更前と矛盾しないかどうかを必ず考慮する必要があります。変更が既存のプロセスに影響する場合は、既存のオブジェクトを編集するのではなく、新規のオブジェクトを作成する必要があります。

関連項目：

3-80 ページ「[Oracle Workflow Builder のオブジェクトの変更](#)」

プロパティ画面の「編集」ボタンの使用

Oracle Workflow Builder でオブジェクトを作成するには、そのオブジェクトのプロパティ画面で情報を入力します。入力する情報には、値リストから選択できるものもあります。Oracle Workflow Builder の他のプロパティ画面で定義された値がドロップ・ダウン・フィールドに表示される場合は、ドロップ・ダウン・リストの右に「編集」ボタンが表示されます。ドロップ・ダウン・リストで値を選択すると、隣の「編集」ボタンを選択し、その値のソース・プロパティ画面を表示して編集できます。ソース・プロパティ画面の設定を完了して「OK」または「キャンセル」をクリックすると、作業していた元のプロパティ画面に戻ります。

たとえば、通知アクティビティを作成する場合は、そのアクティビティの「結果タイプ」を指定する必要があります。「結果タイプ」のドロップ・ダウン・フィールドを使用して、値「なし」または事前定義の選択肢タイプを選択できます。選択肢タイプを選択した場合は、隣の「編集」ボタンを選択し、その選択肢タイプのプロパティ画面を表示できます。選択肢タイプのプロパティ画面を表示または編集した後は、「OK」または「キャンセル」を選択すると、通知アクティビティのプロパティ画面に戻ります。

▶ ボトムアップによるプロセス定義の作成

1. Oracle Workflow Builder を起動するには、「Oracle - <SID NAME>」プログラム・グループの「Application Development」フォルダに入っている「Oracle Workflow Builder」アイコンをダブルクリックします。Windows NT 4.0 以上を使用している場合は、「スタート」メニューから適切なプログラム・フォルダの「Oracle Workflow Builder」アイコンを選択する方法もあります。
2. 「ファイル」メニューから「新規作成」を選択し、新しいプロセス定義のための作業領域を作成します。

提案： クイック・スタート・ウィザードを使用して、新しいプロセス定義の概略を最初に作成することもできます。一度クイック・スタート・ウィザードで新しい項目タイプとプロセス・アクティビティを作成すると、手順 4 に進んで、新しい項目タイプとプロセス・アクティビティでサポートするオブジェクトの定義を開始できます。2-19 ページの「[クイック・スタート・ウィザードの使用](#)」を参照してください。

3. 新しい項目タイプを作成します。項目タイプでは、プロセスで管理する作業項目を分類します。3-6 ページの「[項目タイプの作成](#)」を参照してください。
4. 項目タイプの属性を定義して項目タイプを具体的に記述し、プロセス内のアクティビティでこれらの属性の情報を参照できるようにすることができます。3-8 ページの「[項目タイプまたはアクティビティ属性の定義](#)」を参照してください。
5. 新規の選択肢タイプを作成します。3-21 ページの「[選択肢タイプの作成](#)」を参照してください。

アクティビティを定義する前に、その「結果タイプ」を表す選択肢タイプを定義する必要があります。結果タイプは、アクティビティの完了時に考えられる結果のリストです。選択肢タイプとアクティビティを定義した後、選択肢をナビゲータ・ツリーのアクティビティにドラッグし、その選択肢をアクティビティの結果タイプに設定できます。選択肢タイプは、項目タイプ属性、アクティビティ属性、メッセージまたはメッセージ属性で参照することもできます。

6. 新規のメッセージを作成します。3-38 ページの「[メッセージの作成](#)」を参照してください。

プロセスの通知アクティビティを作成する場合は、通知アクティビティで送信するメッセージを最初に作成する必要があります。新しいメッセージをナビゲータ・ツリーの通知アクティビティにドラッグすると、対象アクティビティにメッセージを割り当てることができます。

メッセージのメッセージ属性も作成できます。動的な内容を表すように実行時にトークンで置き換えられるメッセージには、「送信」タイプのメッセージ属性を組み込むことができます。また、「応答」タイプのメッセージ属性を定義して、通知の受信者が受信時に応答するように設定できます。3-44 ページの「[メッセージ属性の定義](#)」を参照してください。

7. 新規のプロセス・アクティビティ、通知アクティビティ、関数アクティビティ、またはイベント・アクティビティを作成します。標準項目タイプに関連付けられている事前に定義された標準アクティビティを使用することもできます。3-53 ページの「[アクティビティ](#)」および 5-2 ページの「[標準アクティビティ](#)」を参照してください。

上位プロセス・ダイアグラムを表すプロセス・アクティビティを、1つ以上は定義する必要があります。プロセス・ダイアグラムにより、プロセス内のすべてのアクティビティ間の関連が設定されます。

8. プロセス・ダイアグラムを作成します。

該当するプロセス・アクティビティの「プロセス」ウィンドウを表示し、ワークフロー・プロセスを定義するアクティビティとトランジションのダイアグラムを作成します。アクティビティは、ナビゲータ・ツリーから「プロセス」ウィンドウにドラッグできます。4-6 ページの「[プロセスの図示](#)」を参照してください。

9. 「ファイル」メニューから「保存」または「別名保存」を選択し、作業内容を保存します。2-16 ページの「[作業内容の保存](#)」を参照してください。

10. Oracle Workflow Server からアクセスできるデータベース内で、PL/SQL 関数アクティビティによってコールされる PL/SQL ストアド・プロシージャを作成します。この操作には、SQL*Plus または Oracle Procedure Builder を使用できます。『Oracle Workflow API リファレンス』の「Oracle Workflow のプロシージャと関数」および 6-3 ページの「関数アクティビティがコールする PL/SQL プロシージャの標準 API」を参照してください。

関連項目：

- 2-12 ページ「プロセス定義の変更」
- 3-79 ページ「Oracle Workflow Builder のオブジェクトの削除」
- 3-80 ページ「Oracle Workflow Builder のオブジェクトの変更」
- 2-25 ページ「項目タイプの定義」 Web ページ

▶ トップダウンによるプロセス定義の作成

1. Oracle Workflow Builder を起動するには、「Oracle - <SID NAME>」プログラム・グループの「Application Development」フォルダに入っている「Oracle Workflow Builder」アイコンをダブルクリックします。Windows NT 4.0 以上を使用している場合は、「スタート」メニューから適切なプログラム・フォルダの「Oracle Workflow Builder」アイコンを選択する方法もあります。
2. クイック・スタート・ウィザードを使用して、新しいプロセス定義の概略を作成します。新規の項目タイプとプロセス・アクティビティに必要な情報を指定します。2-19 ページの「クイック・スタート・ウィザードの使用」を参照してください。
3. 「プロセス」ウィンドウが表示され、「開始アクティビティ」と「終了アクティビティ」のノードが示されます。開始ノードと終了ノードの間に新しいアクティビティ・ノードを定義し、プロセスのダイアグラムを作成します。4-9 ページの「プロセスのノードの定義」を参照してください。

標準項目タイプに関連付けられている事前に定義された標準アクティビティを使用することもできます。5-2 ページの「標準アクティビティ」を参照してください。

4. アクティビティ間の遷移を示して、プロセスをモデル化します。4-6 ページの「プロセスの図示」を参照してください。
5. 「ファイル」メニューから「保存」または「別名保存」を選択し、作業内容を保存します。2-16 ページの「作業内容の保存」を参照してください。

注意： Oracle Workflow では、作業内容を保存すると、不正なプロセス定義や不足している情報が自動的に検証され、該当するものがあれば「ワークフロー・エラー」検証ウィンドウに表示されます。「ワークフロー・エラー」ウィンドウは非モーダル・ウィンドウで、このウィンドウを開いたまま、プロセスに戻って識別されたエラーを修正できます。また、作業内容をそのまま保存してから、エラーを修正することもできます。他の文書で後で参照できるように、情報をクリップボードに貼り付けるには、「コピー」ボタンを使用して情報をコピーします。エラーを修正せずに作業内容を保存すると、次にこのプロセス定義をオープンしたときに「ワークフロー・エラー」ウィンドウが表示されます。

関連項目

- 2-12 ページ [「プロセス定義の変更」](#)
- 3-79 ページ [「Oracle Workflow Builder のオブジェクトの削除」](#)
- 3-80 ページ [「Oracle Workflow Builder のオブジェクトの変更」](#)
- 2-25 ページ [「項目タイプの定義」 Web ページ](#)

▶ プロセス定義の変更

1. Oracle Workflow Builder を起動するには、「Oracle - <SID NAME>」プログラム・グループの「Application Development」フォルダに入っている「Oracle Workflow Builder」アイコンをダブルクリックします。Windows NT 4.0 以上を使用している場合は、「スタート」メニューから適切なプログラム・フォルダの「Oracle Workflow Builder」アイコンを選択する方法もあります。
2. 「ファイル」メニューから「オープン」を選択し、データベースへの接続をオープンするか、変更するプロセス定義が含まれているファイルをオープンします。2-13 ページの「[既存データ・ストア内のプロセス定義へのアクセス](#)」を参照してください。
3. 変更するプロセス定義に関連付けられている既存の項目タイプを選択し、展開します。
4. 項目タイプ、項目タイプ属性、選択肢、メッセージ、メッセージ属性、プロセス・アクティビティ、通知アクティビティ、関数アクティビティまたはアクティビティ属性を変更できます。3-6 ページの「[項目タイプの作成](#)」、3-8 ページの「[項目タイプまたはアクティビティ属性の定義](#)」、3-21 ページの「[選択肢タイプの作成](#)」、3-38 ページの「[メッセージの作成](#)」、3-44 ページの「[メッセージ属性の定義](#)」または 3-53 ページの「[アクティビティ](#)」を参照してください。
5. プロセス・アクティビティの「プロセス」ウィンドウを表示し、プロセス・ダイアグラムを変更することもできます。4-6 ページの「[プロセスの図示](#)」を参照してください。
6. 「ファイル」メニューから「保存」または「別名保存」を選択し、作業内容を保存します。2-16 ページの「[作業内容の保存](#)」を参照してください。

関連項目：

3-79 ページ [「Oracle Workflow Builder のオブジェクトの削除」](#)

3-80 ページ [「Oracle Workflow Builder のオブジェクトの変更」](#)

2-25 ページ [「項目タイプの定義」 Web ページ](#)

項目タイプのオープンと保存

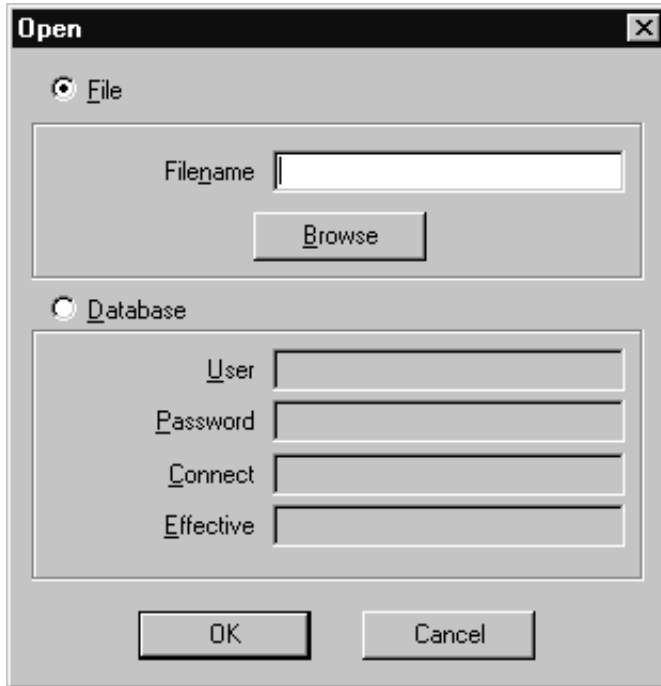
すべてのプロセスは、項目タイプに関連付けられます。1つの項目タイプに、1つ以上のプロセスを含めることができます。項目タイプは、データベースやフラット・ファイルに保存できます。作業結果をデータベースに保存すると、実際には現行データ・ストア内で変更があった情報がすべて保存されます。フラット・ファイルに保存すると、実際には現行データ・ストア内のすべての情報がファイルに保存されます。また、項目タイプをデータベースやフラット・ファイルから Oracle Workflow Builder にロードすることもできます。項目タイプをオープンすると、その項目タイプに関連した属性、メッセージ、選択肢、通知、関数およびプロセスすべてが自動的に取り出されます。

注意： ワークフロー・プロセス定義のコピーを常にフラット・ファイルとして保存し、そのファイルをソース・コントロール・システムにチェックインして、プロセス定義の作業用バージョンを保守してください。データベースにアクセスする他のユーザーが定義を更新できるように、データベースに格納されたプロセス定義は、ソース・コントロール・バージョンとして使用しないでください。

注意： Oracle Workflow Builder からデータベースに接続するには、インストールされている Oracle Workflow Builder の言語が、データベースにインストールされている Oracle Workflow Server の使用可能な言語のいずれかと一致している必要があります。

▶ 既存データ・ストア内のプロセス定義へのアクセス

1. Oracle Workflow Builder を起動するには、「Oracle - <SID NAME>」プログラム・グループの「Application Development」フォルダに入っている「Oracle Workflow Builder」アイコンをダブルクリックします。Windows NT 4.0 以上を使用している場合は、「スタート」メニューから適切なプログラム・フォルダの「Oracle Workflow Builder」アイコンを選択する方法もあります。Oracle Workflow Builder で、「ファイル」メニューから「オープン」を選択します。



2. データベースまたはファイルを選択し、プロセス定義が関連付けられている項目タイプを含むソースに接続します。
3. ファイルを開く場合： ファイルのフルパスを指定して「OK」を選択するか、「参照」をクリックし、ファイル位置を指定してファイル（拡張子 .wft）を開きます。

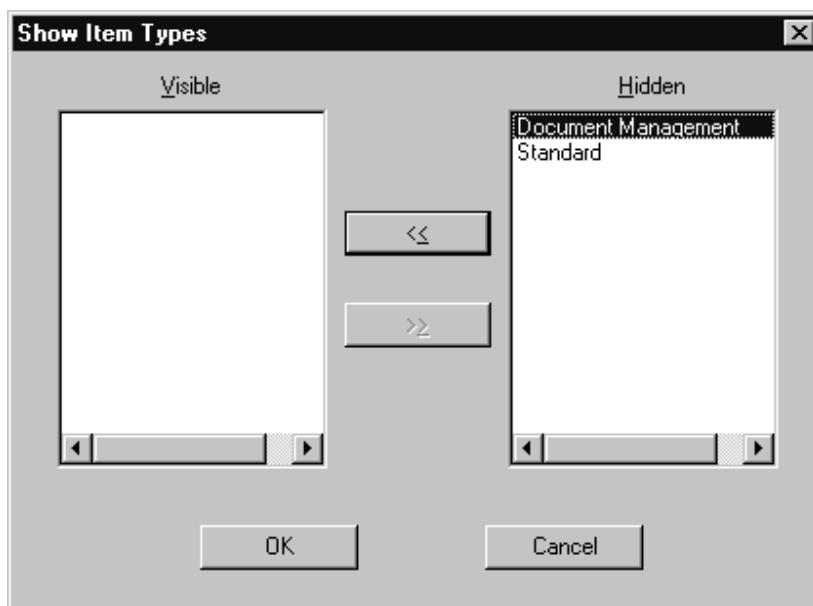
注意： Microsoft Windows 98/2000/XP や Windows NT 4.0 の「エクスプローラ」、または Microsoft Windows NT の「ファイルマネージャ」から、.wft ファイルをナビゲータ・ツリーにドラッグ・アンド・ドロップし、そのファイルを Oracle Workflow Builder で開くこともできます。

注意： 「参照」を使用してファイルを検索してオープンすると、オープンしたファイルが含まれている現行のディレクトリが、次にファイルをオープンするときのデフォルトのディレクトリとなります。このデフォルトのディレクトリは、次に「参照」を使用して他のファイルを検索するまで保持されます。

4. データベース接続を開く場合： データベースのユーザー名とパスワードを入力します。データベースの別名または接続文字列を入力し、「OK」を選択します。

注意： Oracle Applications に組み込まれている Workflow を使用している場合は、APPS スキーマを使用してデータベースに接続します。

5. 特定の日に有効であったプロセス定義を取り出す場合は、「有効日」フィールドで日時を指定して、そのデータをデータベースから取り出します。日時の指定に使用する書式は、Windows の「コントロールパネル」の「地域」設定で定義した日時の書式に基づきます。



6. データ・ストアに複数の項目タイプが存在する場合は、「項目タイプの表示」ウィンドウが表示されます。表示する項目タイプを「非表示」リストから選択し、「<<」を選択して「表示」リストに移動します。「OK」を選択し、選択した項目タイプをナビゲータ・ツリーにロードします。
7. 現行のデータ・ストア内で非表示の項目タイプを表示して変更する場合は、ナビゲータ・ツリーの「項目タイプの非表示」のブランチをダブルクリックして、「項目タイプの表示」ウィンドウを表示し、表示する項目タイプを選択します。また、「項目タイプの表示」ウィンドウを表示するには、「ファイル」メニューから「項目タイプの表示 / 非表示」を選択する方法もあります。

注意： 項目タイプは、相互に参照している場合でも、データ・ストア間では任意の順序でコピーできます。ただし、外部キーの参照によって、検証エラーとなることがあります。このようなエラーは、外部キーの参照を解決するために、場合によっては他の項目タイプを新しいデータ・ストアへコピーする必要があるため、注意が必要です。参照されている項目タイプがすべて、コピー先の新しいデータ・ストアにコピーされると、新しいデータ・ストアの最終プロセス定義が有効になります。

8. 作業の完了後に、「ファイル」メニューから「保存」を選択し、変更内容を保存して即時に有効にします。2-16 ページの「[作業内容の保存](#)」を参照してください。

関連項目：

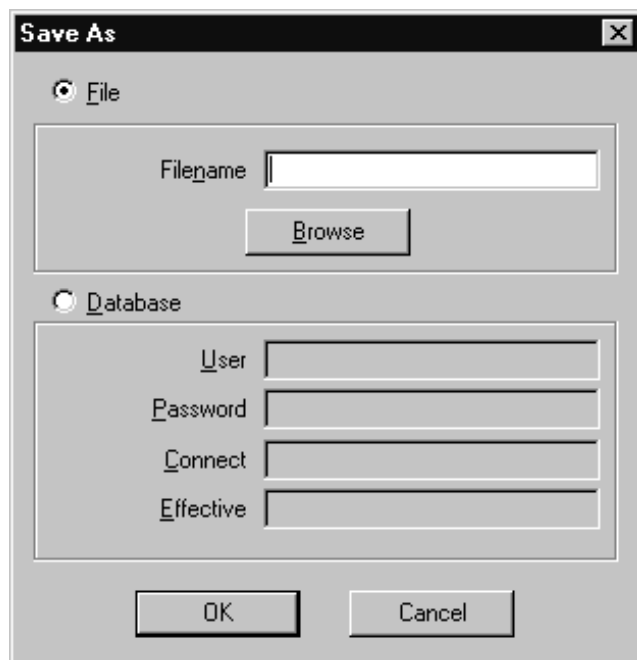
2-18 ページ「[コマンド・プロンプトからの Oracle Workflow Builder の起動](#)」

➤ **作業内容の保存**

1. 「ファイル」メニューから「保存」を選択し、変更内容を保存して即時に有効にします。

「保存」コマンドを使用すると、現在選択しているデータ・ストアの変更済オブジェクトがすべて（非表示でも）データ・ストアに保存されます。特定の項目タイプのみを保存する場合は、新規のデータ・ストアを作成し、保存する項目タイプを新規ストアにコピーして、新規ストアを保存する必要があります。

注意： Oracle Workflow Builder では、2つのモードのどちらかで作業結果をデータベースに保存できます。「ヘルプ」メニューから「Oracle Workflow Builder のバージョン情報」ダイアログ・ボックスを表示すると、「カスタマイズされたオブジェクトの変更を可能にする」というチェック・ボックスが表示されます。このチェック・ボックスをオンにすると、編集結果はアップロード・モードで保存され、以前にカスタマイズしたオブジェクトと、変更のためにアクセスした保護対象のオブジェクトが上書きされます。このチェック・ボックスをオフにすると、Oracle Workflow Builder はアップグレード・モードで実行されます。このモードでは、変更のためにアクセスした保護対象のオブジェクトの編集結果のみが保存され、以前にカスタマイズしたオブジェクトは上書きされません。この2つのモードは、ワークフロー定義ローダーのプログラムのアップグレードおよびアップロード動作と同じです。デフォルトでは、このチェック・ボックスはオフになっています。3-19 ページの「[オブジェクトのアクセス・レベルの設定](#)」および『Oracle Workflow 管理者ガイド』の「ワークフロー定義ローダーの使用」を参照してください。



2. 作業結果を別のデータ・ストア（データベースまたはフラット・ファイル）に保存する場合や、現在のシステム日付とは異なる有効日付でデータベースに保存する場合は、「ファイル」メニューから「別名保存」を選択します。「別名保存」ウィンドウで、プロセス定義を保存するファイルまたはデータベースを指定し、プロセス定義がデータベースで有効になる日付を指定します。「有効日」フィールドを空白にすると、変更を保存した直後から有効にすることができます。『Oracle Workflow API リファレンス』の「バージョン / 有効日付」を参照してください。

注意： 将来の有効日を指定して作業結果をデータベースに保存し、同じ Oracle Workflow Builder セッションでプロセスの変更を続けた後に、「ファイル」メニューから「保存」を選択すると、プロセス定義は以前に指定した有効日を使用して同じデータベースに自動的に保存されます。

3. Oracle Workflow では、作業内容を保存すると、不正なプロセス定義や不足している情報が自動的に検証され、該当するものがあれば「ワークフロー・エラー」検証ウィンドウに表示されるため注意してください。情報を修正してから作業内容を保存することも、作業内容を保存してからエラーを修正することもできます。後で参照するために、クリップボードに情報を貼り付けるには、「コピー」ボタンを使用して「ワークフロー・エラー」ウィンドウの情報をコピーします。エラーを修正せずに作業内容を保存する

と、次にこのプロセス定義をオープンしたときに「ワークフロー・エラー」ウィンドウが再表示されます。

4. 「ファイル」メニューから「ストアを閉じる」を選択し、現行データベースまたはファイル・データ・ストアへの接続を終了します。
5. 「ファイル」メニューから「終了」を選択し、Oracle Workflow Builder を終了します。

注意：「ファイル」メニューから「ストアを閉じる」および「終了」オプションを使用できるのは、「ナビゲータ」ウィンドウが現行ウィンドウの場合のみです。

▶ コマンド・プロンプトからの Oracle Workflow Builder の起動

Windows のアイコンをダブルクリックして Oracle Workflow Builder を起動するのではなく、コマンド・プロンプト (Windows98 では MS-DOS プロンプトと呼ぶ) からコマンドを入力して、接続先のファイルやデータベースを指定することもできます。

1. 「コマンドプロンプト」ウィンドウで、特定のワークフロー・データ・ファイルを指定して次のコマンドを入力し、Oracle Workflow Builder を起動します。

<filename.wft> はデータ・ファイルのフルパス名を表します。

```
wfblldr <filename.wft>
```

2. 特定のデータベース接続を指定して Oracle Workflow Builder を起動するには、コマンド・プロンプトから次のコマンドを入力します。 <username/password@connect> は、接続先データベースのアカウント情報を表します。

```
wfblldr -c <username/password@connect>
```

注意： Oracle Workflow Builder を Microsoft Windows NT 4.0 以上で実行する場合は、Windows の「エクスプローラ」上でワークフロー・データ・ファイル (.wft) をダブルクリックすると、そのファイルが自動的に開き、Oracle Workflow Builder が起動します。

注意： Oracle Applications に組み込まれている Workflow を使用している場合は、APPS スキーマを使用してデータベースに接続します。

3. Oracle Workflow Builder を起動し、データ・ストア内で指定した項目タイプをオープンするには、手順 1 または 2 の適切なコマンドを次のように修正します。 <item_type> は、オープンする項目タイプの内部名です。

```
-E <item_type>
```

たとえば、次のようになります。

```
wfbldr wfdemo.wft -E wfdemo
```

4. Oracle Workflow Builder を起動し、データ・ストア内で指定したプロセス・ダイアグラムをオープンするには、手順 1 または 2 の適切なコマンドを次のように修正します。`<item_type:process>` は、オープンする項目タイプとプロセスの内部名です。

```
-E <item_type:process>
```

たとえば、次のようになります。

```
wfbldr wfdemo.wft -E WFDEMO:NOTIFYAPPROVER
```

関連項目：

『Oracle Workflow 管理者ガイド』の「ワークフロー定義ローダーの使用」

4-25 ページ「ワークフロー・プロセスのショートカット・アイコンの作成」

クイック・スタート・ウィザードの概要

クイック・スタート・ウィザードを使用すると、ワークフロー・プロセスの設計をその場で開始できます。このウィザードでは、最初に `wftemplate.wft` というファイル（ワークフロー・プロセスの作成に必要なすべての必須オブジェクトの概略）がロードされ、プロセスのダイアグラムを作成する「プロセス」ウィンドウが表示されます。クイック・スタート・ウィザードを開始した後は、ボトムアップまたはトップダウンのアプローチを使用して、ワークフロー・プロセスの定義を完了します。

▶ クイック・スタート・ウィザードの使用

1. 「ファイル」メニューから「クイック・スタート・ウィザード」を選択します。

Workflow Quick Start Wizard

Please provide the following information for an automatic definition generation.

New Item Type

Internal Name: NEW_TYPE

Display Name: New Item Type

Persistence Type: Temporary

Number of Days: 0

New Process

Internal Name: NEW_PROCESS

Display Name: New Process

OK Cancel

2. 「Workflow クイック・スタート・ウィザード」ウィンドウに、次の必須情報の入力を求めるプロンプトが表示されます。

- 新規項目タイプ

- 内部名： 大文字のみを使用し、半角英数字で8文字以内の内部名を指定します。すべての Oracle Workflow API、SQL スクリプトおよび PL/SQL プロシージャでは、項目タイプの識別時に内部名が参照されます。

注意： 定義された項目タイプの内部名を更新するには、wfchitt.sql という特別な SQL スクリプトを使用する必要があります。『Oracle Workflow 管理者ガイド』の「wfchitt.sql」を参照してください。

注意： 内部名には、コロン (:) や空白を使用しないでください。

- 表示名： 項目タイプを表す変換可能な表示名を入力します。

- 維持タイプ： 項目タイプの実行ステータス情報として「一時」または「永続」を指定します。
 - 日数：「維持タイプ」が「一時」の場合は、項目タイプのインスタンスが終了してから、実行ステータス情報が削除可能になるまでの日数を指定します。3-4 ページの「維持タイプ」を参照してください。
- 新規プロセス
 - 内部名： 内部名を大文字のみで指定します。

注意： 定義されたアクティビティの内部名を更新するには、wfchact.sql という特別な SQL スクリプトを使用する必要があります。『Oracle Workflow 管理者ガイド』の「wfchact.sql」を参照してください。

注意： 内部名には、コロン (:) や空白を使用しないでください。

- 表示名： プロセス・アクティビティを表す変換可能な表示名を入力します。表示名は、「プロセス」ウィンドウのタイトル・バーにも表示されます。
3. クイック・スタート・ウィザードにより、次の処理が実行されます。
 - 「ナビゲータ」ウィンドウに「タイトルなし-n」というデータ・ストアが作成されます。
 - 「Workflow クイック・スタート・ウィザード」ウィンドウに入力した情報が使用され、データ・ストアに新しい項目タイプとプロセス・アクティビティが作成されます。
 - 作成するプロセスに標準のアクティビティを組み込めるように、標準項目タイプが新しいデータ・ストアにロードされます。
 - 定義した新しいプロセス・アクティビティの「プロセス」ウィンドウが開きます。「プロセス」ウィンドウには、「開始アクティビティ」および「終了アクティビティ」が表示されます。
 4. 次の2つの方法のどちらかを使用して、プロセスの定義をカスタマイズできます。
 - ボトムアップ・デザイン： ワークフロー・ダイアグラムを作成する前に、まずアクティビティを作成し、それらのアクティビティでサポートしているすべてのオブジェクトを作成します。2-9 ページの「ボトムアップによるプロセス定義の作成」を参照してください。
 - トップダウン・デザイン： ワークフロー・ダイアグラムの作成に必要な、最低限の情報が含まれているアクティビティを作成します。後で各アクティビティおよびサポートしているオブジェクトに戻って、具体的な内容を詳しく記述できます。2-11 ページの「トップダウンによるプロセス定義の作成」を参照してください。

各種リリースのサーバーでの Oracle Workflow Builder の使用

Oracle Workflow Builder リリース 2.6.3 は、Oracle Applications に組み込まれている Workflow Server リリース 11i の全リリースだけでなく、Oracle Workflow Server スタンドアロン版のリリース 2.6.3、リリース 2.6.2、リリース 2.6.1、リリース 2.6 およびリリース 2.5 とも互換性があります。

- ビジネス・イベント・システムのコンポーネントや外部 Java 関数アクティビティなど、リリース 2.6 で導入された新機能が含まれているワークフロー・プロセス定義を作成、表示および変更できます。Oracle Workflow Builder では、Oracle Workflow Server リリース 2.6 がインストールされているデータベースに対して、これらのプロセス定義をアップロード / ダウンロードできます。
- リリース 2.6 の機能を利用しない場合は、リリース 2.5 の機能だけが含まれるワークフロー・プロセスを作成、表示および変更できます。Oracle Workflow Builder では、Oracle Workflow Server リリース 2.5 がインストールされているデータベースに対して、これらのプロセス定義をアップロード / ダウンロードできます。

組み込み版 Server リリース 2.6、またはスタンドアロン版 Server リリース 2.6.3、2.6.2 またはリリース 2.6.1 で Oracle Workflow Builder リリース 2.6.3 を使用する

Oracle Applications に組み込まれている Workflow Server リリース 2.6、あるいはスタンドアロン版 Oracle Workflow Server リリース 2.6.3、リリース 2.6.2 またはリリース 2.6.1 で Oracle Workflow Builder を使用している場合は、現在使用可能な機能をすべてワークフロー・プロセスで使用できます。これらのプロセス定義をデータベースに保存しておけば、データベースのプロセス定義をオープンして表示または変更することができます。

また、Oracle Workflow Builder リリース 2.5、リリース 2.6、リリース 2.6.1 またはリリース 2.6.2 を使用して作成した既存のプロセス定義をオープンし、Oracle Applications に組み込まれている Workflow Server リリース 2.6、またはスタンドアロン版 Oracle Workflow Server リリース 2.6.3 がインストールされているデータベースに保存することもできます。

スタンドアロン版 Server リリース 2.6 で Oracle Workflow Builder リリース 2.6.3 を使用する

Oracle Workflow リリース 2.6 スタンドアロン版で Oracle Workflow Builder を使用している場合は、現在使用可能な機能のほとんどをワークフロー・プロセスで使用できます。ただし、「イベント・パラメータ」選択肢コードは使用しないでください。この機能は、Oracle Applications に組み込まれている Workflow リリース 2.6 で追加されましたが、現在は「イベント・プロパティ」選択肢タイプに関連付けられています。「イベント・パラメータ」選択肢コードは、カスタム・アクティビティでは使用しないでください。また、「イベント・パラメータ」プロパティは、次の標準ワークフロー・アクティビティで選択しないでください。

- イベント・パラメータの取得
- イベント・パラメータの設定
- イベント・パラメータの比較

Oracle Workflow Builder リリース 2.6 を使用して作成した既存のプロセス定義をオープンし、リリース 2.6 のコンポーネントだけを使用して表示または変更し、Oracle Workflow Server リリース 2.6 スタンドアロン版がインストールされているデータベースに保存することができます。

また、リリース 2.6 のコンポーネントだけを使用して、新しいワークフロー・プロセスを作成することもできます。ただし、Oracle Workflow Builder リリース 2.6.3 の標準項目タイプには、リリース 2.6 のコンポーネントが一部含まれます。Oracle Workflow Server リリース 2.6 スタンドアロン版がインストールされているデータベースに新しいプロセス定義を保存する場合は、次の手順を実行します。

1. ワークフロー・プロセス定義を新規作成します。
2. 標準項目タイプをデータ・ストアから強制的に削除します。
3. Oracle Workflow Server スタンドアロン版リリース 2.6 がインストールされているデータベースにプロセス定義を強制的に保存します。
4. データベースからこのプロセス定義を再度オープンします。このプロセス定義には、現在リリース 2.6 の標準項目タイプが含まれています。
5. リリース 2.6 のベース・コンポーネントだけを使用して、ワークフロー・プロセスの定義を続行します。

スタンドアロン版 Server リリース 2.5 または組込み版 Server リリース 2.5 で Oracle Workflow Builder リリース 2.6.3 を使用する

Oracle Workflow リリース 2.5 スタンドアロン版または Oracle Applications に組み込まれている Workflow Server リリース 2.5 で Oracle Workflow Builder を使用している場合は、リリース 2.5 の機能だけをワークフロー・プロセスで使用してください。リリース 2.6 で導入された次の新機能は使用しないでください。

- 「イベント」アクティビティ
- イベント・タイプの項目属性
- 「外部 Java」関数アクティビティ

Oracle Workflow Builder リリース 2.5 を使用して作成した既存のプロセス定義をオープンし、リリース 2.5 のコンポーネントだけを使用して表示または変更し、Oracle Workflow Server リリース 2.5 がインストールされているデータベースに保存することができます。

また、リリース 2.5 のコンポーネントだけを使用して、新しいワークフロー・プロセスを作成することもできます。ただし、Oracle Workflow Builder リリース 2.6.3 の標準項目タイプには、リリース 2.6 のコンポーネントが一部含まれます。Oracle Workflow Server リリース 2.5 がインストールされているデータベースに新しいプロセス定義を保存する場合は、次の手順を実行します。

1. ワークフロー・プロセス定義を新規作成します。
2. 標準項目タイプをデータ・ストアから強制的に削除します。
3. Oracle Workflow Server リリース 2.5 がインストールされているデータベースにプロセス定義を強制的に保存します。
4. データベースからこのプロセス定義を再度オープンします。このプロセス定義には、現在リリース 2.5 の標準項目タイプが含まれています。

「項目タイプの定義」 Web ページ

Web ベースの「項目タイプの定義」 ページを使用すると、Oracle Workflow データベースに格納されているワークフロー定義への個別アクセスが可能になります。このページでは、特定の項目タイプに関連した属性、プロセス、通知、関数、イベント、メッセージおよび選択肢タイプの詳細を表示して、ワークフロー・プロセスを表示したり、その構成を検討できます。

項目タイプ定義を表示するには、最初に「項目タイプの検索」 Web ページを使用して項目タイプを問い合わせます。項目タイプは、有効な日付および時刻に基づいて問合せできます。

「項目タイプの定義」 ページが表示されます。情報は、2つのフレームに表示されます。各フレームは Oracle Workflow Builder のようにモデル化されており、内容を簡単に効率よく検討できます。左のフレームには、項目タイプ定義に含まれるすべてのオブジェクトが、展開可能なナビゲータ・ツリーに表示されます。右のフレームには、ナビゲータ・ツリーで選択したオブジェクトの詳細が表示されます。また、いつでも左右どちらかのフレームを選択し、そのフレーム内のすべての情報を Web ブラウザを使用して印刷できます。

▶ 項目タイプの問合せ

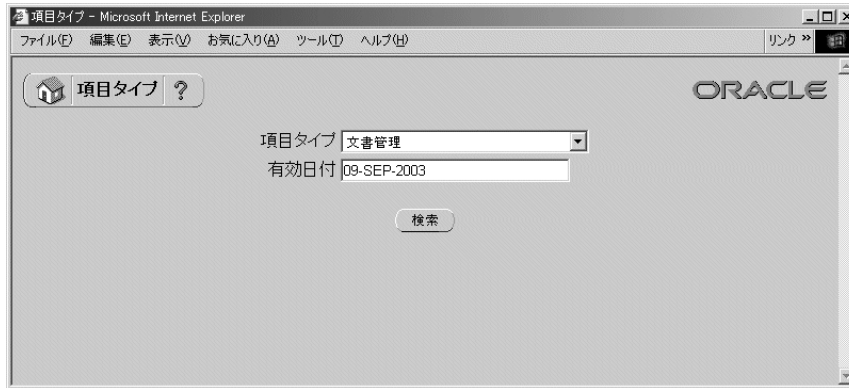
1. Web ブラウザに次の URL を入力します。

```
<webagent>/wf_item_definition.find_item_type
```

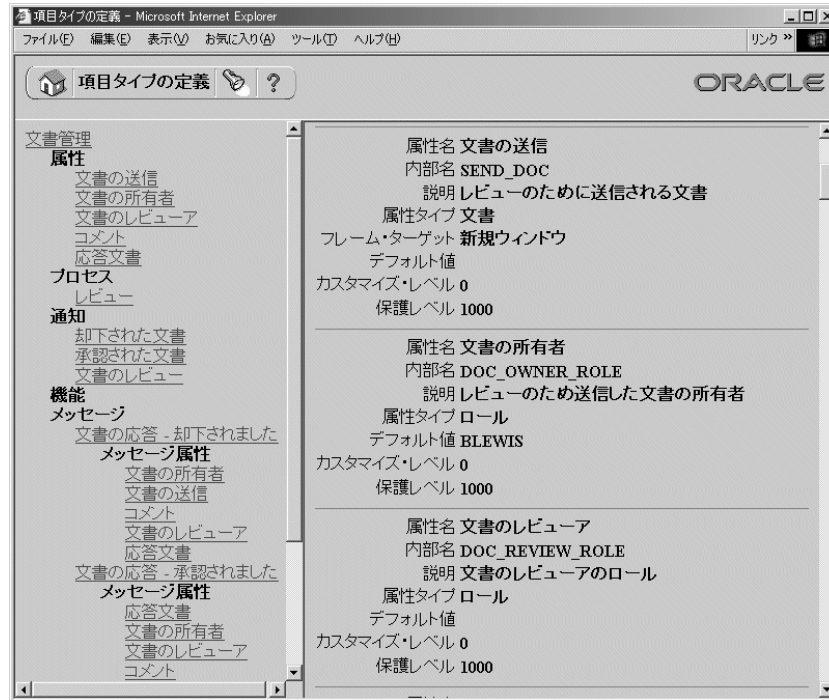
<webagent> は、Web サーバーで Oracle Workflow 用に構成された Web エージェントのベース URL に置き換えてください。『Oracle Workflow 管理者ガイド』の「グローバル・ユーザー設定の設定」を参照してください。

注意： この URL はセキュリティ画面にアクセスするため、現行 Web セッションで有効なユーザーとしてログオンしていない場合は、ページが表示される前に有効なユーザーとしてのログオンを求めるプロンプトが表示されます。

注意： 「項目タイプの検索」 Web ページには、Oracle Workflow ホーム・ページからもアクセスできます。『Oracle Workflow 管理者ガイド』の「Oracle Workflow ホーム・ページへのアクセス」を参照してください。



2. 「項目タイプ」ドロップ・ダウン・フィールドを使用して、項目タイプを選択します。
3. 「ユーザー設定」Web ページの「日付書式」フィールドで指定した書式を使用して、表示する項目タイプ定義の有効日時を指定します。『Oracle Workflow ユーザーズ・ガイド』の「ユーザー設定項目の設定」を参照してください。
4. 「検索」を選択して「項目タイプの定義」Web ページに項目タイプを表示します。



▶ 項目タイプの定義の検討

1. 「項目タイプの定義」Web ページには、2つのフレームが表示されます。左のフレームには、Oracle Workflow Builder のナビゲータ・ツリーと同様の階層形式で、項目タイプ定義のコンポーネントが表示されます。右のフレームには、各コンポーネントの詳細が表示されます。
2. 左側のフレームにあるいずれかのコンポーネントのリンクをクリックします。そのコンポーネントの詳細が右側のフレームに表示されます。

ワークフロー・プロセスのコンポーネントの定義

この章では、Oracle Workflow Builder を使用して、ワークフロー・プロセス・ダイアグラムの構成に必要なコンポーネントを定義する方法について説明します。

ワークフロー・プロセスのコンポーネント

作成するワークフロー・プロセスによっては、プロセスを構成する次のタイプのコンポーネントのすべてまたは一部を定義する必要があります。

- 項目タイプ
- 選択肢タイプ
- メッセージ
- アクティビティ
- 属性
- ロール

項目タイプ

項目タイプは、ワークフロー・プロセスを構成するコンポーネントの分類です。プロセス用に作成するコンポーネント（関数アクティビティやメッセージなど）は、すべて特定の項目タイプに関連付ける必要があります。多くの場合、項目タイプを定義して、ワークフロー・プロセスで管理する項目を記述すると理解しやすくなります。たとえば、購買申請を項目タイプとして定義し、特定の ID 番号で識別される購買申請をその項目タイプの項目にすることができます。3-6 ページの「[項目タイプの作成](#)」を参照してください。

項目タイプ属性

項目タイプ属性は、特定の項目タイプに関連付けられたプロパティです。項目タイプ属性はグローバル変数として機能し、プロセス内のすべてのアクティビティから参照、更新できます。通常、項目タイプ属性は、ワークフロー・プロセスを完了させるために必要な項目の情報を提供します。たとえば、「ワークフロー・デモンストレーション」という項目タイプには、「購買申請金額」という項目タイプ属性があります。この例で、「購買承認申請」プロセスのアクティビティでは、選択された承認者にその金額の購買申請を承認する権限があるかどうかを判別するために、この項目タイプ属性の値が必要です。

関数アクティビティと同様に、アプリケーションでも、Oracle Workflow Engine API を使用して項目タイプ属性を参照したり設定できます。項目タイプに必要な数だけ、項目タイプ属性を定義して保守できます。プロセス内のアクティビティに必要な情報や、通知メッセージでの送信が必要な情報は、すべて項目タイプ属性として定義する必要があります。3-44 ページの「[メッセージ属性の定義](#)」を参照してください。

関連項目：

『Oracle Workflow 管理者ガイド』の「項目属性」

属性タイプ

次のように、属性は 10 のタイプに分かれています。タイプによって、使用可能な値や属性の使用方法が決まります。

- テキスト： 属性値はテキストの文字列です。
- 数値： 属性値は、指定したオプションの書式マスク付きの数値です。
- 日付： 属性値は、指定したオプションの書式マスク付きの日付です。
- 選択肢： 属性値は、指定した選択肢タイプの選択肢コードの値の 1 つです。
- フォーム： 属性値は、Oracle Applications の内部フォーム機能名とオプションのフォーム機能パラメータです。この属性タイプは、Oracle Workflow のスタンドアロン版には関係ありません。

通知メッセージに、メッセージ属性としてフォーム・タイプの属性を含める場合、「通知の詳細」 Web ページから表示する通知には添付フォーム・アイコンが表示され、参照されているフォームまでドリルダウンできます。『Oracle Applications 開発者ガイド』の「メニューおよび機能セキュリティの概要」を参照してください。

- URL： 属性値は、ネットワーク位置の Universal Resource Locator (URL) です。通知メッセージで、メッセージ属性として URL 属性を参照すると、「通知の詳細」 Web ページから、または HTML 形式の電子メールとして表示した通知には、URL 属性で指定した URL へのアンカーが表示されます。ユーザーはこの URL にアクセスして、アクティビティを完了したり、アクティビティ関連の追加情報を表示できます。
- 文書： 属性値は添付文書です。デフォルト値フィールドで、次のタイプの文書を指定できます。
 - PL/SQL 文書： データベースのデータを文字列として表す文書で、PL/SQL プロシージャから生成されます。
 - PL/SQL CLOB 文書： データベースのデータをキャラクタ・ラージ・オブジェクト (CLOB) として表す文書で、PL/SQL プロシージャから生成されます。CLOB には、プレーン・テキスト、HTML、Adobe Acrobat PDF (Portable Document Format) 文書、Microsoft RTF (Rich Text Format) 文書を含めることができ、データベースのバージョンが Oracle9i Database 以降の場合は、Base64 にエンコードされたバイナリ・データも含めることができます。
 - PL/SQL BLOB 文書： データベースのデータをバイナリ・ラージ・オブジェクト (BLOB) として表す文書で、PL/SQL プロシージャから生成されます。BLOB には、イメージやバイナリ・データとして格納されたその他のアプリケーション・ファイルを含めることができます。

関連項目：

3-13 ページ「[文書属性の定義](#)」

- **ロール：** 属性値はロールの内部名です。通知メッセージにロール・タイプのメッセージ属性が含まれる場合、属性は自動的にロールの表示名に設定されるため、ロールの内部名と表示名について別々の属性を保守する必要がなくなります。また、Web ブラウザから通知を表示する場合は、ロールの表示名がそのロールの電子メール・アドレスを示すハイパーテキスト・リンクになります。属性のデフォルト値を設定するには、最初にデータベースからロールをロードする必要があります。4-26 ページの「**ロール**」を参照してください。
- **属性：** 属性値は、プロセスで参照を維持する別の既存項目タイプ属性の内部名です。
- **イベント：** 属性値は、標準 WF_EVENT_T 構造のビジネス・イベント・システムのイベント・メッセージです。『Oracle Workflow API リファレンス』の「イベント・メッセージ構造」を参照してください。

注意： イベント・メッセージをイベント・タイプの項目属性で保存すると、URL タイプの項目属性を作成し、イベント・データを参照する URL 属性値を設定することにより、そのイベント・メッセージ内のイベント・データにアクセスできます。『Oracle Workflow API リファレンス』の「SetItemAttribute」を参照してください。

維持タイプ

項目タイプを定義するときに、その維持タイプも指定する必要があります。維持タイプにより、項目タイプのインスタンスごとに維持される実行ステータス情報の期間を制御します。「維持」を「永続」に設定すると、ランタイム・ステータス情報はプロシージャ WF_PURGE.TotalPerm() をコールして情報を意図的に削除するまで、無期限に保存されます。

項目タイプの「維持」を「一時」に設定した場合は、維持日数（「*n*」）も指定する必要があります。「一時」項目タイプの各インスタンスの実行ステータス情報は、その完了日以後少なくとも *n* 日間は維持されます。*n* 日間維持された後は、WF_PURGE API のいずれかを使用して項目タイプのランタイム・ステータス情報を削除できます。『Oracle Workflow API リファレンス』の「WF_PURGE」を参照してください。

項目タイプの「維持」を「同期」に設定すると、その項目タイプのインスタンスは、項目キー #SYNCH が指定された状態で強制同期プロセスとして実行されます。強制同期プロセスは、単一 SQL セッション内で開始および終了し、データベース表に対して挿入または更新を行いません。強制同期プロセスではランタイム・ステータス情報が保持されないため、通常は、「同期」維持タイプのプロセスを削除する必要はありません。ただし、同期モードでプロセスを実行するときに、テストまたはデバッグを目的として一意の項目キーを使用した場合は、プロセス・インスタンスのランタイム・ステータス情報が保持されます。この情報を削除するには、項目タイプを「維持」から「一時」に変更してから、任意の WF_PURGE API を実行します。実行したら、項目タイプを「一時」から「同期」に戻します。『Oracle Workflow API リファレンス』の「同期プロセス、非同期プロセスおよび強制同期プロセス」および「WF_PURGE」、『Oracle Workflow 管理者ガイド』の「パフォーマンス改善のためのページ」を参照してください。

注意： Oracle Applications に組み込まれている Workflow を使用している場合、不要な項目タイプのランタイム・ステータス情報を削除するには、「ワークフローの不要ランタイム・データのパージ」コンカレント・プログラムも使用する必要があります。このコンカレント・プログラムの実行ファイル名は Oracle Workflow Purge Obsolete Data で、短縮名は FNDWFPR です。『Oracle Workflow API リファレンス』の「ワークフローの不要ランタイム・データのパージ」を参照してください。

項目タイプのセレクト関数

項目タイプに複数の実行可能なプロセス・アクティビティが関連している場合、または今後関連付ける予定の場合は、特定の状況でどのプロセス・アクティビティを実行するかを判別する PL/SQL 関数を定義します。たとえば、同じ項目タイプに 2 つの異なる購買承認申請プロセス・アクティビティが関連付けられている場合があります。Oracle Workflow で実行されるプロセスは、購買申請方法と依頼元に応じて異なる可能性があります。このような場合は、セレクト関数により、特定の状況に適切なプロセスが判別されます。

また、プロセスの実行中に SQL セッションが割込みされた場合に、必要に応じて項目タイプのコンテキスト情報をリセットできるように、セレクト関数を通常のコールバック関数に拡張することもできます。これは、Oracle Applications の使用例で、「通知の詳細」Web ページから通知を表示し、その通知に関連付けられた他のフォームを起動する場合に特に重要です。Oracle Workflow では、Oracle Application Object Library 関数のセキュリティ・システムにフォームの起動が渡される前に、項目タイプに対するセレクト / コールバック関数が TEST_CTX モードでコールされて、Oracle Applications コンテキストがテストされます。TEST_CTX モードでは、セレクト / コールバック関数により、フォームの起動が適切かどうかの判断に必要なあらゆるロジックを実行できます。6-12 ページの「[項目タイプのセレクト関数またはコールバック関数の標準 API](#)」を参照してください。

外部文書の統合

文書は、組織運営に大きな影響力を持ちます。デジタル・メディアとワールド・ワイド・ウェブ (WWW) の急激な発展に伴い、印刷物以外のメディアなど様々な書式の電子文書が出現したため、組織はこれらの文書を管理する必要に迫られています。このような文書の情報は、管理および共有されて初めて価値を持ちます。

ワークフロー・プロセスでは、PL/SQL プロシージャにより生成された文書 (PL/SQL 文書、PL/SQL CLOB 文書または PL/SQL BLOB 文書) を添付できます。文書をワークフロー・プロセスに添付するには、事前定義済の項目属性または「文書」タイプのメッセージ属性で文書を参照します。3-3 ページの「[属性タイプ](#)」、3-8 ページの「[項目タイプまたはアクティビティ属性の定義](#)」および 3-44 ページの「[メッセージ属性の定義](#)」を参照してください。

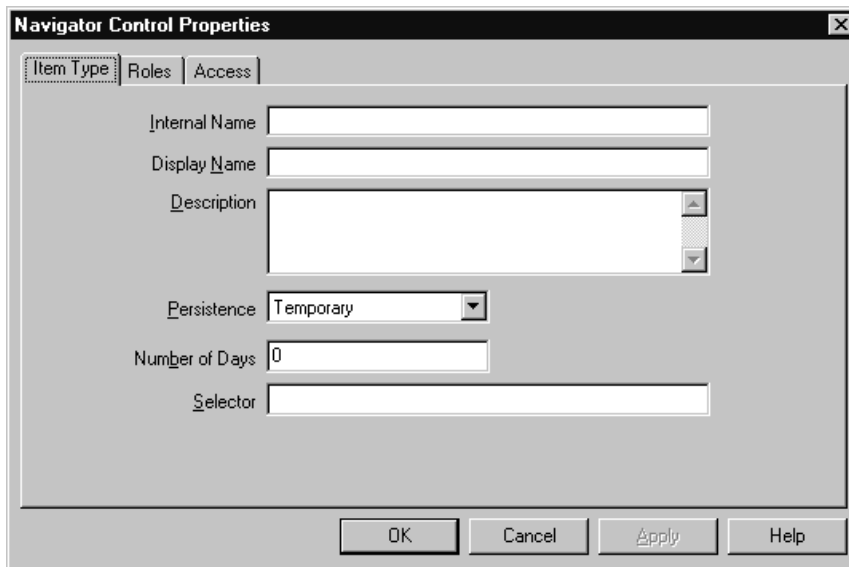
PL/SQL 文書、PL/SQL CLOB 文書または PL/SQL BLOB 文書の場合、項目属性またはメッセージ属性の値は、その文書の生成に使用された PL/SQL パッケージおよびプロシージャの

名前になります。PL/SQL プロシージャは、Oracle Workflow 標準インタフェースに従う必要があります。6-15 ページの「PL/SQL 文書の標準 API」を参照してください。

テキストまたは HTML が含まれた PL/SQL および PL/SQL CLOB 文書の場合、PL/SQL プロシージャで生成された文書は、通知のテキスト内に表示するか、添付ファイルとして含めることができます。その他のタイプの PL/SQL CLOB 文書の内容は、PL/SQL BLOB 文書の場合と同様に、添付ファイルとしてのみ含めることができます。

- 添付ファイルとして含められた PL/SQL CLOB 文書には、テキストまたは HTML に加えて、PDF または RTF 文書を含めることができ、データベースのバージョンが Oracle9i Database 以降であれば、Base64 にエンコードされたバイナリ・データも含めることができます。
- 添付ファイルとして含められた PL/SQL BLOB 文書には、イメージまたはバイナリ・データとして格納されたその他のアプリケーション・ファイルを含めることができます。

▶ 項目タイプの作成



1. まだデータ・ストアをオープンしていない場合は、「ファイル」メニューから「新規作成」を選択し、この新規項目タイプを定義する新規データ・ストアを作成します。次に「編集」メニューから「新規項目タイプ」を選択し、ナビゲータ・ツリー内で新規項目タイプを定義します。「項目タイプ」プロパティ画面が表示されます。
2. 各項目タイプには、すべて大文字の内部名（8 文字以内）があります。すべての Oracle Workflow API、SQL スクリプトおよび PL/SQL プロシージャでは、項目タイプの識別時に内部名が参照されます。

注意： 定義した項目タイプの内部名を更新するには、wfchitt.sql という特別な SQL スクリプトを使用する必要があります。このスクリプトは、設計時に項目タイプの内部名のエラーを訂正する場合にのみ使用してください。プロセスの実行中のインスタンスに関連する項目タイプの名前の変更には、使用しないでください。『Oracle Workflow 管理者ガイド』の「wfchitt.sql」を参照してください。

注意： 内部名には、コロン (:) や空白を使用しないでください。

3. 「表示名」には、変換可能で内部名より長くてもわかりやすい名前を入力します。また、項目タイプの説明も入力できます。
4. 維持タイプとして「一時」または「永続」を指定します。「維持タイプ」を「一時」に設定した場合は、項目のインスタンスが終了してから、実行ステータス情報が削除されるまでの日数を指定します。3-4 ページの「[維持タイプ](#)」を参照してください。
5. 項目タイプに複数のワークフロー・プロセスが関連している場合や、今後関連付ける予定の場合は、<package_name>.<procedure_name> 構文を使用してセレクト関数を指定できます。セレクト関数は PL/SQL のストアド・プロシージャで、この項目タイプに対してワークフローが起動された場合に、ワークフロー・エンジンで実行される特定のプロセス定義を自動的に識別します。また、ワークフロー・エンジンにより新規データベース・セッションが確立されてアクティビティが実行されるたびに、コンテキスト情報がリセットされるように、セレクト関数を一般コールバック関数に拡張することもできます。6-12 ページの「[項目タイプのセレクト関数またはコールバック関数の標準 API](#)」を参照してください。
6. 「適用」を選択して変更内容を保存します。
7. 「ロール」タブを選択し、この項目タイプにアクセスできるロールを指定します。（この機能は、今後のリリースでサポートされます。）
8. 「アクセス」タブを選択し、この項目タイプのアクセス・レベルとカスタマイズ・レベルを設定します。3-18 ページの「[オブジェクトへのアクセス許可の設定](#)」を参照してください。
9. 「適用」を選択して変更内容を保存するか、「OK」を選択し、変更内容を保存してプロパティ画面を閉じるか、「キャンセル」を選択し、変更を取り消してプロパティ画面を閉じます。
10. 作成した項目タイプを表す 2 次ブランチがナビゲータ・ツリーに表示されます。ナビゲータ・ツリーで項目タイプをダブルクリックするか、項目タイプを選択して「編集」メニューから「プロパティ」を選択すると、いつでもこの項目タイプのプロパティを検討または編集できます。

11. プロセス内でグローバル変数として使用する項目タイプ属性を、必要な数だけ定義します。これらの項目タイプ属性を使用して、作成した関数、通知またはイベントの各アクティビティとの間で値を渡すことができます。3-8 ページの「[項目タイプまたはアクティビティ属性の定義](#)」を参照してください。

関連項目：

2-9 ページ「[プロパティ画面の「編集」ボタンの使用](#)」

➤ **項目タイプまたはアクティビティ属性の定義**

1. 項目タイプ属性を作成するには、ナビゲータ・ツリーで項目タイプを選択し、「編集」メニューから「新規属性」を選択します。

The screenshot shows the 'Navigator Control Properties' dialog box. The 'Attribute' tab is selected. The 'Item Type' field contains 'Document Management'. The 'Type' dropdown is set to 'Text'. The 'Default' section has 'Type' set to 'Constant'. The 'Value' field is empty. The dialog includes 'OK', 'Cancel', 'Apply', and 'Help' buttons.

アクティビティ属性を作成するには、ナビゲータ・ツリーでアクティビティを選択し、「編集」メニューから「新規属性」を選択します。

どちらの場合も、「属性」プロパティ画面が表示されます。

2. 内部名をすべて大文字で空白を入れずに入力します。すべての Oracle Workflow API、SQL スクリプトおよび PL/SQL プロシージャでは、属性の識別時に内部名が参照されます。

注意： 定義した属性の内部名を更新するには、wfchita.sql および wfchacta.sql という特別な SQL スクリプトを使用する必要があります。これらのスクリプトは、設計時に属性の内部名のエラーを訂正する場合にのみ使用してください。プロセスの実行中のインスタンスに関連する属性の名前の変更には、使用しないでください。『Oracle Workflow 管理者ガイド』の「wfchita.sql」と「wfchacta.sql」を参照してください。

注意： 内部名には、コロン (:) や空白を使用しないでください。

3. 表示名を入力します。これは、ナビゲータ・ツリーに表示される名前です。
4. オプションの説明を入力します。
5. 属性のデータ型を選択します。アクティビティ属性を定義する場合であれば、フォーム、URL および文書のデータ型は関係ありません。
6. 属性のデータ型に応じて、次のデフォルト値の情報を指定します。

- **テキスト:** テキスト属性の最大長と、オプションでデフォルトのテキスト文字列を指定します。
- **数値:** オプションで、数値の書式マスクとデフォルト値を指定します。
- **日付:** オプションで、日付の書式マスクとデフォルト値を指定します。
- **選択肢:** 値を取り出す事前定義済の選択肢タイプを選択します。その選択肢タイプから、デフォルト値の選択肢コードを選択します。
- **URL:** オプションで、ネットワーク位置への **Universal Resource Locator (URL)** を「デフォルト」の「値」フィールドに指定し、URL のフレーム・ターゲットを指定します。3-12 ページの「[URL 属性の定義](#)」を参照してください。

注意: 「フレーム・ターゲット」フィールドは、URL タイプのメッセージ属性のみに適用されます。項目タイプ属性やアクティビティ属性には使用されません。

- **フォーム:** この属性は、Oracle Applications に組み込まれている Workflow にのみ関係します。
「デフォルト」の「値」フィールドで、開発者フォーム機能名と引数文字列（フォーム機能のパラメータ）を指定します（いずれもオプション）。『Oracle Applications 開発者ガイド』の「メニューおよび機能セキュリティの概要」および 3-13 ページの「[フォーム属性の定義](#)」を参照してください。
- **文書:** 文書を識別する文字列（オプション）を「デフォルト値」の「値」フィールドに入力します。3-13 ページの「[文書属性の定義](#)」を参照してください。

注意: 「フレーム・ターゲット」フィールドは、文書タイプの属性には適用されません。文書属性の「フレーム・ターゲット」フィールドは、今後使用する目的で確保されています。

- **ロール:** ロール名を指定します。4-26 ページの「[ロール](#)」を参照してください。
 - **属性:** 既存の項目タイプ属性のリストから、プロセス内で参照を維持する項目タイプ属性の名前を選択して指定します。
 - **イベント:** 項目タイプ属性を定義している場合は、イベント属性のデフォルト値を指定できません。アクティビティ属性を定義している場合は、イベント項目タイプ属性だけをデフォルト値として指定できます。
7. 項目タイプ属性の場合、オプションのデフォルト値は、入力するか値リストから選択した定数になります。ただし、この定数は、実行時にトークンが置換されるテキスト文字列の場合もあります。

アクティビティ属性の場合は、オプションのデフォルト値として定数または項目タイプ属性を指定できます。デフォルトで項目タイプ属性のすべての値を取得する場合は、「デフォルト」の「値」フィールドで「項目属性」を選択し、隣りにあるドロップ・ダウン・フィールドを使用して項目タイプ属性を選択します。ここで選択する項目タイプ属性は、アクティビティ自体が関連付けられているのと同じ項目タイプに関連付ける必要があります。また、アクティビティ属性と同じデータ型の項目タイプ属性を選択する必要があります。

注意：「テキスト」タイプのアクティビティ属性には、すべての項目属性タイプとの互換性がありますが、他のアクティビティ属性タイプは、いずれも項目属性タイプと完全に一致させる必要があります。

注意：「選択肢」タイプの属性の場合、デフォルト値はその選択肢タイプに属する選択肢コードにする必要があります。

8. 「適用」を選択して変更内容を保存するか、「OK」を選択し、変更内容を保存してプロパティ画面を閉じるか、「キャンセル」を選択し、変更を取り消してプロパティ画面を閉じます。
9. 項目タイプ属性を定義している場合は、「アクセス」タブを選択し、この属性を変更できるアクセス・レベルを設定します。アクティビティ属性では、その親アクティビティのアクセス / 保護レベルが想定されます。3-18 ページの「[オブジェクトへのアクセス許可の設定](#)」を参照してください。
10. 「適用」を選択して変更内容を保存します。
11. 作成した項目タイプ属性は、ナビゲータ・ツリーの「属性」のブランチの下に表示されます。定義した関数アクティビティ属性は、ナビゲータ・ツリーでその定義対象となったアクティビティの下に表示されます。ナビゲータ・ツリーで属性をダブルクリックするか、属性を選択して「編集」メニューから「プロパティ」を選択すると、いつでもその属性のプロパティを検討または編集できます。

注意： これらの属性がナビゲータ・ツリーに表示される順序は、これらの属性の取出し元となる値リストに表示される順序に関連しています。ナビゲータ・ツリーのドラッグ・アンド・ドロップ機能を使用するか、属性を選択して「編集」メニューから「属性を上へ移動」や「属性を下へ移動」を選択して、属性セット内での属性の順序を変更できます。

関連項目：

2-9 ページ「[プロパティ画面の「編集」ボタンの使用](#)」

▶ URL 属性の定義

1. 「属性」プロパティ画面の「デフォルト」の「値」フィールドに、ネットワーク位置への Universal Resource Locator (URL) を指定します。URL には、定数または他の項目属性から戻される値を使用できます。
2. URL には、テキスト文字列の引数文字列を指定できます。また、URL タイプのメッセージ属性を定義している場合は、指定した引数文字列 (トークン) が他のメッセージ属性に置換されます。トークンから置換されるメッセージ属性では、定数値を指定したり、項目タイプ属性から返された値を参照したりできます。3-44 ページの「[メッセージ属性の定義](#)」および 3-51 ページの「[属性のトークン置換](#)」を参照してください。

引数文字列内の他のメッセージ属性についてトークンを置換するには、メッセージ属性を次のように指定します。

```
-&message_attr-
```

たとえば、次の文字列は arg1 および arg2 という 2 つの引数を持つ URL を表し、その引数が実行時にはメッセージ属性の値 msgattr1 および msgattr2 でそれぞれトークンを置き換えます。

```
http://www.oracle.com?arg1=-&msgattr1-&arg2=-&msgattr2-
```

注意： URL タイプのメッセージ属性を定義する場合、引数文字列に `-&#NID-` という特殊トークンを含めることもできます。このトークンはランタイム通知の通知 ID で置換されます。

3. URL 属性に引数文字列が含まれている場合、次の制限事項を厳守してください。
 - その引数文字列について、文書タイプの別の項目属性でトークンを置換することはできません。
 - その引数文字列について別のフォーム属性または URL 属性でトークンを置換できます。ただし、他の属性の引数文字列についてそれ以上トークンを置換することはできません。
4. 日付と時間を引数として URL に渡す場合は、TO_CHAR を使用して文字列を YYYY/MM/DD+HH24:MI:SS の書式にします。同様に、URL がコールする関数の中でも、TO_DATE を使用して対応する書式変換を行う必要があります。このような書式設定が必要なのは、マルチバイトのデータベースでは、書式 DD-MON-YYYY のうち MON (月) に当たる部分が URL 間で使用できない値に変換される可能性があるためです。
5. 操作の完了後に、「OK」を選択します。

▶ フォーム属性の定義

1. フォームの「属性」プロパティ画面の「デフォルト」の「値」フィールドで、開発者フォーム機能名とオプションで引数文字列（フォーム機能のパラメータ）を指定します。
2. デフォルト値は、次の書式で入力してください。

```
function_name:arg1=value1 arg2=value2 ...argN=valueN
```

argN の値は、引用符 (" ") で囲んだテキスト文字列か、または次のいずれかの方法で別の項目タイプ属性により置換されるトークンでもかまいません。&item_attr は項目タイプ属性の内部名を表します。

- `argN="&item_attr"`
- `argN="Value &item_attr"`

関連項目：

3-51 ページ「属性のトークン置換」

注意：「フォーム」タイプのメッセージ属性を定義する場合、引数文字列に &#NID という特殊トークンを含めることもできます。このトークンはランタイム通知の通知 ID で置換されます。

3. フォーム属性に引数文字列が含まれている場合は、次の制限事項を厳守してください。
 - *argN* の値について、「文書」タイプの別の項目属性でトークンを置換することはできません。
 - その引数文字列の値について別のフォーム属性または URL 属性でトークンを置換することはできますが、他の属性の引数文字列についてそれ以上トークンを置換することはできません。
4. 操作の完了後に、「OK」を選択します。

▶ 文書属性の定義

1. 「属性」プロパティ画面の「デフォルト」の「値」フィールドに、文書を識別する文字列を入力します。

文書属性について、次のタイプの文書を識別できます。

- PL/SQL 文書
- PL/SQL CLOB 文書
- PL/SQL BLOB 文書

2. PL/SQL 文書は、データベースのデータを文字列として表すもので、PL/SQL プロシージャにより生成されます。PL/SQL 文書のデフォルト値は、次のように指定します。

```
plsql:<procedure>/<document_identifier>
```

<procedure> は、PL/SQL パッケージおよびプロシージャの名前をピリオド (.) で区切ったもので置き換えます。<document_identifier> は、プロシージャに直接渡される PL/SQL 引数文字列に置き換えます。引数文字列では、文書を識別する必要があります。

注意： PL/SQL プロシージャは、標準 API 書式に準じてください。6-15 ページの「[PL/SQL 文書の標準 API](#)」を参照してください。

たとえば、次の文字列はプロシージャ `po_wf.show_req` から生成された PL/SQL 文書 `po_req:2034` を表しています。

```
plsql:po_wf.show_req/po_req:2034
```

注意： PL/SQL 文書の最大データ長は、32KB です。文書が 32KB を超える場合は、かわりに PL/SQL CLOB 文書を使用してデータを格納する必要があります。

PL/SQL 文書は、通知のメッセージ本文内に表示することも、通知への添付ファイルとして添付することもできます。

3. PL/SQL CLOB 文書は、データベースのデータをキャラクタ・ラージ・オブジェクト (CLOB) として表すもので、PL/SQL プロシージャから生成されます。PL/SQL CLOB 文書のデフォルト値は、次のように指定します。

```
plsqlclob:<procedure>/<document_identifier>
```

<procedure> は、PL/SQL パッケージおよびプロシージャの名前をピリオド (.) で区切ったもので置き換えます。<document_identifier> は、プロシージャに直接渡される PL/SQL 引数文字列に置き換えます。引数文字列では、文書を識別する必要があります。

注意： PL/SQL プロシージャは、標準 API 書式に準じてください。6-15 ページの「[PL/SQL 文書](#)」および 6-22 ページの「[PL/SQL BLOB 文書](#)」を参照してください。

たとえば、次の文字列はプロシージャ `po_wf.show_req_clob` から生成された PL/SQL CLOB 文書 `po_req:2036` を表しています。

```
plsqcllob:po_wf.show_req_clob/po_req:2036
```

テキストまたは HTML が含まれた PL/SQL CLOB 文書は、通知のメッセージ本文内に表示することも、通知への添付ファイルとして添付することもできます。

PL/SQL CLOB 文書では、メッセージ属性のトークンを置換できません。CLOB 内のテキストまたは HTML コンテンツは、PL/SQL プロシージャで生成された状態でメッセージ本文に印刷されます。

- CLOB 内でトークンを使用しないでください。
- 必要な書式は PL/SQL プロシージャで設定してください。

通知に添付された PL/SQL CLOB 文書には、PDF または RTF 文書を含めることができます。また、データベースのバージョンが Oracle9i Database 以降であれば、Base64 でエンコードされたバイナリ・データも含めることができます。そのためには、バイナリ・ラージ・オブジェクト (BLOB) として文書をデータベースに格納してから、CLOB を作成するために使用する PL/SQL プロシージャの一部としてその文書を CLOB に変換する必要があります。6-15 ページの「PL/SQL 文書の標準 API」を参照してください。

UTL_RAW.Cast_To_VARCHAR2 関数を使用して、PDF または RTF データを BLOB から CLOB に書き込むための VARCHAR2 データへ変換できます。データベースのバージョンが Oracle9i Database 以降の場合は、WF_MAIL_UTIL.EncodeBLOB プロシージャを使用してバイナリ・データを Base64 にエンコードすることもできます。『Oracle9i PL/SQL パッケージ・プロシージャおよびタイプ・リファレンス』の「UTL_RAW」および『Oracle Workflow API リファレンス』の「EncodeBLOB」を参照してください。

注意： Oracle8i Database では Base64 のエンコードがサポートされないため、Oracle8i Database を使用している場合は、WF_MAIL_UTIL.EncodeBLOB プロシージャが利用できません。また、PL/SQL CLOB 文書には PDF または RTF 文書以外のバイナリ・データを格納できません。この機能は、Oracle9i Database 以降を使用している場合に限られます。

ただし、Oracle8i Database と Oracle9i Database 以降の両方で UTL_RAW.Cast_To_VARCHAR2 関数がサポートされるので、どのデータベースのバージョンでも、添付される PL/SQL CLOB 文書内に PDF 文書と RTF 文書を格納できます。

4. PL/SQL BLOB 文書は、データベースのデータをバイナリ・ラージ・オブジェクト (BLOB) として表すもので、PL/SQL プロシージャから生成されます。PL/SQL BLOB 文書のデフォルト値は、次のように指定します。

```
plsqblob:<procedure>/<document_identifier>
```

<procedure> は、PL/SQL パッケージおよびプロシージャの名前をピリオド (.) で区切ったもので置き換えます。<document_identifier> は、プロシージャに直接渡される PL/SQL 引数文字列に置き換えます。引数文字列では、文書を識別する必要があります。

注意： PL/SQL プロシージャは、標準 API 書式に準じてください。6-15 ページの「[PL/SQL 文書の標準 API](#)」を参照してください。

たとえば、次の文字列はプロシージャ `po_wf.show_req_blob` から生成された PL/SQL BLOB 文書 `po_req:2038` を表しています。

```
plsqblob:po_wf.show_req_blob/po_req:2038
```

PL/SQL BLOB 文書では、メッセージ属性のトークンを置換できません。

PL/SQL BLOB 文書には、イメージや、PDF 文書、RTF 文書またはその他のタイプのファイルなど、バイナリ・データとして格納されたアプリケーション・ファイルを含めることができます。PL/SQL BLOB 文書は、通知に添付ファイルとして含めることができます。ただしこのような文書は、通知のメッセージ本文に表示することはできません。

5. PL/SQL、PL/SQL CLOB または PL/SQL BLOB 文書の文書識別子を動的に生成する場合は、文書識別子について他の項目タイプ属性でトークンを置換できます。項目属性の名前は、大文字で記述し、コロンで区切る必要があります。3-51 ページの「[属性のトークン置換](#)」を参照してください。

次に例を示します。

```
plsqli:po_wf.show_req/&ITEM_ATTR1:&ITEM_ATTR2
```

注意： 「文書」タイプのメッセージ属性を定義する場合、引数文字列に `&#NID` という特殊トークンを含めることもできます。このトークンはラシタイム通知の通知 ID で置換されます。

6. 操作の完了後に、「OK」を選択します。

▶ 項目タイプのコピー

1. ナビゲータ・ツリーで、コピーする項目タイプを選択します。
2. マウスの左ボタンを押しながら、項目タイプをコピー先のデータ・ストアまたはワークスペースヘドドラッグします。

「編集」メニューの「コピー」および「貼り付け」コマンドも使用できます。

3. この項目タイプを同じデータ・ストアにコピーすると、「項目タイプ」プロパティ画面に、その項目タイプの新規の内部名と表示名の入力を求めるプロンプトが表示されます。これは、項目タイプには一意の内部名と表示名を与える必要があるためです。操作の完了後に「OK」を選択します。

項目タイプをコピーすると、その項目タイプに関連付けられているコンポーネントもすべてコピーされるため注意してください。ほとんどのコンポーネントの内部名と表示名も一意である必要があるため、これらのコンポーネントに関しても、そのプロパティ画面に内部名と表示名の更新を求めるプロンプトが表示されます。

4. 同じ項目タイプの旧バージョンがすでに存在するデータ・ストアに項目タイプをコピーすると、コピー先データ・ストア内の項目タイプの旧バージョンは、コピー元の項目タイプのバージョンの内容で更新されます。

注意： 新規のストアに複数の項目タイプをドラッグする場合は、その順序が重要になります。たとえば、ある項目タイプが標準項目タイプのオブジェクトを参照する場合を考えます。その項目タイプと標準項目タイプを新規データ・ストアにコピーする場合は、標準項目タイプを新規データ・ストアにドラッグしてから、他方の項目タイプをドラッグする必要があります。そうしないと、その項目タイプには標準項目タイプに対する未解決の参照が含まれることとなります。

▶ 属性のコピー

1. ナビゲータ・ツリーで、コピーする属性を選択します。
2. マウスの左ボタンを押しながら、属性をコピー先となるコンポーネントのブランチヘドドラッグします。
3. 属性を同じ項目タイプに関連付けられているコンポーネントにコピーすると、その属性のプロパティ画面が表示されます。

その属性の一意の新しい内部名と表示名を入力します。

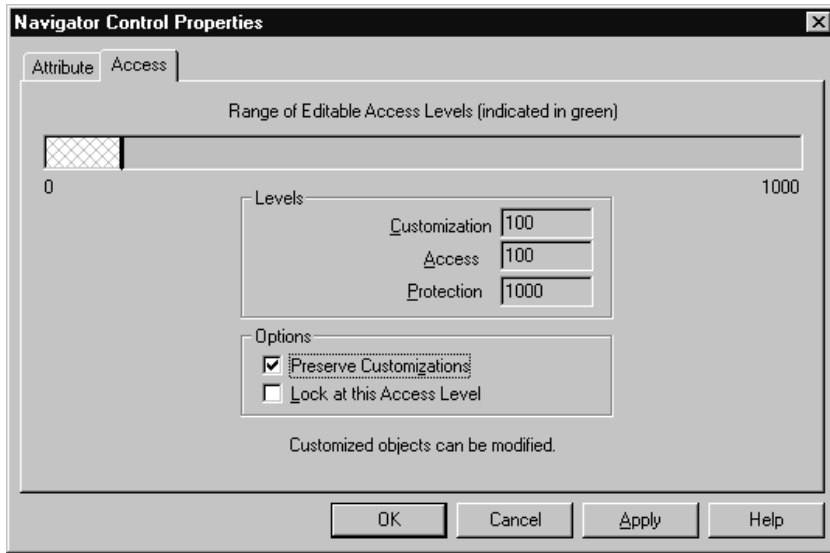
操作の完了後に「OK」を選択します。

注意： 「編集」メニューの「コピー」および「貼り付け」オプションも使用できます。

関連項目：

2-9 ページ [「プロパティ画面の「編集」ボタンの使用」](#)

オブジェクトへのアクセス許可の設定



「アクセス」タブの「編集可能なアクセス・レベルの範囲」インジケータ・バーには、オブジェクトを編集できるアクセス・レベルの範囲が相対表示されます。影付きの部分はオブジェクトを編集できるアクセス・レベルを表し、縦棒は現行のアクセス・レベルを表します。『Oracle Workflow 管理者ガイド』の「Oracle Workflow のアクセス保護の概要」を参照してください。

インジケータ・バーは、影付きの緑色の実線または影付きの緑色の実線と灰色の交差線の組合せとして表示できます。「ヘルプ」メニューの「Oracle Workflow Builder のバージョン情報」ダイアログ・ボックスで、「カスタマイズされたオブジェクトの変更を可能にする」チェック・ボックスのオン/オフによって、表示状態を次のように切り替えることができます。

- オン： 編集可能なアクセス・レベルの範囲を、緑色の実線と灰色の交差線による部分の組合せとして表示できます。灰色の交差線が示すレベルは、通常はカスタマイズ済みのオブジェクトを変更できませんが、Oracle Workflow Builder がアップロード・モードで動作しているため変更可能になっているレベルを表します。アップロード・モードは、変更するためにアクセスした保護対象のオブジェクトのみでなく、以前にカスタマイズしたオブジェクトも上書きして、Oracle Workflow Builder で編集結果を保存できることを意味します。
- オフ： 編集可能なアクセス・レベルの範囲は、緑色の実線部分として表示されます。これは、作業結果を保存すると、Oracle Workflow Builder はアップグレード・モードで動作することを示します。この場合、変更のためにアクセスした保護対象のオブジェ

クトの編集内容のみが保存され、以前にカスタマイズ済のオブジェクトはそのまま残ります。

この2つのモードは、ワークフロー定義ローダーのプログラムのアップグレードおよびアップロード動作と同じです。3-19 ページの「**オブジェクトのアクセス・レベルの設定**」および『Oracle Workflow 管理者ガイド』の「ワークフロー定義ローダーの使用」を参照してください。

▶ **オブジェクトのアクセス・レベルの設定**

1. プロパティ画面の「アクセス」タブを選択します。
2. 「オプション」フィールドの「カスタマイズを保持」および「このアクセス・レベルでロック」チェック・ボックスを使用して、このオブジェクトを変更できるアクセス・レベルを定義します。このフィールドでオンにするオプションは、「レベル」フィールドに表示される値に直接影響します。

次の表は、各オプションの様々な組合せをオンにした場合に、オブジェクトのカスタマイズ・レベルと保護レベルにどのように影響するかを示しています。この表は、オプションのアクセス・レベルが 100 に設定されていることを前提としています。

表 3-1

オンになっている チェック・ボックス	結果のレベル	オブジェクトの変更が許可されるレベル
なし： オブジェクトはいつでもだれでも更新できます。	カスタマイズ = 0、 アクセス = 100、 保護 = 1000	オブジェクトはどのレベル（0 ~ 1000）でも更新が可能です。
カスタマイズを保持： ワークフロー定義のアップグレード中は、カスタマイズされたオブジェクトの上書きを禁止します。	カスタマイズ = 100、 アクセス = 100、 保護 = 1000	アクセス・レベルが 100 ~ 1000 の場合にのみ、オブジェクトの更新が可能です。「カスタマイズされたオブジェクトの変更を可能にする」チェック・ボックスがオンの場合は、インジケータ・バーの灰色の交差線で表されるアクセス・レベル 0 ~ 99 でも、カスタマイズされたオブジェクトを更新できます。
このアクセス・レベルでロック： オブジェクトを現在のアクセス・レベルで保護し、オブジェクトのカスタマイズを許可しません。	カスタマイズ = 0、 アクセス = 100、 保護 = 100	アクセス・レベルが 0 ~ 100 の場合にのみ、オブジェクトの更新が可能です。

表 3-1 (続き)

オンになっている チェック・ボックス	結果のレベル	オブジェクトの変更が許可されるレベル
両方: オブジェクトは、それが保護されているレベルでのみ更新できます。	カスタマイズ = 100、 アクセス = 100、 保護 = 100	アクセス・レベルが 100 以外の場合には、オブジェクトを更新できません。「カスタマイズされたオブジェクトの変更を可能にする」チェック・ボックスがオンの場合は、インジケータ・バーの灰色の交差線で表されるアクセス・レベル 0 ~ 99 でも、カスタマイズされたオブジェクトを更新できます。

3. 「適用」 ボタンを選択して変更内容を保存します。

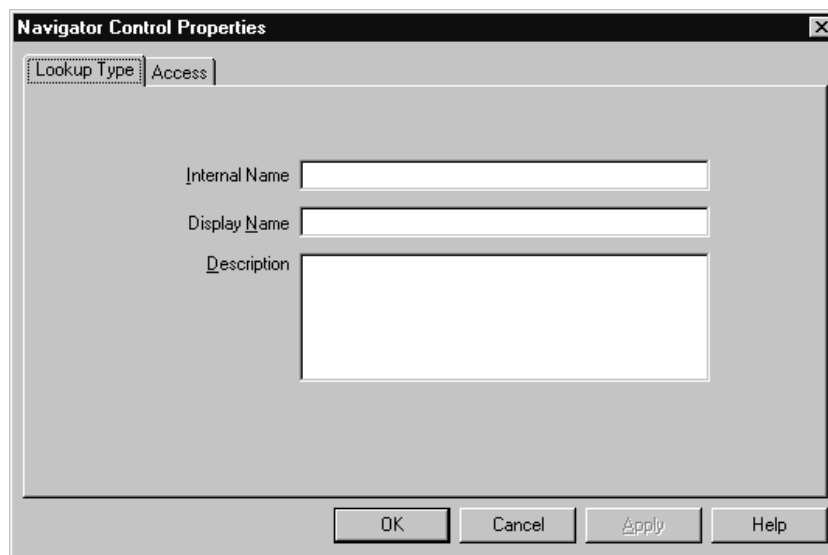
注意: オブジェクトは、ナビゲータ・ツリーに赤い小型のロック付きアイコンとして表示されます。これは、オブジェクトの編集が許可されていないアクセス・レベルで操作中の場合、つまり、アクセス・レベルが「編集可能なアクセス・レベルの範囲」インジケータ・バーのうち白い部分に含まれているときは、そのオブジェクトが読取り専用であることを示します。

選択肢タイプ

選択肢タイプは、静的な値リストです。このリストは、アクティビティ、項目タイプ、メッセージまたはアクティビティの属性から参照できます。たとえば、アクティビティは選択肢タイプで使用できる結果の値を参照し、メッセージ属性は選択肢タイプを参照して通知の実行者が使用可能な応答のリストを提供します。

選択肢タイプを定義してから、それを特定の項目タイプに関連付けます。ただし、選択肢タイプは項目タイプに固有のものではありません。このため、アクティビティまたは属性を作成する場合は、選択肢タイプが関連付けられている項目タイプに関係なく、現行のデータ・ストアのすべての選択肢タイプを参照できます。3-21 ページの「[選択肢タイプの作成](#)」を参照してください。

▶ 選択枝タイプの作成



1. ナビゲータ・ツリーから項目タイプを選択し、「編集」メニューから「新規選択枝タイプ」を選択します。「選択枝タイプ」プロパティ画面が表示されます。
2. 選択枝タイプには、すべて大文字で前後に空白を含まない「内部名」と、変換可能な「表示名」が付いています。すべての Oracle Workflow API、SQL スクリプトおよび PL/SQL プロシージャでは、選択枝タイプの識別時に内部名が参照されます。

注意： 定義した選択枝タイプの内部名を更新するには、wfchlut.sql という特別な SQL スクリプトを使用する必要があります。このスクリプトは、設計時に選択枝タイプの内部名のエラーを訂正する場合にのみ使用してください。プロセスの実行中のインスタンスに関連する選択枝タイプの名前の変更には、使用しないでください。『Oracle Workflow 管理者ガイド』の「wfchlut.sql」を参照してください。

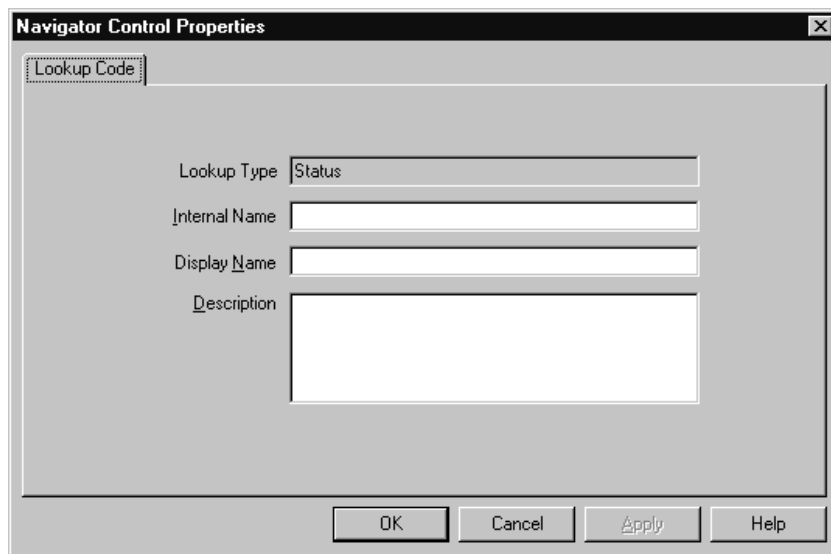
注意： 内部名には、コロン (:) や空白を使用しないでください。

選択枝タイプには任意で説明を加えることもできます。

3. 「アクセス」タブを選択し、この選択枝タイプを変更できるアクセス・レベルを設定します。3-18 ページの「[オブジェクトへのアクセス許可の設定](#)」を参照してください。

4. 「適用」を選択して変更内容を保存するか、「OK」を選択し、変更内容を保存してプロパティ画面を閉じるか、「キャンセル」を選択し、変更を取り消してプロパティ画面を閉じます。
5. 定義した選択肢タイプが、ナビゲータ・ツリーの「選択肢タイプ」のブランチの下に表示されます。ナビゲータ・ツリーで選択肢タイプをダブルクリックするか、選択肢タイプを選択して「編集」メニューから「プロパティ」を選択すると、いつでもこの選択肢タイプのプロパティを検討または編集できます。
6. 作成した選択肢タイプの選択肢コードを定義します。3-22 ページの「[選択肢タイプの選択肢コードの作成](#)」を参照してください。

▶ **選択肢タイプの選択肢コードの作成**



1. ナビゲータ・ツリーから選択肢タイプを選択し、「編集」メニューから「新規選択肢コード」を選択します。「選択肢コード」プロパティ画面が表示されます。
2. 選択肢コードの「内部名」と「表示名」を入力します。内部名の前後に空白を入れることはできません。オプションで説明を入力することもできます。すべての Oracle Workflow API、SQL スクリプトおよび PL/SQL プロシージャでは、選択肢コードの識別時に内部名が参照されます。

注意： 定義した選択肢コードの内部名を更新するには、wfchluc.sql という特別な SQL スクリプトを使用する必要があります。このスクリプトは、設計時に選択肢コードの内部名のエラーを訂正する場合のみ使用してください。プロセスの実行中のインスタンスに関連する選択肢コードの名前の変更には、使用しないでください。『Oracle Workflow 管理者ガイド』の「wfchluc.sql」を参照してください。

注意： 内部名には、コロン (:) や空白を使用しないでください。

3. 「適用」を選択して変更内容を保存するか、「OK」を選択し、変更内容を保存してプロパティ画面を閉じるか、「キャンセル」を選択し、変更を取り消してプロパティ画面を閉じます。
4. 定義した選択肢コードが、ナビゲータ・ツリーで作成した「選択肢タイプ」のブランチの下に表示されます。ナビゲータ・ツリーで選択肢コードをダブルクリックするか、選択肢コードを選択して「編集」メニューから「プロパティ」を選択すると、いつでもこの選択肢コードのプロパティを検討または編集できます。
5. 特定の選択肢タイプの選択肢コードを追加作成する場合は、手順 1 を繰り返します。

▶ 選択肢タイプのコピー

1. ナビゲータ・ツリーで、コピーする選択肢タイプを選択します。
2. 「編集」メニューの「コピー」および「貼り付け」オプションを使用して、選択肢タイプを任意の項目タイプにコピー・アンド・ペーストします。または、選択肢タイプを任意の項目タイプにドラッグ・アンド・ドロップします。
 - 選択肢タイプを同じ項目タイプに再度コピーした場合、または別のデータ・ストアの項目タイプにコピーしたときにその選択肢タイプがいずれかの項目タイプにすでに存在する場合は、プロパティ画面が表示されます。新しい選択肢タイプに対して一意の内部名と表示名を入力してください。操作の完了後に「OK」を選択します。
 - 選択肢タイプを別のデータ・ストアの項目タイプにコピーしたときに、その選択肢タイプがそのデータ・ストアのいずれかの項目タイプに存在しない場合は、元の名前と同じ内部名および表示名で新しい選択肢タイプがコピーされます。新しい名前を入力する必要はありません。

注意： 選択肢タイプをコピーすると、それに割り当てられている選択肢コードもコピーされます。

注意：「コピー」および「貼り付け」オプションを使用して、選択肢タイプを同じデータ・ストアの別の項目タイプにコピーすることはできません。ただし、ドラッグ・アンド・ドロップすることはできます。この場合、選択肢タイプは新しい項目タイプに移動します。

▶ **選択肢コードのコピー**

1. ナビゲータ・ツリーで、コピーする選択肢コードを選択します。
2. マウスの左ボタンを押しながら、選択肢コードをコピー先の選択肢タイプまでドラッグします。
3. 選択肢コードを同じ選択肢タイプに、またはこのコードがすでに存在する選択肢タイプへドラッグすると、マウス・ボタンを放したときに、プロパティ画面が表示されるため、新規の選択肢コードの一意の内部名と表示名を入力します。操作の完了後に「OK」を選択します。

注意：「編集」メニューの「コピー」および「貼り付け」オプションも使用できます。

メッセージ

ナビゲータ・ツリーの「メッセージ」のブランチには、現行の項目タイプに使用可能なワークフロー・メッセージがすべて表示されます。

通知アクティビティがワークフロー・プロセスのロールに送信するのがメッセージです。メッセージでは、ユーザーに応答やアクションを求めるプロンプトを表示して、プロセスの次のアクティビティを決定するための情報を得ることができます。ワークフロー・メッセージの受信者は実行者と呼ばれます。

各メッセージは、特定の項目タイプに関連付けられます。これにより、メッセージでは、その項目タイプの属性を、そのメッセージが送信された場合の実行時に置き換えられるトークンとして参照できます。

メッセージを定義するときに、そのメッセージで受信者に特別な応答値を要求するよう指定できます。その場合、ワークフロー・エンジンでは、この応答値を使用して、プロセス内の次の適格アクティビティに分岐する方法が決定されます。メッセージの件名と本文に項目タイプ属性を参照するメッセージ属性トークンを含めて、状況依存のコンテンツを持つメッセージを作成することができます。メッセージ関数を使用して、書式化されたメッセージ属性の表または通知履歴の表をメッセージの本文に含めることができます。文書全体またはURLを表すメッセージ属性を、通知メッセージに添付することも可能です。さらに、メッセージに固有の応答セクションを生成するメッセージ属性も作成できます。

メッセージを「通知」のブランチにドラッグし、そのメッセージを送信する新しい通知アクティビティを作成できます。また、メッセージを既存の通知アクティビティに直接ドラッグして、そのアクティビティで送られるメッセージを更新できます。

メッセージの結果

通知アクティビティのためのメッセージを作成する場合は、通知アクティビティの結果タイプが指定されているかどうかに注意する必要があります。結果タイプが指定されている場合は、通知アクティビティの結果として解釈される特別な応答を受信者に要求するようにメッセージを作成してください。ワークフロー・エンジンでは、その結果を使用して、プロセス内の次の「適格」アクティビティに分岐する方法が決定されます。

この特別な応答を求めるメッセージを作成するには、メッセージのプロパティ画面の「結果」タブに必要な事項を入力します。入力した情報に基づいて、RESULT という内部名を持つ特別な応答メッセージ属性がそのメッセージに作成されます。RESULT メッセージ属性はデータ型が「選択肢」であり、通知アクティビティの結果タイプと同じ選択肢タイプに設定されている必要があります。これにより、通知の実行者は、通知アクティビティで予測される結果のリストと合致する応答値候補のリストから選択できます。3-25 ページの「送信および応答メッセージ属性」を参照してください。

送信および応答メッセージ属性

メッセージの作成後に、そのメッセージに関して必要な数だけメッセージ属性を定義できます。メッセージ属性は、ナビゲータ・ツリーでメッセージの下に表示されます。

メッセージ属性のソース（送信または応答）により、メッセージ属性の使用方法が決まります。「送信」のソースでメッセージ属性を定義する場合は、トークンの置換のためにメッセージの件名または本文（あるいはその両方）にそのメッセージ属性を埋め込むことができます。さらに、通知の送信時にはメッセージにメッセージ属性を添付できます。

各メッセージ属性には、特定のデータ型があります。「送信」メッセージ属性の値は、定数または同じデータ型の項目タイプ属性により戻される値です。トークンの置換のために、メッセージ属性をメッセージの件名または本文に埋め込むには、件名またはテキスト本文内で書式 &MESGATTR を使用して、メッセージ属性の内部名を指定します。

注意：「文書」メッセージ属性は件名内でトークンの置換ができないため、メッセージの件名には「文書」タイプのメッセージ属性を埋め込まないでください。件名に埋め込まれた「文書」メッセージ属性は無視されます。

ただし、「文書」メッセージ属性は、トークンの置換のためにメッセージ本文に埋め込むことができます。

「応答」ソースを使用して定義されたメッセージ属性は、メッセージの応答セクションの構成要素となります。「応答」メッセージ属性には、受信者に対して応答入力を求めるプロンプトを表示するインストラクションが用意されています。「応答」メッセージ属性を定義する場合は、属性のデータ型を指定する必要があります。また、オプションで応答のデフォルト値を指定することもできます。デフォルト値には、定数または同じデータ型の項目タイプ属性から戻される値を使用できます。

通知の受信者に指定された Web ページまたはフォームを使用して通知に回答してもらいたい場合は、URL またはフォーム・タイプの「応答」メッセージ属性を定義します。このような場合、カスタムの Web ページまたはフォームに通知をクローズとしてマークする必要があります。また、その画面またはフォームには、通知への応答完了をワークフロー・エンジンに知らせるために、`CompleteActivity()` API のコールが含まれている必要があります。また、その他のタイプの「応答」メッセージ属性を定義することはできません。必要な応答値は、指定された Web ページまたはフォームに入力する必要があります。『Oracle Workflow API リファレンス』の「CompleteActivity」を参照してください。

Oracle Applications に組み込まれている Workflow で利用可能な Oracle Applications Framework 対応の「ワークリスト」画面を使用していて、URL またはフォーム・タイプの「応答」メッセージ属性を定義する場合は、「通知の詳細」画面に URL の応答リンクまたはフォームの応答アイコンのみが表示され、その他の応答フィールドは表示されません。ただし、通知の再割当てや詳細情報の要求は、これらのオプションが通知で定義されていれば、通常どおりに行うことができます。

関連項目：

『Oracle Workflow 管理者ガイド』の「メッセージ属性」

#WF_REASSIGN_LOV 属性

Oracle Applications に組み込まれている Workflow で利用可能な Oracle Applications Framework 対応の「ワークリスト」画面を使用している場合は、内部名 `#WF_REASSIGN_LOV` のメッセージ属性を使用して、メッセージ再割当て先のユーザーを指定することができます。通知が再割当てされるときに新しい受信者として許可する可能性があるすべてのユーザーをメンバーとするロールを作成し、`#WF_REASSIGN_LOV` 属性の値としてこのロールを割り当てます。このようにすると、通知の受信者が通知を再度割り当てるときに、このロールに属するユーザーのみが「再割当て」フィールドの値のリストに表示されます。

注意： Oracle Applications に組み込まれている Workflow で利用可能な Oracle Applications Framework 対応の「ワークリスト」画面を使用している場合は、「FND: 通知再割当てモード」プロファイル・オプションの設定に応じて、「通知の再割当て」画面に「再割当て」フィールドのかわりに「委任」フィールドまたは「譲渡」フィールドを含めることができます。`#HIDE_REASSIGN` 属性により、「委任」フィールドと「譲渡」フィールドは「再割当て」フィールドと同様に制御されます。『Oracle Workflow 管理者ガイド』の「FND: 通知再割当てモード」プロファイル・オプションの設定を参照してください。

`#WF_REASSIGN_LOV` 属性はロール・タイプであって、「送信」ソースを所有している必要があります。`#WF_REASSIGN_LOV` 属性の定数値として特定のロールを指定することも、値として項目タイプ属性を指定し、その項目タイプ属性を実行時に動的にロールに設定するロジックをワークフロー・プロセス内に組み込むこともできます。既存のロールが要件に合

わない場合は、実行時にアドホック・ロールを作成してそのロールが必要なユーザーを追加するロジックを使用するプロセス内に含めることができます。『Oracle Workflow API リファレンス』の「CreateAdHocRole」および「AddUsersToAdHocRole」を参照してください。

注意： Oracle Workflow の Web ページは、Oracle Applications Framework のユーザー・インタフェース形式に変換されます。Oracle Workflow のバージョンおよび適用したパッチに応じて、Oracle Workflow の Web ページは新しい形式または以前の形式で表示されます。現時点では、#WF_REASSIGN_LOV 属性がサポートされる新しい「ワークリスト」画面は、Oracle Applications に組み込まれている Workflow で利用できます。

関連項目：

『Oracle Workflow ユーザーズ・ガイド』の「他のユーザーへの通知の再割当て」（Oracle Applications の場合）

#HIDE_REASSIGN 属性

メッセージ属性に内部名 #HIDE_REASSIGN を指定すると、「通知の詳細」Web ページの「再割当て」ボタンを非表示にできます。ユーザーが「ワークリスト」Web ページで通知を表示すると、デフォルトで「通知の詳細」画面の応答セクションに「再割当て」ボタンが表示されます。ユーザーが通知を再割当てしないようにする場合は、#HIDE_REASSIGN 属性を追加して、「再割当て」ボタンを表示するか非表示にするかを制御できます。

注意： Oracle Applications に組み込まれている Workflow で利用可能な Oracle Applications Framework 対応の「ワークリスト」画面を使用している場合は、「FND: 通知再割当てモード」プロファイル・オプションの設定に応じて、「通知の詳細」画面に「再割当て」ボタンかわりに「委任」ボタンまたは「譲渡」ボタンを含めることができます。#HIDE_REASSIGN 属性により、「委任」ボタンと「譲渡」ボタンは「再割当て」ボタンと同様に制御されます。『Oracle Workflow 管理者ガイド』の「FND: 通知再割当てモード」プロファイル・オプションの設定を参照してください。

Oracle Applications に組み込まれている Workflow で利用可能な Oracle Applications Framework 対応の「ワークリスト」画面およびホーム・ページを使用している場合は、#HIDE_REASSIGN 属性によっても、拡張ワークリスト、個人ワークリストまたは Self-Service のホーム・ページ内の「再割当て」ボタンを使用して、通知を再割当てできるかどうかを制御できます。これらの画面で、既に #HIDE_REASSIGN 属性が設定されている通知をユーザーが再割当てしようとする、「再割当て」ボタンは表示されますが、エラー・メッセージが表示されます。

#HIDE_REASSIGN 属性は、テキスト・タイプまたは選択肢タイプのどちらかにする必要があります。「通知の詳細」画面の「再割当」ボタンを非表示にして拡張ワークリスト、個人ワークリストまたは Self-Service のホーム・ページ内から再割当てできないようにする場合は、この属性の値を Y に設定します。「通知の詳細」画面の「再割当」ボタンを表示して、拡張ワークリスト、個人ワークリストまたは Self-Service のホーム・ページ内から再割当てできるようにする場合は、この属性の値を N に設定します。

注意： #HIDE_REASSIGN 属性を選択肢タイプに定義して、標準項目タイプに用意されている事前定義済の「Yes/No」選択肢タイプに割り当てることをお勧めします。「Yes/No」選択肢タイプには、表示名が Yes と No という 2 つの選択肢コードが含まれており、Y の値と N の値をそれぞれ表しています。

- 特定のメッセージを使用する通知の再割当てを常に制限する場合は、値 Y を定数として指定します。
- 特定の場合にのみ再割当てを制限する場合は、値として項目タイプを指定します。次に、再割当てを許可するか制限するかを実行時に動的に決定し、それぞれ項目タイプ属性を Y または N に設定するロジックを、ワークフロー・プロセスに組み込みます。

関連項目：

『Oracle Workflow ユーザーズ・ガイド』の「他のユーザーへの通知の再割当て」

『Oracle Workflow ユーザーズ・ガイド』の「他のユーザーへの通知の再割当て」(Oracle Applications の場合)

『Oracle Workflow ユーザーズ・ガイド』の「拡張ワークリストの通知の表示」

『Oracle Workflow ユーザーズ・ガイド』の「個人ワークリストの通知の表示」

#HIDE_MOREINFO 属性

Oracle Applications に組み込まれている Workflow で利用可能な Oracle Applications Framework 対応の「ワークリスト」画面を使用している場合は、内部名 #HIDE_MOREINFO のメッセージ属性を使用して、「通知の詳細」画面の「要求情報」ボタンを非表示にすることができます。ユーザーが「ワークリスト」画面で応答が必要な通知を表示すると、デフォルトで「通知の詳細」画面の応答フィールドに「要求情報」ボタンが表示されます。ユーザーが通知についての詳細な情報を要求できないようにする場合は、#HIDE_MOREINFO 属性を追加して、「要求情報」ボタンを表示するか非表示にするかを制御できます。

注意： Oracle Workflow のスタンドアロン版および Oracle Applications に組み込まれている Workflow では、#HIDE_MOREINFO 属性によって、HTML 形式の電子メール通知内に「要求情報」リンクを含めるか含めないかも制御されます。ただし、#HIDE_MOREINFO 属性が特定の通知に対して定義されていない場合のデフォルトの動作は、「通知の詳細」画面と HTML 形式の電子メール通知とは異なります。「通知の詳細」画面では #HIDE_MOREINFO 属性が定義されていないと「要求情報」ボタンが表示されるのに対し、HTML 形式の電子メールによる通知に #HIDE_MOREINFO 属性が定義されていないと、デフォルトでは「要求情報」リンクが表示されません。ユーザーが通知にアクセスするすべてのインタフェースでこのオプションを制御するには、明示的に #HIDE_MOREINFO 属性を定義し、設定する必要があります。『Oracle Workflow ユーザーズ・ガイド』の「HTML 形式の電子メール通知への応答」を参照してください。

#HIDE_MOREINFO 属性は、テキスト・タイプまたは選択肢タイプのどちらかにする必要があります。「要求情報」ボタンを非表示にするには、この属性の値を Y に設定し、表示するには N に設定します。

注意： 選択肢タイプの #HIDE_MOREINFO 属性を定義して、標準項目タイプで提供される事前定義済の「Yes/No」選択肢タイプに割り当てることをお勧めします。「Yes/No」選択肢タイプには、表示名 Yes と No が付いた 2 つの選択肢コードが含まれており、Y の値と N の値をそれぞれ表しています。

- 特定のメッセージを使用する通知の「要求情報」ボタンを常に非表示にする場合は、値 Y を定数として指定します。
- 特定の場合にのみ「要求情報」ボタンを非表示にするには、値として項目タイプ属性を指定します。次に、このボタンを表示するかどうかを実行時に動的に決定し、それぞれ項目タイプ属性を Y または N に設定するロジックを、ワークフロー・プロセスに組み込みます。

注意： Oracle Workflow の Web ページは、Oracle Applications Framework のユーザー・インタフェース形式に変換されます。Oracle Workflow のバージョンおよび適用したパッチに応じて、Oracle Workflow の Web ページは新しい形式または以前の形式で表示されます。現時点では、「要求情報」ボタンが含まれている新しい「ワークリスト」画面は、Oracle Applications に組み込まれている Workflow で利用できます。

関連項目：

『Oracle Workflow ユーザーズ・ガイド』の「通知の詳細の表示」(Oracle Applications の場合)

#FROM_ROLE 属性

メッセージ属性に内部名 #FROM_ROLE を指定すると、ロール (通知ソース) を指定できます。たとえば、購買申請を送信したことを承認者に通知する場合、購買申請作成者をメッセージの「ロール」として設定します。

この属性を持つ通知が「ワークリスト」Web ページで再割当てされた場合、新しい受信者への通知の送信時に「ロール」が前の受信者に自動的に設定されます。

各通知の「ロール」は、通知の確認や応答に関する詳細情報として、「ワークリスト」Web ページと電子メール通知に表示されます。また、「通知の検索」画面を使用すれば、「ロール」を基準にして通知を検索できます。

#FROM_ROLE 属性はロール・タイプであって、「送信」ソースを所有している必要があります。属性の表示名と説明は、「ロール」です。#FROM_ROLE 属性の値として項目タイプ属性を指定し、実行時に動的にその項目タイプ属性をロールに設定するロジックをワークフロー・プロセス内に組み込む必要があります。

関連項目：

『Oracle Workflow ユーザーズ・ガイド』の「ワークリストの通知の表示」

『Oracle Workflow ユーザーズ・ガイド』の「通知の検索」

『Oracle Workflow ユーザーズ・ガイド』の「通知の詳細の表示」(Oracle Applications の場合)

#WF_SIG_POLICY 属性

Oracle Applications に組み込まれている Workflow で利用可能な Oracle Applications Framework 対応の「ワークリスト」画面を使用している場合は、内部名 #WF_SIG_POLICY のメッセージ属性を使用して、通知に対するユーザーの応答が電子署名で認証されるようにすることができます。この電子署名は、手書きの署名と似ています。パスワードに基づいた署名が必要であることを通知に定義すると、ユーザーは、「通知の詳細」Web ページからの通知に応答する必要があります。Oracle Applications のユーザー名とパスワードを入力して、応答を確認する必要があります。入力しない場合は、応答が無効とみなされます。

#WF_SIG_POLICY 属性は、テキスト・タイプまたは選択肢タイプのどちらかにする必要があります。パスワードに基づいた署名を要求する場合は、#WF_SIG_POLICY 属性の値を「PSIG_ONLY」に設定します。この値を「DEFAULT」に設定したり、空白のままにしておいた場合、またはメッセージに対して #WF_SIG_POLICY 属性を定義しなかった場合は、署名を要求しないデフォルトの応答処理が行われます。

- 管理を容易にするために、選択肢タイプで #WF_SIG_POLICY 属性を定義し、標準項目タイプの事前定義済の「署名ポリシー」選択肢タイプに割り当てることができます。「署名ポリシー」選択肢タイプには表示名が Password Signature と Default という 2 つの選択肢コードが含まれており、「PSIG_ONLY」の値と「DEFAULT」の値をそれぞれ表しています。
- また、テキスト・タイプの #WF_SIG_POLICY 属性を定義することもできます。この場合は、設定した属性の値を手動で管理する必要があります。

#WF_SIG_POLICY 属性の定数値として「PSIG_ONLY」を指定することも、値として項目タイプ属性を指定し、実行時に動的にその項目タイプ属性を「PSIG_ONLY」または「DEFAULT」のどちらかに設定してパスワードが要求されているかどうかを判断するロジックをワークフロー・プロセス内に組み込むこともできます。

注意： Oracle Workflow の Web ページは、Oracle Applications Framework のユーザー・インタフェース形式に変換されます。Oracle Workflow のバージョンおよび適用したパッチに応じて、Oracle Workflow の Web ページは新しい形式または以前の形式で表示されます。現時点では、パスワードに基づいた電子署名がサポートされる新しい「ワークリスト」画面は、Oracle Applications に組み込まれている Workflow で利用できます。

ただし、Oracle Application Object Library のユーザー名とパスワードに基づいて Workflow ディレクトリ・サービスを管理している場合のみ、パスワードに基づいた電子署名がサポートされます。このオプションは、Oracle Internet Directory (OID) を統合してディレクトリ・サービスを管理している場合、現時点ではサポートされません。

Oracle Workflow の将来のリリースでは、他の値の #WF_SIG_POLICY 属性を使用した、他のタイプの電子署名もサポートされる予定です。

関連項目：

『Oracle Workflow ユーザーズ・ガイド』の「通知の詳細の表示」(Oracle Applications の場合)

『Oracle Workflow API リファレンス』の「NtfSignRequirementsMet」

#WF_SIG_ID 属性

Oracle Applications に組み込まれている Workflow で利用可能な Oracle Applications Framework 対応の「ワークリスト」画面を使用している場合は、#WF_SIG_POLICY メッセージ属性を定義して、通知用に電子署名ポリシーを指定することができます。通知に対するユーザーの応答を検証するために電子署名を要求する署名ポリシーを指定する場合は、通知が署名された後に #WF_SIG_ID という名前の別のメッセージ属性が作成されます。#WF_SIG_ID 属性には署名の識別子が格納されます。後に必要に応じてこの識別子を使用し、署名についての情報を参照することができます。この属性はテキスト・タイプであり、「応答」ソースがあります。

注意： Oracle Workflow では、ユーザーが電子署名を使用して通知への応答を送信するときに、#WF_SIG_ID 属性の値が自動的に作成されて設定されます。手動でこの属性を作成したり設定したりする必要はありません。

ヘッダー属性

Oracle Applications に組み込まれている Workflow で利用可能な Oracle Applications Framework 対応の「ワークリスト」画面を使用している場合は、特殊なヘッダー・メッセージ属性を使用して、「通知の詳細」画面の通知ヘッダー領域に重要な情報を表示することができます。ヘッダー属性は、電子メール通知の先頭にも表示されます。メッセージ属性をヘッダー属性として識別するために、次の書式の #HDR_ で始まる内部名を定義する必要があります。

#HDR_<name>

「通知の詳細」画面または電子メール通知内のヘッダー属性の順序は、Workflow Builder のナビゲーション・ツリー内のヘッダー属性の順序によって決まります。各ヘッダー属性の表示名は、通知のヘッダー領域の属性値の前のラベルとして表示されます。各属性に定数値を指定することも、値として項目タイプ属性を指定し、実行時に動的にその項目タイプ属性値を設定するロジックをワークフロー・プロセス内に組み込むこともできます。

注意： Oracle Workflow の Web ページは、Oracle Applications Framework のユーザー・インタフェース形式に変換されます。Oracle Workflow のバージョンおよび適用したパッチに応じて、Oracle Workflow の Web ページは新しい形式または以前の形式で表示されます。現時点では、ヘッダー属性がサポートされる新しい「ワークリスト」画面は、Oracle Applications に組み込まれている Workflow で利用できます。

関連項目：

『Oracle Workflow ユーザーズ・ガイド』の「通知の詳細の表示」(Oracle Applications の場合)

通知メーラー属性

受信者が電子メール通知を受信することを通知環境で設定している場合に、特殊なメッセージ属性を使用して、通知メーラーが通知電子メールを生成する方法を制御できます。たとえば、特定の部門が送信する通知をカスタマイズする場合は、そのような通知に対してメッセージ・テンプレート属性を定義することができます。

- **#WFM_FROM:** 電子メールの通知メッセージがユーザーに配信されるときにメッセージ・ヘッダーの「送信元」フィールドに表示される値を指定するには、この属性を定義します。**#WFM_FROM** メッセージ属性が通知に定義されていると、メッセージを送信する通知メーラーによって、メーラーの **From Address** パラメータの値を無視してメッセージの「送信元」フィールド内の **#WFM_FROM** の属性値が使用されます。
- **#WFM_REPLYTO:** 電子メール通知の応答が送信される先の着信メッセージを受信する電子メール・アカウントのアドレスを指定するには、この属性を定義します。**#WFM_REPLYTO** メッセージ属性が通知に定義されていると、メッセージを送信する通知メーラーによって、メーラーの **Reply To** パラメータの値を無視してメッセージの応答アドレスとして、**#WFM_REPLYTO** の属性値が使用されます。
- **#WFM_HTMLAGENT:** HTML 通知応答を処理する HTML Web エージェントを識別するベース URL を指定するには、メッセージに対してこの属性を定義します。HTML 添付ファイル付きの電子メール通知をサポートするには、この URL が必須です。デフォルトの URL は、「グローバル・ワークフロー設定」Web ページで指定した **Workflow Web Agent** から取り出されますが、この属性に異なる値を定義すると、デフォルトを上書きできます。**#WFM_HTMLAGENT** メッセージ属性が通知に定義されていると、メッセージを送信する通知メーラーによって、メーラーの **HTML Agent** パラメータの値を無視してメッセージの HTML Web エージェントとして、**#WFM_HTMLAGENT** の属性値が使用されます。

関連項目：

『Oracle Workflow ユーザーズ・ガイド』の「電子メールによる通知の閲覧」

Oracle Applications Manager のオンライン・ヘルプ

Oracle Enterprise Manager のオンライン・ヘルプ

通知メーラーのメッセージ・テンプレート属性

受信者が電子メール通知を受信することを通知環境で設定している場合に、特殊なメッセージ属性を使用して、通知メーラーが通知の電子メールを生成するために使用するメッセージ・テンプレートを指定することができます。たとえば、特定の部門が送信する通知をカスタマイズする場合は、そのような通知に対してメッセージ・テンプレート属性を定義することができます。特定の通知に対してこれらの属性で指定されたテンプレートは、通知メーラーの構成パラメータに指定されたテンプレートより優先されます。

Workflow Builder を使用して項目タイプ内に空のメッセージを定義することで、独自のメッセージ・テンプレートを作成することができます。『Oracle Workflow 管理者ガイド』の「メッセージ・テンプレートの変更」を参照してください。

メッセージ・テンプレート属性の値は、次の書式で指定する必要があります。

```
<item_type_internal_name>:<message_internal_name>
```

たとえば、「システム：メーラー」項目タイプ内に「ワークフロー・オープン・メール」を指定するには、次の値を入力します。

```
WFMAIL:OPEN_MAIL
```

次の属性を使用して、電子メール通知処理の各ステージで通知メッセージに使用されるテンプレートを指定することができます。

- **#WFM_OPEN_MAIL:** テンプレートによる応答方式を使用している場合に、応答が必要な電子メール通知に使用するテンプレートを指定します。
- **#WFM_OPEN_MAIL_DIRECT:** 直接応答方式を使用している場合に、応答が必要な電子メール通知に使用するテンプレートを指定します。
- **#WFM_OPEN_MAIL_FYI:** 応答が不要な電子メール通知に使用するテンプレートを指定します。
- **#ATTACHED_URLS:** URL 属性の「内容の添付」がオンになっている HTML 形式の通知メッセージに対して「通知参照」添付ファイルを作成するテンプレートを指定します。
- **#WFM_CANCELED:** 以前送信された通知が取り消されたことを受信者に知らせるために使用するテンプレートを指定します。

- #WFM_OPEN_INVALID: 通知に対してユーザーが誤って応答した場合にユーザーに送信するために使用するテンプレートを指定します。
- #WFM_CLOSED: 以前送信された通知が完了したことを受信者に知らせるために使用するテンプレートを指定します。
- #WFM_OPEN_MAIL_PSIG_ONLY: ユーザーの応答内に、パスワードに基づいた電子署名が必要な電子メール通知に使用するテンプレートを指定します。
- #WFM_OPEN_SIGN: パスワードに基づいた電子署名が必要な通知内に通知 ID (NID) 行を含む電子メール通知をユーザーが送信しようとした場合に、そのユーザーに警告を発するために使用するテンプレートを指定します。このような通知に対する有効な応答は、「通知の詳細」 Web ページからのみ送信できます。
- #WFM_OPEN_MORE_INFO: ユーザーが別のユーザーに詳細な情報を要求するために使用するテンプレートを指定します。

関連項目 :

『Oracle Workflow 管理者ガイド』の「メッセージ・テンプレートの変更」

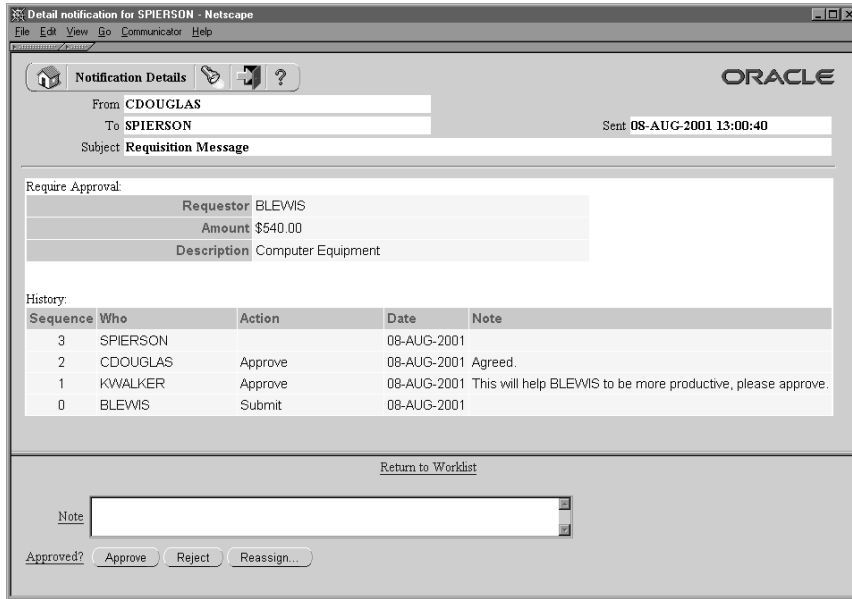
『Oracle Workflow ユーザーズ・ガイド』の「電子メールによる通知の閲覧」

Oracle Applications Manager のオンライン・ヘルプ

Oracle Enterprise Manager のオンライン・ヘルプ

WF_NOTIFICATION() メッセージ関数

メッセージ属性トークンの他に、WF_NOTIFICATION() という特別なメッセージ関数を使用して、状況依存のコンテンツをメッセージの本文に追加することもできます。WF_NOTIFICATION() 関数では、指定したパラメータに応じてメッセージ属性の表または通知履歴の表を生成することができます。表は、Oracle Workflow 標準形式で作成されます。



注意： WF_NOTIFICATION() は PL/SQL 関数ではなく、Oracle Workflow メッセージ本文内でのみコールできる特別なメッセージ関数です。

メッセージ属性表

メッセージ属性の表をメッセージの本文に含めるには、ATTRS オプションの後にメッセージ属性の内部名をカンマで区切って指定して、WF_NOTIFICATION() をコールします。書式は、次のとおりです。

```
WF_NOTIFICATION (ATTRS, <attribute1>, <attribute2>, <attribute3>, ...)
```

注意： WF_NOTIFICATION() のコールには、スペースや改行を含めないでください。リスト内でパラメータを区切るときはカンマのみを使用してください。

メッセージ属性表には、WF_NOTIFICATION() コールにリストされている各メッセージ属性の行が格納され、各属性の表示名および値が示されます。

通知履歴表

通知履歴表をメッセージの本文に含めるには、次のように HISTORY オプションを指定して WF_NOTIFICATION() をコールします。

WF_NOTIFICATION(HISTORY)

通知履歴表には、プロセス内ですでに実行された同じ通知アクティビティに対応する各行、およびプロセスの初期送信に対応する行が格納されます。通知履歴はある通信アクティビティのノードに特有のものであり、プロセスが同じノードに複数回ループするときに最も役立ちます。たとえば、複数の承認者に一定の通知を送信する購買承認申請通知アクティビティの場合、通知履歴表には通知が送信された各承認者の行およびプロセス所有者の行が格納されます。

通知履歴表には、次の列が含まれます。

- 順序： 通知アクティビティの実行順序。プロセス所有者がプロセスを初期送信すると、ゼロ (0) に初期化されます。
- 対象者： 通知の受信者またはプロセス所有者。
- 処理： 通知に対して受信者が行った応答処理。
- 日付： 通知の日付。
- 備考： 受信者からの「追加メモ」。受信者が通知履歴表のメモを追加できるようにするには、内部名が WF_NOTE の「応答」メッセージ属性を作成する必要があります。

次のプロパティを持つメッセージ属性を定義します。

- 内部名： WF_NOTE
- 表示名： 「追加メモ」
- 説明： 「追加メモ」
- タイプ： テキスト
- ソース： 応答

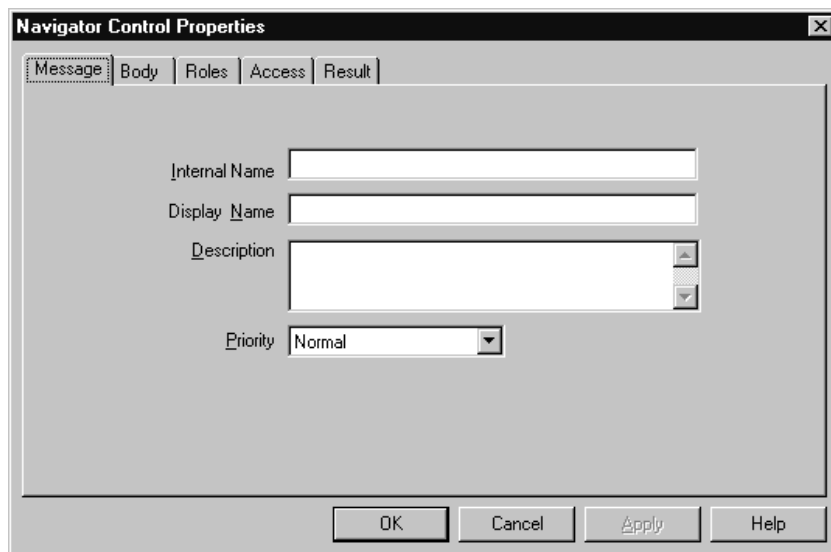
ソースを「応答」として定義した WF_NOTE 属性は、通知応答セクションに表示され、受信者は通知に応答するときにメモを入力できます。WF_NOTIFICATION() 関数は、

WF_NOTE 属性に格納されているメモ・テキストを取得し、それを通知履歴表に表示します。

受信者がメモを入力しなかった場合、または WF_NOTE メッセージ属性が通知に定義されていなかった場合、通知履歴表の「備考」列は空白のままになります。

注意： プロセスの送信時にプロセス所有者が通知履歴表にメモを追加することはできません。通知に応答するときにメモを追加できるのは、通知の受信者のみです。

▶ メッセージの作成



The image shows a dialog box titled "Navigator Control Properties" with a close button (X) in the top right corner. The "Message" tab is selected, and other tabs visible are "Body", "Roles", "Access", and "Result". The dialog contains the following fields and controls:

- Internal Name: A text input field.
- Display Name: A text input field.
- Description: A text area with scroll bars.
- Priority: A dropdown menu currently set to "Normal".

At the bottom of the dialog are four buttons: "OK", "Cancel", "Apply", and "Help".

1. ナビゲータ・ツリーで、メッセージを作成する項目タイプを選択し、「編集」メニューから「新規メッセージ」を選択します。「メッセージ」プロパティ画面が表示されます。
2. メッセージの内部名をすべて大文字で空白を入れずに入力し、表示名を指定します。オプションで説明を入力することもできます。すべての Oracle Workflow API、SQL スクリプトおよび PL/SQL プロシージャでは、メッセージの識別時に内部名が参照されます。

注意： 定義したメッセージの内部名を更新するには、wfchmsg.sql という特別な SQL スクリプトを使用する必要があります。このスクリプトは、設計時にメッセージの内部名のエラーを訂正する場合にのみ使用してください。プロセスの実行中のインスタンスに関連するメッセージの名前の変更には、使用しないでください。『Oracle Workflow 管理者ガイド』の「wfchmsg.sql」を参照してください。

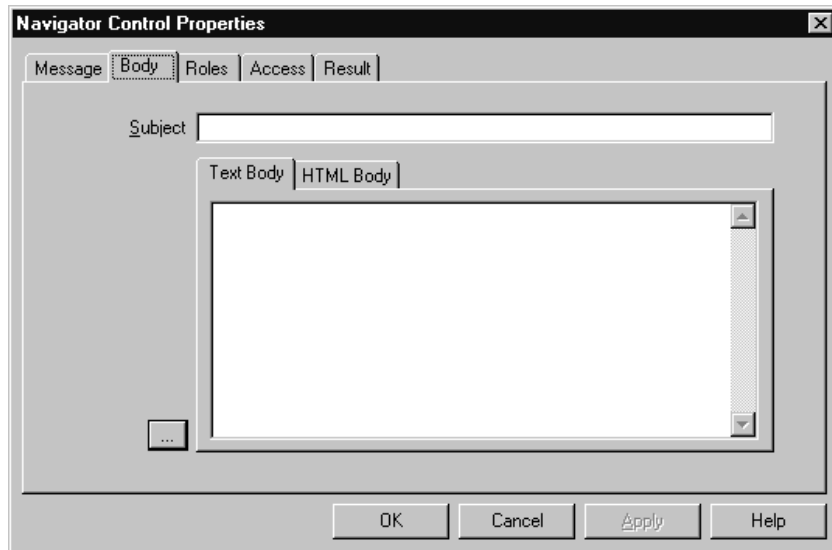
注意： 内部名には、コロン (:) や空白を使用しないでください。

3. メッセージのデフォルトの優先度に「高」、「標準」および「低」を選択します。これは、メッセージの優先度を宛先に伝えるためのものです。メッセージの処理や配信には影響しません。

注意： このメッセージを通知アクティビティに割り当て、通知アクティビティをノードとしてプロセス・ダイアグラムに組み込むと、このデフォルト優先度を固定の、または実行時に動的に決定される新規の優先度で上書きできます。4-9 ページの「プロセスのノードの定義」を参照してください。

注意： Oracle Workflow の旧バージョンでは、メッセージの優先度は 1 (高) ~ 99 (低) の数値で表されていました。このリリースの Oracle Workflow では、旧バージョンで定義されていたすべてのメッセージ定義の優先度が 1 ~ 33= 高、34 ~ 66= 標準、67 ~ 99= 低として自動的に変換されます。

4. 「適用」を選択して変更内容を保存します。



5. 「本文」タブを選択し、メッセージの「本文」プロパティ画面を表示します。
6. 「件名」には、「メッセージ」タブで入力した表示名がデフォルト値として取得されます。デフォルトの件名を選択するか、またはメッセージの新しい件名を入力します。件名には、メッセージの配信時にトークンをランタイム値で置き換えるメッセージ属性を含めることができます。件名にメッセージ属性を含めるには、アンパサンド (&) に続けてメッセージ属性の内部名を指定します。3-25 ページの「[送信および応答メッセージ属性](#)」および 3-44 ページの「[メッセージ属性の定義](#)」を参照してください。

提案： メッセージ属性には、参照先の項目タイプ属性と同じ名前を付けると、わかりやすくなります。

7. 「テキスト本文」フィールドに、プレーン・テキストのメッセージ本文を入力します。省略記号「...」ボタンを選択すると、「件名」および「テキスト本文」フィールドの表示を別のウィンドウに展開できます。

Oracle Workflow では、「テキスト本文」フィールドに入力した内容を使用して、通知メッセージのプレーン・テキスト版が生成されます。プレーン・テキスト・メッセージは、プレーン・テキスト・メッセージを表示する電子メール・ソフトウェアで表示できます。

注意：「テキスト本文」フィールドには、プレーン・テキストのメッセージ本文を入力してください。「テキスト本文」が空の場合、プレーン・テキスト用電子メール・ソフトウェアでメッセージ表示したときに通知が空になります。

8. 「HTML Body」タブを選択して内容を入力するか、「インポート」を選択して .HTM または .HTML ファイルから内容をインポートすると、オプションで「HTML Body」フィールドに HTML 形式のメッセージ本文を入力できます。また、省略記号「...」ボタンを選択すると、「件名」および「HTML Body」フィールドの表示を別のウィンドウに展開できます。

注意： HTML メッセージ本文を入力またはインポートするときには、`<Body>...</Body>` の HTML タグを含める必要はありません。これらのタグを使用すると、Oracle Workflow ではタグで囲まれた内容が抽出され、HTML メッセージ本文として使用されます。そのため、`<Body>` タグより前にある HTML タグや内容は無視されることとなります。

注意： Oracle Workflow Builder では、メッセージ本文の HTML 形式は検証されません。

Oracle Workflow では、「HTML Body」フィールドに入力した内容を使用して、通知メッセージの HTML 形式版が生成されます。HTML 形式の通知メッセージは、「通知の詳細」Web ページから、または HTML 形式のメッセージや HTML 形式のメッセージ添付ファイルを表示する電子メール・ソフトウェアから表示できます。

注意：「HTML Body」が空の場合、Oracle Workflow では「テキスト本文」に入力されたメッセージ本文を使用して通知メッセージが生成されません。この場合は、`<pre>...</pre>` の HTML タグ間にプレーン・テキストが挿入されます。

注意： Oracle Workflow では、HTML メッセージ本文中のアイコンおよび画像ファイルへの参照が完全にはサポートされていません。使用中の Web サーバーではこれらのファイル位置を検出して「通知の詳細」Web ページに正しく表示できる場合でも、通知メーラーやサード・パーティの電子メール・アプリケーションでは、ユーザーが電子メール通知の HTML 版を表示するときに、アイコンまたは画像ファイルの位置を識別できません。

9. テキストまたは HTML 本文にメッセージ属性を埋め込むことができます。Oracle Workflow トークンにより、メッセージ属性は通知の配信時にランタイム値で置き換えられます。メッセージ属性を埋め込むには、「&」に続けてメッセージ属性の内部名を指定します。

注意： メッセージ本文中のテキストは、4000 バイト以内にする必要があります。テキストにトークンの置換のためのメッセージ属性を挿入すると、最終的なメッセージ本文は最大で 32000 バイトまで増えることがあります。

注意： メッセージの件名または本文には、&#NID という特殊トークンも挿入できます。このトークンは、ランタイム通知の通知 ID に置き換えられます。

また、メッセージ関数 WF_NOTIFICATION() を使用して、書式化されたメッセージ属性の表または通知履歴表をテキスト・メッセージや HTML メッセージの本文に含めることができます。

10. 「適用」を選択して変更内容を保存します。
11. 「ロール」タブを選択し、このメッセージにアクセスできるロールを指定します。（この機能は、今後のリリースでサポートされます。）
12. 「アクセス」タブを選択し、このメッセージを変更できるアクセス・レベルを設定します。3-18 ページの「[オブジェクトへのアクセス許可の設定](#)」を参照してください。
13. 通知メッセージによって実行者に応答値を要求し、Oracle Workflow でその応答値を通知アクティビティの結果として解釈させるには、「結果」タブを選択して必要な情報を入力します。ここで指定した情報を使用して、RESULT という特殊な応答メッセージ属性が作成されます。3-25 ページの「[メッセージの結果](#)」を参照してください。

The screenshot shows the 'Navigator Control Properties' dialog box with the 'Result' tab selected. The dialog has several tabs: 'Message', 'Body', 'Roles', 'Access', and 'Result'. The 'Result' tab contains the following fields and controls:

- Display Name:** A text input field.
- Description:** A text input field.
- Lookup Type:** A dropdown menu with an 'Edit' button to its right.
- Default:** A sub-dialog containing:
 - Type:** A dropdown menu.
 - Value:** A text input field with an 'Edit' button to its right.
- Clear:** A button located below the Default sub-dialog.
- OK, Cancel, Apply, Help:** Standard dialog buttons at the bottom.

RESULT の表示名および説明を指定します。ドロップ・ダウン・フィールドから選択肢タイプを選択します。ここでは、通知アクティビティの結果タイプに指定した選択肢タイプと同一の選択肢タイプを選択する必要があります。「デフォルト」の「値」フィールドで選択肢コードを選択します。その選択肢コードが、RESULT メッセージ属性のデフォルト値として表示されます。

注意： 他のタイプのメッセージ属性の作成については、3-44 ページの「メッセージ属性の定義」を参照してください。

14. 「適用」を選択して変更内容を保存するか、「OK」を選択し、変更内容を保存してプロパティ画面を閉じるか、「キャンセル」を選択し、変更を取り消してプロパティ画面を閉じます。
15. 定義したメッセージが、ナビゲータ・ツリーの「メッセージ」のブランチの下に表示されます。ナビゲータ・ツリーでメッセージをダブルクリックするか、メッセージを選択して「編集」メニューから「プロパティ」を選択すると、いつでもこのメッセージのプロパティを検討または編集できます。

メッセージに「結果」が定義されている場合、ナビゲータ・ツリーではそのメッセージ・アイコンに赤い疑問符が重なって表示されるため、「結果」定義を持たないメッセージと区別しやすくなります。
16. このメッセージの件名と本文に含めたメッセージ属性を、すべて定義する必要があります。

17. 項目タイプ属性を参照するメッセージ属性を作成するには、参照される項目タイプ属性をナビゲータ・ツリーで選択し、マウスの左ボタンを押したままメッセージにドラッグします。

表示されたプロパティ画面を編集して、メッセージ属性の「ソース」が正しいかどうか確認します。「デフォルト」の「値」フィールドが自動的に「項目属性」に設定され、元の項目属性が参照されます。

18. 既存の項目タイプ属性に無関係なメッセージ属性を作成することもできます。3-44 ページの「[メッセージ属性の定義](#)」を参照してください。

▶ メッセージ属性の定義

The screenshot shows the 'Navigator Control Properties' dialog box. It is divided into two main sections: 'Attribute' and 'Default'.
In the 'Attribute' section:
- 'Message' is set to 'Document Response - Approved'.
- 'Internal Name', 'Display Name', and 'Description' are empty text boxes.
- 'Type' is a dropdown menu set to 'Text'.
- 'Source' is a dropdown menu set to 'Send'.
- 'Length' is an empty text box.
In the 'Default' section:
- 'Type' is a dropdown menu set to 'Constant'.
- 'Value' is an empty text box.
At the bottom of the dialog are four buttons: 'OK', 'Cancel', 'Apply', and 'Help'.

1. 既存の項目タイプ属性を参照しないメッセージ属性を作成するには、ナビゲータ・ツリーでメッセージを選択し、「編集」メニューから「新規属性」を選択します。
「属性」プロパティ画面が表示されます。
2. 内部名をすべて大文字で空白を入れずに入力します。すべての Oracle Workflow API、SQL スクリプトおよび PL/SQL プロシージャでは、属性の識別時に内部名が参照されます。

注意： 定義したメッセージ属性の内部名を更新するには、wfchmsga.sql という特別な SQL スクリプトを使用する必要があります。このスクリプトは、設計時にメッセージ属性の内部名のエラーを訂正する場合にのみ使用してください。プロセスの実行中のインスタンスに関連するメッセージ属性の名前の変更には、使用しないでください。『Oracle Workflow 管理者ガイド』の「wfchmsga.sql」を参照してください。

注意： メッセージ属性の内部名には、空白やコロン (:)、アンパサンド (&) または番号記号 (#) を使用しないでください。また、メッセージ属性の内部名には空白を使用しないことをお勧めします。

3. 「ソース」フィールドで「送信」または「応答」を指定し、この属性で通知メッセージの受信者に情報を提供するか、応答を求めるプロンプトを表示するかを指定します。
4. 表示名を入力します。これは、ナビゲータ・ツリーに表示される名前です。このメッセージ属性のソースを「応答」にした場合、この表示名は応答を求めるプロンプトとしても使用されます。

注意： 「送信」メッセージ属性の場合は、URL のドリルダウン先を記述する「URL」タイプの属性の場合にのみ、プレーン・テキストの電子メール通知に属性の表示名が表示されます。

5. オプションの説明を入力します。「応答」メッセージ属性の場合は、このフィールドを使用して応答インスタクションを指定します。
6. 属性のデータ型を選択します。
7. 属性の「タイプ」に応じて、次のデフォルト情報を指定します。
 - **テキスト：** テキスト属性の最大長を指定します。
 - **数値：** オプションで、数値の書式マスクとデフォルト値を指定します。
 - **日付：** オプションで、日付の書式マスクとデフォルト値を指定します。
 - **選択肢：** 値を取り出す事前定義済の選択肢タイプの名前を選択します。デフォルト値の選択肢コードを選択します。
 - **URL：** 「デフォルト」の「値」フィールドに、ネットワーク位置への Universal Resource Locator (URL) を指定します。3-12 ページの「URL 属性の定義」を参照してください。

注意： URL タイプの「応答」メッセージ属性は、プレーン・テキスト用電子メール・ソフトウェアから通知を表示すると正しく表示されません。URL タイプの「応答」メッセージ属性を持つメッセージを作成する場合は、ワークフローのユーザーに「通知の詳細」 Web ページから通知を表示するように指示する必要があります。

注意： URL タイプの単一の「応答」メッセージ属性により、「通知の詳細」 Web ページの応答フレームが置き換えられ、通知の受信者はカスタム HTML ページにリンクさせられて通知への応答を完了します。カスタム HTML の応答ドキュメントには、特殊な RESULT 属性（定義されている場合）など、すべての「応答」メッセージ属性の参照が含まれている必要があります。また、通知への応答完了をワークフロー・エンジンに知らせるために、ワークフロー・エンジンの CompleteActivity() API のコールも含まれている必要があります。

- **フォーム：** この属性は、Oracle Applications に組み込まれている Workflow にのみ関係します。

「デフォルト」の「値」フィールドで、開発者フォーム関数名とオプションの引数文字列（フォーム関数のパラメータ）を指定します。『Oracle Applications 開発者ガイド』の「メニューおよび機能セキュリティの概要」および 3-13 ページの「[フォーム属性の定義](#)」を参照してください。

注意： 「フォーム」タイプの「送信」および「応答」メッセージ属性が表示されるのは、「通知」 Web ページが Oracle Applications から起動される場合のみです。通知メッセージに「フォーム」タイプの「送信」メッセージ属性が含まれている場合は、添付フォーム・アイコンが使用可能になります。通知の受信者は、添付フォーム・アイコンをクリックし、メッセージ属性で定義されたフォーム関数にドリルダウンできます。

注意： メッセージに「フォーム」タイプの「応答」メッセージ属性が含まれている場合は、通知の「応答」セクションに表示される添付フォーム・アイコンを選択するだけで、指定されたフォームに直接ドリルダウンできます。このフォームは、通知応答が完了したことがワークフロー・エンジンに通知されるように、ワークフロー・エンジンの CompleteActivity() API のコールを使用してコーディングする必要があります。『Oracle Workflow API リファレンス』の「[Workflow Engine API](#)」を参照してください。

- **文書**： 文書を識別する文字列を「デフォルト」の「値」フィールドに入力します。3-13 ページの「[文書属性の定義](#)」を参照してください。
- **ロール**： ロール名を指定します。通知メッセージにロール・タイプのメッセージ属性が含まれる場合、属性は自動的にロールの表示名に設定されるため、ロールの内部名と表示名について別々の属性を保守する必要がなくなります。また、Web ブラウザから通知を表示する場合は、ロールの表示名がそのロールの電子メール・アドレスを示すハイパーテキスト・リンクになります。属性のデフォルト値を設定するには、最初にデータベースからロールをロードする必要があります。4-26 ページの「[ロール](#)」を参照してください。
- **イベント**： デフォルト値としてイベント項目タイプ属性を指定します。

注意： メッセージ属性のデータ型として「属性」を指定しないでください。このデータ型は、通知メッセージには無意味であり、ワークフロー通知システムでもサポートされません。

注意： 「応答」メッセージ属性のタイプが「日付」、「数値」、「テキスト」、「文書」または「ロール」の場合は、通知の受信者に対して、それぞれ日付、数値、テキスト値、文書、ロール（内部名または表示名）による応答を求めるプロンプトが表示されます。

「応答」メッセージ属性のタイプが選択肢の場合は、値リストから応答を選択するように求めるプロンプトが通知の受信者に表示されます。

8. メッセージ属性のタイプが「URL」の場合は、「フレーム・ターゲット」を指定します。このメッセージ属性をメッセージ内で参照し、メッセージを受信したユーザーがメッセージ本文内で URL リンクをクリックすると、フレーム・ターゲットに何を指定したかに応じて、URL がオープンします。フレーム・ターゲットは、次のとおりです。
 - **新規ウィンドウ**： URL は新しい名称未設定のブラウザ・ウィンドウにロードされます。
 - **同じフレーム**： URL は、その URL 属性を参照する要素と同じフレームにロードされます。
 - **親フレーム**： URL は現在のフレームのすぐ上位の FRAMESET にロードされます。現在のフレームに上位フレームがなければ、この値は「同じフレーム」と等価です。
 - **フル・ウィンドウ**： URL は、元のフル・ウィンドウにロードされ、他のフレームがすべて取り消されます。現在のフレームに上位フレームがなければ、この値は「同じフレーム」と等価です。

注意： Oracle Applications に組み込まれている Workflow で利用可能な Oracle Applications Framework 対応の「ワークリスト」画面を使用している場合は、URL 属性のフレーム・ターゲットとして、「新規ウィンドウ」または「フル・ウィンドウ」のみを選択する必要があります。このリリースのワークリストでは、「通知の詳細」 Web ページで通知を表示するのにフレームは使用されません。したがってこの場合は、「同じフレーム」または「親フレーム」を選択しても、「フル・ウィンドウ」を選択したことと同じになります。

9. メッセージの属性が「送信」で、タイプが「文書」の場合は、「内容の添付」をオンにして、その属性の内容を通知メッセージに添付できます。「通知」 Web ページ・インタフェースから通知を表示すると、通知メッセージ本文に続いて文書アイコンが表示され、このアイコンをクリックすると添付されたメッセージ属性の内容が示されます。電子メールから通知を表示した場合、添付ファイルの表示方法は、電子メール通知の環境設定によって異なります。『Oracle Workflow ユーザーズ・ガイド』の「電子メールによる通知の閲覧」を参照してください。

注意： 通知メッセージには、文書属性の添付も埋込み（トークンの置換を使用）もでき、両方行うことも可能です。

10. メッセージの属性が「送信」で、タイプが「URL」の場合は、「内容の添付」をオンにして、「通知参照」という添付ファイルを通知メッセージに添付できます。この添付ファイルには、「内容の添付」がオンになっているメッセージの各 URL 属性へのリンクが含まれています。URL へのリンクを選択すると、その URL に移動できます。

注意： 通知メッセージには、URL 属性の添付も埋込み（トークンの置換を使用）もでき、両方行うことも可能です。

11. メッセージ属性の場合は、デフォルト値として定数または項目タイプ属性を指定できます。デフォルトで項目タイプ属性の値全体を直接参照する場合は、「項目属性」を選択し、ドロップ・ダウン・フィールドを使用して項目タイプ属性を選択します。ここで選択する項目タイプ属性は、メッセージ自体が関連付けられているのと同じ項目タイプに関連付ける必要があります。また、メッセージ属性と同じデータ型の項目タイプ属性を選択する必要があります。

注意： 「テキスト」タイプのメッセージ属性には、すべての項目属性タイプとの互換性がありますが、他のメッセージ属性タイプは、いずれも項目属性タイプと完全に一致させる必要があります。

12. 「適用」を選択して変更内容を保存するか、「OK」を選択し、変更内容を保存してプロパティ画面を閉じるか、「キャンセル」を選択し、変更を取り消してプロパティ画面を閉じます。
13. 定義したメッセージ属性は、ナビゲータ・ツリーで定義対象となったメッセージの下に表示されます。ナビゲータ・ツリーで属性をダブルクリックするか、属性を選択して「編集」メニューから「プロパティ」を選択すると、いつでもその属性のプロパティを検討または編集できます。ナビゲータ・ツリーの「応答」メッセージ属性アイコンには、「送信」メッセージ属性アイコンと区別できるように、赤い疑問符が表示されます。

注意： メッセージ属性では、その親メッセージのアクセス / 保護レベルが想定されます。

注意： ナビゲータ・ツリーで「応答」メッセージ属性を表示する順序は、これらの属性が通知メッセージの応答セクションに表示される順序に関連します。ナビゲータ・ツリーのドラッグ・アンド・ドロップ機能を使用するか、属性を選択して「編集」メニューから「属性を上へ移動」や「属性を下へ移動」を選択して、属性セット内での属性の順序を変更できます。

関連項目：

3-50 ページ「[「応答」メッセージ属性の例](#)」

2-9 ページ「[プロパティ画面の「編集」ボタンの使用](#)」

『Oracle Workflow ユーザーズ・ガイド』の「[電子メールによる通知の閲覧](#)」

「応答」メッセージ属性の例

次の例は、デフォルト値のない「応答」メッセージ属性のサンプル・セットを使用して、電子メール通知の「応答」セクションが通知システムによって生成される状況を示しています。

次の表に、「応答」メッセージ属性の例をいくつか示します。

表 3-2

内部名	型	書式 / 選択肢タイプ	表示名	説明
RESULT	選択肢	WFSTD_APPROVAL	アクション	承認するかどうか。
COMMENT	テキスト	2000	レビュー・コメント	
REQDATE	日付	DD-MON-YYYY	必須の日付	必須の日付がない場合は空白のままにします。
MAXAMT	数値		最大金額	これは承認額の最大値です。

テンプレートによる応答方式の場合は、次の定型テキストを使用して、電子メール通知の「応答」テンプレート・セクションが生成されます。

```
<Description>
<Display Name>: " "
    <list of lookup codes>
```

テンプレートによる応答の電子メール通知に表示される応答テンプレートの部分

```
承認するかどうか。
アクション: ""
    承認する
    却下する

レビュー・コメント: ""

必須の日付がない場合は空白のままにします。
必須の日付: ""

これは承認額の最大値です。
最大金額: ""
```

直接応答方式の場合は、次の定型テキストを使用して、電子メール通知の「応答」セクションが生成されます。

Enter the *<Display Name>* on line *<Sequence>*. *<Description>* *<Type_Hint>*

<Display Name> は、メッセージ属性の表示名に置き換えられます。*<Sequence>* は、ナビゲータ・ツリーにすべての「応答」メッセージ属性とともに表示される相対順序番号に置き換えられます（つまり、順序の決定時には、「送信」メッセージ属性の存在は無視されません）。*<Description>* は、メッセージ属性の説明に置き換えられます。また、*<Type_Hint>* は、メッセージ属性がヒント付きのデータ型と一致する場合は、ヒント文で置換されます。次の表は、ヒント付きのデータ型と対応するヒント文を示しています。

表 3-3

型	型のヒント
選択肢	値は次のどちらかにしてください。 < 選択肢コードのリスト >
日付	値は日付である必要があります [形式は " <i><format></i> "]。
数値	値は数値である必要があります [形式は " <i><format></i> "]。
テキスト	値は <i><format></i> バイト以内にします。

直接応答の電子メール通知に表示される応答セクションの部分

1 行目にアクションを入力してください。承認しますか? 値は次のどちらかにしてください。

承認する
却下する

2 行目にレビュー・コメントを入力してください。値は 2000 バイト以内にします。

3 行目に必須の日付を入力してください。必須の日付がない場合は空白のままにします。値は「DD-MON-YYYY」形式の日付にします。

4 行目に最大金額を入力してください。これは承認額の最大値です。値は数値にします。

➤ 属性のトークン置換

- Oracle Workflow では、属性のランタイム・トークンの置換がサポートされています。属性は、他の属性の中に埋め込むことも、メッセージの件名および本文に埋め込むこともできます。属性を埋め込むには、トークンの置換に使用する属性を、*&attr_name* のように指定します。*attr_name* は、属性の内部名です。

トークンの置換を実行すると、Oracle Workflow によって属性の内部名とその値がフェッチされます。トークンの置換が必要な属性が、他の属性とネストされている場合、属性の内部名の長さに従って、ネストされた属性リストが整理され、最も長い内部名から順に属性の置換が開始されます。

注意： 必要なメッセージ本文の内容を表示するために、2階層より深くメッセージ属性をネストする必要がある場合、PL/SQL 文書タイプのメッセージ属性を作成することを検討してください。3-5 ページの「[外部文書の統合](#)」を参照してください。

▶ **メッセージのコピー**

1. ナビゲータ・ツリーで、コピーするメッセージを選択します。
2. マウスの左ボタンを押しながら、メッセージをコピー先の「項目タイプ」ブランチまでドラッグします。
3. マウス・ボタンを離すと、新規メッセージのプロパティ画面が表示されます。

注意： 「編集」メニューの「コピー」および「貼り付け」オプションも使用できます。

4. 新規の内部名と表示名を入力します。
5. メッセージのプロパティをさらに変更します。
6. 操作の完了後に「OK」を選択します。

注意： メッセージをコピーすると、そのメッセージに割り当てられているメッセージ属性もコピーされます。

アクティビティ

アクティビティは、プロセスを完了させるために行う作業の単位です。アクティビティには、通知アクティビティ、関数アクティビティ、イベント・アクティビティまたはプロセス・アクティビティがあります。通知アクティビティは、ワークフローのユーザーにメッセージを送信します。メッセージには、単にユーザーに情報を提供するものと、なんらかのアクションを要求するものがあります。関数アクティビティは、PL/SQL のストアード・プロシージャまたは外部プログラムをコールして、自動化関数を実行します。イベント・アクティビティは、ビジネス・イベントを受信、発生または送信します。プロセス・アクティビティはモデル化されたワークフロー・プロセスで、他のプロセス内にアクティビティとして組み込んで、サブプロセスとして使用できます。

アクティビティは、ナビゲータ・ツリーでそれぞれの「プロセス」、「通知」、「関数」または「イベント」ヘッダーの下に編成されます。ナビゲータ・ツリーでアクティビティの定義を作成、編集および削除できます。また、アクティビティをツリーから「プロセス」ウィンドウにドラッグし、プロセス・ダイアグラムにそのアクティビティの新しい使用方法を作成することもできます。各アクティビティは、プロセス・ダイアグラム内でアイコンとして表示されます。

Oracle Workflow には、「標準」という項目タイプが用意されており、定義するあらゆるプロセスに使用できるように一般的なアクティビティが含まれています。たとえば、2つの値を比較するなど、標準的な機能を実行するアクティビティがあります。5-2 ページの「[標準アクティビティ](#)」を参照してください。

また、Oracle Workflow には、「システム:エラー」という項目タイプも用意されており、そこに含まれる標準エラー・プロセスとアクティビティを使用して、カスタムのエラー・プロセスを作成できます。エラー・プロセスをプロセス・アクティビティに割り当てることができます。エラーが発生すると、エラー・プロセスによってエラーの対処方法が Oracle Workflow に通知されます。11-2 ページの「[ワークフロー・プロセスのエラー処理](#)」を参照してください。

通知アクティビティ

ワークフロー・エンジンは、通知アクティビティに達すると、通知システムに `Send()` API コールを発行し、割り当てられた実行者にメッセージを送信します。通知によって送信されるメッセージはユーザーが定義します。メッセージとして、情報メモや、実行者に応答入力を求めるプロンプトを使用できます。実行者が通知アクティビティに応答すると、その応答が通知システムにより処理され、通知アクティビティを完了したため次の「適格」アクティビティの処理を継続するように、ワークフロー・エンジンに通知されます。3-59 ページの「[通知アクティビティの作成](#)」を参照してください。

プロセスに通知アクティビティをノードとして組み込む場合は、通知アクティビティの実行者を指定します。実行者として、特定のロールまたは動的にロール名を戻す項目タイプ属性を指定できます。4-9 ページの「[プロセスのノードの定義](#)」および 4-26 ページの「[ロール](#)」を参照してください。

通知アクティビティを定義するときに、オプションで次のことも可能です。

- 「ロールの拡張」をオンにして、ロール内の各ユーザーに、通知メッセージの個別コピーを送信できます。ユーザーが応答するか閉じるまで、通知はユーザーの通知キューに残ります。

注意： ロールの全ユーザーが閲覧する必要のあるブロードキャスト・タイプのメッセージを送信する場合は、ロールを拡張してください。

通知アクティビティのロールを拡張しなければ、割り当てた実行者ロールに通知メッセージが 1 通送信され、その通知はロールの全ユーザーの通知キューで表示されます。ロールのユーザーが 1 人でも通知に応答するか、または通知を閉じると、この通知はロールの全ユーザーの通知キューから削除されます。

- 通知の送信後に通知の状態更新に応じてワークフロー・エンジンにより実行される、通知後関数を指定します。ワークフロー・エンジンでは、受信者が通知に応答したか、通知を転送したか、通知を譲渡したか、または通知がタイムアウトになったかに応じて、通知後関数が `RESPOND`、`FORWARD`、`TRANSFER` または `TIMEOUT` のいずれかのモードで実行されます。通知システムが `RESPOND` モードで通知後関数の実行を終了すると、ワークフロー・エンジンは `RUN` モードで通知後関数を再実行します。

たとえば、通知転送先のロールを制限する場合は、受信者が通知の転送を試行すると `FORWARD` モードで通知後関数が実行されるように指定できます。通知後関数はロールを監査して、転送を許可するか、またはエラーを戻して転送を拒否します。『[Oracle Workflow API リファレンス](#)』の「通知後関数」および「通知モデル」を参照してください。

通知後関数を作成するには、関数アクティビティの場合と同じ `PL/SQL` API を使用する必要があります。6-3 ページの「[関数アクティビティがコールする PL/SQL プロシージャの標準 API](#)」を参照してください。

「ロールの拡張」をオンにして、通知後関数を指定すると、独自の投票集計アクティビティを作成できます。3-73 ページの「[投票アクティビティ](#)」を参照してください。

関数アクティビティ

関数アクティビティは、PL/SQL ストアド・プロシージャ、またはそれによりコールされる外部プログラムによって定義されます。通常、関数アクティビティは、プロセスにおいて完全に自動化された手順を実行するために使用します。PL/SQL ストアド・プロシージャであるため、関数アクティビティは通常の引数を受け入れて実行後の結果を戻すことができます。

ストアド・プロシージャのパラメータを渡すと、パラメータをアクティビティ属性として外部から識別できます。アクティビティ属性の値は、アクティビティをプロセスのノードとして定義するときに設定できます。このようなアクティビティ属性は、現行のアクティビティにのみ有効で、項目タイプ属性のようにグローバルではないため注意してください。4-19 ページの「[アクティビティ属性値の定義](#)」を参照してください。

外部プログラムとして、関数アクティビティではペイロード情報を Oracle Advanced Queuing の送信キューにエンキューでき、外部エージェントがそれをデキューして利用します。外部エージェントは同様に、更新した属性と完了結果を受信キューにエンキューし、ワークフロー・エンジンがこれを使用して処理します。

外部 Java プログラムとして、関数アクティビティではペイロード情報を Oracle Advanced Queuing の送信キューにエンキューでき、Java 関数アクティビティ・エージェントがそれをデキューして利用します。Java プログラムの結果は、受信キューにエンキューされ、ワークフロー・エンジンがこれを使用して処理します。現在、この機能を使用できるのは Oracle Workflow のスタンドアロン版のみです。3-61 ページの「[関数アクティビティの作成](#)」を参照してください。

イベント・アクティビティ

イベント・アクティビティは、ワークフロー・プロセス内のビジネス・イベント・システムのビジネス・イベントを表します。関数を実行したり事前定義済エージェントにイベントを送信したりするときに、イベント・サブスクリプションの標準オプションより複雑なビジネス・イベントの処理やルーティング・ロジックをモデル化するには、イベント・アクティビティをワークフロー・プロセスに組み込みます。8-2 ページの「[ビジネス・イベントの管理](#)」を参照してください。

イベント・アクティビティは、ビジネス・イベントを受信、発生または送信できます。

- 受信イベント・アクティビティは、特定のプロセスに対して「開始」アクティビティとしてマークすることができます。このプロセスでは、常にイベントを受信できます。また、受信イベント・アクティビティをそのプロセス内に配置すれば、プロセスが「受信」アクティビティに遷移したときにだけイベントを受信します。

受信アクティビティがイベントを受信すると、ワークフロー・エンジンはノードのイベントの詳細の指定に従って、イベント名、イベント・キーおよびイベント・メッセージを項目タイプ属性に格納し、イベント・メッセージのパラメータ・リストに含まれるすべてのパラメータをプロセスの項目タイプ属性として設定し、パラメータに対応する属性がまだ存在しない場合は新しい項目属性を作成し、イベント・アクティビティからスレッドの実行を続行します。「受信」アクティビティがイベントをすでに受信している場合は、「再開封時」フラグによってワークフロー・エンジンがアクティビティを再実行するかどうかが決まります。3-70 ページの「[オプションのアクティビティ詳細の定義](#)」を参照してください。

イベントが別のワークフロー・プロセスの呼出しイベント・アクティビティによって発生した場合は、そのプロセスの項目タイプと項目キーがイベント・メッセージ内のパラメータ・リストに追加されます。ワークフロー・エンジンは、イベントを受信するプロセスの親として、指定されたプロセスを自動的に設定し、既存の親の設定を上書きします。『Oracle Workflow API リファレンス』の「[SetItemParent](#)」を参照してください。

- 呼出しイベント・アクティビティは、イベントに関する情報を取り出し、ビジネス・イベント・システムに対してイベントを発行します。ビジネス・イベント・システムは、そのイベントに対してサブスクリプションを実行します。「呼出し」アクティビティは、ノードのイベントの詳細に従って、イベント名、イベント・キーおよびイベント・データを取り出します。イベントの詳細は、項目タイプ属性を使用して実行時に動的に決定されます。イベント名は、イベント・アクティビティ・ノードの事前定義済の定数として指定することもできます。

「呼出し」アクティビティは、定義されているアクティビティ属性の名前と値を取り出し、イベント・メッセージのパラメータ・リストに設定します。別のプロセスがイベント・メッセージを後で受信した場合、ワークフロー・エンジンはイベント・パラメータをそのプロセスの項目タイプ属性として設定します。『Oracle Workflow API リファレンス』の「[イベント・メッセージ構造](#)」を参照してください。

「呼出し」アクティビティは、現行のワークフロー・プロセスの項目タイプと項目キーをイベント・メッセージのパラメータ・リストに自動的に設定します。別のプロセスがイベント・メッセージを後で受信した場合、ワークフロー・エンジンはその項目タイプ

と項目キーを使用して、イベントを受信するプロセスの親として、イベントを呼び出したプロセスを自動的に設定します。『Oracle Workflow API リファレンス』の「SetItemParent」を参照してください。

- 送信イベント・アクティビティは、ノードのイベントの詳細に従って、イベント名、イベント・キー、イベント・メッセージ、アウトバウンド・エージェントおよびインバウンド・エージェントを取り出します。相関 ID がイベント・メッセージに指定されていない場合は、Oracle Workflow によって相関 ID がプロセスの項目キーに自動的に設定されます。次に、送信イベント・アクティビティは、アウトバウンド・エージェントのイベントをインバウンド・エージェントに直接送信します。イベントの詳細は、項目タイプ属性を使用して実行時に動的に決定されます。イベント名、アウトバウンド・エージェントおよびインバウンド・エージェントは、イベント・アクティビティ・ノードの事前定義済みの定数として指定することもできます。3-65 ページの「[イベント・アクティビティの作成](#)」および 4-14 ページの「[イベント・ノードのイベントの詳細の定義](#)」を参照してください。

注意：「送信」エージェント・アクティビティでは、ビジネス・イベント・システムに対してイベントを発行しないため、サブスクリプション処理は実行されません。

プロセス・アクティビティ

プロセス・アクティビティは、特定の関連を持つアクティビティの集まりを表します。プロセス・アクティビティが他のプロセスに含まれている場合、そのプロセスはサブプロセスと呼ばれます。つまり、プロセス内のアクティビティは、それ自身がプロセスになることもあります。この階層の深さには制限はありません。3-68 ページの「[プロセス・アクティビティの作成](#)」を参照してください。

注意： Oracle Workflow では、1 つのプロセス階層内でサブプロセス・アクティビティを複数回使用することはできません。

関連項目：

『Oracle Workflow 管理者ガイド』の「サブプロセス」

アクティビティ・コスト

各関数アクティビティおよびイベント・アクティビティには、コストが関連付けられています。コストは、ワークフロー・エンジンがアクティビティを実行するために必要な秒数を表す値です。ワークフロー・エンジンがそのアクティビティを実行する所要時間が不明な場合は、コストの見積りを入力し、後でパフォーマンスに関する情報が集まってから更新できます。通常、複雑で長時間かかるアクティビティには、高いコストを割り当てる必要があります。

コストの有効範囲は、0.00 ～ 1,000,000.00 です。

注意： Oracle Workflow Builder にコストを秒単位で入力して表示しても、実際には 100 分の 1 秒単位に換算されてデータベースに格納されます。

通常の処理では、ワークフロー・エンジンは 1 つのアクティビティの実行を完了してから次のアクティビティに移ります。場合によっては、処理に長時間がかかるため、バックグラウンド処理の方が適しているアクティビティもあります。

指定したしきい値よりコストが高いアクティビティをバックグラウンド処理にまわして延期するように、ワークフロー・エンジンを定義できます。ワークフロー・エンジンでは、次に適格な保留中のアクティビティの処理が続行されます。このアクティビティは、プロセスの別の並列分岐で発生することがあります。

ワークフロー・エンジンのデフォルトのしきい値は、50/100 秒です。これより高いコストのアクティビティは、バックグラウンド・エンジンにまわされます。バックグラウンド・エンジンは、特定タイプのアクティビティのみを実行するようにカスタマイズできます。ワークフロー・エンジンのしきい値は、SQL*Plus で設定できます。『Oracle Workflow 管理者ガイド』の「バックグラウンドのワークフロー・エンジンの設定」、「エンジンのしきい値の設定」および「アクティビティの遅延」を参照してください。

▶ 通知アクティビティの作成

The screenshot shows the 'Navigator Control Properties' dialog box with the 'Activity' tab selected. The fields are as follows:

- Internal Name: [Empty text box]
- Display Name: [Empty text box]
- Description: [Empty text box]
- Icon: NOTIFY.ICO (dropdown menu) with a 'Browse' button and an envelope icon.
- Function Name: [Empty text box]
- Function Type: PL/SQL (dropdown menu)
- Result Type: <None> (dropdown menu) with an 'Edit' button.
- Message: [Empty text box] with an 'Edit' button.
- Expand Roles: Expand Roles

Buttons at the bottom: OK, Cancel, Apply, Help.

1. ナビゲータ・ツリーで、通知を作成する項目タイプを選択し、「編集」メニューから「新規通知」を選択します。「アクティビティ」プロパティ画面で、通知アクティビティを定義します。

また、ナビゲータ・ツリーでメッセージを選択し、同じ項目タイプの「通知」ブランチにドラッグ・アンド・ドロップして、そのメッセージを送信する通知アクティビティを作成することもできます。

2. 通知アクティビティには、内部名（すべて大文字で空白を含みません）と表示名が必要です。表示名は、プロセス・ダイアグラムに表示される変更可能な名前です。説明を使用して、このアクティビティに関する説明を入力します。

注意： 定義したアクティビティの内部名を更新するには、wfchact.sql という特別な SQL スクリプトを使用する必要があります。このスクリプトは、設計時にアクティビティの内部名のエラーを訂正する場合にのみ使用してください。プロセスの実行中のインスタンスに関連するアクティビティの名前の変更には、使用しないでください。『Oracle Workflow 管理者ガイド』の「wfchact.sql」を参照してください。

注意： 内部名には、コロン (:) や空白を使用しないでください。

3. このアクティビティの結果タイプ（事前に定義された選択肢タイプ）を指定します。結果タイプによって、アクティビティから戻される可能性がある結果のリストが表示されます。完了したアクティビティからの戻り値に応じて、ワークフロー・ダイアグラムが分岐することがあります。3-21 ページの「[選択肢タイプの作成](#)」を参照してください。

アクティビティから値が戻されない場合や、ワークフロー・プロセスが戻り値に依存しない場合は、結果タイプとして「<なし>」を選択できます。

4. この通知を送信するメッセージの名前を選択します。3-38 ページの「[メッセージの作成](#)」を参照してください。
5. この通知を複数のユーザーで構成されるロールに割り当てる予定があり、この通知のコピーをロールの各ユーザーに送信する場合は、「ロールの拡張」をオンにします。「ロールの拡張」をオフにすると、通知のコピーはロールに対して 1 通のみ送信されます。3-54 ページの「[通知アクティビティ](#)」を参照してください。
6. オプションで、「関数名」フィールドに PL/SQL ストアド・プロシージャを指定できます。このプロシージャは通知後関数と呼ばれ、通知アクティビティに処理のロジックをまとめることができます。ワークフロー・エンジンでは、受信者が通知に応答したか、通知を転送したか、通知を移動したか、または通知がタイムアウトになったかに応じて、この通知後関数が RESPOND、FORWARD、TRANSFER または TIMEOUT のいずれかのモードで実行されます。通知システムが RESPOND モードで通知後関数の実行を終了すると、ワークフロー・エンジンは RUN モードで通知後関数を再実行します。6-3 ページの「[関数アクティビティがコールする PL/SQL プロシージャの標準 API](#)」および『Oracle Workflow API リファレンス』の「通知後関数」を参照してください。

「ロールの拡張」をオンにし、この通知アクティビティに特殊な結果を含むメッセージを割り当て、「関数名」フィールドで、この通知の各受信者から返信された応答を集計するカスタム PL/SQL ストアド・プロシージャの名前を指定します。プロシージャを指定するには、書式 `<package_name>.<procedure_name>` を使用します。3-73 ページの「[投票アクティビティ](#)」を参照してください。

7. アクティビティを識別するアイコンを選択します。.ico ファイルに保存されているアイコンを使用して、アクティビティの処理をシンボル化できます。『Oracle Workflow 管理者ガイド』の「Oracle Workflow へのカスタム・アイコンの追加」を参照してください。

「参照」を選択すると、ワークフロー・アイコン・サブディレクトリ内のアイコン・ファイルが表示されます。

また、Windows の「エクスプローラ」や「ファイル マネージャ」からアイコン・ファイルをナビゲータ・ツリーのアクティビティ上にドラッグ・アンド・ドロップし、アクティビティにアイコンを設定することもできます。

8. 「適用」を選択して変更内容を保存します。
9. 「詳細」タブを選択し、アクティビティの詳細オプションを表示して変更します。3-70 ページの「[オプションのアクティビティ詳細の定義](#)」を参照してください。

10. 「ロール」タブを選択し、この通知アクティビティにアクセスできるロールを指定します。（この機能は、今後のリリースでサポートされます。）
11. 「アクセス」タブを選択し、この通知を変更できるアクセス・レベルを設定します。3-18ページの「[オブジェクトへのアクセス許可の設定](#)」を参照してください。
12. 「OK」を選択して変更を保存し、プロパティ画面を閉じます。
13. 通知アクティビティが、ナビゲータ・ツリーで「通知」の下に表示されます。ナビゲータ・ツリーでアクティビティをダブルクリックするか、アクティビティを選択して「編集」メニューから「プロパティ」を選択するか、キーボードで [Enter] キーを押すと、いつでもこのアクティビティのプロパティを検討または編集できます。

関連項目：

2-9 ページ「[プロパティ画面の「編集」ボタンの使用](#)」

▶ 関数アクティビティの作成

The screenshot shows the 'Navigator Control Properties' dialog box with the 'Activity' tab selected. The fields are as follows:

- Internal Name: [Empty text box]
- Display Name: [Empty text box]
- Description: [Empty text box]
- Icon: FUNCTION.ICO (dropdown menu)
- Function Name: [Empty text box]
- Function Type: PL/SQL (dropdown menu)
- Result Type: <None> (dropdown menu)
- Cost: 0.00 (text box)

Buttons: Browse, Edit, OK, Cancel, Apply, Help.

1. ナビゲータ・ツリーで、関数を作成する項目タイプを選択し、「編集」メニューから「新規関数」を選択します。「アクティビティ」プロパティ画面で、関数アクティビティを定義します。
2. 関数アクティビティには、内部名（すべて大文字で空白を含みません）と表示名が必要です。表示名は、プロセス・ダイアグラムに表示される変換可能な名前です。説明を使用して、このアクティビティに関する説明を入力します。

注意： 定義したアクティビティの内部名を更新するには、wfchact.sql という特別な SQL スクリプトを使用する必要があります。このスクリプトは、設計時にアクティビティの内部名のエラーを訂正する場合にのみ使用してください。プロセスの実行中のインスタンスに関連するアクティビティの名前の変更には、使用しないでください。『Oracle Workflow 管理者ガイド』の「wfchact.sql」を参照してください。

注意： 内部名には、コロン (:) や空白を使用しないでください。

3. このアクティビティで実行する関数の名前を入力します。「関数タイプ」フィールドで、この関数のタイプ（「PL/SQL」関数、「外部」関数または「外部 Java」関数）を指定します。

PL/SQL 関数の場合は、関数タイプを「PL/SQL」に設定し、関数を `<package_name>.<procedure_name>` として指定します。この関数は、標準 API に従って定義する必要があります。6-3 ページの「[関数アクティビティがコールする PL/SQL プロシージャの標準 API](#)」を参照してください。

外部関数アクティビティの場合は、関数タイプを「外部」に設定します。ワークフロー・エンジンで「送信」キューにエントリがエンキューされ、そのエントリと、Workflow スキーマ名および項目タイプで構成される値との相関値が次の書式で設定されます。

`<schema_name><item_type>`

関連項目：

『Oracle Workflow API リファレンス』の「Workflow QUEUE API」

この種のレコードを「送信」キュー内で検索するには、独自のキュー・ハンドラを作成する必要があります。このキュー・ハンドラでは、レコードに関連する処理を実行し、処理結果を「受信」キューにシードする必要があります。バックグラウンド・エンジンにより、受信キューにあるメッセージが処理され、元のワークフロー・プロセスが再開されます。『Oracle Workflow API リファレンス』の「遅延処理」を参照してください。

外部 Java 関数アクティビティの場合は、関数タイプを「外部 Java」に設定し、カスタム Java クラスのクラス名を関数名として入力します。現在、この機能を使用できるのは Oracle Workflow のスタンドアロン版のみです。カスタム・クラスがパッケージ内にある場合は、次の書式でクラス名の先頭にパッケージ名を付けます。

`<customPackage>.<customClass>`

Java クラスは、標準 API に従って定義する必要があります。6-8 ページの「[関数アクティビティがコールする Java プロシージャの標準 API](#)」を参照してください。

ワークフロー・エンジンがエントリを「送信」キューにエンキューします。Java 関数アクティビティ・エージェントは、このタイプのメッセージをデキューし、Java プログラムを実行して、結果を「受信」キューにエンキューします。バックグラウンド・エンジンにより、受信キューにあるメッセージが処理され、元のワークフロー・プロセスが再開されます。『Oracle Workflow 管理者ガイド』の「Java 関数アクティビティ・エージェントの設定」および『Oracle Workflow API リファレンス』の「遅延処理」を参照してください。

注意： これらの「送信」キューおよび「受信」キューは、ビジネス・イベント・システムに使用するキューとは異なります。『Oracle Workflow API リファレンス』の「Workflow QUEUE API」を参照してください。

注意： 外部 Java 関数アクティビティを実行するには、CLASSPATH に JAR ファイルを指定する必要があります。

4. このアクティビティの結果タイプ（事前に定義された選択肢タイプ）を指定します。結果タイプによって、アクティビティから戻される可能性がある結果のリストが表示されます。完了したアクティビティからの戻り値に応じて、ワークフロー・ダイアグラムが分岐することがあります。3-21 ページの「[選択肢タイプの作成](#)」を参照してください。

アクティビティから値が戻されない場合や、ワークフロー・プロセスが戻り値に依存しない場合は、結果タイプとして「<なし>」を選択できます。

5. この関数アクティビティのコストを指定します。3-58 ページの「[アクティビティ・コスト](#)」を参照してください。
6. アクティビティを識別するアイコンを選択します。.ico ファイルに保存されているアイコンを使用して、アクティビティの処理をシンボル化できます。『Oracle Workflow 管理者ガイド』の「[Oracle Workflow へのカスタム・アイコンの追加](#)」を参照してください。

「参照」を選択すると、ワークフロー・アイコン・サブディレクトリ内のアイコン・ファイルが表示されます。

また、Windows の「エクスプローラ」や「ファイル マネージャ」からアイコン・ファイルをナビゲータ・ツリーのアクティビティ上にドラッグ・アンド・ドロップし、アクティビティにアイコンを設定することもできます。

7. 「適用」を選択して変更内容を保存します。
8. 「詳細」タブを選択し、アクティビティのオプションの詳細を表示して変更します。3-70 ページの「[オプションのアクティビティ詳細の定義](#)」を参照してください。
9. 「ロール」タブを選択し、この関数アクティビティにアクセスできるロールを指定します。（この機能は、今後のリリースでサポートされます。）

10. 「アクセス」タブを選択し、この関数を変更できるアクセス・レベルを設定します。3-18 ページの「[オブジェクトへのアクセス許可の設定](#)」を参照してください。
11. 関数アクティビティが、ナビゲータ・ツリーで「関数」の下に表示されます。ナビゲータ・ツリーでアクティビティをダブルクリックするか、アクティビティを選択して「編集」メニューから「プロパティ」を選択するか、キーボードで [Enter] キーを押すと、いつでもこのアクティビティのプロパティを検討または編集できます。
12. 作成する関数に入力引数が必要な場合は、その引数を関数アクティビティの属性として Oracle Workflow Builder に渡すことができます。関数アクティビティ属性は、プロセスでそのアクティビティが使用されるたびに変更可能な値を持つパラメータとして機能します。関数アクティビティ属性は、関数アクティビティに限定され、プロセスに対してグローバルではありません。3-8 ページの「[項目タイプまたはアクティビティ属性の定義](#)」を参照してください。

項目タイプ属性を参照する関数アクティビティ属性を作成するには、参照される項目タイプ属性をナビゲータ・ツリーで選択し、マウスの左ボタンを押したまま関数アクティビティにドラッグします。「デフォルト」の「値」フィールドが自動的に「項目属性」に設定され、元の項目属性が参照されます。

プロセスにノードとして関数アクティビティを含めると、そのノードに固有の値を関数アクティビティ属性に割り当てることができます。4-19 ページの「[アクティビティ属性値の定義](#)」を参照してください。

関連項目：

2-9 ページ「[プロパティ画面の「編集」ボタンの使用](#)」

▶ イベント・アクティビティの作成

The screenshot shows a dialog box titled "Navigator Control Properties" with a close button (X) in the top right corner. It has four tabs: "Activity", "Details", "Roles", and "Access". The "Activity" tab is active. The fields are as follows:

- Internal Name: []
- Display Name: []
- Description: []
- Icon: [EVENT.ICO] (with a lightning bolt icon and a "Browse" button)
- Event Action: [Receive]
- Event Filter: []
- Cost: [0.00]

At the bottom, there are four buttons: "OK", "Cancel", "Apply", and "Help".

1. ナビゲータ・ツリーで、イベントを作成する項目タイプを選択し、「編集」メニューから「新規イベント」を選択します。「アクティビティ」プロパティ画面で、イベント・アクティビティを定義します。
2. イベント・アクティビティには、内部名（すべて大文字で空白を含みません）と表示名が必要です。表示名は、プロセス・ダイアグラムに表示される変更可能な名前です。説明を使用して、このアクティビティに関する説明を入力します。

注意： 定義したアクティビティの内部名を更新するには、wfchact.sql という特別な SQL スクリプトを使用する必要があります。このスクリプトは、設計時にアクティビティの内部名のエラーを訂正する場合にのみ使用してください。プロセスの実行中のインスタンスに関連するアクティビティの名前の変更には、使用しないでください。『Oracle Workflow 管理者ガイド』の「wfchact.sql」を参照してください。

注意： 内部名には、コロン (:) や空白を使用しないでください。

3. アクティビティを識別するアイコンを選択します。.ico ファイルに保存されているアイコンを使用して、アクティビティの処理をシンボル化できます。『Oracle Workflow

管理者ガイド』の「Oracle Workflow へのカスタム・アイコンの追加」を参照してください。

「参照」を選択すると、ワークフロー・アイコン・サブディレクトリ内のアイコン・ファイルが表示されます。

また、Windows の「エクスプローラ」や「ファイル マネージャ」からアイコン・ファイルをナビゲータ・ツリーのアクティビティ上にドラッグ・アンド・ドロップし、アクティビティにアイコンを設定することもできます。

4. アクティビティの「イベント・アクション」を選択します。
 - 受信： ビジネス・イベント・システムからイベントを受信します。
 - 呼出し： ビジネス・イベント・システムにイベントを発行します。
 - 送信： ビジネス・イベント・システムにイベントを再発行せずに、特定のイベント・エージェントのイベントを別のエージェントに直接送信します。

注意： 選択したイベント・アクションに応じて、次のイベントの詳細の一部またはすべてに対して項目タイプ属性を定義する必要があります。

- イベント名
- イベント・キー
- イベント・メッセージ
- イベント・データ
- 送信エージェント
- 宛先エージェント

イベント・アクティビティをノードとしてプロセスに含める場合は、項目タイプ属性を使用して、そのノードに必要なイベントの詳細情報の格納場所または検索場所を指定できます。イベントの詳細に使用する項目タイプ属性は、イベント・アクティビティが関連付けられている項目タイプと同じ項目タイプに関連付ける必要があります。3-8 ページの「[項目タイプまたはアクティビティ属性の定義](#)」および 4-14 ページの「[イベント・ノードのイベントの詳細の定義](#)」を参照してください。

5. 受信イベント・アクティビティを定義している場合は、「イベント・フィルタ」を入力すれば、受信できるイベントを指定できます。
 - 指定したイベントだけを受信するには、完全な内部イベント名を入力します。

注意： イベント・フィルタには 1 つのイベントだけを指定します。イベント・フィルタにイベント・グループを指定することはできません。

- すべてのイベントをアクティビティで受信するには、「イベント・フィルタ」フィールドを空白のままにします。

関連項目：

8-7 ページ [「イベントの定義」](#)

6. アクティビティのオプションのコストを入力します。イベント・アクションが「呼出し」または「送信」のイベント・アクティビティでは、コストを使用して、実行に時間がかかるアクティビティをバックグラウンド・エンジンに委任することができます。3-58 ページの [「アクティビティ・コスト」](#) を参照してください。
7. 「適用」を選択して変更内容を保存します。
8. 「詳細」タブを選択し、アクティビティのオプションの詳細を表示して変更します。3-70 ページの [「オプションのアクティビティ詳細の定義」](#) を参照してください。
9. 「ロール」タブを選択し、この関数アクティビティにアクセスできるロールを指定します。（この機能は、今後のリリースでサポートされます。）
10. 「アクセス」タブを選択し、このイベントを変更できるアクセス・レベルを設定します。3-18 ページの [「オブジェクトへのアクセス許可の設定」](#) を参照してください。
11. イベント・アクティビティが、ナビゲータ・ツリーで「イベント」の下に表示されます。ナビゲータ・ツリーでアクティビティをダブルクリックするか、アクティビティを選択して「編集」メニューから「プロパティ」を選択するか、キーボードで [Enter] キーを押すと、いつでもこのアクティビティのプロパティを検討または編集できます。
12. 呼出しイベント・アクティビティでは、発行したイベントのイベント・メッセージにパラメータを追加する場合、それらのパラメータを呼出しイベント・アクティビティの属性として定義できます。イベントが発生すると、アクティビティ属性がパラメータとしてイベント・メッセージのパラメータ・リストに追加されます。別のプロセスがイベント・メッセージを後で受信した場合、ワークフロー・エンジンはイベント・パラメータをそのプロセスの項目タイプ属性として設定します。アクティビティ属性の値は、プロセス内での呼出しイベント・アクティビティの使用方法にあわせて変更できます。イベント・アクティビティ属性は、イベント・アクティビティに限定され、プロセスに対してグローバルではありません。3-8 ページの [「項目タイプまたはアクティビティ属性の定義」](#) を参照してください。

項目タイプ属性を参照するイベント・アクティビティ属性を作成するには、参照される項目タイプ属性をナビゲータ・ツリーで選択し、マウスの左ボタンを押したままイベント・アクティビティにドラッグします。「デフォルト」の「値」フィールドが自動的に「項目属性」に設定され、元の項目属性が参照されます。

プロセスにノードとしてイベント・アクティビティを含めると、そのノードに固有の値をイベント・アクティビティ属性に割り当てることができます。4-19 ページの [「アクティビティ属性値の定義」](#) を参照してください。

注意： 呼出しイベント・アクティビティでは、現行のワークフロー・プロセスの項目タイプと項目キーも、イベント・メッセージのパラメータ・リストに自動的に設定されます。別のプロセスがイベント・メッセージを後で受信した場合、ワークフロー・エンジンはその項目タイプと項目キーを使用して、イベントを受信するプロセスの親として、イベントを呼び出したプロセスを自動的に設定します。『Oracle Workflow API リファレンス』の「SetItemParent」を参照してください。

▶ プロセス・アクティビティの作成

ワークフロー・プロセス・ダイアグラムを作成する前に、ナビゲータ・ツリーでプロセス・ダイアグラムを表すプロセス・アクティビティを作成する必要があります。

1. ナビゲータ・ツリーで、プロセス・アクティビティを作成する項目タイプを選択し、「編集」メニューから「新規プロセス」を選択します。「アクティビティ」プロパティ画面で、プロセス・アクティビティを定義します。

プロセス・アクティビティが閉じており、それを再表示する場合は、ナビゲータ・ツリーでそのプロセス・アクティビティを選択して、[Enter] キーを押すか、またはマウスを右クリックしてポップアップ・メニューから「プロパティ」を選択します。

2. プロセス・アクティビティには、内部名（すべて大文字で空白を含みません）と表示名が必要です。表示名は、プロセス・ダイアグラムに表示される変換可能な名前です。説明を使用して、このアクティビティに関する説明を入力します。

注意： 定義したアクティビティの内部名を更新するには、`wfchact.sql` という特別な SQL スクリプトを使用する必要があります。このスクリプトは、設計時にアクティビティの内部名のエラーを訂正する場合にのみ使用してください。プロセスの実行中のインスタンスに関連するアクティビティの名前の変更には、使用しないでください。『Oracle Workflow 管理者ガイド』の「`wfchact.sql`」を参照してください。

注意： 内部名には、コロン (:) や空白を使用しないでください。

3. このアクティビティの結果タイプ（事前に定義された選択肢タイプ）を指定します。結果タイプによって、このプロセスから戻される可能性がある結果のリストが表示されます。3-21 ページの「[選択肢タイプの作成](#)」を参照してください。

プロセスの完了に関して特定の結果を記録する必要がない場合は、結果タイプとして「<なし>」を選択できます。

4. アクティビティを識別するアイコンを選択します。ico ファイルに保存されているアイコンを使用して、アクティビティの処理をシンボル化できます。『Oracle Workflow 管理者ガイド』の「[Oracle Workflow へのカスタム・アイコンの追加](#)」を参照してください。

「参照」を選択すると、ワークフロー・アイコン・サブディレクトリ内のアイコン・ファイルが表示されます。

また、Windows の「エクスプローラ」や「ファイル マネージャ」からアイコン・ファイルをナビゲータ・ツリーのアクティビティ上にドラッグ・アンド・ドロップし、アクティビティにアイコンを設定することもできます。

5. 「実行可能」をオンにすると、このアクティビティが表すプロセスを上位レベルのプロセスとして開始し、単独で実行できます。プロセス・アクティビティがサブプロセスであり、上位レベルのプロセスからコールされた場合にかぎり実行するのであれば、「実行可能」をオフにします。『Oracle Workflow API リファレンス』の「[CreateProcess](#)」を参照してください。

注意： Oracle Workflow では、1つのプロセス階層内でサブプロセス・アクティビティを複数回使用することはできません。1つのプロセス内でサブプロセスを2回以上使用する場合は、必要なインスタンスごとにサブプロセスの個別コピーを作成する必要があります。

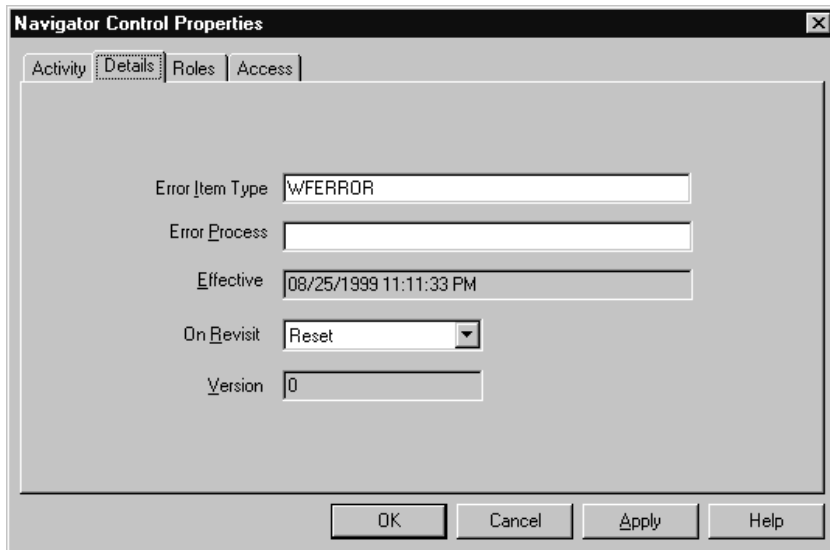
6. 「適用」を選択して変更内容を保存します。
7. 「詳細」タブを選択し、アクティビティのオプションの詳細を表示して変更します。3-70 ページの「[オプションのアクティビティ詳細の定義](#)」を参照してください。

8. 「アクセス」タブを選択し、このプロセスを変更できるアクセス・レベルを設定します。プロセス・アクティビティに対して設定したアクセス・レベルによって、そのプロセス・ダイアグラムにアクセスして編集できるユーザーが決定されます。3-18 ページの「オブジェクトへのアクセス許可の設定」を参照してください。
9. 「適用」を選択して変更内容を保存するか、「OK」を選択し、変更内容を保存してプロパティ画面を閉じるか、「キャンセル」を選択し、変更を取り消してプロパティ画面を閉じます。
10. プロセス・アクティビティが、ナビゲータ・ツリーで「プロセス」の下に表示されます。アクティビティを選択して「編集」メニューから「プロパティ」を選択するか、キーボードで [Enter] キーを押すと、いつでもこのアクティビティのプロパティを参照および編集できます。

関連項目：

2-9 ページ「プロパティ画面の「編集」ボタンの使用」

➤ **オプションのアクティビティ詳細の定義**



1. アクティビティのプロパティ画面の「詳細」タブを選択します。
2. プロセス・アクティビティを作成する場合、現在のプロセスでエラーが発生した場合に実行するエラー・プロセスを指定できます。エラー・プロセスを持つ項目タイプの内部名を「エラー項目タイプ」に入力し、実行するエラー・プロセス・アクティビティの内部名を「エラー・プロセス」に入力してください。エラー・プロセスの項目タイプは、現在の Oracle Workflow Builder セッションでオープンしていなくても、ここで定義で

きることに注意してください。11-2 ページの「ワークフロー・プロセスのエラー処理」を参照してください。

注意： エラーが発生したときに起動されるエラー・プロセスに対して、エラー項目タイプとエラー・プロセスの両方を指定する必要があります。エラー項目タイプのデフォルトは、「システム:エラー」項目タイプの内部名である WFERROR です。この項目タイプ内に用意されている事前定義済のエラー・プロセスの1つを起動したい場合は、そのプロセスの内部名を入力する必要があります。また、カスタムのエラー項目タイプおよびプロセスを入力することもできます。

事前定義済のエラー・プロセスの1つを自分のアクティビティに割り当てる場合は、独自の項目タイプ内に2つの特殊な項目タイプ属性を定義することで、そのエラー・プロセスの動作をカスタマイズすることができます。WF_ADMINISTRATOR 属性を使用して Oracle Workflow がエラー通知を送信する先のロールを指定し、ERROR_TIMEOUT 属性を使用してエラー通知がタイムアウトしたかどうかを指定します。11-4 ページの「項目タイプのエラー通知のカスタマイズ」を参照してください。

3. 「有効日」には、このバージョンのアクティビティをワークフロー・エンジンで実行できるようになる日付が表示されます。「有効日」フィールドが空白の場合、アクティビティはただちに有効になります。

有効日は、「ファイル」メニューの「別名保存」オプションを使用して変更内容を保存するときに設定します。アクティビティに対するすべての変更内容には、保存するときに同じ有効日が適用されます。

4. 「再開封時」に値を指定して、このアクティビティに2度以上推移した場合のワークフロー・エンジンでの処理を決定します。このアクティビティがループ内で再実行される最初のアクティビティの場合は、「再開封時」を設定し、ワークフロー・エンジンによるループの処理方法を指定する必要があります。ループ内の最初のアクティビティは、ピボット・アクティビティとも呼ばれます。ループ内のそれ以外のアクティビティについては、「再開封時」の値は無関係です。

「再開封時」が「無視」に設定されている場合、ワークフロー・エンジンではアクティビティが1回のみ実行され、その後の再実行では無視されます。

「再開封時」が「リセット」に設定されている場合、ワークフロー・エンジンは、ピボット・アクティビティから本来の順序でループを通過して、各アクティビティをCANCEL モードで実行しながらループ内の完了したアクティビティをリセットします。各関数のCANCEL モードには、直前の操作を取り消す特別なロジックを含めることができます。その後、ワークフロー・エンジンはループ内を本来の順序で通過し、ピボット・アクティビティから始めて各アクティビティをRUN モードで再実行します。

「再開封時」が「ループ」に設定されている場合、ピボット・アクティビティと、それに続くループ内の全アクティビティが、リセットされず最初の実行どおりに再度実行されます。『Oracle Workflow API リファレンス』の「ループ」を参照してください。

5. 検証中のアクティビティの改訂が、バージョン番号で識別されます。アクティビティに対する最新の更新が確実に使用されるように、そのアクティビティの有効な最新バージョン番号が使用されます。
6. 「適用」を選択して変更内容を保存します。

▶ **アクティビティのコピー**

1. ナビゲータ・ツリーで、コピーするアクティビティを選択します。
2. マウスの左ボタンを押しながら、アクティビティをコピー先の「項目タイプ」ブランチまでドラッグします。
3. アクティビティを同じ項目タイプ内でコピーすると、プロパティ画面が表示され、コピーしたアクティビティについて一意の新規内部名と表示名の入力を求めるプロンプトが表示されます。

注意：「編集」メニューの「コピー」および「貼り付け」オプションも使用できます。

4. 操作の完了後に「OK」を選択します。

注意：関数アクティビティ、イベント・アクティビティまたは通知アクティビティをコピーすると、それに関連付けられた属性やメッセージもコピーされます。

投票アクティビティ

ロール内のユーザー・グループに通知を送信し、ユーザーからの応答を集計する、投票アクティビティを作成できます。集計結果によって、次のプロセスに進むアクティビティが決まります。

投票アクティビティは通知アクティビティの1つで、最初にユーザー・グループに通知メッセージを送信してから、PL/SQLの通知後関数を実行してユーザーの応答（投票）を集計します。

定義するアクティビティ属性と、通知アクティビティのプロパティ画面にある次の4つのフィールドによって、投票の動作が決定されます。

- 「メッセージ」フィールド
- 「結果タイプ」フィールド
- 「ロールの拡張」チェック・ボックス
- 「関数」フィールド

注意： 投票の通知を受信したユーザーは、その通知を再度割り当てることはできません。

▶ 投票アクティビティの作成

1. 投票アクティビティで集計する応答を含む投票選択肢タイプを作成します。3-21 ページの「[選択肢タイプの作成](#)」を参照してください。
2. 受信者に対して、投票選択肢タイプから1つを選択して応答するように求める投票メッセージを作成します。メッセージの「結果」タブで必要事項を入力します。「結果」タブの選択肢タイプを、手順1で定義した投票選択肢タイプに設定します。3-38 ページの「[メッセージの作成](#)」を参照してください。
3. ナビゲータ・ツリーで、投票アクティビティを作成する項目タイプを選択し、「編集」メニューから「新規通知」を選択します。
4. 内部名（すべて大文字で空白を含みません）と表示名を指定します。説明を使用して、この投票アクティビティに関する説明を入力します。

注意： 定義したアクティビティの内部名を更新するには、wfchact.sql という特別な SQL スクリプトを使用する必要があります。このスクリプトは、設計時にアクティビティの内部名のエラーを訂正する場合にのみ使用してください。プロセスの実行中のインスタンスに関連するアクティビティの名前の変更には、使用しないでください。『Oracle Workflow 管理者ガイド』の「wfchact.sql」を参照してください。

注意： 内部名には、コロン (:) や空白を使用しないでください。

5. 「結果タイプ」フィールドには、投票アクティビティで集計する応答を列挙した選択肢タイプが含まれている必要があります。これは手順 1 で定義した投票選択肢タイプです。
6. 投票アクティビティを識別するアイコンを選択します。
7. 「メッセージ」フィールドで、手順 2 で作成した投票メッセージの名前を選択します。投票メッセージによって、受信者は応答を求められます。応答では、投票選択肢タイプで指定した事前定義済みの値から 1 つが選択されます。
8. ワークフロー・エンジンで、最初に応答したユーザーのみでなくロール内の複数ユーザーからの応答が集計されるように、「ロールの拡張」をオンにします。3-54 ページの「[通知アクティビティ](#)」を参照してください。
9. 「関数」フィールドに、ユーザーからの投票を集計する関数を指定します。標準の「はい / いいえ投票」アクティビティによりコールされる PL/SQL プロシージャ `WF_STANDARD.VOTEFORRESULTTYPE` を使用できます。
`WF_STANDARD.VOTEFORRESULTTYPE` は、一般的な集計関数です。関数により集計される応答候補は、投票アクティビティに指定した結果タイプによって定義されます。関数による応答の集計方法は、投票アクティビティに定義したアクティビティ属性によって決定されます。5-11 ページの「[「はい / いいえ投票」アクティビティ](#)」を参照してください。

別の方法として、カスタムの集計関数を指定することもできますが、これは関数アクティビティの標準 API に準拠している必要があります。プロシージャを指定するには、書式 `<package_name>.<procedure_name>` を使用します。6-3 ページの「[関数アクティビティがコールする PL/SQL プロシージャの標準 API](#)」を参照してください。
10. 「適用」を選択して変更内容を保存します。
11. 「詳細」タブを選択し、アクティビティの「詳細」プロパティ画面を表示して変更します。3-70 ページの「[オプションのアクティビティ詳細の定義](#)」を参照してください。
12. 「ロール」タブを選択し、この通知アクティビティにアクセスできるロールを指定します。（この機能は、今後のリリースでサポートされます。）
13. 「アクセス」タブを選択し、この通知を変更できるアクセス・レベルを設定します。3-18 ページの「[オブジェクトへのアクセス許可の設定](#)」を参照してください。
14. `WF_STANDARD.VOTEFORRESULTTYPE` 集計関数を使用する場合は、投票に予測される応答ごとに「数値」タイプの「カスタム」アクティビティ属性を作成します。投票に予測される各応答は、投票アクティビティの結果タイプに関連付けられた選択肢コードであることに注意してください。そのため、「カスタム」アクティビティ属性を定義する場合は、アクティビティ属性の内部名が選択肢コードの内部名、つまり応答値と一致する必要があります。

アクティビティ属性の値には、空白か、特定の結果に必要な票数のパーセンテージを表す数値を使用できます。パーセンテージを指定すると、その応答の実際の集計パーセンテージの方が指定したパーセンテージより大きい場合に、結果が一致します。アクティビティ属性値を空白にすると、ワークフロー・エンジンではそのアクティビティ属性の応答がデフォルトとして扱われます。つまり、投票の集計後に特定のパーセンテージに達しなければ、空白のアクティビティ属性に関連付けられている応答のうち最高票数を獲得した応答が結果となります。

注意： 集計された投票結果がどの応答パーセンテージにも達しない場合に、デフォルトの応答（空白のアクティビティ属性）が指定されていない場合は、結果は #NOMATCH になります。投票アクティビティからの「一致しない」トランジションが存在する場合、ワークフロー・エンジンではこのトランジションが選択され、それ以外は「デフォルト」が選択されます。「デフォルト」トランジションが存在しない場合は、その結果に使用可能なトランジションがないことを示すエラー (ERROR:#NOTTRANSITION) が発生します。

注意： 集計された投票結果が複数の応答パーセンテージに達するか、応答パーセンテージには達成しないが、デフォルトの応答のうち同じパーセンテージのものがある場合、結果は #TIE になります。投票アクティビティからの「一致」トランジションが存在する場合、ワークフロー・エンジンではこのトランジションが選択され、それ以外は「デフォルト」が選択されます。「デフォルト」トランジションが存在しない場合は、その結果に使用可能なトランジションがないことを示すエラー (ERROR:#NOTTRANSITION) が発生します。

15. 集計関数 WF_STANDARD.VOTEFORRESULTTYPE を使用する場合は、一連の「カスタム」アクティビティ属性を定義するのみでなく、内部名 VOTING_OPTION を指定して「投票オプション」というアクティビティ属性を定義する必要があります。「投票オプション」アクティビティ属性を、「はい/いいえ投票」標準アクティビティからコピーすることもできます。

「投票オプション」アクティビティ属性では、投票の集計方法を指定します。指定できる値は、次のとおりです。

- 「参加者全員の投票を待つ」： ワークフロー・エンジンは、すべての投票が行われるまで待つから、通知を受けた全ユーザーに対する割合として結果を集計します。タイムアウト条件が発生すると、タイムアウト発生前に出された投票の合計に対する割合として投票結果が計算されます。
- 「投票のつど集計」： ワークフロー・エンジンは、通知を受けたすべてのユーザーに対する累積割合を、応答があるたびに集計します。タイムアウト条件が発生すると、応答は合計投票数に対する割合として集計されます。カスタム「応答」アク

アクティビティ属性のどれかの値が空白値の場合、このオプションは意味がなくなるため注意してください。

- 「参加者全員の投票が必要」：すべての投票が行われた後にかぎり、通知を受けた全ユーザーに対する割合で応答が評価されます。タイムアウト条件が発生すると、標準的なタイムアウト・トランジションに従って進むか、使用可能なトランジションが1つも存在しない場合はエラーが発生し、投票結果は集計されません。

関連項目：

5-11 ページ 「はい / いいえ投票」 アクティビティ」

投票方法の例

1. 単純過半数

次の表に、「単純過半数」投票方法の各応答に割り当てられている、カスタムの「応答」アクティビティ属性の値を示します。

表 3-4

応答	カスタムの「応答」アクティビティ属性の値
A	50
B	50
C	50

50 パーセントを超える票を獲得した応答が結果となります。50 パーセントを超える応答が1つもなければ、結果は該当なしとなります (#NOMATCH)。

2. デフォルト付き単純過半数

次の表に、「デフォルト付き単純過半数」投票方法の各応答に割り当てられている、カスタムの「応答」アクティビティ属性の値を示します。

表 3-5

応答	カスタムの「応答」アクティビティ属性の値
A	50
B	50
C	空白

応答 A が 50 パーセントを超える票を得た場合、A が結果となります。同様に、応答 B が 50 パーセントを超える票を得た場合は、B が結果となります。A と B のどちらの得票も 50 パーセントを超えなかった場合は、C が結果となります。

3. 複数のデフォルト付きの単純過半数

次の表に、「複数のデフォルト付きの単純過半数」投票方法の各応答に割り当てられている、カスタムの「応答」アクティビティ属性の値を示します。

表 3-6

応答	カスタムの「応答」アクティビティ属性の値
A	50
B	空白
C	空白

応答 A が 50 パーセントを超える票を得た場合、A が結果となります。A の得票率が 50 パーセント以下の場合は、B または C のうち得票数の高い方が結果となります。A が 50 パーセント以下の得票率で、B と C 両方が同じ数の票を得た場合、結果は一致となります (#TIE)。

4. 人気投票

次の表に、「人気投票」投票方法の各応答に割り当てられている、カスタムの「応答」アクティビティ属性の値を示します。

表 3-7

応答	カスタムの「応答」アクティビティ属性の値
A	空白
B	空白
C	空白

最高得票の応答が結果となります。

5. 反対投票

次の表に、「反対投票」投票方法の各応答に割り当てられている、カスタムの「応答」アクティビティ属性の値を示します。

表 3-8

応答	カスタムの「応答」アクティビティ属性の値
YES	100
NO	0

応答 NO への投票が 1 票でもあれば、結果は NO となります。

6. 評決

次の表に、「評決」投票方法の各応答に割り当てられている、カスタムの「応答」アクティビティ属性の値を示します。

表 3-9

応答	カスタムの「応答」アクティビティ属性の値
GUILTY	100
NOT_GUILTY	100

満場一致の応答が要求され、それ以外は該当なしとなります（#NOMATCH）。

関連項目：

5-11 ページ [「はい / いいえ投票」アクティビティ](#)

Oracle Workflow Builder のオブジェクトの削除

Oracle Workflow Builder のオブジェクトは、他のオブジェクトによって参照されている場合でも、カスタマイズに対して保護されていないければ削除できます。削除するオブジェクトが他のオブジェクトによって参照されている場合は、「ワークフロー・エラー」ダイアログ・ボックスが表示され、失われる外部キー参照についての警告が示されます。そのまま続行してオブジェクトを削除するか、削除のアクションを取り消すことができます。削除を選択すると、ワークフロー・プロセス定義を保存または検証するときに「ワークフロー・エラー」ダイアログ・ボックスが表示され、定義内に存在する失われた外部キー参照がすべて報告されます。

この処理の結果、無効な外部キーを持つワークフロー定義を Oracle Workflow Builder にロードして修正できるようになります。Oracle Workflow Builder では、欠落している外部キーに対する元の内部名参照が保持され、プロセス定義のロード時にそれが検証エラー・メッセージに表示されます。削除したオブジェクトを、元の内部名を使用して元の項目タイプで再作成すると、プロセス定義内で失われた外部キー参照をリストアできます。

注意： Oracle Workflow Builder の項目タイプ定義全体を削除することもできます。

Oracle Workflow Builder のオブジェクトの変更

ワークフロー・オブジェクトの定義を変更する前に、その定義に基づくアクティブな作業項目に悪影響を及ぼさないことを確認する必要があります。Oracle Workflow のオブジェクトに対する変更がアクティブな作業項目に及ぼす影響は、そのオブジェクトがバージョン管理をサポートしているかどうかに応じて異なります。

ワークフロー・オブジェクトの場合、バージョン管理とは、オブジェクト自体またはそれを所有するオブジェクトにより、データベース内の同じオブジェクトの複数のオカレンスがバージョン番号、開始日付および終了日付のみで区別され、サポートされることを意味します。たとえば、次の表の投票アクティビティは、WF_ACTIVITIES 表に同時に存在させることができます。

表 3-10

名前	バージョン	開始日付	終了日付	メッセージ	選択肢タイプ (Lookup Type)
投票	1	1998 年 1 月 1 日	1998 年 1 月 1 日	投票メッセージ	はい / いいえ
投票	2	1998 年 1 月 1 日	< 空白 >	新規投票メッセージ	承認

バージョン管理をサポートしているワークフロー・オブジェクトを変更すると、元のバージョンと新バージョンの両方がデータベースに存在することになります。そのオブジェクトを参照するアクティブな作業項目は、その開始時に有効だったのと同じバージョンを引き続き使用して完了まで進行します。新バージョンを使用するのは、変更後に開始された新規作業項目のみです。

前述の例で、1998 年 1 月 1 日～1998 年 12 月 31 日に開始される作業項目では、1999 年 1 月 1 日までに完了するかどうかに関係なく、結果オプション Yes または No を持つメッセージ「投票メッセージ」が送信されます。結果オプション「承認」または「否認」を含むメッセージ「新規投票メッセージ」が送信されるのは、1999 年 1 月 1 日以後に開始される作業項目のみです。

注意： プロセス定義情報は、すべてバージョン管理対象となります。

ただし、バージョン管理をサポートしていないワークフロー・オブジェクトを変更すると、そのオブジェクトの以前の定義が更新され、データベースには変更後の定義のみが存在することになります。そのオブジェクトを参照するアクティブな作業項目では、いずれも変更後のオブジェクトが使用されます。

変更後のオブジェクトに、作業項目で使用されるワークフロー定義の残りの部分との互換性がなくなると、エラーになることがあります。このようなエラーを回避するには、変更によって互換性が失われないように、変更の計画時にオブジェクトの参照をすべて考慮する必要があります。

注意： 可能であれば、アクティブな作業項目をすべて強制終了し、変更後に再起動して、下位互換性が失われる危険性を回避できます。この方法では、再起動した作業項目では、バージョン管理をサポートするかどうかを問わず、すべてのワークフロー・オブジェクトの変更後の定義が使用されます。

バージョン管理をサポートするワークフロー・オブジェクト

次のワークフロー・オブジェクトでは、バージョン管理がサポートされます。

- 通知
- 関数
- イベント
- プロセスとサブプロセス
- プロセス・アクティビティ（ノード）
- アクティビティ属性
- アクティビティ属性値
- アクティビティ・トランジション

Oracle Workflow Builder の前述のオブジェクトのいずれかを変更してデータベースに保存すると、ワークフロー定義ローダーでは、そのオブジェクトの既存の定義は更新されません。かわりに、そのオブジェクトまたは所有オブジェクトの新バージョンが作成されます。

その結果、変更前に開始されていたアクティブな作業項目には影響を与えずに、これらのオブジェクトを変更できます。

次に例を示します。

- 新規メッセージを参照するように通知アクティビティを更新する場合、その通知はバージョン管理対象となります。
- 新規選択肢タイプを参照するように関数アクティビティを更新する場合、その関数はバージョン管理対象となります。
- 新規 API を参照するように関数アクティビティを更新する場合、その関数はバージョン管理対象となります。
- プロセス・ダイアグラムからプロセス・アクティビティ（ノード）を削除する場合は、そのプロセス内に存在するすべてのプロセス・アクティビティのみでなく、所有プロセスもバージョン管理対象となります。
- アクティビティにアクティビティ属性を追加する場合は、所有アクティビティがバージョン管理対象となります。

前述のどの例でも、変更はそれ以後に開始される作業項目にのみ影響します。

バージョン管理をサポートしないワークフロー・オブジェクト

次のワークフロー・オブジェクトでは、バージョン管理がサポートされません。

- 項目属性
- メッセージ
- 選択肢
- 関数アクティビティにより参照される PL/SQL コード

Oracle Workflow Builder の項目属性、メッセージまたは選択肢のいずれかを変更してデータベースに保存すると、ワークフロー定義ローダーでは、そのオブジェクトの既存の定義が更新されます。また、関数アクティビティの既存の PL/SQL API を変更すると、その関数アクティビティではデータベースに格納された更新後の API が参照されます。

その結果、前述のオブジェクトのいずれかを変更すると、変更結果はそのオブジェクトを参照するアクティブな作業項目すべてに即時に反映されます。変更によって下位互換性が失われないように慎重に計画を立ててください。

注意： Oracle Workflow Builder では、PL/SQL コードの編集はサポートされません。PL/SQL コードは、単に Oracle Workflow の関数アクティビティで参照されるコードを変更した場合の結果を説明するためにのみ列挙してあります。

項目属性

作業項目が開始されると、Oracle Workflow では、その項目タイプに定義された各項目属性のランタイム・コピーが作成されます。ワークフロー・エンジンでは、ワークフロー・プロセス内の関数アクティビティの PL/SQL コードで項目属性が参照されるたびに、これらのランタイム・コピーが参照されます。

作業項目の開始後に新規項目属性を追加しても、アクティブな作業項目には反映されません。ただし、これらの作業項目には、`AddItemAttr()` または `AddItemAttributeArray` API をコールして属性を意図的に作成しないかぎり、新規項目属性は含まれません。ワークフロー・プロセス内の既存の PL/SQL コードで新規項目属性の参照も追加すると、その参照が原因で、新規項目属性を含んでいないアクティブな作業項目にエラーが発生することがあります。

たとえば、`Workflow Engine` API をコールして、関数アクティビティの PL/SQL API を新規項目属性と通信するように変更すると、新規に定義された項目属性を持たないアクティブな作業項目については、関数アクティビティが失敗します。

ワークフロー・プロセス内で既存の PL/SQL コードを変更する場合は、新規項目属性の参照を追加するのは、それを必要とするアクティブな作業項目についても項目属性を作成する場

合に限るように、慎重に計画する必要があります。3-85 ページの「PL/SQL コード」を参照してください。

注意： ただし、ワークフロー・プロセスを起動する API には、新規項目属性の参照を追加でき、アクティブな作業項目に関して特に項目属性を作成する必要はありません。たとえば、`SetItemAttribute` または `SetItemAttributeArray` API をコールして、プロセスの開始時に新規項目属性を作成できます。アクティブな作業項目はすでにこのコードを通過しているため、この種の変更の影響を受けません。

メッセージ

ワークフロー・エンジンが通知システムに対してメッセージの送信を要求すると、通知システムは各メッセージ属性の通知表に通知属性を作成します。通知属性の行には、件名行や本文のテキストなど、実行時にメッセージ本文にトークンが置換される変数データが含まれています。

ただし、メッセージ本文は通知表にコピーされません。かわりに、メッセージ本文は、通知がユーザーに表示されるときに、様々な通知システム API により実行時に参照されます。したがって、メッセージ本文の変更は、変更後に送信される通知のみでなく、変更前に送信されたアクティブな作業項目の通知にも影響します。

非互換性エラーが発生する危険性を回避して、メッセージ本文に特定のタイプの変更を加えることができます。これには、次のような変更が含まれます。

- 静的テキストの追加
- 静的テキストの編集
- 静的テキストの削除
- メッセージ属性トークンの削除

たとえば、メッセージ本文に「5 日以内に承認してください。」などの静的センテンスを追加すると、アクティブな作業項目内のすべての通知に、アクセス時にこのセンテンスが組み込まれます。通知システムでは、追加のセンテンスの解決には他の情報は不要なため、変更後のメッセージ本文をエラーなしで表示できます。

ただし、新規に作成されたメッセージ属性用のトークンを追加するなど、不適切な変更を加えると、アクティブな作業項目内の通知が正常に表示されなくなることがあります。変更は、エラーを回避できるように慎重に計画してください。

新規メッセージ属性のトークンをメッセージ本文に追加する必要がある場合は、既存のメッセージを変更するのではなく、新規メッセージを作成して変更を実装する必要があります。通知アクティビティではバージョン管理がサポートされるため、アクティブな作業項目に影響を与えずに、既存の通知アクティビティに新規メッセージを添付できます。

たとえば、「承認者名」などの新規メッセージ属性を作成し、その属性のトークンをメッセージ本文に追加すると、アクティブな作業項目内のすべての通知には、アクセス時に新規

トークンが組み込まれます。ただし、変更前に送信された通知には、通知属性として新規メッセージ属性「承認者名」は組み込まれません。通知システムでは、新規メッセージ属性の参照を解決できず、かわりに通知にはトークン「&APPROVER_NAME」が表示されません。

この例では、元のメッセージ本文を変更するかわりに、新規メッセージ属性を含む新規メッセージを作成し、新規属性のトークンを新規メッセージの本文に追加して、新規メッセージを通知アクティビティに添付する必要があります。変更結果を保存すると、通知アクティビティがバージョン管理対象となります。その後、アクティブな作業項目は引き続き通知アクティビティの旧バージョンを参照し、互換性のないトークンは通知に表示されなくなります。

選択肢タイプおよびコード

Oracle Workflow では、選択肢タイプには次の重要な用途があります。

- ワークフロー・アクティビティに考えられる完了ステータス（選択肢コード）を決定します。
- 「応答」メッセージ属性に考えられる完了ステータス（選択肢コード）を決定します。

選択肢タイプに不適切な変更を加えると、その選択肢タイプを参照するアクティブな作業項目が失敗することがあります。

下位互換性の消失によるエラーを回避するには、アクティブな作業項目により参照される選択肢タイプについて、次のガイドラインに従ってください。

- 選択肢タイプを削除しないこと。
- 既存の選択肢タイプから選択肢コードを削除しないこと。
- 既存の選択肢タイプに選択肢コードを追加しないこと。

前述のいずれかの変更を行う必要がある場合は、既存の選択肢タイプを変更するのではなく、新規の選択肢タイプを作成して変更を実装する必要があります。

アクティビティではバージョン管理がサポートされるため、既存のアクティビティには、アクティブな作業項目に影響を与えずに新規の選択肢タイプを添付できます。ただし、ワークフロー・メッセージではバージョン管理がサポートされないため、既存のメッセージ属性には新規の選択肢タイプを添付しないでください。

次の例は、選択肢タイプに不適切な変更を加えたために発生する一部のエラーを示しています。

- 「応答」メッセージ属性で参照される選択肢タイプに選択肢コードを追加すると、ユーザーは通知への応答として新規の選択肢コードを選択できるようになります。ただし、通知アクティビティの旧バージョンでは、新規の選択肢コード用の分離ロジックがモデル化されていません。ユーザーが新規コードを選択すると、プロセスは「ERROR」になります。

- 「応答」メッセージ属性で参照される選択肢タイプから選択肢コードを削除すると、ユーザーはその結果コードを通知への応答として選択できなくなります。

PL/SQL コード

関数アクティビティではバージョン管理がサポートされますが、基礎となる PL/SQL コードでは、コード自体のバージョン管理を実装しないかぎり、バージョン管理がサポートされません。バージョン管理なしで PL/SQL コードを変更すると、そのコードを参照するアクティブな作業項目のビジネス・フローが変化します。不適切な変更を行うと、これらの作業項目が失敗することがあります。

関数アクティビティに関する PL/SQL API の変更がアクティブな作業項目に影響しないようにするには、既存の API を変更するのではなく、新規 API を作成して変更を実装する必要があります。関数アクティビティではバージョン管理がサポートされるため、アクティブな作業項目に影響を与えずに、既存の関数アクティビティに新規 API を添付できます。

既存の API を変更する必要があり、かわりに新規 API を作成できない場合は、変更によって下位互換性が失われないように慎重に計画する必要があります。

たとえば、API の戻り値のグループに選択肢コードを追加する予定であれば、最初に関数アクティビティ・ノードに「デフォルト」などの出力トランジションがあり、API がその値を戻すときにワークフロー・エンジンでたどれることを確認する必要があります。この条件が満たされていないければ、その値が戻されるとプロセスが「ERROR」になります。適切な出力トランジションがない場合は、新規 API に変更を実装し、追加の戻り値について分岐ロジックをモデル化するようにプロセスを更新する必要があります。

たとえば、Workflow Engine API をコールして、関数アクティビティの既存の PL/SQL API を新規項目属性と通信するように変更する場合は、アクティブな作業項目について新規項目属性も作成する必要があります。そうでないと、新規項目属性が定義されていないアクティブな作業項目については、関数アクティビティが失敗します。

新規作成された項目属性をパラメータとして持つ次の API のいずれかをコールする場合に、アクティブな作業項目にその項目属性が定義されていなければ、関数アクティビティは失敗することがあります。

- `wf_engine.SetItemAttrText`
- `wf_engine.SetItemAttrNumber`
- `wf_engine.SetItemAttrDate`
- `wf_engine.SetItemAttrEvent`
- `wf_engine.SetItemAttrFormattedDate`
- `wf_engine.SetItemAttrDocument`
- `wf_engine.SetItemAttrTextArray`
- `wf_engine.SetItemAttrNumberArray`

- wf_engine.SetItemAttrDateArray
- wf_engine.GetItemAttrText
- wf_engine.GetItemAttrNumber
- wf_engine.GetItemAttrDate
- wf_engine.GetItemAttrEvent
- wf_engine.GetItemAttrDocument
- wf_engine.GetItemAttrClob
- wf_engine.GetItemAttrInfo

アクティブな作業項目について新規項目属性のコピーを作成するには、AddItemAttr() か、AddItemAttributeArray API の 1 つをコールします。このコールは、カスタム・アップグレード・スクリプトまたは例外ハンドラに挿入できます。

- **アップグレード・スクリプト:** API を変更する前に、その API を参照するアクティブな作業項目について、新規項目属性を作成して挿入するカスタム・アップグレード・スクリプトを記述し、実行します。次の例は、アップグレード・スクリプトの記述方法を示しています。

```
for <each active work item>
  begin
    wf_engine.AddItemAttr(itemtype,
                          itemkey,
                          '<new_attribute_name>');
    wf_engine.SetItemAttrText(itemtype,
                              itemkey,
                              '<new_attribute_name>',
                              '<New attribute value>');
  end;
end loop;
```

注意: アクティブな作業項目は、WF_ITEMS.END_DATE が NULL である項目として識別されます。

- **例外ハンドラ:** 変更後の API で、新規項目属性の参照の後に、属性が存在しない場合に作成して挿入する例外ハンドラを追加します。次の例は、このような例外ハンドラの記述方法を示しています。

```
procedure <procedure_name>(
  itemtype in varchar2,
  itemkey in varchar2,
  actid in number,
  funcmode in varchar2,
```

```
    result in out varchar2)
is
begin
    --
    -- RUN mode - normal process execution
    --
    if (funcmode = 'RUN') then
        -- your run code goes here
        null;
        wf_engine.SetItemAttrText(itemtype,
                                   itemkey,
                                   '<existing_attribute_name>',
                                   '<Existing attribute value>');
        begin
            wf_engine.SetItemAttrText(itemtype,
                                       itemkey,
                                       '<new_attribute_name>',
                                       '<New attribute value>');
        exception
        when others then
            if (wf_core.error_name = 'WFENG_ITEM_ATTR') then
                wf_engine.AddItemAttr(itemtype,
                                       itemkey,
                                       '<new_attribute_name>');
                wf_engine.setitemattrtext(itemtype,
                                           itemkey,
                                           '<new_attribute_name>',
                                           '<New attribute value>');
            else
                raise;
            end if;
        end;
        -- example completion
        result := 'COMPLETE: ';
        return;
    end if;

    --
    -- CANCEL mode - activity 'compensation'
    --
    -- This is in the event that the activity must be undone,
    -- for example when a process is reset to an earlier point
    -- due to a loop back.
    --
    if (funcmode = 'CANCEL') then
```

```
-- your cancel code goes here
null;
-- no result needed
result := 'COMPLETE';
return;
end if;

--
-- Other execution modes may be created in the future. Your
-- activity will indicate that it does not implement a mode
-- by returning null
--
result := '';
return;

exception
when others then
-- The line below records this function call in the error
-- system in the case of an exception.
wf_core.context('<package_name>',
                '<procedure_name>',
                itemtype,
                itemkey,
                to_char(actid),
                funcmode);
raise;
end <procedure_name>;
```

関連項目：

3-82 ページ [「項目属性」](#)

ワークフロー・プロセス・ダイアグラムの定義

この章では、Oracle Workflow Builder を使用してワークフロー・プロセス・ダイアグラムを定義する方法と、特定のロールに通知アクティビティを割り当てることができるように、データベースからロールをロードする方法について説明します。

「プロセス」ウィンドウ

Oracle Workflow Builder の「プロセス」ウィンドウには、特定のプロセスのアクティビティ（アイコン）とトランジション（矢印）がグラフィカルに表示されます。各アクティビティはノードであり、プロセス完了のために実行される論理的な手順です。

ナビゲータ・ツリーからアクティビティを「プロセス」ウィンドウにドラッグ・アンド・ドロップするか、または「プロセス」ウィンドウに直接アクティビティを作成します。アクティビティ・ノードのプロパティは、「プロセス」ウィンドウのノードをマウスの左ボタンでダブルクリックして表示したり編集します。マウスの右ボタンを使用してアクティビティのノード間を矢印で結び、トランジションを定義します。

プロセス・ノードは、通知アクティビティ、関数アクティビティ、イベント・アクティビティおよびプロセス・アクティビティで構成されます。プロセスのダイアグラムで、プロセスにプロセス・アクティビティが含まれている場合、そのプロセス・アクティビティをサブプロセスと呼びます。この階層の深さには制限はありません。サブプロセス・ダイアグラムを「プロセス」ウィンドウに表示するには、上位の「プロセス」ウィンドウでサブプロセス・アクティビティ・ノードをダブルクリックします。

トランジション

トランジションは、ダイアグラムには矢印で表示され、あるアクティビティが完了して別のアクティビティがアクティブになることを示します。結果タイプが「<なし>」で完了したアクティビティの場合、そこからのトランジションは単に次のアクティビティへの矢印として表示されます。これは、元のアクティビティが完了すると、プロセスが次のアクティビティに進むことを示します。

結果タイプが定義されているアクティビティの場合は、作成するトランジションの矢印をアクティビティに考えられる結果のうちの 1 つに関連付ける必要があります。アクティビティの完了時に戻される結果によって、次に適切なアクティビティが決定されます。次に適切なアクティビティは、完了したアクティビティを起点とする結果ベースのトランジションによって定義されます。たとえば、「承認者に通知」の結果が「否認済」の場合は、「購買申請を却下」に進みます。10-15 ページの「[購買申請プロセス・アクティビティ](#)」を参照してください。

また、結果タイプが定義されているアクティビティについては、デフォルト、任意またはタイムアウトのトランジションを作成できます。ワークフロー・エンジンは、完了結果と一致するトランジションが他になければ、デフォルト・トランジションに従います。任意トランジションには、アクティビティが戻す完了結果にかかわらず従います。そのため、特定の結果に従うアクティビティと並行してワークフロー・エンジンで実行される一般的なアクティビティをプロセスに含めることが可能です。ワークフロー・エンジンは、通知アクティビティが完了前にタイムアウトになると、タイムアウト・トランジションに従います。『Oracle Workflow 管理者ガイド』の「[バックグラウンドのワークフロー・エンジンの設定](#)」を参照してください。

アクティビティは 1 つの結果に対して複数のトランジションを持ち、並列分岐を作成できます。

タイムアウト・トランジション

通知アクティビティが指定した時間内に完了しない場合に、プロセスで別のアクティビティを強制的に実行させるには、その通知アクティビティと別のアクティビティの間をタイムアウト・トランジションで結びます。4-9 ページの「[プロセスのノードの定義](#)」を参照してください。

アクティビティがタイムアウトになると、Oracle Workflow ではそのアクティビティがタイムアウトになったとしてマークされ、そのアクティビティに関連するすべての通知が取り消されます。取り消された通知に応答が必要で、実行者が通知を電子メールで受信する場合にのみ、通知システムでは実行者に取消メッセージを送信します。

注意： Oracle Applications に組み込まれている Workflow を使用している場合は、オプションで「WF: メーラー取消 E メール」プロファイル・オプションを使用して、通知メーラーが通知キャンセルのメッセージを送信しないようにすることができます。『Oracle Workflow 管理者ガイド』の「[通知メーラーの設定](#)」を参照してください。

その後、処理はプロセス定義で指定されたとおりタイムアウト・トランジションに沿って続行されます。タイムアウトになったアクティビティに、そのアクティビティを起点とするタイムアウト・トランジションがなければ、Oracle Workflow では、そのアクティビティまたはその親プロセスに関連したエラー・プロセスが実行されます。3-70 ページの「[オプションのアクティビティ詳細の定義](#)」を参照してください。

注意： タイムアウトになったアクティビティを処理するには、事前にバックグラウンド・エンジンを設定する必要があります。『Oracle Workflow 管理者ガイド』の「[バックグラウンドのワークフロー・エンジンの設定](#)」を参照してください。

単一アクティビティへの複数トランジションの作成

プロセス・ダイアグラムで、1つのアクティビティに至る複数のトランジションを作成できます。このような複数のトランジションは、プロセスが1つのノードに進む流れが複数あっても、そのノードを実行するのは1度のみであることを示す場合があります。

また、複数のトランジションは、そのアクティビティがループの開始点になるため、そこに複数回進むことを示す場合もあります。このような場合は、そこに到達するたびにそのアクティビティを再実行します。

アクティビティの「再開封時」フラグによって、そのアクティビティに複数回到達した場合に、それを再実行するかどうかが決まります。これは、ループのピボット・アクティビティに関して重要なフラグです。「再開封時」は、アクティビティの「詳細」プロパティ画面で初期設定されます。ただし、プロセスでアクティビティを使用するたびに、「ノード」プロパティ画面でそのノードに対する「再開封時」の設定を変更できます。プロセスがループ内を進む回数を制御する初期アクティビティとして、標準の「ループ・カウンタ」アクティビティも使用できます。『Oracle Workflow API リファレンス』の「ループ」および5-7ページの「[「ループ・カウンタ」アクティビティ](#)」を参照してください。

提案： 並列分岐からの複数の入力トランジションがある場合は、常にAND、OR またはカスタムの「結合」アクティビティを使用して、これらの分岐をマージする必要があります。特に、並列分岐をマージした後で、プロセスにループを作成する場合は、この操作が必要です。「結合」アクティビティを使用して並列分岐をマージし、後続のアクティビティをループの開始点として指定すると、エンジンによって実行されるプロセスをさらに簡素化して作成できます。5-2 ページの「[標準アクティビティ](#)」を参照してください。

「開始」アクティビティと「終了」アクティビティの指定

各プロセスには、そのプロセスの開始点を識別する「開始」アクティビティが必要です。論理的にプロセスを開始するノードであれば、どんなノードでも「開始」アクティビティに指定できます。プロセスの開始時に、ワークフロー・エンジンは入力トランジションがない（そのアクティビティを指す矢印がない）「開始」アクティビティから開始します。複数の「開始」アクティビティがある場合は、実行可能な各「開始」アクティビティが実行され、「終了」の結果に到達するまでプロセス内を進みます。実行できる「開始」アクティビティは、任意の順序で実行されます。プロセスには、それぞれ「終了」アクティビティを持つ複数の分岐が含まれていることがあります。ワークフロー・エンジンが「終了」アクティビティに到達すると、まだ処理中の並列分岐があっても、プロセス全体が終了します。

「終了」アクティビティは、プロセスの完了結果を表す結果に戻す必要があります。結果は、そのプロセス・アクティビティの結果タイプの値の1つです。

「プロセス」ウィンドウでは、「開始」アクティビティには小さい緑の矢印マーク、「終了」アクティビティにはアクティビティ・ノードのアイコンの右下隅に赤の矢印マークが付いた状態で表示されます。

プロセスの実行

ワークフロー・プロセスは、アプリケーションがワークフロー・エンジンの `CreateProcess()` および `StartProcess()` API をコールするか、ビジネス・イベント・システムのサブスクリプションがプロセスを起動するイベントを送信したときに開始されます。サブプロセスは、ワークフロー・エンジンがサブプロセスを表すプロセス・アクティビティに進んだときに開始されます。

ビジネス・イベント・システムを使用してワークフロー・プロセスを起動するには、次の手順に従います。

1. ビジネス・イベントを定義します。
2. このビジネス・イベントに対するサブスクリプションを定義します。サブスクリプションのプロパティには、起動するワークフローの項目タイプとプロセスを指定します。

デフォルトでは、起動するワークフロー・プロセスの項目キーとして、イベント・キーが使用されます。カスタム・ルールに基づいて項目キーを生成する場合は、任意の項目キーを使用して関連 ID をイベント・メッセージに生成する関数を作成し、サブスクリプションのルール関数として割り当てます。

3. ワークフロー・プロセスの起動場所で、`Raise()` API をカスタム・アプリケーション・コードに追加します。

注意： プロセス・インスタンスの項目キーには、1バイトの文字のみを含めることができます。複数バイトの値を含めることはできません。

関連項目：

『Oracle Workflow API リファレンス』の「Workflow Engine API」

『Oracle Workflow API リファレンス』の「Raise」

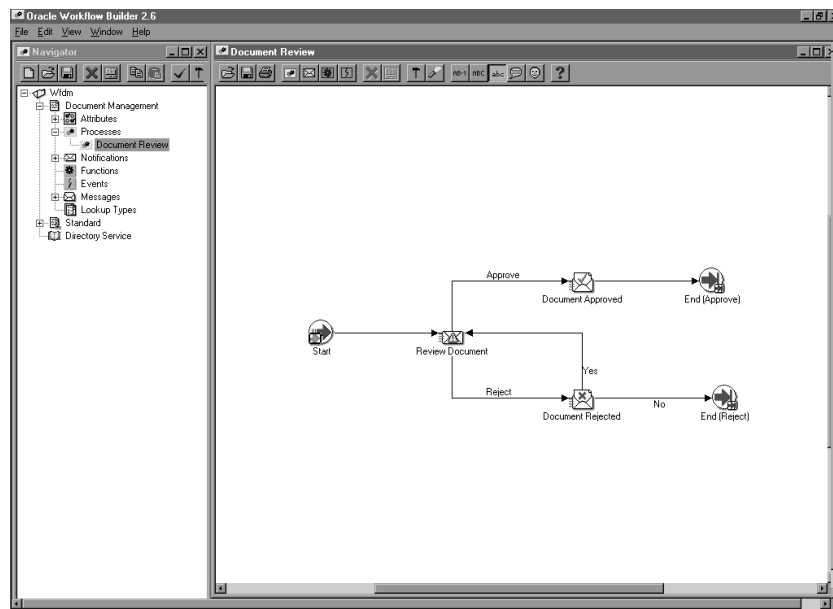
8-2 ページ「ビジネス・イベントの管理」

プロセスの図示

ここでは、「プロセス」ウィンドウでワークフロー・プロセスを作成して定義する方法について説明します。

- 4-7 ページ「ワークフロー・プロセスへのノードの追加」
- 4-9 ページ「プロセスのノードの定義」
- 4-14 ページ「イベント・ノードのイベントの詳細の定義」
- 4-19 ページ「アクティビティ属性値の定義」
- 4-20 ページ「トランジションの作成と編集」
- 4-21 ページ「プロセス概要の表示」
- 4-22 ページ「プロセスの印刷」
- 4-22 ページ「プロセス・ダイアグラムをクリップボードにコピー」
- 4-23 ページ「プロセス定義の検証」

▶ ワークフロー・プロセスへのノードの追加



1. プロセス・ダイアグラムの作成を開始するには、最初にプロセス・アクティビティの「プロセス」ウィンドウを表示します。「プロセス」ウィンドウは、次のいずれかの方法でオープンできます。
 - ナビゲータ・ツリーで、事前定義済みのプロセス・アクティビティをダブルクリックします。
 - 事前定義済みのプロセス・アクティビティを選択して、[Ctrl]+[E] キーを押します。
 - 事前定義済みのプロセス・アクティビティを選択し、「編集」メニューから「プロセスの詳細」を選択します。
 - クイック・スタート・ウィザードを使用して、新規のプロセス・アクティビティを作成します。2-19 ページの「[クイック・スタート・ウィザードの使用](#)」を参照してください。「プロセス」ウィンドウが表示され、ウィンドウ・タイトルにプロセス名が表示されます。
2. 次のいずれかの方法を使用して、プロセスに新規ノードを作成します。
 - ナビゲータ・ツリーから「プロセス」ウィンドウへ、通知、関数、イベントまたはプロセス・アクティビティをドラッグ・アンド・ドロップします。ドラッグするア

クティビティは、ドロップ先のプロセスと同じデータ・ストアに属している必要があります。

注意： ドロップ先のプロセスと異なるデータ・ストアにあるアクティビティをプロセスにドラッグする場合は、最初にそのアクティビティが属する項目タイプを、このプロセスと同じデータ・ストアにコピーしてください。

- 「新規関数」、「新規プロセス」、「新規イベント」または「新規通知」のツールバー・ボタンを選択して、新規のアクティビティを作成します。
 - カーソルを「プロセス」ウィンドウに置き、マウスの右ボタン・メニューから「アクティビティ作成」を選択して新規のアクティビティ・ノードを作成します。
3. また、マウスの右ボタン・メニューを使用して新規ノードを作成することもできます。新規の関数、通知、イベントまたはプロセスを作成できます。「アクティビティ」プロパティ画面が表示され、このノードのアクティビティを選択します。4-9 ページの「[プロセスのノードの定義](#)」を参照してください。
 4. 「プロセス」ウィンドウで、カーソルをアクティビティに合せると、そのアクティビティに関する情報を表示できます。「ラベル名」、「内部名」、「表示名」、「コメント」および「実行者」は、ポップアップ・ヒントの形式で表示されます。
 5. 「プロセス」ウィンドウでアクティビティ・ノードをシングル・クリックすると、ナビゲータ・ツリーが拡張され、選択したノードのマスター・アクティビティがハイライト表示されます。
 6. マウスの右ボタンを押したままソース・アクティビティから宛先アクティビティへドラッグし、2つのアクティビティ・ノード間にトランジション（矢印）を作成します。
 7. ソース・アクティビティに結果コードが関連付けられていなければ、デフォルトではトランジションにラベルは表示されません。この種のトランジションのラベルを表示するように特定して選択すると、「デフォルト」というラベルが表示されます。4-20 ページの「[トランジションの作成と編集](#)」を参照してください。

ソース・アクティビティに結果コードが関連付けられている場合は、宛先アクティビティへのトランジションを作成しようとする時、選択肢の値リストが表示されます。トランジションに割り当てる値を選択します。また、アクティビティから他のトランジションの結果と一致しない結果が戻された場合や、アクティビティからなんらかの結果が戻された場合、またはアクティビティがタイムアウトになった場合のトランジションを、それぞれ値「デフォルト」、「任意」または「タイムアウト」を選択して定義することもできます。

ナビゲータ・ツリーから「プロセス」ウィンドウの既存のトランジションに選択肢コードをドラッグ・アンド・ドロップし、そのトランジションの結果を変更することもできます。ドラッグ・アンド・ドロップする選択肢コードは、置き換える選択肢コードと同じデータ・ストアに属し、同じ選択肢タイプである必要があります。

- プロセス・ダイアグラムで、複数のアクティビティ・ノードとトランジションを含むリージョン全体を選択し、[Ctrl] または [Shift] キーを押したまま選択範囲を「プロセス」ウィンドウの別の位置にドラッグして、選択範囲のコピーを作成できます。

注意： Oracle Workflow では、1つのプロセス階層内でサブプロセス・アクティビティを複数回使用することはできません。1つのプロセス内でサブプロセスを2回以上使用する場合は、必要なインスタンスごとにサブプロセスの個別コピーを作成する必要があります。

- 「表示」メニューで「スナップのグリッド」をオンにして、ダイアグラムの完成時にアクティビティ・アイコンをグリッドに貼り付ける必要があります。デフォルトでは、「スナップのグリッド」はオンになっており、設定を変更するとその設定がデフォルトになります。

関連項目：

B-9 ページ 「「プロセス」ウィンドウのツールバー」

▶ **プロセスのノードの定義**

- プロセス・アクティビティの「プロセス」ウィンドウをオープンします。
- 新規の関数、通知、イベントまたはプロセス・ノードを作成するには、まず「プロセス」ウィンドウのツールバーから「新規関数」、「新規通知」、「新規イベント」または「新規プロセス」のアイコンを選択します。次に、「プロセス」ウィンドウ内で、この新規ノードを配置する位置をクリックします。新規ノードのプロパティ画面が表示されます。

注意： 新規ノードを作成するには、事前に定義されたアクティビティを、ナビゲータ・ツリーから「プロセス」ウィンドウにドラッグ・アンド・ドロップする方法もあります。この方法では、ノードのプロパティ画面に事前定義済の情報が自動的に作成されます。ノードのプロパティ画面を編集する場合は、そのノードをダブルクリックして手順5に進んでください。

- 「項目タイプ」フィールドで、このアクティビティ・ノードを関連付ける項目タイプを選択します。
- 次のどちらかの方法を選択し、ノードの残りの情報を定義します。
 - 事前に定義されたアクティビティの内部名または表示名を選択します。Oracle Workflow Builder により、「ナビゲータ」ウィンドウに表示されたマスター・アクティビティから、すべてのフィールドに事前定義済の情報が移入されます。
 - または、「新規作成」ボタンを選択して新規のアクティビティを定義します。プロパティ画面の他のタブに必要事項を入力する場合は、次の項を参照してください。

- 「プロセス」： 3-68 ページ「プロセス・アクティビティの作成」
- 「関数」： 3-61 ページ「関数アクティビティの作成」
- 「通知」： 3-59 ページ「通知アクティビティの作成」
- 「イベント」： 3-65 ページ「イベント・アクティビティの作成」
- 「詳細」： 3-70 ページ「オプションのアクティビティ詳細の定義」
- 「ルール」： このタブの情報は、現時点ではサポートされていません。
- 「アクセス」： 3-19 ページ「オブジェクトのアクセス・レベルの設定」

注意： これらのタブで行った変更はすべて、マスター・アクティビティに自動的に伝播されるため、そのアクティビティの他のインスタンスがすべて影響を受けます。ただし、「ノード」と「ノード属性」タブ、およびイベント・アクティビティの「イベントの詳細」タブで行った変更は、現在のノード・アクティビティだけに適用されます。

The screenshot shows the 'Navigator Control Properties' dialog box with the 'Node' tab selected. The dialog has several sections:

- Label:** A text input field.
- Start/End:** A dropdown menu currently set to 'Normal'.
- Comment:** A text input field.
- Timeout:** A section containing a 'Type' dropdown menu set to 'No Timeout'.
- Performer:** A section containing a 'Type' dropdown menu set to 'Constant' and a 'Value' dropdown menu set to '<None>'. There is an 'Edit' button next to the 'Value' field.

At the bottom of the dialog are four buttons: 'OK', 'Cancel', 'Apply', and 'Help'.

5. 「ノード」タブを選択し、このノードに固有の情報を指定します。ノードのラベルを指定します。アクティビティは任意のプロセスで複数回使用できるため、「ラベル」フィールドを使用すると、プロセス内で特定のアクティビティのインスタンスに一意の名前を指定できます。デフォルトでは、ラベル名はアクティビティ名ですが、そのアクティビティがプロセスで複数回使用されると、アクティビティ名の後ろに -N が追加されます。N は、使用中のアクティビティの N 番目のインスタンスを表します。

注意：ほとんどの Oracle Workflow API をコールする場合に、アクティビティ名ではなくアクティビティのラベル名を渡す必要があります。『Oracle Workflow API リファレンス』の「Workflow Engine API」を参照してください。

6. 「開始」または「終了」を選択し、現行ノードがプロセスの「開始」アクティビティか「終了」アクティビティかを指定します。どちらでもない場合、デフォルトは「標準」です。プロセスには、複数の「開始」ノードと「終了」ノードを含めることができます。

「開始」アクティビティには（開始）というマークが付き、アクティビティ・アイコンに小さい緑の矢印が表示されます。「終了」アクティビティには（終了）というマークが付き、アクティビティ・アイコンに赤の矢印が表示されます。

注意：「開始 / 終了」フィールドは、デフォルトではすべてのアクティビティ・ノードについて常に「標準」に設定されます。「Standard Start」または「Standard End」アクティビティを使用する場合にも、「開始 / 終了」フィールドは手動でそれぞれ「開始」または「終了」に編集する必要があります。

7. 「終了」ノードの場合、プロセス・アクティビティ全体に結果タイプが関連付けられているときは、プロセスの最終結果の値も選択する必要があります。プロセスの最終結果の値リストは、プロセス・アクティビティの結果タイプとして指定された選択肢タイプから導出されます。
8. このノードに関するコメントを入力できます。
9. 応答が必要な通知、他のプロセス内のサブプロセスであるプロセス・アクティビティ、イベント・アクションが「受信」のイベント・アクティビティ、またはブロック関数アクティビティの場合は、指定した期限までに完了する必要があるかどうかを指定します。指定した期限までにアクティビティが完了しなかった場合、親プロセスを別のアクティビティに進めるようにリダイレクトできます。4-3 ページの「タイムアウト・トランジション」を参照してください。

アクティビティを指定期限までに完了する必要がなければ、「タイムアウトなし」を選択します。

固定の相対時間までにアクティビティを完了する必要がある場合は、「相対時間」を選択します。日数、時間、分の任意の組合せを指定して、アクティビティがタイムアウトとなる期限を指定します。入力した値は、アクティビティ開始日からの分単位の相対オフセット値として解釈されます。相対タイムアウト値を 0 にすると、タイムアウトなしになります。

実行時に動的に算出される一定の相対時間までにアクティビティを完了する必要がある場合は、「項目属性」を選択します。最初に、算出されるタイムアウト値を格納する数値タイプの項目属性を作成し、その項目属性をここで参照する必要があるため注意してください。3-2 ページの「[項目タイプ属性](#)」および 3-8 ページの「[項目タイプまたはアクティビティ属性の定義](#)」を参照してください。

注意： この属性に格納される動的タイムアウト値は、アクティビティ開始日からの分単位の相対オフセット値として解釈されます。相対タイムアウト値を NULL または 0 にすると、タイムアウトなしになります。

10. 通知アクティビティ・ノードの場合、またはイベント・アクションが「送信」のイベント・アクティビティ・ノードの場合は、そのアクティビティのメッセージに割り当てられた優先度を上書きできます。メッセージのデフォルト優先度を維持する場合は、「デフォルト」を選択してください。

デフォルトの優先度を、新しく指定する優先度で上書きする場合は「定数」を選択します。

デフォルトの優先度を、実行時に動的に決定される新しい優先度で上書きする場合は「項目属性」を選択します。最初に、算出される優先度値を格納する数値タイプの項目属性を作成し、その項目属性をここで参照する必要があるため注意してください。3-2 ページの「[項目タイプ属性](#)」および 3-8 ページの「[項目タイプまたはアクティビティ属性の定義](#)」を参照してください。

注意： 算出される優先度値は、1～99 の任意の数値です。Oracle Workflow ではこの数値が自動的に優先度 1～33= 高、34～66= 標準、67～99= 低に変換されます。

11. 通知アクティビティ・ノードの場合は、アクティビティの実行者を指定します。実行者とは、通知送信先のロールです。固定のロール名、または実行時にロールを動的に決定する項目タイプ属性を選択できます。最初に、算出されるロール名を格納するロール・タイプの項目属性を作成し、その項目属性をここで参照する必要があるため注意してください。3-2 ページの「[項目タイプ属性](#)」、3-8 ページの「[項目タイプまたはアクティビティ属性の定義](#)」および 4-26 ページの「[ロール](#)」を参照してください。

注意：「実行者」の「タイプ」を「定数」に設定し、データベースに接続してルールをロードしている場合は、「実行者」ドロップ・ダウン・リストから固定のルール名を選択できます。データベースへの接続をオープンせずにファイルのデータ・ストアで作業している場合は、「実行者」フィールドに有効なルールの表示名を直接入力できます。このファイルをデータベースにアップロードすると、ルールは入力したルール表示名に基づいて、データベースに格納されている適切なルール・データに解決されます。

注意： 通知をマルチ・ユーザーのルールに割り当てると、実際に通知に応答するルールの各ユーザーが追跡されます。『Oracle Workflow API リファレンス』の「Respond」を参照してください。

注意： Oracle Workflow Builder では、どんなタイプのノード・アクティビティにも実行者を指定できますが、Oracle Workflow では通知アクティビティ・ノードに対する実行者の値のみが考慮されます。

12. 「適用」を選択して変更内容を保存するか、「OK」を選択し、変更内容を保存してプロパティ画面を閉じるか、「キャンセル」を選択し、変更を取り消してプロパティ画面を閉じます。

プロパティ画面を保存して閉じると、「プロセス」ウィンドウの指定した位置に、アクティビティ・ノードが表示されます。これが新規作成したアクティビティであれば、ナビゲータ・ツリーの適切なブランチに、対応するマスター・アクティビティも作成されます。
13. ノードがイベント・アクティビティの場合は、「イベントの詳細」タブを選択して必要なイベント情報を追加できます。4-14 ページの「[イベント・ノードのイベントの詳細の定義](#)」を参照してください。
14. ノードが関数、通知またはイベント・アクティビティのときに、そのアクティビティにアクティビティ属性が指定されている場合は、「ノード属性」タブを選択してアクティビティ属性に値を割り当てることができます。4-19 ページの「[アクティビティ属性値の定義](#)」を参照してください。
15. ノードがプロセス・アクティビティの場合は、プロセス・アクティビティ・アイコンの右上隅に、小型のサブプロセス・アイコンが表示されます。サブプロセス・アイコンは、そのノードがプロセス・ダイアグラム内のサブプロセスであることを示します。

関連項目：

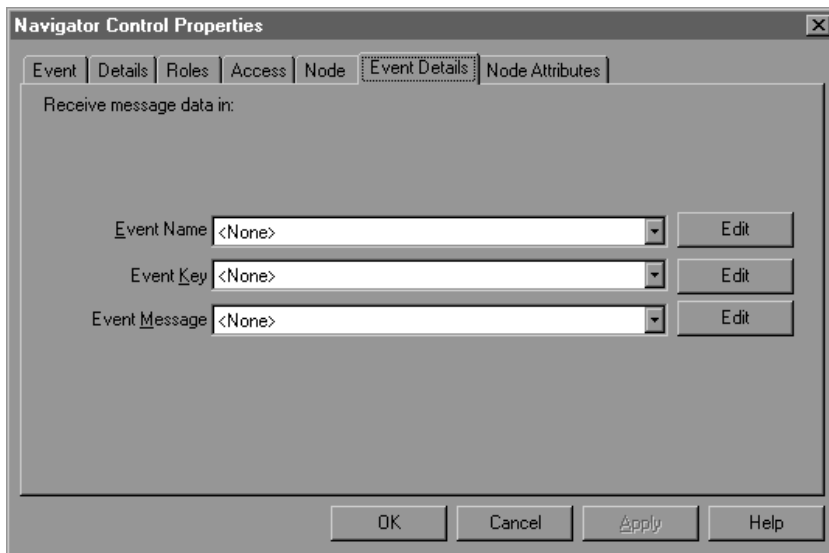
2-7 ページ「ナビゲータ・ツリーでのオブジェクトの検索」

2-9 ページ「プロパティ画面の「編集」ボタンの使用」

▶ **イベント・ノードのイベントの詳細の定義**

イベント・アクティビティに定義したイベント・アクションに基づいて、イベント・ノードにイベントの詳細を定義します。項目タイプ属性を使用して、格納または取得するイベントの詳細情報を定義します。一部の項目は、必ず定義する必要があります。事前定義済の項目タイプ属性を参照するには、適切なタイプの項目タイプ属性をあらかじめ作成する必要があります。イベントの詳細に使用する項目タイプ属性は、イベント・アクティビティが関連付けられている項目タイプと同じ項目タイプに関連付ける必要があります。3-2 ページの「[項目タイプ属性](#)」および 3-8 ページの「[項目タイプまたはアクティビティ属性の定義](#)」を参照してください。

1. イベント・アクティビティ・ノードのプロパティ画面を表示します。「イベントの詳細」タブを選択します。



2. イベント・アクションが「受信」のアクティビティの場合、次のイベントの詳細を入力します。
 - 「イベント名」：（オプション）テキスト・タイプの項目タイプ属性を選択します。ノードが受信したイベント名を格納します。

注意： 受信イベント・アクティビティは、イベント・フィルタとして指定したイベント名と一致するイベントだけを受信できます。3-65 ページの「[イベント・アクティビティの作成](#)」を参照してください。

- 「イベント・キー」：（オプション）テキスト・タイプの項目タイプ属性を選択します。ノードが受信したイベント・キーを格納します。
- 「イベント・メッセージ」：（オプション）イベント・タイプの項目タイプ属性を選択します。ノードが受信したイベント・メッセージを格納します。

注意： 受信アクティビティがイベントを受信すると、ワークフロー・エンジン は、指定されたイベント名、イベント・キーおよびイベント・メッセージを項目タイプ属性に格納します。パラメータに対応する属性がまだ存在しない場合は、新しい項目属性を作成して、イベント・メッセージのパラメータ・リストに含まれるすべてのパラメータをプロセスの項目タイプ属性として設定します。3-70 ページの「[オプションのアクティビティ詳細の定義](#)」を参照してください。

また、イベントが別のワークフロー・プロセスの呼出しイベント・アクティビティによって発生した場合は、そのプロセスの項目タイプと項目キーがイベント・メッセージ内のパラメータ・リストに追加されます。ワークフロー・エンジンは、イベントを受信するプロセスの親として、指定されたプロセスを自動的に設定し、既存の親の設定を上書きします。『Oracle Workflow API リファレンス』の「SetItemParent」を参照してください。

3. イベント・アクションが「呼出し」のアクティビティの場合、次のイベントの詳細を入力します。
 - 「イベント名」： ノードが呼び出すイベントの名前。定形のイベント名を指定するか、テキスト・タイプの項目タイプ属性を選択します。後者の場合、実行時にイベント名を動的に決定できます。

注意： 呼び出せるイベントは1つだけです。イベント・グループは呼び出せません。

- 「イベント・キー」： テキスト・タイプの項目タイプ属性を選択します。ノードが呼び出したイベントのイベント・キーを格納します。
- 「イベント・データ」： (オプション) 項目タイプ属性を選択します。ノードが呼び出したイベントのイベント・データを格納します。イベント・データは、テキスト、数値、日付、選択肢、ロール、または属性タイプの項目タイプ属性に保存することができます。ただし、イベント・データはイベント属性タイプのための形式であるイベント・メッセージ構造の一部にすぎないため、イベント・タイプの項目タイプ属性を使用することはできません。

テキスト属性に入力できるデータ長は、最大 4000 バイトです。イベント・データが 4000 バイトを超える場合は、テキスト属性でイベント・データを渡さず、イベント定義にジェネレート関数を割り当ててイベント・データを生成する必要があります。8-7 ページの「[イベントの定義](#)」を参照してください。

注意：「呼出し」アクティビティでは、「イベント名」および「イベント・キー」が必須です。

注意： 呼出しイベント・アクティビティでは、これらのイベントの詳細以外に、ノードのアクティビティ属性を使用して、イベント・メッセージのパラメータ・リストに追加するパラメータを指定できます。別のプロセスがイベント・メッセージを後で受信した場合、ワークフロー・エンジンはイベント・パラメータをそのプロセスの項目タイプ属性として設定します。4-19 ページの「[アクティビティ属性値の定義](#)」を参照してください。

また、呼出しイベント・アクティビティでは、現行のワークフロー・プロセスの項目タイプおよび項目キーが、イベント・メッセージのパラメータ・リストに自動的に追加されます。別のプロセスがイベント・メッセージを後で受信した場合、ワークフロー・エンジンはその項目タイプと項目キーを使用して、イベントを受信するプロセスの親として、イベントを呼び出したプロセスを自動的に設定します。『Oracle Workflow API リファレンス』の「SetItemParent」を参照してください。

The screenshot shows the 'Navigator Control Properties' dialog box with the 'Event Details' tab selected. The dialog has several sections for configuring event details:

- Send message as:** A dropdown menu for 'Event Message' set to '<None>' with an 'Edit' button.
- Event Name:** A section with a 'Type' dropdown set to 'Constant' and a 'Value' text input field.
- Event Key:** A dropdown menu for 'Event Key' set to '<None>' with an 'Edit' button.
- Out Agent:** A section with a 'Type' dropdown set to 'Constant' and a 'Value' text input field.
- To Agent:** A section with a 'Type' dropdown set to 'Constant' and a 'Value' text input field.

At the bottom of the dialog are buttons for 'OK', 'Cancel', 'Apply', and 'Help'.

4. イベント・アクションが「送信」のアクティビティの場合、次のイベントの詳細を入力します。

- 「イベント・メッセージ」： イベント・タイプの項目タイプ属性を選択します。ノードが送信したイベント・メッセージを格納します。
- 「イベント名」： (オプション) ノードが送信したイベントの名前を入力します。定形のイベント名を指定するか、テキスト・タイプの項目タイプ属性を選択します。後者の場合、実行時にイベント名を動的に決定できます。ここで入力したイベント名によって、イベント・メッセージに入力されているイベント名の値が上書きされます。
- 「イベント・キー」： (オプション) テキスト・タイプの項目タイプ属性を選択します。ノードが送信したイベントのイベント・キーが格納されます。ここで入力したイベント・キーによって、イベント・メッセージに入力されているイベント・キーの値が上書きされます。
- 「送信エージェント」： (オプション) ノードがイベントを送信したときのアウトバウンド・エージェントを入力します。次の書式を使用して、エージェントのエージェント名およびシステム名を指定します。

```
<agent_name>@<system_name>
```

定形の送信エージェント名を指定するか、テキスト・タイプの項目タイプ属性を選択します。後者の場合、実行時に送信エージェント名を動的に決定できます。ここで入力したアウトバウンド・エージェントによって、イベント・メッセージに入力されている送信エージェントの値が上書きされます。

- 「宛先エージェント」： (オプション) ノードがイベントを送信したときのインバウンド・エージェントを入力します。次の書式を使用して、エージェントのエージェント名およびシステム名を指定します。

```
<agent_name>@<system_name>
```

定形の宛先エージェント名を指定するか、テキスト・タイプの項目タイプ属性を選択します。後者の場合、実行時に宛先エージェント名を動的に決定できます。ここで入力したインバウンド・エージェントによって、イベント・メッセージに入力されている宛先エージェントの値が上書きされます。

注意： 送信イベント・アクティビティでは、「イベント・メッセージ」が必須です。また、イベント・メッセージ内に宛先エージェントまたは送信元エージェントを追加するか、このノードのイベントの詳細に宛先エージェントまたは送信エージェントを指定する必要があります。インバウンド・エージェントおよびアウトバウンド・エージェントのどちらも指定しない場合は、イベントを送信できません。

注意： 相関 ID がイベント・メッセージに指定されていない場合は、Oracle Workflow によって相関 ID がプロセスの項目キーに自動的に設定されます。

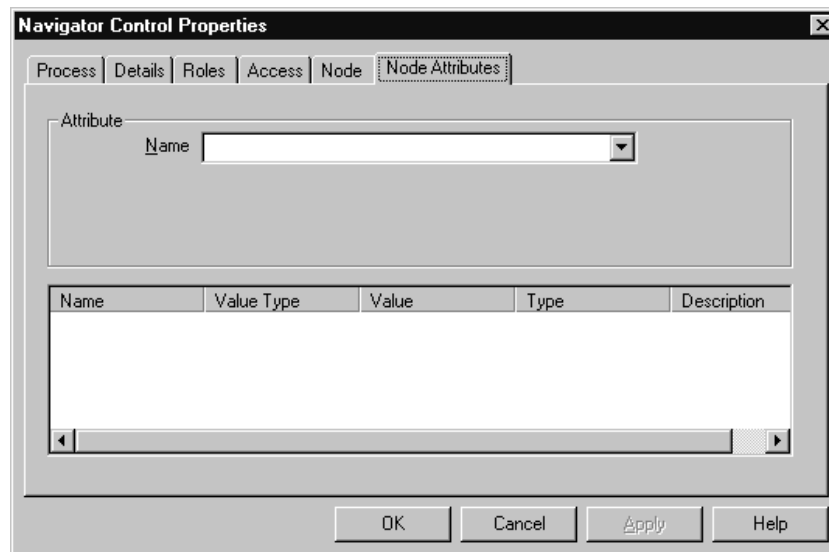
5. 「適用」を選択して変更内容を保存するか、「OK」を選択し、変更内容を保存してプロパティ画面を閉じるか、「キャンセル」を選択し、変更を取り消してプロパティ画面を閉じます。

関連項目：

- 2-9 ページ「プロパティ画面の「編集」ボタンの使用」
- 3-56 ページ「イベント・アクティビティ」
- 3-65 ページ「イベント・アクティビティの作成」

➤ **アクティビティ属性値の定義**

関数または通知アクティビティのアクティビティの属性値は、そのアクティビティがコールする PL/SQL のストアド・プロシージャによって使用されます。呼出しイベント・アクティビティのアクティビティ属性値は、パラメータとしてイベント・メッセージのパラメータ・リストに追加されます。3-8 ページの「項目タイプまたはアクティビティ属性の定義」を参照してください。



1. アクティビティ・ノードのプロパティ画面を表示します。「ノード属性」タブを選択します。
2. 属性を選択します。
3. 「値」フィールドに、この属性の値を入力します。値には、定数、または項目タイプ属性に保存されている値を使用できます。

入力する値は、アクティビティ属性と同じデータ型で、そのアクティビティに関連した PL/SQL 関数で定義されている、実際のアクティビティ・パラメータ自体のデータ型と一致している必要があります。属性タイプは、各属性の名前、説明、値の型および値とともに「attributes summary」フィールドに表示されます。

4. 「適用」を選択して変更内容を保存するか、「OK」を選択し、変更内容を保存してプロパティ画面を閉じるか、「キャンセル」を選択し、変更を取り消してプロパティ画面を閉じます。

▶ トランジションの作成と編集

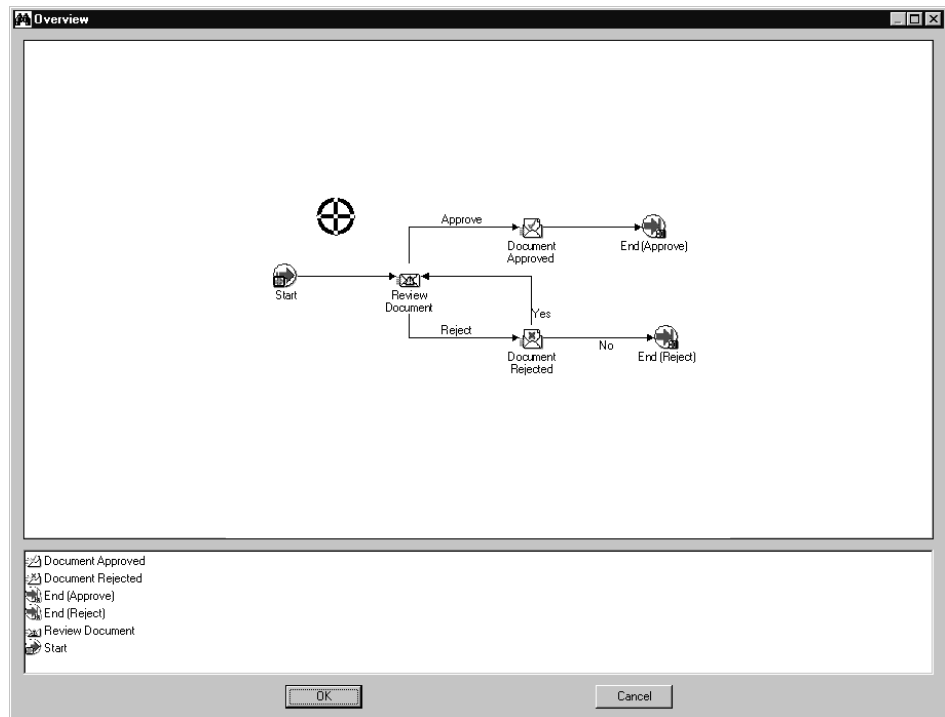
1. マウスの右ボタンを押したまま、ソース・アクティビティから宛先アクティビティへドラッグし、2つのアクティビティ間にトランジションを作成します。

注意： 重複するトランジションは、単一の重複しないトランジションとは異なる色で表示されます。

2. トランジションを編集するには、そのトランジションを選択します。
3. トランジションのラベルを移動するには、単にマウスでラベルを選択して新しい位置にドラッグします。ラベルがトランジションに貼り付けられます。
4. マウスでトランジションを選択して右ボタンをクリックすると、いつでも次の編集オプション・メニューを表示できます。
 - 「トランジションの削除」： 選択したトランジションが削除されます。
 - 「ロック」： さらに編集するかどうかに応じて、トランジションのロックとロック解除を切り替えます。トランジションがロックされていると、トランジションに対する支点の追加や削除はできませんが、トランジションの削除は可能です。
 - 「ラベルを非表示」： トランジション・ラベルの表示と非表示が切り替わります。
 - 「直線化」： トランジションを曲げている余分な支点が削除され、トランジションが直線になります。
 - 「結果」： トランジションに結果が割り当てられている場合に、このオプションを使用してトランジションの結果ラベルを変更します。ユーザーが選択できる結果ラベルのリストが、別のメニューに表示されます。

5. トランジションを曲げるには、そのトランジションを選択し、マウスの左ボタンを押したままドラッグして支点を作成します。支点を自由に移動して、トランジションの線を曲げることができます。
6. ソース・アクティビティ・ノードにループするトランジションを、次のどちらかの方法で作成できます。
 - マウスの右ボタンを押したまま、ソース・アクティビティからそれ自身へドラッグして、自己ループを作成します。
 - ソース・アクティビティ・ノードから、他の任意のアクティビティ・ノードへのトランジションを作成します。支点を追加してトランジションの線を折り曲げます。次にトランジションの矢印の先を選択し、ソースのアクティビティ・ノードにドラッグします。必要に応じてさらに支点を作成し、ループのトランジションを見やすくします。
7. 1つの支点をトランジションから削除するには、その支点を選択し、他の支点までドラッグして2つの点を結合します。

▶ プロセス概要の表示



1. カーソルをプロセス・ウィンドウにあわせて、マウスの右ボタンで表示するメニューから「概要」を選択します。
2. プロセスの「概要」ウィンドウが表示されます。
ウィンドウの上部ペインにはプロセス全体のサイズが縮小されたスケッチが表示され、下部ペインにはプロセスに含まれるアクティビティのリストが表示されます。
3. 「概要」ウィンドウのサイズは、プロセス・スケッチが見やすいように変更できます。
4. プロセス・スケッチ・ペインに、ドラッグできるように十字型のカーソルが表示されます。十字型のカーソルを使用して、プロセスのうち「プロセス」ウィンドウに表示する部分を選択します。
5. 下部ペイン内でアクティビティをクリックし、十字型のカーソルをスケッチ内のそのアクティビティに合せます。「OK」を選択してウィンドウを閉じ、「プロセス」ウィンドウ内でそのアクティビティにジャンプします。

また、上部ペイン内で十字型のカーソルをドラッグしてダブルクリックし、「プロセス」ウィンドウ内で目的の領域にジャンプすることもできます。

▶ プロセスの印刷

1. 印刷するプロセスを含む「プロセス」ウィンドウを表示します。
2. 「プロセス」ウィンドウをアクティブ・ウィンドウとして選択し、「ファイル」メニューまたはマウスの右ボタンで表示するメニューから「ダイアグラムの印刷」を選択します。

「ダイアグラムの印刷」オプションを選択すると、プロセス・ダイアグラムはピクチャ（メタファイル）として取り込まれ、印刷用に正しいサイズに拡大され、プリンタに送信されます。ダイアグラムが大きい場合は、印刷時に複数ページにまたがる場合があります。ただし、使用中のプリンタ・ドライバによっては、「印刷」ダイアログ・ボックスで印刷時に1ページに収まるようにイメージを縮小できます。

注意： プロセス・ダイアグラムに使用されているフォントがプリンタで見つからない場合は、プリンタ・ドライバによって類似するフォントに置き換えられたり、テキストが印刷されないことがあります。

▶ プロセス・ダイアグラムをクリップボードにコピー

1. コピーするプロセスを含む「プロセス」ウィンドウを表示してアクティブにします。
2. 「編集」メニューまたはマウスの右ボタンで表示されるメニューから、「ダイアグラムのコピー」を選択します。

これで、プロセスがメタファイルとビットマップの形式でクリップボードにコピーされます。

3. プロセス・ダイアグラムのメタファイル版またはビットマップ版を別のアプリケーションのウィンドウに貼り付けるには、メタファイルまたはビットマップを貼り付ける方法を、そのアプリケーションのマニュアルで確認する必要があります。

ビットマップ・イメージを編集するには、ビットマップを編集できるアプリケーションにイメージを貼り付ける必要があります。

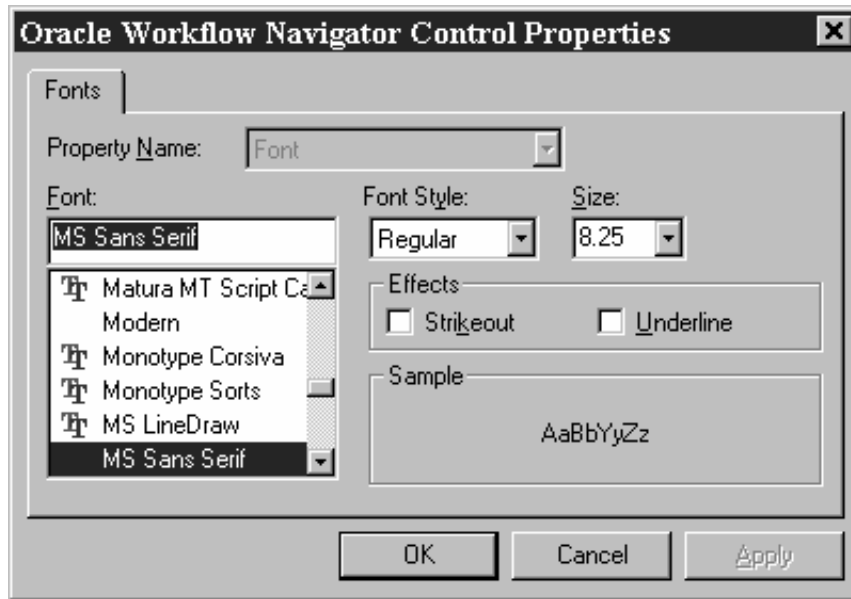
▶ プロセス定義の検証

1. 「ファイル」メニューから「検証」を選択し、現在選択されているデータ・ストアのすべてのプロセス定義を検証します。
2. 次のリストは、「検証」コマンドで実行される検証の例を示しています。
 - プロセスに、1つ以上の「開始」アクティビティと「終了」アクティビティがあるかどうかチェックされます。
 - プロセスに、そのプロセス自体がプロセス・アクティビティとして含まれていないかどうか検証されます。
 - 1つのプロセス内で、同じサブプロセスを2度使用できないように制限されます。
 - アクティビティの結果がすべて出力トランジションとしてモデル化されているかどうか検証されます。アクティビティが完了し、その結果が出力トランジションに関連付けられておらず、そのアクティビティの「デフォルト」トランジションが存在しなければ、アクティビティは「エラー」になります。
 - 終了ノードとしてマークされているアクティビティ・ノードに、出力トランジションがないかどうか検証されます。
 - 通知アクティビティの結果タイプが、そのメッセージの「RESULT」メッセージ属性に定義されている選択肢タイプと一致しているかどうか検証されます。
 - メッセージ本文でトークンの置換のために参照されるメッセージ属性が、メッセージ定義内に存在しているかどうか検証されます。
 - 別の項目タイプからオブジェクトを参照するプロセスについては、参照される項目タイプに関連付けられた前提条件の項目属性が存在するかどうか検証されます。

注意： 定義を正常に実行できなくする潜在的な問題が含まれていないかどうかを識別できるように、新規のプロセス定義を作成した場合に、必ず検証する必要があります。

Oracle Workflow Builder のフォントの変更

Oracle Workflow Builder のウィンドウで使用されるフォントを変更できます。変更は、プログラム内のすべてのウィンドウに適用されます。



▶ フォントの変更

1. 「表示」メニューから「フォント」を選択し、「フォント」プロパティ画面を表示します。
2. アイコンのラベルに使用するフォントを選択します。このフォントは、Oracle Workflow Builder のすべてのアイコンに使用されます。「サンプル」フィールドに、選択したフォントの表示イメージが表示されます。
3. フォント・スタイルとして「標準」、「太字」、「斜体」または「太字斜体」を選択します。フォントの中には、フォント・スタイルが制限されるものもあります。
4. 必要なフォント・サイズを指定します。フォントには、フォント・サイズが制限されるものもあります。
5. 「下線」または「取り消し線」チェック・ボックスをオンにして適用します。
6. 操作の完了後に、「OK」を選択します。これらのフォント設定は即時に有効になり、次回に Oracle Workflow Builder を起動するときにも使用されます。

ワークフロー・プロセスのショートカット・アイコンの作成

Windows のデスクトップに、Oracle Workflow Builder へのショートカットを作成できます。ショートカットを使用すると、指定したデータ・ストアに自動的に接続し、そのデータ・ストアから特定の「プロセス」ウィンドウを開いて、Oracle Workflow Builder を起動できます。

▶ Oracle Workflow Builder のショートカットの作成

1. Oracle Workflow Builder を起動します。
2. 「ファイル」メニューから「オープン」を選択してデータ・ストアを開きます。
3. 必要であれば、「プロセス」のブランチを拡張し、1つ以上のプロセス・アクティビティをダブルクリックして、そのプロセスのプロセス・ウィンドウを開きます。
4. 「ファイル」メニューから「ショートカットの作成」を選択します。
5. デスクトップに表示するショートカット名を入力します。
6. デスクトップ上で新規のショートカット・アイコンをダブルクリックすると、Oracle Workflow Builder が自動的に起動し、ショートカットの作成時に選択されていたデータ・ストアと、開いていたプロセス・ウィンドウが開きます。

ショートカットのデータ・ストアがデータベースの場合は、ショートカットをダブルクリックすると、データベースのパスワード入力を求めるプロンプトが表示されます。

ロール

Oracle Workflow のロールは、Oracle Workflow ディレクトリ・サービスのデータベースに格納されています。現在、Oracle Workflow Builder では新規のワークフロー・ロールを作成できませんが、データベースに格納されているロールを表示して参照することはできません。

ワークフロー・プロセスのロールの参照

ワークフロー・プロセスでロールが参照される方法の一例は、通知アクティビティをプロセスにノードとして組み込むときです。そのノードを実行者に割り当てる必要があります。実行者は、指定したロール、または動的にロールの値を戻す項目タイプ属性です。実行者にロールを割り当てるには、最初に Oracle Workflow データベースから Oracle Workflow Builder セッションにロールをロードする必要があります。『Oracle Workflow 管理者ガイド』の「Oracle Workflow のディレクトリ・サービスの設定」および 4-9 ページの「プロセスのノードの定義」を参照してください。

注意： 現在、Oracle Workflow Builder ではワークフロー・プロセス内でロールを参照できますが、特定のワークフロー・オブジェクトのプロパティ画面に表示される「ロール」タブは、将来のリリースまでサポートされません。「ロール」タブの目的は、ロールに特定のオブジェクトへのアクセス権を付与することです。

アドホックのユーザーおよびロール

Oracle Workflow では、ワークフロー・プロセス内で新規のアドホック・ユーザーおよびロールを作成して、ディレクトリ・サービスに追加できます。そのためには、サーバー側で適切な WF_DIRECTORY API をコールする関数アクティビティを定義し、その関数アクティビティをプロセス・ダイアグラムに含めます。6-3 ページの「関数アクティビティがコールする PL/SQL プロシージャの標準 API」および『Oracle Workflow API リファレンス』の「Workflow ディレクトリ・サービス API」を参照してください。

関連項目：

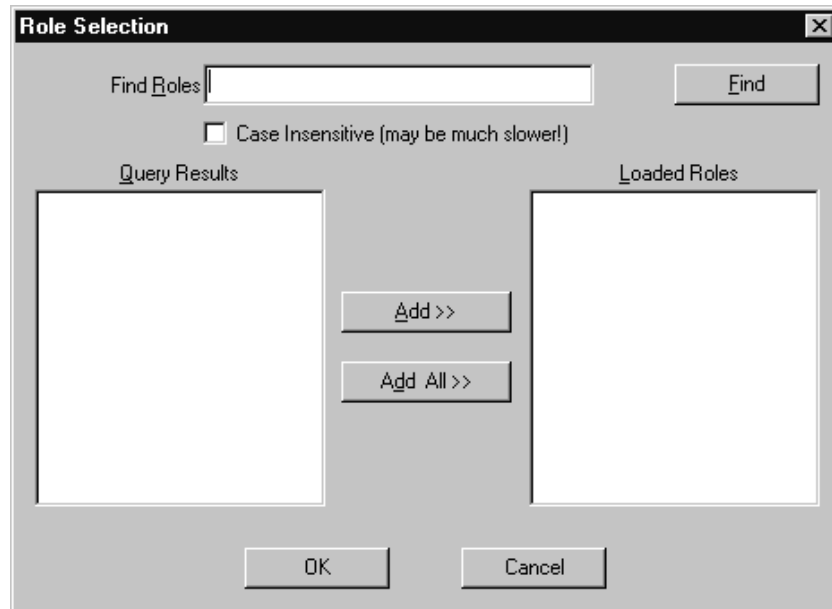
4-26 ページ「[ロールのロード](#)」

4-28 ページ「[Oracle Workflow Builder での Builder ディレクトリ・サービスの表示](#)」

➤ **ロールのロード**

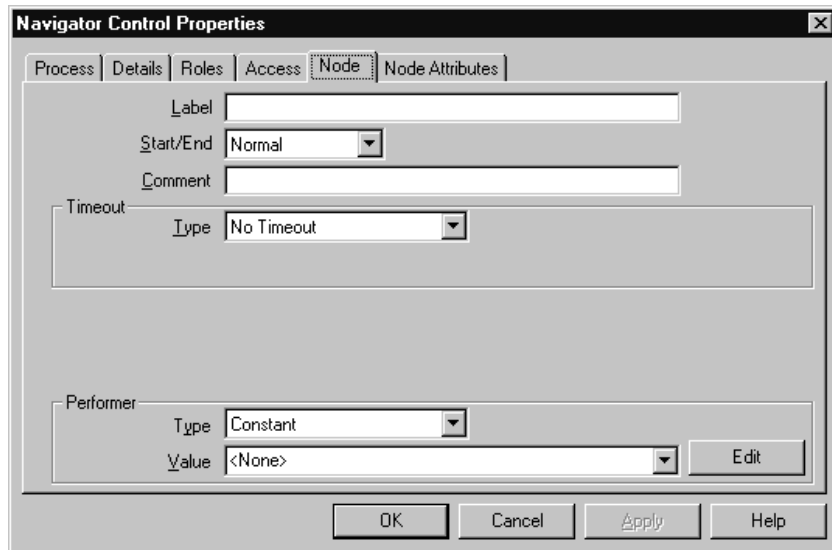
1. Oracle Workflow データベースに接続していない場合は、「ファイル」メニューから「オープン」を選択し、データベースに接続して項目タイプを開きます。
2. 「ファイル」メニューから「データベースからロールをロード」を選択します。「ロール選択」ウィンドウが表示されます。SQL 問合せ構文を使用し「ロール検索」フィールドに検索基準を入力して、ロールのサブセットを検索できます。また、検索基準を指定し

ないで「検索」を選択すると、すべてのルールを識別できます。「ルール選択」ウィンドウで、指定したルールが検索され、「問合せ結果」リスト・ボックスに表示されます。



3. 「問合せ結果」リストでロードするルールを選択し、「追加 >>」を選択して「ロードされたルール」リストに移動します。また、「すべて追加 >>」を選択すると、「問合せ結果」リスト内のすべてのルールが「ロードされたルール」リストに移動されます。「OK」を選択すると、選択したルールが **Oracle Workflow Builder** にロードされ、開いている項目タイプのワークフロー・オブジェクトに使用可能になります。

ルール情報の参照を必要とするワークフロー・オブジェクトの場合は、そのプロパティ画面に特定のフィールドが表示されます。これらのフィールドは、次の「ノード」プロパティ画面の例のように、データベースからロードしたルールのリストが表示されるドロップ・ダウン・フィールドです。



4. これらのドロップ・ダウン・フィールドからロールを選択するときに、そのフィールドの右側の「編集」ボタンを選択し、選択したロールのプロパティ・シートを表示することもできます。
5. 選択したロールの情報が、「ロール」プロパティ画面に読み取り専用で表示されます。

注意： 保存したプロセス定義を Oracle Workflow Builder で再度開くと、プロセス定義をファイルから開いたためにデータベースに接続されていない場合も、そのプロセスによって参照されるロール情報が自動的にロードされます。

➤ **Oracle Workflow Builder での Builder ディレクトリ・サービスの表示**

1. Oracle Workflow Builder でデータベースからロールをロードすると、ナビゲータ・ツリーにディレクトリ・サービス情報が表示されます。4-26 ページの「[ロールのロード](#)」を参照してください。
2. ナビゲータ・ツリーで、「ディレクトリ・サービス」ブランチを拡張します。データベースからロードした全ロールが表示されます。
3. ロールをダブルクリックし、次に示すようなロールに関する読み取り専用情報を表示します。現在、「ディレクトリ・サービス」ブランチでは、ロールの特定ユーザーは表示できないため注意してください。

Navigator Control Properties [X]

Role

Internal Name BLEWIS

Display Name BLEWIS

Description BLEWIS

Language AMERICAN Territory AMERICA

Email Address BLEWIS Fax

Notification Preference

Status ACTIVE Expiration

OK Cancel Apply Help

5

Oracle Workflow の事前に定義された アクティビティ

この章では、Oracle Workflow の事前に定義されたアクティビティの使用方法について説明します。

標準アクティビティ

Oracle Workflow には、プロセスの管理に使用できる一般的なアクティビティがいくつか用意されています。これらのアクティビティは、標準項目タイプに関連付けられていますが、ユーザーが定義するプロセスで使用できます。標準項目タイプは、Oracle Workflow Server に自動的にインストールされます。また、標準項目タイプには、`<ORACLE_HOME>/WF/data/<language>/` サブディレクトリにある `wfstd.wft` ファイルからもアクセスできます。

注意： 事前に定義されたアクティビティは、Oracle Applications および Oracle Self-Service Web Applications に付属の事前定義済のワークフローにも使用可能です。Oracle Applications に固有のワークフロー・アクティビティの詳細は、該当する Oracle Applications 製品のドキュメントまたはヘルプを参照してください。

注意： ドロップ先のプロセスと異なるデータ・ストアにあるアクティビティをプロセスにドラッグする場合は、最初にそのアクティビティが属する項目タイプを、このプロセスと同じデータ・ストアにコピーしてください。たとえば、`wfexample.wft` に格納されているプロセスを変更し、`wfstd.wft` に保存されている標準アクティビティをそのプロセスに追加するとします。最初に、両方のファイルを Oracle Workflow Builder でデータ・ストアとして開き、`wfstd` にある標準項目タイプをコピーして、`wfexample` データ・ストアに貼り付ける必要があります。これで、`wfexample` データ・ストアの標準アクティビティをプロセスにドラッグできます。

「実行時間の比較」アクティビティ

「実行時間の比較」アクティビティには、プロセス実行時の経過時間を定数のテスト時間と比較する標準的な方法が用意されています。

「実行時間の比較」アクティビティは、WF_STANDARD.COMPAREEXECUTIONTIME という PL/SQL プロシージャをコールします。

アクティビティ属性

「実行時間の比較」アクティビティには、次の 2 つのアクティビティ属性があります。

- テスト実行時間：経過した実行時間と比較する時間（秒）。
- 上位タイプ：選択肢コードで、「ルート」または「上位」のどちらかの値を取得します。値「ルート」の場合は、現行のルート・プロセスで経過した実行時間とテスト時間が比較されます。値「上位」の場合は、すぐ上の上位プロセスで経過した実行時間とテスト時間が比較されます。この場合の上位プロセスは、サブプロセスでもかまいません。

このアクティビティでは、結果コードに「比較」選択肢タイプを使用します。使用可能な値は、「より大きい」、「より小さい」、「等しい」、「NULL」（テスト時間が NULL の場合）です。4-19 ページの「[アクティビティ属性値の定義](#)」を参照してください。

「待機」アクティビティ

「待機」アクティビティは、指定された期間のみプロセスを休止します。次のいずれかの期限まで待機できます。

- 指定日時
- 指定月日
- 指定曜日
- このアクティビティに到達後の一定時間

このアクティビティは、WF_STANDARD.WAIT という PL/SQL プロシージャをコールします。

注意： 待機時間が経過したことを判断するために、遅延アクティビティ用のバックグラウンド・エンジンを実行する必要があります。バックグラウンド・エンジンによって「待機」アクティビティが完了するので、プロセスを続行できます。『Oracle Workflow 管理者ガイド』の「バックグラウンドのワークフロー・エンジンの設定」を参照してください。

アクティビティ属性

「待機」アクティビティには、次の6つのアクティビティ属性があります。

- 待機モード： この属性を使用して、待機時間の計算方法を指定します。次の待機モードから1つを選択できます。
 - 絶対日：「絶対日」アクティビティ属性で指定された日付に達するまで、アクティビティを休止します。
 - 相対時間：「相対時間」アクティビティ属性で指定された日数が経過するまで、アクティビティを休止します。
 - 毎月何日：「毎月何日」アクティビティ属性で指定された月の特定日付まで、アクティビティを休止します。
 - 毎週何曜日：「毎週何曜日」アクティビティ属性で指定された特定の曜日まで、アクティビティを休止します。
- 絶対日：「待機モード」が「絶対日」に設定されている場合は、絶対日を入力します。
- 相対時間：「待機モード」が「相対時間」に設定されている場合は、相対時間を <日数>.<日数（小数点以下）> の形式で入力します。たとえば、待機時間が半日（12時間）の場合は、0.5 と入力します。
- 毎月何日：「待機モード」が「毎月何日」に設定されている場合は、リストから月の日を選択します。選択した日が今月はすでに過ぎている場合、アクティビティは翌月のその日まで待機します。
- 毎週何曜日：「待機モード」が「毎週何曜日」に設定されている場合は、リストから曜日を選択します。選択した曜日が今週はすでに過ぎている場合、アクティビティは翌週のその曜日まで待機します。
- 毎日何時：「毎日何時」アクティビティ属性を使用して「待機」アクティビティの休止期限を午前0時以外の時刻に設定しないかぎり、「待機」アクティビティは常に指定日の午前0時まで休止します。

関連項目：

4-19 ページ「[アクティビティ属性値の定義](#)」

「ブロック」アクティビティ

「ブロック」アクティビティでは、なんらかの外部プログラムまたは手動手順が完了し、CompleteActivity Workflow Engine API がコールされるまで、プロセスを休止できます。「ブロック」アクティビティを使用して、コンカレント・プログラムの完了など、なんらかの条件が満たされるまでプロセスを延期できます。プログラムが完了し、「ブロック」アクティビティでプロセスを再開する場合は、プログラムで CompleteActivity コールを発行するようにしてください。『Oracle Workflow API リファレンス』の「CompleteActivity」を参照してください。

このアクティビティは、WF_STANDARD.BLOCK という PL/SQL プロシージャをコールします。

「スレッドの遅延」アクティビティ

「スレッドの遅延」アクティビティは、バックグラウンド・キューに対して後続のプロセス・スレッドを延期します。その場合に、対象スレッドの各アクティビティのコストを、ワークフロー・エンジンのしきい値を超えるように変更する必要はありません。このアクティビティは、スレッドがすでに遅延している場合でも、現行のデータベース・セッションを切断し、プロセスのスレッドを中断します。

このアクティビティは、WF_STANDARD.DEFER という PL/SQL プロシージャをコールします。

「プロセスの開始」アクティビティ

「プロセスの開始」アクティビティを使用して、現行のプロセスから他のワークフロー・プロセスを開始できます。このアクティビティは、WF_STANDARD.LAUNCHPROCESS という PL/SQL プロシージャをコールします。

アクティビティ属性

「プロセスの開始」アクティビティには、次の6つのアクティビティ属性があります。

- 項目タイプ： 開始するプロセスの項目タイプ。ここでは、項目タイプの内部名を指定します。このアクティビティ属性には必ず値を指定します。
- 項目キー： 開始するプロセスの項目キー。値を指定しなければ、項目キーのデフォルトは `<current_item_type>:<current_item_key>-<n>` となります。 `current_item_type` と `current_item_key` は現行のプロセス・インスタンスを表し、 `n` は現行のプロセス・インスタンスで開始されるプロセス番号を表します。このプロセス番号は、1 が初期値となります。

注意： プロセス・インスタンスの項目キーには、シングルバイト文字のみを含めることができます。マルチバイトの値を含めることはできません。

- プロセス名： 開始するプロセスの内部名。プロセス名を指定しなければ、「プロセスの開始」アクティビティでは、開始するプロセスの項目タイプ・セレクト関数でプロセス名をチェックします。
- ユーザー・キー： 開始するプロセスのユーザー定義キー。
- 所有者： 開始するプロセスの所有者として指定されているロール。
- 即時に遅延： 「YES」または「NO」を選択し、開始するプロセスをバックグラウンド・エンジンに対して即時に遅延するかどうか決定します。デフォルトは「NO」で、プロセスがいったん開始されると、そのプロセスが完了するまで、またはプロセスのアクティビティのいずれかが延期されるまで続行します。4-19 ページの「[アクティビティ属性値の定義](#)」を参照してください。

「Noop」アクティビティ

「Noop」アクティビティは、何の処理も実行しないプロセスホルダ・アクティビティとして機能します。処理を実行しないノードを配置する場所に、このアクティビティを使用できます。このアクティビティをプロセスに含める場合には、将来このアクティビティを何に使用するかを忘れないように、表示名をわかりやすい名前に変更できます。このアクティビティは、WF_STANDARD.NOOP という PL/SQL プロシージャをコールします。

「ループ・カウンタ」アクティビティ

「ループ・カウンタ」アクティビティを使用して、プロセス内の特定のパスを通過するワークフロー・エンジンのトランジション回数を制限します。「ループ・カウンタ」アクティビティの結果として「ループ」または「終了」を指定できます。

この「ループ・カウンタ」アクティビティは、WF_STANDARD.LOOPCOUNTER という PL/SQL プロシージャをコールします。

アクティビティ属性

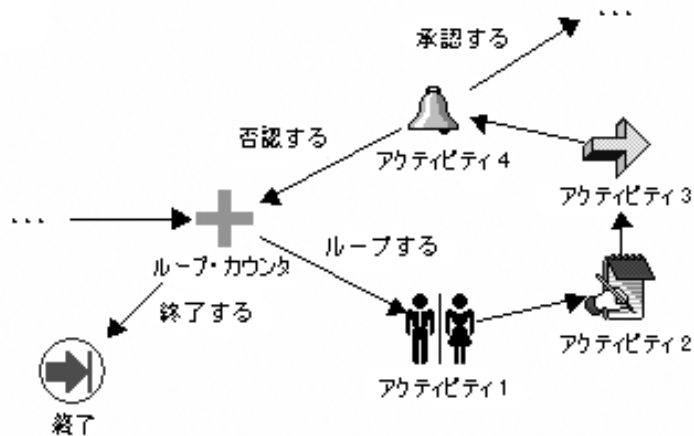
「ループ・カウンタ」アクティビティには、「ループの限度」というアクティビティ属性があります。ワークフロー・エンジンが「ループ・カウンタ」アクティビティを通過した回数が「ループの限度」で指定した値よりも小さければ、その「ループ・カウンタ」アクティビティは戻り値「ループ」を戻して完了し、ループ内の次のアクティビティへ進みます。ワークフロー・エンジンが「ループ・カウンタ」アクティビティを通過した回数が、「ループの限度」の値より大きければ、アクティビティは戻り値「終了」を戻して完了し、ループを終了して新しいアクティビティへ進みます。

たとえば、ワークフロー・プロセスに複数のソース・アクティビティから通過できるアクティビティの分岐が含まれていて、特定の分岐をプロセス内で一度だけ実行するとします。この場合、その分岐内の最初のアクティビティとして「ループ・カウンタ」アクティビティを含め、「ループの制限」アクティビティ属性の値として1を指定します。また、「ループ・カウンタ」アクティビティが複数回通過する場合は、「ループ・カウンタ」アクティビティ

とエンジンが実行するアクティビティの間を「終了」トランジションで結びます。次のダイアグラムを参照してください。

たとえば、次のダイアグラムのように、ループ内の最初のアクティビティに「ループ・カウンタ」アクティビティを使用することもできます。「ループの限度」アクティビティ属性に指定する値に、そのループを通過できる回数を指定します。「ループ・カウンタ」アクティビティを通過した回数が、「ループの限度」に指定した値より大きければ、そのプロセスは「終了」トランジションに沿って指定されたアクティビティへ移動します。4-19 ページの「[アクティビティ属性値の定義](#)」を参照してください。

この例では、エンジンは「ループ・カウンタ」アクティビティからループ内のアクティビティ 1、2、3 および 4 を移動します。アクティビティ 4 の結果が「承認する」の場合、プロセスは「承認する」トランジションを移動します。アクティビティ 4 の結果が「否認する」の場合、プロセスは「否認する」トランジションを移動し、「ループ・カウンタ」アクティビティに戻ります。項目が複数回拒否され、「ループ・カウンタ」アクティビティを通過した回数が「ループの限度」値を超えると、プロセスは「終了」トランジションを移動して終了します。



「開始」アクティビティ

「開始」アクティビティは、プロセスの開始を示すもので、処理は何も実行しません。このアクティビティは必須ではありませんが、プロセス・ダイアグラムに「開始」アクティビティを入れると、独立したノードとしてビジュアルにプロセスの開始マークを付けることができます。このアクティビティは、WF_STANDARD.NOOP という PL/SQL プロシージャをコールします。

「終了」アクティビティ

「終了」アクティビティは、プロセスの終了を示すもので、処理は何も実行しません。このアクティビティを使用して、アクティビティの「結果タイプ」を指定し、完了したプロセスの結果を戻すことができます。このアクティビティは必須ではありませんが、プロセス・ダイアグラムに「終了」アクティビティを入れると、独立したノードとしてビジュアルにプロセスの終了マークを付けることができます。このアクティビティは、WF_STANDARD.NOOP という PL/SQL プロシージャをコールします。

「ロール解決」アクティビティ

「ロール解決」アクティビティによって、複数のユーザーで構成されるロールから 1 人のユーザーを識別できます。プロセス・ダイアグラムで、通知アクティビティの前に「ロール解決」アクティビティを配置し、その通知アクティビティの実行者を複数のユーザーからなるロールに設定します。「ロール解決」アクティビティは、そのロールから 1 人のユーザーを選択し、そのユーザーに通知アクティビティを割り当てます。

このアクティビティは、WF_STANDARD.ROLERESOLUTION という PL/SQL プロシージャをコールします。

アクティビティ属性

「ロール解決」アクティビティの「方法」アクティビティ属性を使用して、ロールの解決方法を指定します。属性の値に「ロード・バランス」が設定されている場合、候補の各ユーザーについて、そのアクティビティから受信したオープン通知がいくつあるかを比較し、オープン通知が最も少ないユーザーを選択します。属性の値に「順次処理」が設定されている場合は、最後にそのアクティビティから通知を受信してからの期間が最も長いユーザーを調べ、ロールからユーザーを順に選択します。4-19 ページの「[アクティビティ属性値の定義](#)」を参照してください。

「通知」アクティビティ

「通知」関数アクティビティでは通知を送信でき、送信されるメッセージは直前の関数アクティビティによって実行時に動的に決定されます。「通知」アクティビティを使用するには、「通知」アクティビティによって送信される事前定義済メッセージのうちから1つを選択するプロセスに、前提条件となる関数アクティビティをあらかじめ用意する必要があります。

注意：「通知」アクティビティはアクセス・レベル0では変更できないようにロックされているため、結果タイプの値は「<なし>」から変更できません。したがって、関数アクティビティによって動的に選択されるメッセージは、結果タイプを持つことがないため、応答を要求しない情報メッセージになります。

注意：「通知」アクティビティで応答が必要なメッセージを送信するには、「通知」アクティビティをコピーして独自バージョンを作成する必要があります。独自の「通知」アクティビティでは、複数のメッセージ（応答属性付き）のどれでも送信できるため、プロセスには、戻される可能性のあるすべての通知結果を用意する必要があります。

注意：常に同じメッセージを送信するアクティビティを定義する場合には、この「通知」関数アクティビティを使用せずに、通知アクティビティを定義する必要があります。

この「通知」アクティビティは、WF_STANDARD.NOTIFY という PL/SQL プロシージャをコールします。

アクティビティ属性

「通知」アクティビティには、次の3つのアクティビティ属性があります。

- **メッセージ名：**送信する事前定義済メッセージの名前。送信するメッセージを判断する前提条件となる関数アクティビティには、項目属性にこの名前が格納されている必要があります。その項目属性が「メッセージ名」アクティビティ属性で参照され、送信するメッセージ名が判断されます。
- **実行者：**通知メッセージの送信先となるロールの名前。データベースからロールをロードした場合、実行者として固定ロールを選択できます。または、実行時にロール名を戻す項目属性に実行者を設定することもできます。
- **ロールの拡張：**値として、選択肢コード「Yes」または「No」を取ります。ロール内の各ユーザーに通知メッセージの個別コピーを送信する場合は、「ロールの拡張」を

「Yes」に設定します。4-19 ページの「[アクティビティ属性値の定義](#)」を参照してください。

「はい/いいえ投票」アクティビティ

「はい/いいえ投票」アクティビティでは、ロール内のユーザーのグループに通知を送信し、ユーザーからのはい/いいえ応答を集計できます。集計結果によって、次のプロセスに進むアクティビティが決まります。

「はい/いいえ投票」アクティビティは通知アクティビティの一種です。最初にユーザー・グループに通知メッセージを送信し、次に PL/SQL の通知後関数を実行してユーザーの応答（投票）を集計します。

アクティビティ属性

「はい/いいえ投票」アクティビティには、次の2つの属性があります。

- 「はい」の割合：「はい」の投票の割合を計算します。対象アクティビティの投票結果がすべて「はい」になると完了します。
- 「いいえ」の割合：「いいえ」の投票の割合を計算します。対象アクティビティの投票結果がすべて「いいえ」になると完了します。

注意： 最高得票の応答が結果となるように「人気投票」投票方法を使用するには、「はい」の割合および「いいえ」の割合属性の値をどちらも NULL として定義します。3-76 ページの「[投票方法の例](#)」を参照してください。

- 投票オプション： 次の2の値のいずれかを選択し、投票の集計方法を指定します。
 - 「参加者全員の投票を待つ」： ワークフロー・エンジンは、すべての投票が行われるまで待つから、通知を受けた全ユーザーに対する割合として結果を集計します。タイムアウト条件が発生すると、タイムアウト発生前に出された投票の合計に対する割合として投票結果が計算されます。
 - 「投票のつど集計」： ワークフロー・エンジンは、通知を受けたすべてのユーザーに対する累積割合を、応答があるたびに集計します。タイムアウト条件が発生すると、応答は合計投票数に対する割合として集計されます。カスタム「応答」アクティビティ属性のどれかの値が空白値の場合、このオプションは意味がなくなるため注意してください。
 - 「参加者全員の投票が必要」： すべての投票が行われた後にかぎり、通知を受けた全ユーザーに対する割合で応答が評価されます。タイムアウト条件が発生すると、標準的なタイムアウト・トランジションに従って進むか、使用可能なトランジションが1つも存在しない場合はエラーが発生し、投票結果は集計されません。4-19 ページの「[アクティビティ属性値の定義](#)」を参照してください。

関連項目：

3-73 ページ「投票アクティビティ」

「マスター/ディテール連携」アクティビティ

「マスター/ディテール連携」アクティビティでは、マスター・プロセスとディテール・プロセスのフローを連携できます。たとえば、マスター・プロセスは、各ディテール・プロセスがフローの特定の点に到達したときのみ（またはその逆）マスター・プロセスを継続するように、連携を必要とする詳細プロセスを作成できます。

Oracle Workflow のマスター・プロセスからディテール・プロセスを作成すると、実際には、一意の項目タイプと項目キーを持つ別のプロセスが作成されます。ディテール・プロセスの作成時には、`CreateProcess` API をコールしてから `StartProcess` API をコールする前に、ワークフロー・エンジンの `SetItemParent` API をコールして、2つのプロセス間にマスター/ディテール関係を定義します。『Oracle Workflow API リファレンス』の「`SetItemParent`」を参照してください。

その後、後述する2つのアクティビティを使用して、マスター・プロセスとディテール・プロセスのフローを連携できます。一方はプロセスを休止するアクティビティ、他方は停止中のプロセスに継続するようにシグナルを送信するアクティビティです。これらのアクティビティを使用するには、一方のアクティビティをマスター・プロセスに、他方を各ディテール・プロセスに配置します。

どちらのアクティビティにも、他のプロセスの連携アクティビティの識別に使用する2つのアクティビティ属性が含まれます。

「フロー待ち」アクティビティ

このアクティビティをマスター・プロセスまたはディテール・プロセスに入れて、対応する他方のディテール・プロセスまたはマスター・プロセスで指定したアクティビティが完了するまでフローを休止します。このアクティビティは、WF_STANDARD.WAITFORFLOW という PL/SQL プロシージャをコールします。

アクティビティ属性

「フロー待ち」アクティビティには、次の 2 つの属性があります。

- 継続フロー： 対応するマスター・プロセスまたはディテール・プロセスが完了するまで、このアクティビティが待機するかどうかを指定します。
- 継続アクティビティ： 現行のプロセスを継続する前に、対応するプロセス内で完了させる必要のあるアクティビティ・ノードのラベルを指定します。デフォルト値は CONTINUEFLOW です。4-19 ページの「[アクティビティ属性値の定義](#)」を参照してください。

「継続フロー」アクティビティ

このアクティビティを使用して、停止しているプロセスにマークを付け、対応するディテール・プロセスまたはマスター・プロセスの完了時にその位置から処理を継続します。このアクティビティは、WF_STANDARD.CONTINUEFLOW という PL/SQL プロシージャをコールします。

アクティビティ属性

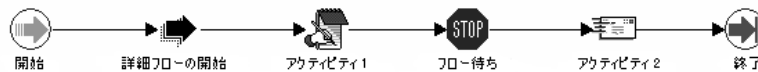
「継続フロー」アクティビティには、次の 2 つの属性があります。

- 待機中フロー： このアクティビティの完了を待機して停止しているプロセスが、「マスター」または「ディテール」フローのどちらであるかを指定します。
- 待機中アクティビティ： このアクティビティの完了を待機して停止しているプロセスの、アクティビティ・ノードのラベルを指定します。4-19 ページの「[アクティビティ属性値の定義](#)」を参照してください。

例

次の図は、これらの連携アクティビティがどのように使用されるかを示しています。マスター・プロセスの例では、マスター・プロセスが「開始」アクティビティから開始した後、「詳細フローの開始」アクティビティが複数のディテール・プロセスを開始します。その後、マスター・プロセスは「アクティビティ 1」を完了してから、「フロー待ち」アクティビティで休止します。「フロー待ち」は、すべてのディテール・プロセスが「継続フロー」アクティビティを完了するまで待ってから、マスター・プロセス「アクティビティ 2」に進み、最後に終了するように定義されています。次のディテール・プロセスの一例は、ディテール・プロセスが「開始」アクティビティから開始した後、「アクティビティ A」を完了することを示しています。「継続フロー」アクティビティに到達すると、マスター・プロセスを「フロー待ち」アクティビティから継続するように、ワークフロー・エンジンにシグナルが送信されます。その後、ディテール・プロセスは「アクティビティ B」に進み、最後に終了します。

マスター・プロセス



ディテール・プロセス



注意： 対応する 1 つ以上のディテール・プロセスに「継続フロー」アクティビティを使用せずに、マスター・プロセスに「フロー待ち」アクティビティを組み込むことができます。ワークフロー・エンジンは、「継続フロー」アクティビティを含む他のすべてのディテール・プロセスが「継続フロー」アクティビティを完了すると、即時にマスター・プロセスを継続します。

マスター・プロセスを継続する前にいつディテール・プロセスが完了してもかまわない（または、すべてのディテール・プロセスを継続する前に、いつマスター・プロセスが完了してもかまわない）場合は、単にマスター・プロセスまたはディテール・プロセスから連携する両方のアクティビティを除外します。

注意： プロセスに「継続フロー」アクティビティを含める場合は、「継続フロー」アクティビティのアクティビティ属性によって定義されるように、対応するマスターまたはディテール・プロセスに「フロー待ち」アクティビティも含める必要があります。

「割当」アクティビティ

「割当」アクティビティでは、項目属性に値を割り当てることができます。このアクティビティは、WF_STANDARD.ASSIGN という PL/SQL プロシージャをコールします。

アクティビティ属性

「割当」アクティビティには、「項目属性」というアクティビティ属性があります。「項目属性」を使用して、値を割り当てる項目属性を選択します。項目属性の書式タイプに応じて、「日付値」、「数値」または「テキスト値」アクティビティ属性を使用し、項目属性に割り当てる値を指定します。

「モニター URL」アクティビティ

「モニター URL」アクティビティでは、ワークフロー・モニターのダイアグラム・ウィンドウの URL が生成され、指定した項目属性に格納されます。このアクティビティは、WF_STANDARD.GETURL という PL/SQL プロシージャをコールします。

アクティビティ属性

「モニター URL」アクティビティには、次の 2 つのアクティビティ属性があります。

- 項目属性： ワークフロー・モニター・ウィンドウの URL を保存するための項目属性の名前を選択します。
- 管理モード： ワークフロー・モニター・ウィンドウで、URL をどのように表示するかを定義します。「管理モード」を「Yes」に設定すると、URL はワークフロー・モニターに ADMIN モードで表示されます。その他の設定の場合、URL はワークフロー・モニターに USER モードで表示されます。4-19 ページの「[アクティビティ属性値の定義](#)」を参照してください。

「イベント・プロパティの取得」アクティビティ

「イベント・プロパティの取得」アクティビティでは、ビジネス・イベント・システムからイベント・メッセージのプロパティを取り出し、そのプロパティ値を項目属性に格納します。このアクティビティは、WF_STANDARD.GETEVENTPROPERTY という PL/SQL プロシージャをコールします。

アクティビティ属性

「イベント・プロパティの取得」アクティビティには、次の4つのアクティビティ属性があります。

- イベント： イベント・タイプの項目属性を選択します。このイベント・メッセージから、プロパティを取り出します。
- プロパティ： 値を取り出すイベント・プロパティ。この属性の値は、「イベント・プロパティ」選択肢タイプの選択肢コードです。有効な値は、「優先度」、「送信日」、「受信日」、「相関 ID」、「イベント・パラメータ」、「イベント名」、「イベント・キー」、「送信元エージェント」、「送信元エージェント名」、「送信元エージェント・システム」、「宛先エージェント」、「宛先エージェント名」および「宛先エージェント・システム」です。『Oracle Workflow API リファレンス』の「イベント・メッセージ構造」を参照してください。
- イベント・パラメータ： 「プロパティ」属性で「イベント・パラメータ」プロパティを選択した場合は、値を取り出すパラメータの名前を入力します。Oracle Workflow では、この名前を使用して、イベント・メッセージのパラメータ・リスト内のパラメータを識別します。「イベント・パラメータ」以外のプロパティを選択した場合は、この属性を空白のままにします。
- 項目属性： イベント・プロパティ値を格納する項目属性。4-19 ページの「[アクティビティ属性値の定義](#)」を参照してください。

「イベント・プロパティの設定」アクティビティ

「イベント・プロパティの設定」アクティビティでは、ビジネス・イベント・システムのイベント・メッセージのプロパティ値を設定します。このアクティビティは、WF_STANDARD.SETEVENTPROPERTY という PL/SQL プロシージャをコールします。

アクティビティ属性

「イベント・プロパティの設定」アクティビティには、次の6つのアクティビティ属性があります。

- イベント： イベント・タイプの項目属性を選択します。このイベント・メッセージのプロパティを設定します。
- プロパティ： 値を設定するイベント・プロパティ。この属性の値は、「イベント・プロパティ」選択肢タイプの選択肢コードです。有効な値は、「優先度」、「送信日」、「受信日」、「関連 ID」、「イベント・パラメータ」、「イベント名」、「イベント・キー」、「送信元エージェント」、「送信元エージェント名」、「送信元エージェント・システム」、「宛先エージェント」、「宛先エージェント名」および「宛先エージェント・システム」です。『Oracle Workflow API リファレンス』の「イベント・メッセージ構造」を参照してください。
- イベント・パラメータ： 「プロパティ」属性の「イベント・パラメータ」プロパティを選択した場合は、値を設定するパラメータの名前を入力します。Oracle Workflow では、この名前を使用して、イベント・メッセージのパラメータ・リスト内のパラメータを識別します。「イベント・パラメータ」以外のプロパティを選択した場合は、この属性を空白のままにします。
- 日付値： イベント・プロパティに設定する日付タイプの値（「送信日」または「受信日」プロパティを選択した場合）
- 数値： イベント・プロパティに設定する数値タイプの値（「優先度」プロパティを選択した場合）
- テキスト値： イベント・プロパティに設定するテキスト・タイプの値（「関連 ID」、「イベント・パラメータ」、「イベント名」、「イベント・キー」、「送信元エージェント名」、「送信元エージェント・システム」、「宛先エージェント名」または「宛先エージェント・システム」プロパティを選択した場合）。4-19 ページの「[アクティビティ属性値の定義](#)」を参照してください。

注意： アクティビティ属性に設定する値は、選択したイベント・プロパティのデータ型と一致している必要があります。

「イベント・プロパティの比較」アクティビティ

「イベント・プロパティの比較」アクティビティでは、ビジネス・イベント・システムのイベント・メッセージのプロパティと、指定したテスト値を比較します。このアクティビティは、WF_STANDARD.COMPAREEVENTPROPERTY という PL/SQL プロシージャをコールします。

アクティビティ属性

「イベント・プロパティの比較」アクティビティには、次の6つのアクティビティ属性があります。

- イベント： イベント・タイプの項目属性を選択します。このイベント・メッセージのプロパティと、テスト値を比較します。
- プロパティ： テスト値と比較する値を持つイベント・プロパティ。この属性の値は、「イベント・プロパティ」選択肢タイプの選択肢コードです。有効な値は、「優先度」、「送信日」、「受信日」、「相関ID」、「イベント・パラメータ」、「イベント名」、「イベント・キー」、「送信元エージェント」、「送信元エージェント名」、「送信元エージェント・システム」、「宛先エージェント」、「宛先エージェント名」および「宛先エージェント・システム」です。『Oracle Workflow API リファレンス』の「イベント・メッセージ構造」を参照してください。
- イベント・パラメータ： 「プロパティ」属性の「イベント・パラメータ」プロパティを選択した場合は、テスト値と比較する値を持つパラメータの名前を入力します。Oracle Workflow では、この名前を使用して、イベント・メッセージのパラメータ・リスト内のパラメータを識別します。「イベント・パラメータ」以外のプロパティを選択した場合は、この属性を空白のままにします。
- 日付値： イベント・プロパティ値と比較する日付タイプのテスト値（「送信日」または「受信日」プロパティを選択した場合）
- 数値： イベント・プロパティ値と比較する数値タイプのテスト値（「優先度」プロパティを選択した場合）
- テキスト値： イベント・プロパティ値と比較するテキスト・タイプのテスト値（「相関ID」、「イベント・パラメータ」、「イベント名」、「イベント・キー」、「送信元エージェント名」、「送信元エージェント・システム」、「宛先エージェント名」または「宛先エージェント・システム」プロパティを選択した場合）

「イベント・プロパティの比較」アクティビティでは、結果コードに「比較」選択肢タイプを使用します。有効な値は、「より大きい」、「より小さい」、「等しい」、「NULL」（テスト・アクティビティ属性値が NULL の場合）です。イベント・プロパティ値とテスト値との比較方法に基づいて、ワークフロー・プロセスを進めることが可能です。4-19 ページの「[アクティビティ属性値の定義](#)」を参照してください。

注意： アクティビティ属性に入力するテスト値は、選択したイベント・プロパティのデータ型と一致している必要があります。データ型が一致しないアクティビティ属性にテスト値を入力すると、「イベント・プロパティの比較」アクティビティから「NULL」結果コードが返されます。

「XML タグ値の取得」アクティビティ

「XML タグ値の取得」アクティビティでは、ビジネス・イベント・システムのイベント・メッセージの内容から、データを取り出します。現在、この機能を使用できるのは Oracle Workflow のスタンドアロン版のみです。このアクティビティでは、イベント・メッセージの特殊な XML タグ・セットに囲まれたデータを取り出し、指定した項目属性に格納します。「XML タグ値の取得」アクティビティは、`oracle.apps.fnd.wf.XMLGetTagValue` という外部 Java 関数をコールします。

注意： ワークフロー・エンジンでは、外部 Java 関数アクティビティを検出すると、エントリを「送信」キューに入れます。アクティビティの実行を続行するには、Java 関数アクティビティ・エージェントを実行して、該当する Java 関数をコールし、結果を「受信」キューに入れる必要があります。次に、バックグラウンド・エンジンを実行して「受信」キューを処理し、関数アクティビティを完了する必要があります。『Oracle Workflow 管理者ガイド』の「Java 関数アクティビティ・エージェントの設定」および「バックグラウンドのワークフロー・エンジンの設定」を参照してください。

アクティビティ属性

「XML タグ値の取得」アクティビティには、次の 3 つのアクティビティ属性があります。

- イベント： イベント・タイプの項目属性を選択します。このイベント・メッセージから、データを取り出します。
- タグ： データを取り出すイベント・メッセージが囲まれているタグ・セット。タグ・セットは、XPath 表記で指定します。たとえば、発注を含む XML 文書の場合は、次の書式で発注番号タグの XML パスを指定できます。

```
/order/header/ordernumber
```

次のパスの例では、発注書の 3 行目に ITEMNO ノードがあります。

```
/order/orderlines/line[3]/itemno
```

次のパスの例では、通貨属性が「AUD」に設定されている COST ノードが、発注書の 2 行目にあります。「//」は、指定したノードがルート・ノードの下位であることを示しています。

```
//line[2]/cost[@currency="AUD"]
```

詳細は、W3C 勧告の「XPath」を参照してください。

- 項目属性： データを格納する日付、数値またはテキスト・タイプの項目属性を選択します。項目属性と取り出すデータのデータ型は、一致している必要があります。

関連項目：

4-19 ページ「[アクティビティ属性値の定義](#)」

「XML タグ値の比較」アクティビティ

「XML タグ値の比較」アクティビティでは、ビジネス・イベント・システムを介して受信したイベント・メッセージのデータを、テスト値と比較します。現在、この機能を使用できるのは Oracle Workflow のスタンドアロン版のみです。「XML タグ値の比較」アクティビティでは、イベント・メッセージ内で特殊な XML タグに囲まれているデータを、指定したテスト値と比較します。

「XML タグ値の比較（日付）」	このアクティビティでは、日付値が比較されます。
「XML タグ値の比較（数値）」	このアクティビティでは、数値が比較されます。
「XML タグ値の比較（テキスト）」	このアクティビティでは、テキスト値が比較されます。すべての「XML タグ値の比較」アクティビティは、 <code>oracle.apps.fnd.wf.XMLCompareTag</code> という外部 Java 関数をコールします。

注意： ワークフロー・エンジンでは、外部 Java 関数アクティビティを検出すると、エントリを「送信」キューに入れます。アクティビティの実行を続行するには、Java 関数アクティビティ・エージェントを実行して、該当する Java 関数をコールし、結果を「受信」キューに入れる必要があります。次に、バックグラウンド・エンジンを実行して「受信」キューを処理し、関数アクティビティを完了する必要があります。『Oracle Workflow 管理者ガイド』の「[Java 関数アクティビティ・エージェントの設定](#)」および「[バックグラウンドのワークフロー・エンジンの設定](#)」を参照してください。

アクティビティ属性

各「XML タグ値の比較」アクティビティには、次の3つのアクティビティ属性があります。

- **イベント**: イベント・タイプの項目属性を選択します。このイベント・メッセージのデータと比較します。
- **タグ**: テスト値と比較するデータが含まれる、イベント・メッセージを囲むタグ・セット。タグ・セットは、XPath 表記で指定します。たとえば、発注を含む XML 文書の場合、次の書式で発注番号タグの XML パスを指定できます。

```
/order/header/ordernumber
```

次のパスの例では、発注書の3行目に ITEMNO ノードがあります。

```
/order/orderlines/line[3]/itemno
```

次のパスの例では、通貨属性が「AUD」に設定されている COST ノードが、発注書の2行目にあります。「//」は、指定したノードがルート・ノードの下位にあることを示しています。

```
//line[2]/cost[@currency="AUD"]
```

詳細は、W3C 勧告の「XPath」を参照してください。

- **値**: イベント・メッセージのデータと比較する日付、数値またはテキスト・タイプのテスト値。

「XML タグ値の比較」アクティビティでは、結果コードに「比較」選択肢タイプを使用します。有効な値は、「より大きい」、「より小さい」、「等しい」、「NULL」（テスト・アクティビティ属性値が NULL の場合）です。イベント・メッセージのデータとテスト値との比較方法に基づいて、ワークフロー・プロセスを進めることが可能です。4-19 ページの「[アクティビティ属性値の定義](#)」を参照してください。

「XML 変換」アクティビティ

「XML 変換」アクティビティでは、ビジネス・イベント・システムのイベント・メッセージのペイロードに対して、XML スタイル・シートを適用します。現在、この機能を使用できるのは Oracle Workflow のスタンドアロン版のみです。適用した文書は、イベント・タイプの項目属性に格納されます。このアクティビティは、`oracle.apps.fnd.wf.XSLTTransform` という外部 Java 関数をコールします。

注意： ワークフロー・エンジンでは、外部 Java 関数アクティビティを検出すると、エントリを「送信」キューに入れます。アクティビティの実行を続行するには、Java 関数アクティビティ・エージェントを実行して、該当する Java 関数をコールし、結果を「受信」キューに入れる必要があります。次に、バックグラウンド・エンジンを実行して「受信」キューを処理し、関数アクティビティを完了する必要があります。『Oracle Workflow 管理者ガイド』の「Java 関数アクティビティ・エージェントの設定」および「バックグラウンドのワークフロー・エンジンの設定」を参照してください。

アクティビティ属性

「XML 変換」アクティビティには、次の 3 つのアクティビティ属性があります。

- イベント： イベント・タイプの項目属性を選択します。このイベント・メッセージを変換します。
- スタイルシート： 適用するスタイル・シートの格納場所への参照。この参照は URL として指定します。
- 新規文書： イベント・タイプの項目属性を選択します。ここに、スタイル・シートを適用して作成された新しい文書を格納します。4-19 ページの「[アクティビティ属性値の定義](#)」を参照してください。

コンカレント・マネージャの標準アクティビティ

Oracle Applications には、Oracle Applications に組み込まれている Workflow を使用する場合に、プロセス制御に使用できるように、汎用アクティビティが用意されています。これらのアクティビティは、「コンカレント・マネージャ機能」項目タイプに関連付けられていますが、定義する任意のプロセスで使用できます。

- 「コンカレント・プログラムの実行」アクティビティ
- 「コンカレント・プログラムの発行」アクティビティ
- 「コンカレント・プログラム待ち」アクティビティ

「コンカレント・マネージャ機能」項目タイプは、Oracle Applications Workflow Server に自動的にインストールされます。この項目タイプには、\$FND_TOP/admin/import サブディレクトリにあるファイル fndwfaol.wft からアクセスできます。

「コンカレント・プログラムの実行」アクティビティ

「コンカレント・プログラムの実行」アクティビティを使用できるのは、Oracle Applications に組み込まれている Workflow のみです。このアクティビティでは、ワークフロー・プロセスから Oracle Applications のコンカレント・プログラムを発行してその完了を待ち、完了時にアクティビティのステータスを更新し、ワークフロー・プロセスの実行をバックグラウンド・エンジンに戻します。コンカレント・プログラムは、「コンカレント・プログラムのステータス」選択肢タイプで定義されたとおり、NORMAL、ERROR、WARNING、CANCELLED または TERMINATED のうちのどれかの結果で完了してもかまいません。これらの結果はすべて、あらかじめプロセス・ダイアグラムに用意する必要があります。

注意：「コンカレント・プログラムの実行」アクティビティを使用するには、バックグラウンド・エンジンが実行用に設定されていることを確認する必要があります。

注意：通常、Oracle Applications フォームからセッションを開始すると、プロセスの項目タイプのコンテキストが常に設定されます。ただし、通知やアクティビティのブロックなどによってセッションが中断された場合は、セレクト関数またはコールバック関数内で、`FND_GLOBAL.APPS_INITIALIZE(user_id, resp_id, resp_appl_id)` を SET_CTX モードでコールし、コンテキストが設定されていることを確認する必要があります。6-12 ページの「[項目タイプのセレクト関数またはコールバック関数の標準 API](#)」および『Oracle Applications 開発者ガイド』の「[FNDSQF ルーチン API](#)」を参照してください。

「コンカレント・プログラムの実行」アクティビティでは、標準の Oracle Application Object Library API である FND_WF_STANDARD.EXECUTECONCPROGRAM がコールされます。

アクティビティ属性

「コンカレント・プログラムの実行」アクティビティには、次のアクティビティ属性があります。

- アプリケーション短縮名： コンカレント・プログラムが登録されているアプリケーションの短縮名。
- プログラム短縮名： 実行するコンカレント・プログラムの短縮名。
- 引数の数： コンカレント・プログラムの必須の引数の数。
- 項目属性名： コンカレント・プログラムの要求 ID を格納する、オプションの項目属性名。
- 引数 1、引数 2、... 引数 100： 各コンカレント・プログラムの引数の値が、コンカレント・プログラムの正しい構文に従った順序で示されます。ここでは最大 100 個の引数を使用できますが、このアクティビティの「引数の数」属性で定義したのと同じ数の引数値を指定する必要があります。4-19 ページの「[アクティビティ属性値の定義](#)」を参照してください。

「コンカレント・プログラムの発行」アクティビティ

「コンカレント・プログラムの発行」アクティビティを使用できるのは、Oracle Applications に組み込まれている Workflow のみです。このアクティビティでは、ワークフロー・プロセスから Oracle Applications のコンカレント・プログラムを発行しますが、このプログラムが実行または完了するまで待機するわけではありません。このアクティビティで、コンカレント要求を発行すると、ワークフロー・エンジンでは、プロセス内の次のアクティビティが続けて処理されます。

注意： 通常、Oracle Applications フォームからセッションを開始すると、プロセスの項目タイプのコンテキストが常に設定されます。ただし、通知やアクティビティのブロックなどによってセッションが中断された場合は、セレクト関数またはコールバック関数内で、`FND_GLOBAL.APPS_INITIALIZE(user_id, resp_id, resp_appl_id)` を SET_CTX モードでコールし、コンテキストが設定されていることを確認する必要があります。6-12 ページの「[項目タイプのセレクト関数またはコールバック関数の標準 API](#)」を参照してください。

「コンカレント・プログラムの発行」アクティビティでは、標準の Oracle Application Object Library API である FND_WF_STANDARD.SUBMITCONCPROGRAM をコールします。

アクティビティ属性

「コンカレント・プログラムの発行」アクティビティには、次のアクティビティ属性があります。

- アプリケーション短縮名: コンカレント・プログラムが登録されているアプリケーションの短縮名。
- プログラム短縮名: 実行するコンカレント・プログラムの短縮名。
- 引数の数: コンカレント・プログラムの必須の引数の数。
- 項目属性名: コンカレント・プログラムの要求 ID を格納する項目属性の名前。
- 引数 1、引数 2、... 引数 100: 各コンカレント・プログラムの引数の値が、コンカレント・プログラムの正しい構文に従った順序で示されます。ここでは最大 100 個の引数を使用できますが、このアクティビティの「引数の数」属性で定義したのと同じ数の引数値を指定する必要があります。4-19 ページの「[アクティビティ属性値の定義](#)」を参照してください。

「コンカレント・プログラム待ち」アクティビティ

「コンカレント・プログラム待ち」アクティビティを使用できるのは、Oracle Applications に組み込まれている Workflow のみです。ワークフロー・プロセスからコンカレント・プログラムを発行する場合は、「コンカレント・プログラム待ち」アクティビティを使用して、コンカレント・プログラムが完了するまで、それ以降のプロセスが実行されないようにできます。コンカレント・プログラムが完了すると、このアクティビティでは、アクティビティのステータスを更新してブロックを消去し、ワークフロー・プロセスの実行の制御をバックグラウンド・エンジンに戻します。コンカレント・プログラムは、「コンカレント・プログラムのステータス」選択肢タイプで定義されたとおり、NORMAL、ERROR、WARNING、CANCELLED または TERMINATED のうちどの結果で完了してもかまいません。これらの結果はすべて、あらかじめプロセス・ダイアグラムに用意する必要があります。

注意: 「コンカレント・プログラム待ち」アクティビティを使用するには、バックグラウンド・エンジンが実行用に設定されていることを確認する必要があります。

「コンカレント・プログラム待ち」アクティビティでは、標準の Oracle Application Object Library API である FND_WF_STANDARD.WAITFORCONCPROGRAM がコールされます。

アクティビティ属性

「コンカレント・プログラム待ち」アクティビティには、「要求 ID」アクティビティ属性があります。この属性には、完了を待機しているコンカレント・プログラムの要求 ID を設定する必要があります。4-19 ページの「[アクティビティ属性値の定義](#)」を参照してください。

6

Oracle Workflow のプロシージャおよび関数の定義

この章では、Oracle Workflow に対して PL/SQL および Java のプロシージャと関数を使用するときの標準 API について説明します。

Oracle Workflow のプロシージャおよび関数の定義

Oracle Workflow では、ワークフロー・プロセスやビジネス・イベント・システムの特定の場所に、PL/SQL および Java のカスタム・プロシージャと関数を取り込むことができます。カスタム・コードを Oracle Workflow で正しく動作させるには、次の標準 API に従ってプロシージャおよび関数を作成します。

- 6-3 ページ [「関数アクティビティがコールする PL/SQL プロシージャの標準 API」](#)
- 6-8 ページ [「関数アクティビティがコールする Java プロシージャの標準 API」](#)
- 6-12 ページ [「項目タイプのセレクト関数またはコールバック関数の標準 API」](#)
- 6-15 ページ [「PL/SQL 文書の標準 API」](#)
- 6-24 ページ [「イベント・データ・ジェネレート関数の標準 API」](#)
- 6-26 ページ [「キュー・ハンドラの標準 API」](#)
- 6-29 ページ [「イベント・サブスクリプションのルール関数の標準 API」](#)

関数アクティビティがコールする PL/SQL プロシージャの標準 API

Oracle ワークフロー・プロセスの関数アクティビティまたは通知アクティビティでコールされる PL/SQL スタアド・プロシージャはすべて、ワークフロー・エンジンでアクティビティを正しく実行できるように、次の標準 API の書式に従う必要があります。

注意： ワークフロー・エンジンでは、各関数アクティビティの前にセーブポイントを設定し、関数アクティビティによって生成されるエラーが検出されるようにしています。アクティビティで、処理されない例外が生成された場合、ワークフロー・エンジンはセーブポイントまでロールバックし、アクティビティのステータスを **ERROR** に設定します。このため、関数アクティビティの PL/SQL プロシージャでは、ユーザーがコミットすることはありません。コミットは、コール側のアプリケーションが行うため、ワークフロー・エンジンではコミットを発行しません。

データベース・トリガーや分散トランザクションなど、セーブポイントを使用できない環境では、ワークフロー・エンジンは自動的に「セーブポイントの使用不可」エラーを検出し、バックグラウンド・エンジンに対してアクティビティの実行を遅延させます。

この項の例には、参照しやすいように「1⇒」のように番号が付いています。この番号と矢印自体はプロシージャの一部ではありません。

```

1⇒ procedure <procedure name> (itemtype in varchar2,
                                itemkey in varchar2,
                                actid in number,
                                funcmode in varchar2,
                                resultout out varchar2) is
2⇒   <local declarations>
3⇒   begin
        if ( funcmode = 'RUN' ) then
            <your RUN executable statements>
            resultout := 'COMPLETE:<result>';
            return;
        end if;
4⇒   if ( funcmode = 'CANCEL' ) then
            <your CANCEL executable statements>
            resultout := 'COMPLETE';
            return;
        end if;
5⇒   if ( funcmode = 'RESPOND' ) then
            <your RESPOND executable statements>
            resultout := 'COMPLETE';
            return;
        end if;

```

```
6⇒ if ( funcmode = 'FORWARD' ) then
    <your FORWARD executable statements>
    resultout := 'COMPLETE';
    return;
end if;
7⇒ if ( funcmode = 'TRANSFER' ) then
    <your TRANSFER executable statements>
    resultout := 'COMPLETE';
    return;
end if;
8⇒ if ( funcmode = 'TIMEOUT' ) then
    <your TIMEOUT executable statements>
    if (<condition_ok_to_proceed>) then
        resultout := 'COMPLETE';
    else
        resultout := wf_engine.eng_timedout;
    end if;
    return;
end if;
9⇒ if ( funcmode = '<other funcmode>' ) then
    resultout := ' ';
    return;
end if;
10⇒ exception
    when others then
        WF_CORE.CONTEXT ('<package name>', '<procedure name>', <itemtype>,
            <itemkey>, to_char(<actid>), <funcmode>);
        raise;
11⇒ end <procedure name>;
```

1⇒ ワークフロー・エンジンは、関数アクティビティのストアド・プロシージャをコールするときに、プロシージャに 4 つのパラメータを渡し、プロシージャの完了時に結果を求める場合があります。次のパラメータを定義します。

itemtype	項目タイプの内部名。項目タイプは、Oracle Workflow Builder で定義します。
itemkey	項目タイプの主キーを表す文字列。主キーは、ワークフローで使用可能なアプリケーションによって生成されます。この文字列により、項目タイプの項目が一意に識別されます。
actid	このプロシージャのコール元となるアクティビティの ID 番号。

funcmode	アクティビティの実行モード。アクティビティが関数アクティビティの場合は、RUN または CANCEL モードとします。アクティビティが通知アクティビティの場合で、通知後関数を使用している場合、モードは RESPOND、FORWARD、TRANSFER、TIMEOUT または RUN のいずれかです。今後、これ以外の実行モードが追加されることもあります。
resultout	<p>Oracle Workflow Builder の「アクティビティ」プロパティ画面で、アクティビティの結果タイプが指定されている場合、このパラメータはプロシージャの完了時に戻される結果を表します。戻される結果は、次のとおりです。</p> <p>COMPLETE:<result_code>: アクティビティは指定された結果コードを戻して完了します。結果コードは、関数アクティビティの結果タイプで指定されている結果コードの 1 つと一致する必要があります。</p> <p>WAITING: アクティビティが保留中、つまり他のアクティビティが完了するまで待機している状態です。たとえば、標準の「AND」アクティビティなどです。</p> <p>DEFERRED:<date>: バックグラウンド・エンジンによるアクティビティの実行は、指定した日付まで延期されます。<date> は、次の書式で指定する必要があります。</p> <p style="padding-left: 2em;">to_char(<date_string>, wf_engine.date_format)</p> <p>NOTIFIED:<notification_id>:<assigned_user>: 処理の実行が必要なことが、外部エンティティに通知されます。必要であれば、この結果で通知 ID と割当て済のユーザーを戻すことができます。外部エンティティは、処理が完了した時点で CompleteActivity() をコールして、ワークフロー・エンジンに知らせる必要があることに注意してください。</p> <p>ERROR:<error_code>: アクティビティにエラーが発生し、指定のエラー・コードが戻されます。</p>

2⇒ このセクションでは、このプロシージャで使用されるローカル引数を宣言します。

3⇒ プロシージャ本体は、このセクションの if 文から始まります。このセクションには、funcmode の値が RUN の場合に実行される 1 つ以上の実行文が含まれます。実行文の 1 つは、プロシージャの結果を戻すことができます。たとえば、結果として「COMPLETE:APPROVED」を戻すことがあります。

注意: 通知システムが RESPOND モードで通知後関数の実行を完了すると、ワークフロー・エンジンではその通知後関数が自動的に RUN モードで実行されます。RUN モードの実行文では、投票集計などの処理を実行し、通知アクティビティに戻す結果を決定できます。

4⇒ このセクションでは、アクティビティを消去します。funcmode の値が CANCEL の場合に実行される 1 つ以上の実行文を含めることができます。通常、このセクションには実行文が含まれず、NULL 値が戻されるのみですが、必要であれば処理を取り消すことも可能で

す。そのアクティビティがループの一部として再実行される場合は、`funcmode` に `CANCEL` を指定できます。

ループ内の最初のアクティビティの場合は、「アクティビティ」プロパティの「詳細」ページで「ループ・リセット」オプションを常にオンにする必要があります。ワークフロー・エンジンが実行済のアクティビティに到達すると、そのアクティビティの「ループ・リセット」オプションが設定されているかどうかを検証されます。このオプションが「リセット」に設定されている場合は、そのループに属するアクティビティが識別され、そのアクティビティの `funcmode` が `CANCEL` に設定されます。次に、エンジンはループ内を本来の順序で進み、`CANCEL` モードの各アクティビティを実行し、そのアクティビティの以前の結果をすべて消去して、再度実行できるようにします。『Oracle Workflow API リファレンス』の「ループ」および 5-7 ページの「[「ループ・カウンタ」アクティビティ](#)」を参照してください。

5⇒ このセクションは、通知後関数の場合にのみ必要です。このセクションを使用して、`funcmode` が `RESPOND` の場合、つまり `RESPOND` の処理が行われた場合に処理される実行文を指定します。たとえば、通知の応答を検証する実行文を指定できます。通知システムが `RESPOND` モードで通知後関数を実行した後、その通知後関数はワークフロー・エンジンにより `RUN` モードで再実行されます。『Oracle Workflow API リファレンス』の「通知後関数」を参照してください。

6⇒ このセクションは、通知後関数の場合にのみ必要です。このセクションを使用して、`funcmode` が `FORWARD` の場合、つまり通知のステータスが `FORWARD` に変更された場合に処理される実行文を指定します。たとえば、通知の転送先のルールを検証する実行文を指定できます。

7⇒ このセクションは、通知後関数の場合にのみ必要です。このセクションを使用して、`funcmode` が `TRANSFER` の場合、つまり通知のステータスが `TRANSFER` に変更された場合に処理される実行文を指定します。たとえば、通知の譲渡先のルールを検証する実行文を指定できます。

注意： `funcmode` が `RESPOND`、`FORWARD` または `TRANSFER` の場合は、戻り値が「`ERROR%`」の場合を除いて、`resultout` パラメータは無視されます。したがって、通知後関数を実行した後で、`Respond`、`Forward` または `Transfer` の処理を行わない場合は、次のどちらかの方法で対処します。

- `resultout` パラメータに `ERROR:<errcode>` を戻して、このメッセージの `errcode` を一般的な例外に変換します。
 - プロシージャ内で、詳細なエラー・メッセージを付加した例外を直接発生させます。『Oracle Workflow API リファレンス』の「通知後関数」および「通知モデル」を参照してください。
-

8⇒ このセクションは、通知後関数の場合にのみ必要です。このセクションを使用して、通知アクティビティがタイムアウト値に達したときに処理される実行文を指定します。ワークフローが正常に進行できるかどうかをテストし、問題がなければワークフローが次に進める

ようにアクティビティを完了するロジックを指定できます。たとえば、投票アクティビティで、すべての宛先が応答する前にタイムアウト値に達した場合に、現行の応答プールに基づいて応答の解釈方法を決定し、適切な結果でアクティビティを完了するロジックを指定できます。

ワークフローが正常に進行できない場合に、`wf_engine.eng_timeout` の結果を戻すロジックも指定する必要があります。別のアクティビティへのタイムアウト・トランジションを使用し、プロセス・ダイアグラムの後続の動作をモデル化してください。ワークフロー・エンジンは、結果 `wf_engine.eng_timeout` が戻されると、タイムアウト・トランジションに従います。

9⇒ このセクションでは、RUN、CANCEL、RESPOND、FORWARD、TRANSFER または TIMEOUT 以外の実行モードを処理します。今後、これ以外の実行モードが追加されることもあります。アクティビティでは、これらのモードを実装する必要がないため、NULL が戻されます。

10⇒ このセクションでは、エラー・スタックにコンテキスト情報を含めてエラーの原因箇所を見つけやすいように、例外が発生した場合に `WF_CORE.CONTEXT()` をコールします『Oracle Workflow API リファレンス』の「CONTEXT」を参照してください。

関数アクティビティがコールする Java プロシージャの標準 API

Oracle Workflow プロセスでは、カスタム Java クラスを作成し、外部 Java 関数アクティビティからコールすることができます。現在、この機能を使用できるのは Oracle Workflow のスタンドアロン版のみです。関数アクティビティからコールする Java プロシージャは、WFFunctionAPI クラスの拡張クラスとして実装します。カスタム Java クラスは、Oracle Workflow の Java 関数アクティビティ・エージェントで正しく実行できるように、標準 API の書式に従う必要があります。

注意： ワークフロー・エンジンでは、各関数アクティビティの前にセーブポイントを設定し、関数アクティビティによって生成されるエラーが検出されるようにしています。アクティビティで、処理されない例外が生成された場合、ワークフロー・エンジンはセーブポイントまでロールバックし、アクティビティのステータスを **ERROR** に設定します。このため、PL/SQL プロシージャと同様に、関数アクティビティの Java プロシージャ内でコミットしないでください。コミットは、コール側のアプリケーションが行うため、ワークフロー・エンジンではコミットを発行しません。

ワークフロー・エンジンおよび Workflow 通知システムのほとんどの API には、Java メソッドが対応付けられています。Java プログラムからこれらの Java メソッドをコールすれば、Oracle Workflow と通信できます。WF 関数 API および WF 属性クラスには、Java プログラムからコールして項目タイプとアクティビティ属性にアクセスできるメソッドも含まれています。『Oracle Workflow API リファレンス』の「Oracle Workflow Java インタフェース」、「Workflow 関数 API」および「Workflow 属性 API」を参照してください。

ワークフロー・プロセス内からカスタム Java クラスを呼び出すには、そのクラスをコールする外部 Java 関数アクティビティを作成します。3-61 ページの「[関数アクティビティの作成](#)」を参照してください。

Java 関数アクティビティは、外部プロシージャとして実装されます。ワークフロー・エンジンでは、外部 Java 関数アクティビティを検出すると、メッセージを Workflow の「送信」キューに入れます。Java 関数アクティビティ・エージェントは、このキューを監視し、関数アクティビティの「機能名」プロパティに指定されたクラスをコールします。Java プロシージャが完了すると、Java 関数アクティビティ・エージェントは結果を「受信」キューにエンキューします。『Oracle Workflow 管理者ガイド』の「[Java 関数アクティビティ・エージェントの設定](#)」を参照してください。

注意： これらの「送信」キューおよび「受信」キューは、ビジネス・イベント・システムに使用するキューとは異なります。今後のリリースでは、この関数処理はビジネス・イベント・システム内に実装され、専用のキュー・ハンドラを使用してデキュー/エンキュー操作が行われる予定です。『Oracle Workflow API リファレンス』の「[Workflow QUEUE API](#)」を参照してください。

Java プロシージャが完了したら、バックグラウンド・エンジンを実行して受信キューを処理し、関数アクティビティを実行する必要があります。実行しない場合は、関数アクティビティのステータスが DEFERRED のままになります。『Oracle Workflow 管理者ガイド』の「バックグラウンドのワークフロー・エンジンの設定」を参照してください。

カスタム・クラスを Java 関数アクティビティ・エージェントで利用するには、カスタム・クラスを格納した JAR ファイルを CLASSPATH に追加する必要があります。カスタム・クラス・ファイルは、Java 関数アクティビティ・エージェントが実行されるのと同じプラットフォームに常駐させる必要があります。ただし、Java 関数アクティビティ・エージェントをデータベースと同じ層に常駐させる必要はありません。

この項の例には、参照しやすいように「1⇒」のように番号が付いています。この番号と矢印自体はプロシージャの一部ではありません。

```

1⇒ package oracle.apps.fnd.wf;
2⇒ import java.io.*;
   import java.sql.*;
   import java.math.BigDecimal;
   import oracle.sql.*;
   import oracle.jdbc.driver.*;

   import oracle.apps.fnd.common.*;
   import oracle.apps.fnd.wf.engine.*;
   import oracle.apps.fnd.wf.*;

3⇒ public class className extends WFFunctionAPI {
4⇒     public boolean execute(WFContext pWCtx) {
5⇒         ErrorStack es = pWCtx.getWFErrorStack();
           try
           {
6⇒             WFAttribute lAAttr = new WFAttribute();
             WFAttribute lIAttr = new WFAttribute();

7⇒             loadActivityAttributes(pWCtx, itemType, itemKey, actID);
             loadItemAttributes(pWCtx);

8⇒             lAAttr = getActivityAttr("AATTR");
             lIAttr = getItemAttr("IATTR");
9⇒             <your executable statements>

10⇒            lIAttr.value((Object) "NEWVALUE");
             setItemAttrValue(pWCtx, lIAttr);

11⇒         }
           catch (Exception e)
           {
               es.addMessage("WF", "WF_FN_ERROR");
               es.addToken("MODULE", this.getClass().getName());
           }
       }

```

```

        es.addToken("ITEMTYPE", itemType);
        es.addToken("ITEMKEY", itemKey);
        es.addToken("ACTID", actID.toString());
        es.addToken("FUNCMODE", funcMode);
        es.addToken("ERRMESSAGE", e.getMessage());
        return false;
    }
12⇒    return true;
    }
}

```

1⇒ デフォルトでは、Oracle Workflow が提供する Java クラスは `oracle.apps.fnd.wf` パッケージに入っています。このセクションはオプションです。

2⇒ 正しく操作するには、これらのパッケージを組み込む必要があります。

3⇒ カスタム Java クラスは、WFFunctionAPI クラスの拡張である必要があります。このクラスには、関数アクティビティの操作に必要なクラス変数およびメソッドが用意されています。

通常、PL/SQL 関数アクティビティに渡されるパラメータは、カスタム・クラスのクラス変数として利用できます。それらのパラメータは、`boolean execute()` メソッドをコールする前に初期化されます。次に、`resultOut` および `errorStack` が Oracle Workflow Engine に返されます。

完了したアクティビティのステータスは、`errorStack` 変数に値がない場合、COMPLETE に設定されます。この変数に値が入っている場合、アクティビティのステータスは ERROR に設定されます。`errorStack` 変数の内容は、WFContext クラスの `ErrorStack` クラスを使用して設定できます。例外の取込みについては、この API のセクション 5 および 11 も参照してください。

次のクラス変数が、事前に定義されています。

itemType	項目タイプの内部名。項目タイプは、Oracle Workflow Builder で定義します。
itemKey	項目タイプの主キーを表す文字列。主キーは、ワークフローで使用可能なアプリケーションによって生成されます。この文字列により、項目タイプの項目が一意に識別されます。
ActID	このプロシージャのコール元となるアクティビティの ID 番号。
funcMode	アクティビティの実行モード。外部 Java 関数アクティビティでは現在、「RUN」モードのみがサポートされています。
resultOut	Oracle Workflow Builder の「アクティビティ」プロパティ画面で、アクティビティの結果タイプが指定されている場合、このパラメータはプロシージャの完了時に戻される結果を表します。

注意： 関数アクティビティからコールされる PL/SQL プロシージャの `resultout` とは異なり、Java プロシージャの `resultOut` にはステータス・コードがありません。Java API では、結果タイプ値のみが必須です。アクティビティのステータスは、`errorStack` 変数に値が入っているかどうかに応じて、ワークフロー・エンジンが自動的に設定します。

- 4⇒ カスタム Java クラスには、`boolean execute()` メソッドを実装する必要があります。このメソッドは、Java クラスのメイン・エントリ・ポイントとなります。このメソッドは、正常に完了すると `TRUE` を返します。
- 5⇒ カスタム Java クラスの例外を取り込み、`ErrorStack` クラスを介してワークフロー・エンジンに返します。例外の取込みについては、この API のセクション 11 も参照してください。
- 6⇒ 項目属性およびアクティビティ属性を利用できるように、`WFAttribute` クラスが用意されています。
- 7⇒ 項目属性の値を Java クラスで利用するには、明示的または暗黙的にロードする必要があります。項目属性の値は、要求時にロードされます。明示的にロードするには、`void loadItemAttributes(WFContext)` または `void loadActivityAttributes(WFContext)` メソッドを使用します。`WFAttribute getItemAttr(String)` または `WFAttribute getActivityAttr(String)` メソッドをコールした場合は、暗黙的にロードされます。このセクションはオプションです。
- 8⇒ 項目属性およびアクティビティ属性の実際の値は、`WFAttribute getItemAttr(String)` および `WFAttribute getActivityAttr(String)` メソッドを介して利用します。これらの属性の値を明示的にロードしなかった場合は、この時点で自動的にロードされます。
- 9⇒ このセクションには、任意の実行文を記述します。通常は、必要な項目属性およびアクティビティ属性の詳細を取り出してから（セクション 8）、項目属性の値を設定する（セクション 10）までに追加します。
- 10⇒ `void setItemAttrValue(WFContext, WFAttribute)` メソッドを使用して項目属性の値を設定すると、ローカルの `WFAttribute` の値がデータベースに書き込まれます。`WFAttribute` クラスの値を設定するには、`WFAttribute.value(Object)` メソッドを使用する必要があります。
- 11⇒ カスタム Java クラス内の例外を取り込み、`ErrorStack` クラスを介してワークフロー・エンジンに返します。
- 外部 Java 関数アクティビティが正常に実行されなかった場合は、`FALSE` を返します。
- `WFContext.wErrorStack` クラス変数のメッセージは、すべてワークフロー・エンジンに返されます。外部 Java 関数アクティビティの完了ステータスは、この時点で `ERROR` になります。
- 12⇒ 外部 Java 関数アクティビティが正常に実行された場合は、`TRUE` を返します。

項目タイプのセレクト関数またはコールバック関数の標準 API

特定の項目タイプについて、セレクト関数とコールバック関数の両方として機能する 1 つの関数を定義できます。セレクト関数は PL/SQL プロシージャの一種で、特定の項目タイプでプロセス名が指定されていないワークフローが開始されると、実行対象の特定のプロセス定義を自動的に識別します。Oracle Workflow では、コールバック関数を使用して、項目タイプのコンテキスト情報をリセットまたはテストする機能もサポートしています。次の標準 API を使用して、1 つの PL/SQL プロシージャに、セレクト関数とコールバック関数の機能の両方を定義できます。

Oracle Workflow では、次のコマンドでセレクト関数またはコールバック関数をコールできます。

- **RUN:** 次の 2 つの条件のどちらかが発生したときに開始する適切なプロセスを選択します。
 - プロセスが WF_ENGINE.CreateProcess に明示的に渡されない場合
 - 前に WF_ENGINE.CreateProcess がコールされず、WF_ENGINE.CompleteActivity によってプロセスが暗黙的に開始される場合
- **SET_CTX:** 項目タイプの関数アクティビティを実行するために必要な項目タイプと項目キーの組合せについて、コンテキスト情報を設定します。ワークフロー・エンジンでは、新規の項目タイプと項目キーの組合せが見つかるたびに、このコマンドでセレクトまたはコールバック関数がコールされ、適切なコンテキスト情報が常に設定されていることが確認されます。
- **TEST_CTX:** 関数を実行する前に、現行の項目タイプのコンテキスト情報が正しいかどうかを判断します。たとえば、セレクトまたはコールバック関数を TEST_CTX モードで使用すると、「通知の詳細」 Web ページで参照フォームを起動する直前に、現行のコンテキスト情報でフォームを起動できるかどうかをチェックできます。コンテキストが正しくなければ、フォームは起動されず、その旨を示すメッセージが表示されます。『Oracle Workflow ユーザーズ・ガイド』の「通知の詳細の表示」を参照してください。

セレクト関数またはコールバック関数の標準 API は、次のとおりです。この項の例には、参照しやすいように「1⇒」のように番号が付いています。この番号と矢印自体はプロシージャの一部ではありません。

```

1⇒ procedure <procedure name> (item_type in varchar2,
                               item_key in varchar2,
                               activity_id in number,
                               command in varchar2,
                               resultout in out varchar2) is
2⇒   <local declarations>
3⇒   begin
       if ( command = 'RUN' ) then
           <your RUN executable statements>
           resultout := '<Name of process to run>';
       return;

```

```

end if;
4⇒ if ( command = 'SET_CTX' ) then
    <your executable statements for establishing context information>
    return;
end if;
5⇒ if ( command = 'TEST_CTX' ) then
    <your executable statements for testing the validity of the current
    ccontext information>

    resultout := '<TRUE or FALSE> ';
    return;
end if;
6⇒ if ( command = '<other command>' ) then
    resultout := ' ';
    return;
end if;
7⇒ exception
    when others then
        WF_CORE.CONTEXT ('<package name>', '<procedure name>', <itemtype>,
            <itemkey>, to_char(<actid>), <command>);
        raise;
8⇒ end <procedure name>;

```

1⇒ ワークフロー・エンジンは、セレクト関数またはコールバック関数をコールするときに、プロシージャに4つのパラメータを渡し、プロシージャの完了時に結果を求める場合があります。次のパラメータを定義します。

itemtype	項目タイプの内部名。項目タイプは、Oracle Workflow Builder で定義します。
itemkey	項目タイプの主キーを表す文字列。主キーは、ワークフローで使用可能なアプリケーションによって生成されます。この文字列により、項目タイプの項目が一意に識別されます。
actid	このプロシージャのコール元となるアクティビティの ID 番号。プロシージャが RUN コマンドを使用してコールされ、セレクト関数を実行する場合、このパラメータは常に NULL になることに注意してください。
command	セレクト関数またはコールバック関数の実行方法を決定するコマンド。「RUN」、「SET_CTX」または「TEST_CTX」のいずれかです。今後、他のコマンドが追加される可能性があります。

resultout

セレクタ関数またはコールバック関数のコールに使用するコマンドによっては、結果が戻されることがあります。

関数が「RUN」でコールされる場合は、実行されるプロセスの名前を `resultout` パラメータを介して戻す必要があります。関数が「SET_CTX」でコールされる場合、戻り値は期待されません。関数が「TEST_CTX」でコールされる場合は、コンテキストが正しければコードで「TRUE」を戻し、正しくなければ「FALSE」を戻す必要があります。それ以外の値が戻されると、Oracle Workflow では、このコマンドがコールバックによって実装されていないものと見なされます。

2⇒ セクションでは、このプロシージャで使用されるローカル引数を宣言します。

3⇒ シーシージャ本体は、このセクションの `if` 文から始まります。このセクションには、セレクタ関数を構成する 1 つ以上の実行文が含まれます。実行文は、`command` の値が「RUN」のときに実行されます。実行文の 1 つは、実行されるプロセスを反映するプロシージャの結果を戻す必要があります。たとえば、プロセス・アクティビティ名である「REQUISITION_APPROVAL」という結果を戻すことがあります。

4⇒ このセクションには、`command` の値が「SET_CTX」のときに項目タイプのコンテキスト情報を設定する、1 つ以上の実行文が含まれます。ワークフロー・エンジンでは、新規の項目タイプと項目キーの組合せが見つかるたびに、このコマンドでセレクタまたはコールバック関数がコールされてから、その組合せの関数アクティビティが実行されます。このコマンドは、データベース・セッションのアクティビティを意図したとおりに実行する前に、そのセッションで項目タイプのコンテキスト情報を設定する必要がある場合に役立ちます。たとえば、複数組織データによって左右される関数アクティビティの場合は、職責と組織のコンテキストを設定する必要があります。

5⇒ このセクションには、`command` の値が「TEST_CTX」のときに項目タイプのコンテキスト情報を検証する、1 つ以上の実行文が含まれます。ワークフロー・エンジンでは、アクティビティが実行される前に、このコマンドを使用してセレクタ関数またはコールバック関数がコールされ、現行のデータベース・セッションのコンテキストを許容できるかどうかを検証されます。たとえば、このコールバック機能は、「通知の詳細」Web ページで参照フォームが起動される直前に実行されます。このセクションのコードは、コンテキストが正しければ「TRUE」を、コンテキストが正しくなければ「FALSE」を戻す必要があります。コンテキストが正しくない場合は、例外を発生させ、WF_CORE エラー・システムにメッセージを出して、コンテキストが無効な理由を示すことができます。発生した例外は、フォームにもエラー・メッセージとして出力されます。

6⇒ このセクションでは、RUN、SET_CTX または TEST_CTX 以外の実行モードを処理します。今後、他のモードも追加される可能性があります。関数では、これらのコマンドを実装する必要がないため、NULL が戻されます。

7⇒ このセクションでは、エラー・スタックにコンテキスト情報を含めてエラーの原因箇所を見つけやすいように、例外が発生した場合に `WF_CORE.CONTEXT()` をコールします『Oracle Workflow API リファレンス』の「CONTEXT」を参照してください。

PL/SQL 文書の標準 API

項目タイプ、メッセージまたはアクティビティの属性として「文書」タイプを定義すると、ワークフロー・プロセスに文書を統合できます。Oracle Workflow は、「PL/SQL」文書、「PL/SQL CLOB」文書および「PL/SQL BLOB」文書と呼ばれる文書タイプに対応しています。PL/SQL 文書はデータを文字列で表現し、PL/SQL CLOB 文書はデータをキャラクター・ラージ・オブジェクト（CLOB）で表現し、PL/SQL BLOB 文書はデータをバイナリ・ラージ・オブジェクト（BLOB）で表現します。

Oracle Workflow では、作成する文書タイプの属性によって、文書を生成する PL/SQL プロシージャの動的コールをどのように作成するかが決定されます。メッセージ本文に、PL/SQL または PL/SQL CLOB 文書タイプのメッセージ属性を埋め込んで、通知に文書を表示できます。また、PL/SQL、PL/SQL CLOB または PL/SQL BLOB 文書タイプのメッセージ属性をメッセージに添付して、通知への添付ファイルとして文書を含めることができます。

PL/SQL、PL/SQL CLOB および PL/SQL BLOB 文書を生成する PL/SQL プロシージャは、標準 API の書式に従う必要があります。

PL/SQL 文書

PL/SQL 文書を生成する PL/SQL プロシージャには、次の標準 API を定義する必要があります。

```
procedure <procedure name> (document_id in varchar2,
                             display_type in varchar2,
                             document in out varchar2,
                             document_type in out varchar2)
```

このプロシージャの引数は、次のとおりです。

document_id	文書を一意に識別する文字列。これは、PL/SQL 文書 (plsql:<procedure>/<document_identifier>) の「属性」プロパティ画面のデフォルト値フィールドに指定する値と同じ文字列です。<procedure> は、「package.procedure」の形式で PL/SQL パッケージおよびプロシージャの名前に置き換えます。<document_identifier> は、プロシージャに直接渡される PL/SQL 引数文字列に置き換えます。引数文字列では、文書を識別する必要があります。たとえば、plsql:po_wf.show_req/2034 となります。PL/SQL 引数文字列の値を動的に生成する場合は、PL/SQL 引数の文字列を使用するかわりに、別の項目属性を作成し、そこから対象の項目属性を「&ITEM_ATTRIBUTE」として参照します。次に、別に作成した項目属性を参照するアクティビティが実行される前に、WF_ENGINE.SetItemAttribute API をコールして、PL/SQL 引数の文字列値を動的に設定します。たとえば、plsql:po_wf.show_req/&POREQ_NUMBER のように指定します。
--------------------	---

display_type	<p>通知表示に使用されるコンテンツ・タイプを表す 3 つの値のうちの 1 つ。要求タイプとも呼ばれます。</p> <p>text/plain: 文書は、電子メール・メッセージから表示される通知のプレーン・テキスト表現に埋め込まれます。メール・メッセージ全体を 32KB 以内にする必要があるため、電子メール・テンプレートのサイズによっては、プロシージャによって生成されるプレーン・テキスト文書の一部が切り捨てられることがあります。『Oracle Workflow 管理者ガイド』の「メッセージ・テンプレートの変更」を参照してください。</p> <p>text/html: 文書は、「通知」 Web ページまたは電子メール・メッセージの HTML 添付ファイルから表示される通知の HTML 表現に埋め込まれます。プロシージャでは、文書の HTML 表現を 32KB 以内で生成する必要があります。ただし、文書の挿入先となる HTML ページには、すでに <HTML> や <BODY> などの最上位レベルの HTML タグが含まれているため、これらのタグは除きます。最上位レベルの HTML タグが誤って入力されている場合は、文書属性がメッセージ本文内で参照されるときに、これらのタグが Oracle Workflow によって削除されます。また、通知システムがプレーン・テキストの前後を適切な HTML タグで自動的に囲んで書式を保持するため、プロシージャではプレーン・テキスト文書も生成できます。</p> <p>空白: 文書は、通知の別の添付ファイルとして表示されます。任意のコンテンツ・タイプを戻すことができます。</p>
document	32KB 以内の文書テキストが戻されるアウトバウンド・テキスト・バッファ。
document_type	文書のコンテンツ・タイプが戻されるアウトバウンド・テキスト・バッファ。戻りタイプとも呼ばれます。タイプを指定しない場合は、「text/plain」と見なされます。

関連項目:3-13 ページ [「文書属性の定義」](#)

PL/SQL CLOB 文書

PL/SQL CLOB 文書を生成する PL/SQL プロシージャには、次の標準 API を定義する必要があります。

```
procedure <procedure name> (document_id in varchar2,  
                             display_type in varchar2,  
                             document in out clob,  
                             document_type in out varchar2)
```

通知に添付された PL/SQL CLOB 文書には、PDF または RTF 文書を含めることができます。また、データベースのバージョンが Oracle9i Database 以降であれば、Base64 でエンコードされたバイナリ・データを含めることができます。そのためには、バイナリ・ラージ・オブジェクト (BLOB) として文書をデータベースに格納してから、CLOB を生成する PL/SQL プロシージャの一部としてその文書を CLOB に変換する必要があります。

UTL_RAW.Cast_To_VARCHAR2 関数を使用して、BLOB のデータを CLOB に書き込むための VARCHAR2 データへ変換できます。データベースのバージョンが Oracle9i Database 以降の場合は、WF_MAIL_UTIL.EncodeBLOB プロシージャを使用してバイナリ・データを Base64 にエンコードすることもできます。『Oracle9i PL/SQL パッケージ・プロシージャおよびタイプ・リファレンス』の「UTL_RAW」および『Oracle Workflow API リファレンス』の「EncodeBLOB」を参照してください。

注意： WF_NOTIFICATION.WriteToClob() をコールして、文字データの文字列を CLOB に追加することにより、CLOB を簡単に作成できます。『Oracle Workflow API リファレンス』の「WriteToClob」を参照してください。

注意： Oracle8i Database では Base64 のエンコードがサポートされないため、Oracle8i Database を使用している場合は、WF_MAIL_UTIL.EncodeBLOB プロシージャが利用できません。また、PL/SQL CLOB 文書には PDF または RTF 文書以外のバイナリ・データを格納できません。この機能は、Oracle9i Database 以降を使用している場合に限られます。

ただし、Oracle8i Database と Oracle9i Database 以降の両方で UTL_RAW.Cast_To_VARCHAR2 関数がサポートされるので、どのデータベースのバージョンでも、添付される PL/SQL CLOB 文書内に PDF 文書と RTF 文書を格納できます。

このプロシージャの引数は、次のとおりです。

document_id	<p>文書を一意に識別する文字列。これは、PL/SQL CLOB 文書 (plsqcllob:<procedure>/<document_identifier>) の「属性」プロパティ画面のデフォルト値フィールドに指定する値と同じ文字列です。<procedure> は、「package.procedure」の形式で PL/SQL パッケージおよびプロシージャの名前に置き換えます。<document_identifier> は、プロシージャに直接渡される PL/SQL 引数文字列に置き換えます。引数文字列では、文書を識別する必要があります。たとえば、</p> <pre>plsqcllob:po_wf.show_req_clob/2036</pre> となります。PL/SQL 引数文字列の値を動的に生成する場合は、PL/SQL 引数の文字列を使用するかわりに、別の項目属性を作成し、そこから対象の項目属性を「&ITEM_ATTRIBUTE」として参照します。次に、別に作成した項目属性を参照するアクティビティが実行される前に、WF_ENGINE.SetItemAttribute API をコールして、PL/SQL 引数の文字列値を動的に設定します。たとえば、 <pre>plsqcllob:po_wf.show_req_clob/&POREQ_NUMBER</pre> のように指定します。
display_type	<p>通知表示に使用されるコンテンツ・タイプを表す 3 つの値のうちの 1 つ。要求タイプとも呼ばれます。</p> <p>text/plain: 文書は、通知のプレーン・テキスト表現に埋め込まれます。</p> <p>text/html: 文書は、「通知」Web ページから表示される通知の HTML 表現に埋め込まれます。プロシージャでは、文書の HTML 表現を生成する必要があります。ただし、文書の挿入先となる HTML ページには、すでに <HTML> や <BODY> などの最上位レベルの HTML タグが含まれているため、これらのタグは除きます。最上位レベルの HTML タグが誤って入力されている場合は、文書属性がメッセージ本文内で参照されるときに、これらのタグが Oracle Workflow によって削除されます。また、通知システムがプレーン・テキストの前後を適切な HTML タグで自動的に囲んで書式を保持するため、プロシージャではプレーン・テキスト文書も生成できます。</p> <p>空白: 文書は、通知の別の添付ファイルとして表示されます。任意のコンテンツ・タイプを戻すことができます。</p>
document	<p>文書テキストが格納される場所を指定するアウトバウンド LOB ロケータ。このロケータは一時的な LOB ロケータなので、このロケータの値を置き換えるのではなく、文書テキストを書き込む必要があります。この値が上書きされると、一時 LOB は暗黙的には解放されません。詳細は、『Oracle9i アプリケーション開発者ガイド - ラージ・オブジェクト』の「一時 LOB」を参照してください。</p>

document_type

文書のコンテンツ・タイプが戻されるアウトバウンド・テキスト・バッファ。戻りタイプとも呼ばれます。タイプを指定しない場合は、「text/plain」と見なされます。PDF 文書または RTF 文書の場合、この引数は、application/pdf または application/rtf などの適切な MIME (Multi-purpose Internet Mail Extension) を指定する必要があります。また、この引数の一部として添付ファイルのファイル名を指定することもできます。引数内の前の値とファイル名を区切るためにセミコロン (;) を使用し、「name=<filename>」の形式でファイル名を指定します。等号 (=) の前後には空白を入れしないでください。たとえば、次のコマンドで document_type の値を設定できます。

```
document_type := 'application/pdf; name=filename.pdf';
```

注意： データベースのバージョンが Oracle9i Database 以降で、PL/SQL CLOB 文書にデータを格納するために WF_MAIL_UTIL.EncodeBLOB API を使用してバイナリ・データを Base64 にエンコードする場合は、application/doc、application/pdf、image/jpg または image/gif のように、application または image のプライマリ・タイプとともに、適切な MIME タイプを document_type パラメータに指定する必要があります。MIME タイプの後にはセミコロン (;) を入れ、その後にエンコード仕様の encoding=base64 を指定します。等号 (=) の前後には空白を入れしないでください。また、この引数の一部として添付ファイルのファイル名を指定することもできます。引数内の前の値とファイル名を区切るためにセミコロン (;) を使用し、「name=<filename>」の形式でファイル名を指定します。等号 (=) の前後には空白を入れしないでください。たとえば、次のコマンドで document_type の値を設定できます。

```
document_type := 'image/jpg; encoding=base64; name=filename.jpg';
```

例

次の例は、PDF または RTF 文書を含む PL/SQL CLOB 文書を作成するためのプロシージャです。UTL_RAW.Cast_To_VARCHAR2 関数を使用して、BLOB を CLOB へ変換しています。

```
procedure cdoc (document_id in varchar2,
               display_type in varchar2,
               document in out clob,
               document_type in out varchar2)

is

    bdoc blob;
    cdoc clob;
    content_type varchar2(100);

    lob_id number;
    amount number;

    bdoc_size number;
    block number := 10000;
    blockCount number;
    rawBuff RAW(32000);
    pos number;
    charBuff varchar2(32000);
    charBuff_size number;

    filename varchar2(200);
    location varchar2(200);
    data_type varchar2(100);

begin
    dbms_lob.createTemporary(cdoc, FALSE, dbms_lob.call);

    -- Determine the document to display from the
    -- Document ID
    lob_id := to_number(document_id);

    -- Obtain the BLOB version of the document
    select filename, location, content_type, data_type, bdata
    into filename, location, content_type, data_type, bdoc
    from sjm_lobs
    where id = lob_id;

    -- recast the BLOB to a CLOB
    bdoc_size := dbms_lob.getLength(bdoc);
    if block < bdoc_size then
```

```
    blockCount := round((bdoc_size/block)+0.5);
else
    blockCount := 1;
end if;
pos := 1;
for i in 1..blockCount loop
    dbms_lob.read(bdoc, block, pos, rawBuff);
    charBuff := utl_raw.cast_to_varchar2(rawBuff);
    charbuff_size := length(charBuff);
    dbms_lob.writeAppend(cdoc, charbuff_size, charBuff);
    pos := pos + block;
end loop;

-- Now copy the content to the document
amount := dbms_lob.getLength(cdoc);
dbms_lob.copy(document, cdoc, amount, 1, 1);

-- Set the MIME type as a part of the document_type.
document_type := content_type||'; name=||filename;

exception
when others then
    wf_core.context('LOBDOC_PKG', 'cdoc',
                   document_id, display_type);
    raise;
end cdoc;
```

関連項目：

3-13 ページ [「文書属性の定義」](#)

PL/SQL BLOB 文書

PL/SQL CLOB 文書を生成する PL/SQL プロシージャには、次の標準 API を定義する必要があります。

```
procedure <procedure name> (document_id in varchar2,  
                             display_type in varchar2,  
                             document in out blob,  
                             document_type in out varchar2)
```

このプロシージャの引数は、次のとおりです。

document_id	<p>文書を一意に識別する文字列。これは、PL/SQL BLOB 文書 (plsqlblob:<procedure>/<document_identifier>) の「属性」プロパティ画面のデフォルト値フィールドに指定する値と同じ文字列です。<procedure> は、「package.procedure」の形式で PL/SQL パッケージおよびプロシージャの名前に置き換えます。<document_identifier> は、プロシージャに直接渡される PL/SQL 引数文字列に置き換えます。引数文字列では、文書を識別する必要があります。たとえば、</p> <pre>plsqlblob:po_wf.show_req_blob/2038</pre> となります。PL/SQL 引数文字列の値を動的に生成する場合は、PL/SQL 引数の文字列を使用するかわりに、別の項目属性を作成し、そこから対象の項目属性を「&ITEM_ATTRIBUTE」として参照します。次に、別に作成した項目属性を参照するアクティビティが実行される前に、WF_ENGINE.SetItemAttribute API をコールして、PL/SQL 引数の文字列値を動的に設定します。たとえば、 <pre>plsqlblob:po_wf.show_req_blob/&POREQ_NUMBER</pre> のように指定します。
display_type	<p>PL/SQL BLOB 文書の場合、通知表示に使用されるコンテンツ・タイプを表すには、この値を空白にする必要があります。このタイプは、要求タイプとも呼ばれます。</p> <p>空白： 文書は、通知の別の添付ファイルとして表示されます。任意のコンテンツ・タイプを戻すことができます。</p>
document	<p>文書テキストが格納される場所を指定するアウトバウンド LOB ロケータ。このロケータは一時的な LOB ロケータなので、このロケータの値を置き換えるのではなく、文書テキストを書き込む必要があります。この値が上書きされると、一時 LOB は暗黙的には解放されません。詳細は、『Oracle9i アプリケーション開発者ガイド - ラージ・オブジェクト』の「一時 LOB」を参照してください。</p>

document_type

文書のコンテンツ・タイプが戻されるアウトバウンド・テキスト・バッファ。戻りタイプとも呼ばれます。タイプを指定しない場合は、「text/plain」と見なされます。この引数には、application/doc、application/pdf、image/jpg、または image/gif などのように、application または image のどちらかのプライマリ・タイプとともに、適切な MIME タイプを指定する必要があります。また、この引数の一部として添付ファイルのファイル名を指定することもできます。引数内の前の値とファイル名を区切るためにセミコロン (;) を使用し、「name=<filename>」の形式でファイル名を指定します。等号 (=) の前後には空白を入れしないでください。たとえば、次のコマンドで document_type の値を設定できます。

```
document_type := 'image/jpg; name=filename.jpg';
```

関連項目：

3-13 ページ [「文書属性の定義」](#)

イベント・データ・ジェネレート関数の標準 API

ビジネス・イベント・システムにイベントを定義するときに、ジェネレート関数をイベントに割り当てることにより、イベント名、イベント・キーおよびオプションのパラメータ・リストから完全なイベント・データを生成することができます。イベント・データには、発生するイベントおよび作成される XML 文書の詳細が追加されます。イベントを呼び出すアプリケーションがイベント・データを生成しない場合は、ジェネレート関数を指定する必要があります。

イベントがローカルで呼び出されると、イベント・マネージャはサブスクリプションを実行する前に、各サブスクリプションにイベント・データが必要かどうかをチェックします。必要なイベント・データが渡されていない場合、イベント・マネージャはそのイベントに対してジェネレート関数をコールし、イベント・データを生成します。ジェネレート関数は、キャラクタ・ラージ・オブジェクト (CLOB) 形式のイベント・データを返します。

注意： イベント・データを必要とするイベントにジェネレート関数が定義されていない場合、イベント名およびイベント・キーを使用してデフォルトのイベント・データが作成されます。

注意： ジェネレート関数の負荷が大きいため、イベントの呼び出し後に制御をできるだけ早くコール側アプリケーションに戻りたい場合は、完全なイベント・データを必要とするすべてのサブスクリプションを遅延させることができます。後でそれらのサブスクリプションを実行するまで、イベント・マネージャはジェネレート関数を実行しません。8-49 ページの「[遅延サブスクリプション処理](#)」を参照してください。

イベント・データを生成する PL/SQL 関数には、次の標準 API を定義する必要があります。

```
function <function_name> (p_event_name in varchar2,  
                          p_event_key in varchar2  
                          p_parameter_list in wf_parameter_list_t  
                          default null)  
  return clob;
```

この関数の引数は、次のとおりです。

p_event_name	イベントの内部名。
p_event_key	プログラムまたはアプリケーション内でイベントが発生したときに、生成される文字列。このイベント・キーにより、イベントの特定のインスタンスが一意に識別されます。
p_parameter_list	(オプション) イベントの追加したパラメータの、名前と値の組合せのリスト。

関連項目：

8-7 ページ [「イベントの定義」](#)

8-52 ページ [「イベント・サブスクリプションの定義」](#)

『Oracle Workflow API リファレンス』の「パラメータ・リスト構造」

キュー・ハンドラの標準 API

ビジネス・イベント・システムでエージェントを定義したときは、そのエージェントにキュー・ハンドラを割り当てる必要があります。キュー・ハンドラとは、WF_EVENT_T データ型によって定義される標準の Workflow イベント・メッセージ形式と、エージェントに関連付けられたキューに必要なメッセージ形式との間で、変換を行うパッケージのことです。

Oracle Workflow には、WF_EVENT_T 形式を使用するキューに対して、WF_EVENT_QH (通常の処理用) および WF_ERROR_QH (エラー処理用) という、2つの標準キュー・ハンドラが用意されています。Oracle Workflow には、WF_EVENT_OJMSTEXT_QH という名前の標準キュー・ハンドラも用意されています。このキュー・ハンドラでは、ペイロード形式として JMS テキスト・メッセージが使用されます。また、WF_EVENT_OMB_QH という標準キュー・ハンドラも用意されています。Oracle8i Database に Oracle Message Broker を実装してシステム間でイベント・メッセージを伝播する場合には、このキュー・ハンドラを設定して使用できます。

他の形式を使用するキューに対して、独自のカスタム・キュー・ハンドラを作成することもできます。カスタム・キュー・ハンドラを作成する場合は、パッケージ内で次の標準 API を定義する必要があります。

- 6-27 ページ [「エンキュー」](#)
- 6-28 ページ [「デキュー」](#)

関連項目：

『Oracle Workflow API リファレンス』の「イベント・メッセージ構造」

8-25 ページ [「エージェント」](#)

『Oracle Workflow 管理者ガイド』の「WF_EVENT_OMB_QH キュー・ハンドラの設定」

『Oracle Workflow API リファレンス』の
「WF_EVENT_T および SYS.AQ\$_JMS_TEXT_MESSAGE 間のマッピング」

エンキュー

キュー・ハンドラ・パッケージのエンキュー・プロシージャでは、アウトバウンド・エージェントに関連付けられたキューにイベント・メッセージをエンキューする必要があります。オーバーライド・エージェントを指定して、そこにイベント・メッセージをエンキューすることもできます。指定しない場合、イベント・メッセージはメッセージ内に指定されている送信元エージェントにエンキューされます。エンキュー・プロシージャでは、必要に応じてイベント・メッセージのヘッダー情報を変換し、キューが必要とする形式でメッセージをエンキューします。

イベント・メッセージが送信されると、WF_EVENT.Enqueue 汎用プロシージャは指定されているアウトバウンド・エージェントから対応するキュー・ハンドラを判断し、そのキュー・ハンドラのエンキュー・プロシージャをコールしてメッセージをエンキューします。

PL/SQL エンキュー・プロシージャには、次の標準 API を定義する必要があります。

```
procedure enqueue (p_event in WF_EVENT_T,  
                  p_out_agent_override in WF_AGENT_T);
```

このプロシージャの引数は、次のとおりです。

p_event	イベント・メッセージ。
p_out_agent_override	アウトバウンド・エージェント。ここに関連付けられているキューに対して、イベント・メッセージをエンキューします。

デキュー

キュー・ハンドラ・パッケージのデキュー・プロシージャでは、デキューするメッセージをメッセージ優先度の順に選択して、指定されているインバウンド・エージェントに関連付けられたキューからイベント・メッセージをデキューする必要があります。このプロシージャでは、必要に応じてイベント・メッセージのヘッダー情報を変換し、標準の `WF_EVENT_T` 構造でイベント・メッセージを返します。また、デキュー・プロシージャでは、メッセージがイベント・メッセージの `RECEIVE_DATE` 属性にデキューされる日時を指定できます。

イベント・メッセージが着信すると、`WF_EVENT.Listen` プロシージャは指定されているインバウンド・エージェントから使用するキュー・ハンドラを判断し、そのキュー・ハンドラのデキュー・プロシージャをコールしてメッセージをデキューします。

PL/SQL Dequeue プロシージャには、次の標準 API を定義する必要があります。

```
procedure dequeue (p_agent_guid in raw,  
                  p_event out WF_EVENT_T);
```

このプロシージャの引数は、次のとおりです。

p_agent_guid	インバウンド・エージェントのグローバル一意識別子。このインバウンド・エージェントに、イベント・メッセージをデキューするキューが関連付けられています。
p_event	イベント・メッセージ。

イベント・サブスクリプションのルール関数の標準 API

イベント・サブスクリプションを定義すると、イベント・メッセージに対してルール関数と呼ばれる PL/SQL 関数を実行できます。Oracle Workflow には、基本的なサブスクリプション処理を実行するための標準のデフォルト・ルール関数が用意されています。この関数は、サブスクリプションに対してルール関数が指定されていない場合に、デフォルトで実行されます。デフォルトのルール関数では、次の処理が行われます。

- ワークフロー・プロセスへのイベント・メッセージの送信（サブスクリプション定義で指定されている場合）
- エージェントへのイベント・メッセージの送信（サブスクリプション定義で指定されている場合）

関連項目：

『Oracle Workflow API リファレンス』の「Default_Rule」

Oracle Workflow には、テストおよびデバッグ、またはその他の目的で使用可能な標準のルール関数もいくつか用意されています。『Oracle Workflow API リファレンス』の「イベント・サブスクリプションのルール関数の API」を参照してください。

サブスクリプション処理は、カスタム・ルール関数を作成することで拡張できます。カスタム・ルール関数は、標準 API に従って定義する必要があります。

ルール関数は、イベント・メッセージに対して読取りまたは書込みを行ったり、任意のデータベース処理を実行したりします。ただし、ルール関数内ではコミットしないでください。コミットは、コール側のアプリケーションが行うため、イベント・マネージャではコミットを発行しません。また、ルール関数を使用して、セキュリティや NLS の設定などの接続コンテキストを変更しないでください。

注意： ルール関数を使用してイベント・メッセージに書き込んだ場合、そのイベントに対して実行した後続のサブスクリプションは、変更したメッセージにアクセスします。

特定のイベントに対してサブスクリプション処理を実行するときに、連続したステップがいくつか含まれている場合は、再利用できない複雑で特別なルール関数を作成するのではなく、再利用できる単純なルール関数を作成して複数のサブスクリプションをイベントに定義することをお勧めします。複数のサブスクリプションの実行順序を指定するには、サブスクリプションに対してフェーズ値を入力します。

デフォルトでは、イベント・マネージャではイベント・メッセージの相関 ID としてイベント・キーが使用されます（他の相関 ID が指定されていない場合）。別の相関 ID を指定する場合は、WF_EVENT_FUNCTIONS_PKG.AddCorrelation を使用して相関 ID をイベント・メッセージに追加します。カスタム・ルール関数からこの関数をコールするか、WF_EVENT_FUNCTIONS_PKG.AddCorrelation をルール関数として使用する別のサブス

リブジョンを定義してください。『Oracle Workflow API リファレンス』の「AddCorrelation」を参照してください。

イベント・メッセージに対してカスタム・コードを実行してから、そのメッセージをワークフロー・プロセスまたはエージェントに送信する場合は、ルール関数に送信処理を組み込むか、デフォルト・ルール関数を使用して送信処理を実行するサブスクリプションを個別に定義する必要があります。

- WF_ENGINE.Event() をコールすると、イベント・メッセージがワークフロー・プロセスに送信されます。
- WF_EVENT.Send() をコールすると、イベント・メッセージがエージェントに送信されます。
- WF_RULE.Default_Rule() をコールすると、ワークフロー・プロセスおよびエージェントにイベント・メッセージを送信できる、デフォルトのサブスクリプション処理が組み込まれます。

注意： イベント・マネージャにサブスクリプションを定義するときに、ルール関数を定義する以外に、ワークフロー項目タイプ、ワークフロー・プロセス名、送信エージェント、宛先エージェント、優先度および送信処理のパラメータを定義できます。すべてのルール関数でこれらの送信属性を利用できますが、デフォルト・ルール関数を使用しない場合は、カスタム・ルール関数に送信処理を明示的に組み込む必要があります（同一のサブスクリプションからイベントを送信する場合）。

ルール関数の標準 API は、次のとおりです。この項の例には、参照しやすいように「1⇒」のように番号が付いています。この番号と矢印自体はプロシージャの一部ではありません。

```

1⇒ function <function_name> (p_subscription_guid in raw,
                             p_event in out WF_EVENT_T) return varchar2 is
2⇒   <local declarations>
3⇒   begin
4⇒     <your executable statements>
5⇒     <optional code for WARNING>
6⇒     WF_CORE.CONTEXT('<package name>', '<function name>',
                       p_event.getEventName(), p_subscription_guid);
       WF_EVENT.setErrorInfo(p_event, 'WARNING');
       return 'WARNING';
5⇒   return 'SUCCESS';
6⇒   exception
       when others then
         WF_CORE.CONTEXT('<package name>', '<function name>',
                           p_event.getEventName(), p_subscription_guid);
         WF_EVENT.setErrorInfo(p_event, 'ERROR');
         return 'ERROR';
7⇒   end;
```

1⇒ イベント・マネージャがルール関数をコールすると、2つのパラメータがルール関数に渡されます。関数が完了すると、リターン・コードが返されます。次のパラメータを定義します。

p_subscription_guid サブスクリプションのグローバル意識別子。
p_event イベント・メッセージ。

この関数は、次のステータス・コードのいずれかを返す必要があります。

- **SUCCESS:** ルール関数が正常に完了しました。
- **WARNING:** 警告状態が発生しました。ルール関数は、エラー処理用の **Workflow** コア API を使用して警告メッセージを報告し、警告情報をイベント・メッセージに設定します。イベント・マネージャは、イベント・メッセージのコピーを **WF_ERROR** キューに格納し、サブスクリプション処理を続行します。
- **ERROR:** エラーが発生しました。ルール関数は、エラー処理用の **Workflow** コア API を使用してエラー・メッセージを報告し、エラー情報をイベント・メッセージに設定します。イベント・マネージャは、このイベントのサブスクリプション処理を停止し、イベントに対して実行済のサブスクリプションをすべてロールバックし、イベント・メッセージを **WF_ERROR** キューに格納します。

2⇒ このセクションでは、この関数で使用されるローカル引数を宣言します。

3⇒ プロシージャ本体は、このセクションから始まります。このセクションには、ルール関数を構成する 1 つ以上の実行文を記述します。

4⇒ このセクション (オプション) では、エラー・スタックにコンテキスト情報を含めてエラーの原因箇所を見つけやすいように、警告状態が発生した場合に **WF_CORE.CONTEXT()** をコールします。また、警告情報をイベント・メッセージに設定して、ステータス・コード「**WARNING**」を返します。『Oracle Workflow API リファレンス』の「**CONTEXT**」を参照してください。

5⇒ このセクションでは、ルール関数の通常処理が正常に完了したときに、ステータス・コード「**SUCCESS**」を返します。

6⇒ このセクションでは、エラー・スタックにコンテキスト情報を含めてエラーの原因箇所を見つけやすいように、例外が発生した場合に **WF_CORE.CONTEXT()** をコールします。また、エラー情報をイベント・メッセージに設定して、ステータス・コード「**ERROR**」を返します。『Oracle Workflow API リファレンス』の「**CONTEXT**」を参照してください。

注意: ルール関数で例外が発生した場合は、イベント・マネージャによってそのイベントのサブスクリプション処理がすべてロールバックされ、コール側のアプリケーションにエラーが発生します。この場合、イベント・メッセージは **WF_ERROR** キューに格納されません。

関連項目：

8-52 ページ [「イベント・サブスクリプションの定義」](#)

『Oracle Workflow API リファレンス』の「イベント・メッセージ構造」

『Oracle Workflow API リファレンス』の「Workflow CORE API」

『Oracle Workflow API リファレンス』の「イベント・サブスクリプションのルール関数の API」

『Oracle Workflow API リファレンス』の「Event」

『Oracle Workflow API リファレンス』の「Send」

『Oracle Workflow API リファレンス』の「Default_Rule」

『Oracle Workflow API リファレンス』の「SetErrorInfo」

7

ワークフロー定義のテスト

この章では、Oracle Workflow の「プロセスの開始」Web ページを使用してワークフロー定義をテストする方法について説明します。

ワークフロー定義のテスト

Oracle Workflow には「プロセスの開始」と呼ばれる Web ベースのインタフェースがあり、これを使用して、データベースに定義し保存したあらゆるワークフロー定義をテストします。「プロセスの開始」にアクセスできるのは、ワークフロー管理者のロールに属しているユーザーのみです。

注意： Oracle Applications に組み込まれている Workflow で使用できる Oracle Applications Framework ユーザー・インタフェース形式での開発者スタジオについての詳細は、7-6 ページの「[開発者スタジオによるワークフロー定義のテスト](#)」を参照してください。

あらゆる Oracle Workflow データベースに対して「プロセスの開始」 Web ページを実行できますが、テスト目的の場合は別の環境を作成することをお勧めします。ワークフロー定義をテストするには、テスト環境で次の項目を設定する必要があります。

- テストするユーザーとロールを定義します。Oracle Workflow のデモンストレーション・データ・モデルで事前定義されているユーザーとロールに対してテストできます。10-6 ページの「[購買申請データ・モデルのインストール](#)」を参照してください。
- Oracle Workflow のスタンドアロン版を使用していて、通知の表示に「通知」 Web ページを使用する場合は、テストするユーザーとロールを Web セキュリティ・システムで定義する必要があります。詳細は、使用している Web サーバーのドキュメントを参照してください。
- 通知の表示に電子メールを使用する場合は、通知メーカーを構成するときに Test Address 構成パラメータを設定し、すべての通知を 1 つのテスト用電子メール・アドレスに送ります。詳細は、Oracle Applications に組み込まれている Workflow の場合は Oracle Applications Manager オンライン・ヘルプの Oracle Workflow Manager の項、スタンドアロン版の Oracle Workflow の場合は Oracle Enterprise Manager オンライン・ヘルプの Oracle Workflow Manager の項を参照してください。

▶ ワークフロー定義のテスト

1. Web ブラウザを使用して、次の URL で Oracle Workflow ホーム・ページに接続します。『Oracle Workflow 管理者ガイド』の「[Oracle Workflow ホーム・ページへのアクセス](#)」を参照してください。
2. 「プロセスの開始」リンクを選択し、「プロセスの開始」 Web ページを表示します。

項目タイプ	内部名	説明
ドキュメントのレビュー	REVIEW	
定期警告	WFALEERT	\$Header: wfalect.wft 26.3 2001/07/30 22:07:45 kma ship \$ Periodic Alert process sample. Checks for errors and sends summary FYI notification when errors are detected.
購買申請	WFDDEMO	\$Header: wfdemo.wft 26.3 2001/07/30 22:12:28 kma ship \$ Requisition Approval demo for Oracle Workflow Version 2.5 Standalone
文書管理	WFDMM	\$Header: wfdm.wft 26.3 2001/07/30 22:05:46 kma ship \$ Document Management Integration demo for Oracle Workflow
イベントシステムのデモ	WFEVDEME	イベントシステムのデモ。あるビジネス・イベント・システムがBuyer Systemとなり、別のビジネス・イベント・システムがSupplier Systemとなります。
ワークフロー・エージェントのPing/確認	WFPING	完了前に全インバウンド・エージェントにPingし、承認メッセージを待機するマスター・デテール・プロセス
ワークフロー送信プロトコル	WFSNDPRT	イベントシステムからメッセージを受信し、エージェントをイベント・サブスクリプション定義にルートします。ワークフロー・プロセスは確認メッセージを待機送信します。
製品アンケート	WFSURV	\$Header: wfsrv.wft 26.3 2001/07/30 22:10:08 kma ship \$ Product Survey demo for Oracle Workflow

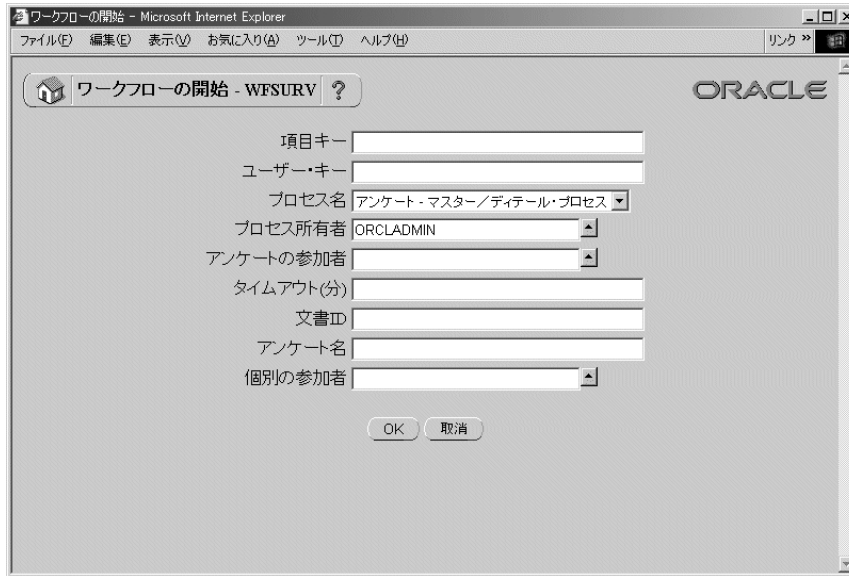
注意： 保護されている次の URL を使用し、直接このページに接続する方法もあるので注意してください。

```
<webagent>/wf_initiate.ItemType
```

<webagent> は、Web サーバーで Oracle Workflow 用に構成された Web エージェントのベース URL を表します。『Oracle Workflow 管理者ガイド』の「グローバル・ユーザー設定の設定」を参照してください。

注意： このページにはセキュリティが適用されるため、現行 Web セッションで有効なワークフロー管理者としてログオンしていない場合は、ページが表示される前に有効なワークフロー管理者としてのログオンを求めるプロンプトが表示されます。

- 「プロセスの開始」ページには、「システム：エラー」、「システム：メーラー」および「標準」など、Oracle Workflow がシードした項目タイプを除き、データベースに格納されている項目タイプ定義がすべて表示されます。各項目タイプの内部名と説明も表示されます。テストするワークフロー・プロセスの定義を所有する項目タイプを選択します。



- 「ワークフローの開始」Web ページを使用して、開始するプロセスの詳細を指定します。ワークフロー・プロセスのインスタンスを実行するには、次の項目を指定する必要があります。

- プロセス・インスタンスの一意の項目キー

注意： プロセス・インスタンスの項目キーには、シングルバイト・キャラクターのみ使用できます。マルチバイトの値を含めることはできません。

- プロセスの識別に使用するユーザー定義のキー
- テストするプロセスの名前
- オプションのプロセス所有者
- プロセスの項目タイプに関連した、あらゆる項目タイプ属性の値

「OK」を選択します。ワークフロー・プロセスを開始するために、「ワークフローの開始」Web ページでは、指定した項目タイプや項目キーに対して、ワークフロー・エンジンの `CreateProcess` や `StartProcess` などの API がコールされます。ワークフロー・エンジンの `SetItemOwner` や `SetItemAttr` などの API もコールされ、プロセス所有者とすべての項目タイプ属性が指定の値に設定されます。

- 実行したプロセス・インスタンスに対するワークフロー・モニターのアクティビティ・リストが表示されます。アクティビティ・リストには、実行されているアクティビティ

のステータスが示されます。「ダイアグラム表示」ボタンを選択し、ワークフロー・モニターにプロセスのステータスをグラフィカルに表示することもできます。『Oracle Workflow 管理者ガイド』の「「アクティビティ・リスト」のアクティビティのフィルタ」を参照してください。

6. テストしているプロセスに通知が含まれている場合は、Workflow ホーム・ページに戻り、「通知の検索」リンクを選択して、プロセスを完了するために応答を必要とする未処理の通知を検索します。また、通知の応答を電子メールでテストする場合は、通知メーラーで指定したテスト用電子メール・アカウントに接続し、プロセスの未処理の通知に応答します。

開発者スタジオによるワークフロー定義のテスト

注意： Oracle Workflow の Web ページは、Oracle Applications Framework のユーザー・インタフェース形式に変換されます。Oracle Workflow のバージョンおよび適用したパッチに応じて、新しい形式または以前の形式でも Oracle Workflow の Web ページを表示できます。現在、Oracle Applications に組み込まれている Workflow で、開発者スタジオを使用できます。

開発者スタジオを使用すると、データベースに格納されたワークフロー定義を表示することができます。ワークフロー管理者権限がある場合は、ワークフロー定義のテスト・ワークフロー・プロセスを実行することもできます。ワークフロー管理者権限は、「グローバル・ワークフロー設定」ページで割り当てられます。『Oracle Workflow 管理者ガイド』の「グローバル・ユーザー設定の設定」を参照してください。

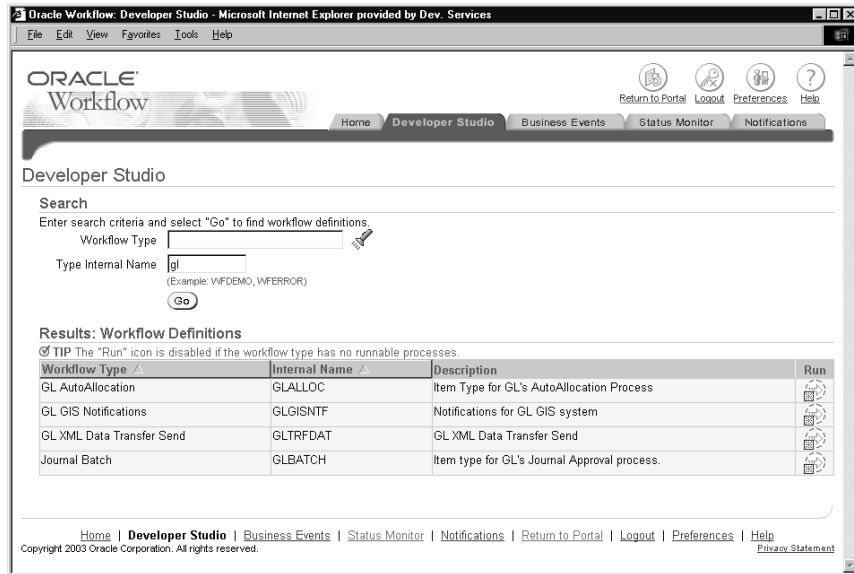
あらゆる Oracle Workflow データベースに対してテスト・ワークフロー・プロセスを実行できますが、テスト目的の場合は別の環境を作成することをお勧めします。テストを始める前に、テスト環境にテスト用のユーザーとロールを設定する必要があります。

注意： 通知の表示に電子メールを使用する場合は、通知メーラーを構成するときに Test Address 構成パラメータを設定し、すべての通知を 1 つのテスト用電子メール・アドレスに送ります。詳細は、Oracle Applications に組み込まれている Workflow の場合は Oracle Applications Manager オンライン・ヘルプの Oracle Workflow Manager の項、スタンドアロン版の Oracle Workflow の場合は Oracle Enterprise Manager オンライン・ヘルプの Oracle Workflow Manager の項を参照してください。

▶ 開発者スタジオによるワークフロー定義の検索

1. システム管理者によって指定された職責およびナビゲーション・パスを使用して、Web ブラウザで開発者スタジオにアクセスします。A-2 ページの「[Oracle Workflow 開発者のナビゲーション・パス](#)」を参照してください。

Oracle Workflow の管理者用 Web ページの「開発者スタジオ」タブまたは下部にある「開発者スタジオ」リンクを選択して、開発者スタジオへ移動することができます。



2. 特定のワークフロー定義を検索する場合は、「検索」領域に検索基準を入力します。検索基準として、次の項目を使用できます。
 - Workflow タイプ： 検討するワークフローの項目タイプを選択します。フィールドの検索アイコンを選択すると、選択可能な値のリストが表示されます。ワークフロー・タイプの表示名、内部名または説明を基準として検索を実行できます。選択したワークフロー・タイプの表示名が「Workflow タイプ」フィールドに、内部名が「タイプ内部名」フィールドに入ります。8-85 ページの「[値リストの使用](#)」を参照してください。
 - タイプ内部名： 内部名の値を選択しないで直接入力する場合は、表示するワークフロー・タイプの内部名を入力します。内部名の始めの数文字を入力しても、ワークフロー・タイプを検索することができます。
3. 「実行」ボタンを選択して、検索を実行します。
4. 「結果」領域に、検索基準と一致したワークフロー定義が表示されます。ワークフロー定義ごとに、ワークフロー・タイプの表示名、内部名および説明が表示されます。「Workflow タイプ」または「内部名」の列見出しを選択すると、その列を基準にしてリストがソートされます。
5. ワークフロー管理者権限を持っている場合は、ワークフロー定義の「実行」列にあるアイコンを選択して、テスト・ワークフロー・プロセスを実行することができます。7-8 ページの「[テスト・ワークフロー・プロセスの実行](#)」を参照してください。

注意： ワークフロー定義に実行可能なプロセスが含まれていない場合、「実行」アイコンは使用不可になります。

また、ワークフロー・プロセスが「受信」イベント・アクティビティで始まっている場合は、プロセスのテストに開発者スタジオを使用できません。テストを行うには、イベント・マネージャからテスト・イベントを呼び出し、プロセスを起動するイベント・サブスクリプションをトリガーする必要があります。8-84 ページの「[テスト・イベントの呼出し](#)」を参照してください。

▶ テスト・ワークフロー・プロセスの実行

1. 「ワークフローの実行」ページへ移動します。

The screenshot shows the Oracle Workflow 'Run Workflow' page. The browser title is 'Oracle Workflow: Run Workflow - Microsoft Internet Explorer provided by Dev. Services'. The page has a navigation bar with 'Home', 'Developer Studio', 'Business Events', 'Status Monitor', and 'Notifications'. The main content area is titled 'Run Workflow: Requisition' and includes instructions: 'Identify the workflow process you're about to launch, enter any attributes that you need to set and select the "Run Workflow" button to begin processing.' Below this are several sections of input fields:

- Workflow Identifier:** Includes 'Workflow Owner' (with a search icon), 'Item Key', and 'User Key'.
- Process:** Set to 'Requisition Approval'.
- Workflow Attributes:** Includes 'Forward From Username', 'Forward To Username', 'Requestor Username', and 'Requisition Amount' (with a format hint: 'Enter a number in the following format: \$,999,999,999.99').

2. 「ワークフロー識別子」領域で、実行するワークフロー・プロセスの識別情報を指定します。

- ワークフロー所有者： オプションで、プロセスの所有者を指定します。デフォルト値は、自分のユーザー名です。フィールドの検索アイコンを選択すると、選択可能な値のリストが表示されます。ユーザーの表示名、内部名または電子メール・アドレスを基準に検索することができます。8-85 ページの「[値リストの使用](#)」を参照してください。
- 項目キー： プロセス・インスタンスを識別する一意の項目キー。

注意： プロセス・インスタンスの項目キーには、シングルバイト・キャラクターのみ使用できます。マルチバイトの値を含めることはできません。

- ユーザー・キー： オプションの、ユーザーがわかりやすい識別子。ステータス・モニターやその他の Oracle Workflow のユーザー・インタフェース・コンポーネント内でワークフロー・プロセスを検索するのに使用できます。
- プロセス： テストするワークフロー・プロセスの名前。ワークフロー項目タイプに含まれている実行可能なプロセスが 1 つのみの場合は、そのプロセスの名前が自動的に表示されます。ワークフロー項目タイプに複数の実行可能なプロセスが含まれている場合は、テストするプロセスを選択します。

3. 「ワークフロー属性」領域には、対象のワークフロー項目タイプに関連するすべての属性が表示されます。プロセスを開始するのに必要な項目属性の値を入力します。
4. 「ワークフローの実行」ボタンを選択します。ワークフロー・プロセスを開始するために、「ワークフローの実行」ページでは、指定した項目タイプと項目キーに対応するワークフロー・エンジンの `CreateProcess()` と `StartProcess()` API がコールされます。また、ワークフロー・エンジンの `SetItemOwner()` と `SetItemAttr()` API もコールされ、プロセス所有者とすべての項目タイプ属性が指定した値に設定されます。

また、「取消」ボタンを選択し、ワークフロー・プロセスを実行せずに前のページに戻ることもできます。

5. ワークフロー・プロセスのステータスを表示する場合は、ワークフローの検索ページで「ステータス・モニター」タブを選択し、プロセスを検索します。『Oracle Workflow 管理者ガイド』の「管理者モニターへのアクセス」を参照してください。
6. テストするプロセスに通知が含まれている場合は、「通知」タブを選択し、「通知」ワークリスト内の通知に応答します。『Oracle Workflow ユーザーズ・ガイド』の「拡張ワークリストの通知の表示」を参照してください。

または、受信者の通知環境設定に電子メール通知を受信するように設定した場合は、電子メール・ソフトウェアを使用して通知を表示し、応答することができます。通知メーラーにテスト用電子メール・アドレスを指定した場合は、その電子メール・アカウントに接続して、プロセスのすべての通知にアクセスすることができます。『Oracle Workflow ユーザーズ・ガイド』の「電子メールによる通知の閲覧」を参照してください。

ビジネス・イベントの管理

この章では、Oracle Workflow の「イベント・マネージャ」Web ページを使用してビジネス・イベントを管理する方法について説明します。

ビジネス・イベントの管理

Oracle Workflow Business Event System は、Oracle Advanced Queuing (AQ) のインフラストラクチャを利用して、システム間でビジネス・イベントを交換するためのアプリケーション・サービスです。ビジネス・イベント・システムは、イベント・マネージャとワークフロー・プロセス・イベント・アクティビティで構成されます。

イベント・マネージャには、ビジネス・イベント、システム、システム内の名前付き通信エージェント、およびイベントと特定のシステムの関連付けを示すサブスクリプションが登録されています。イベントはローカルで呼び出すか、または AQ を介して外部システムまたはローカル・システムから受信します。ローカル・イベントが発生すると、サブスクリプションが遅延されていないかぎり、イベントを呼び出したコードと同じトランザクション内でサブスクライバのコードが実行されます。

サブスクリプションには、次のタイプの処理を指定できます。

- イベント情報に対するカスタム・コードの実行
- ワークフロー・プロセスへのイベント情報の送信
- 他のキューまたはシステムへのイベント情報の送信

ビジネス・イベントは、ワークフロー・プロセス内ではイベント・アクティビティとして表されます。イベント・アクティビティをワークフロー・プロセスに含めることにより、事前定義済みの関数を直接実行したり、事前定義済みの受信者にイベントを送信するだけでなく、ビジネス・イベントに関する複雑な処理またはルーティング・ロジックをモデル化できます。3-56 ページの「[イベント・アクティビティ](#)」を参照してください。

ビジネス・イベント・システムには、次のような機能があります。

- **システム統合メッセージ・ハブ**： Oracle Workflow とビジネス・イベント・システムを統合すれば、複雑なシステム統合シナリオのメッセージ・ハブとして機能します。イベント・マネージャは、イベントおよび呼び出し元に基づいてシステム間のルーティングを密接に連結するために使用できます。ワークフロー・プロセス・イベント・アクティビティは、より高度なルーティング、コンテンツベースのルーティング、変換、エラー処理などのモデル化に使用できます。
- **分散アプリケーション・メッセージ機能**： アプリケーションは、ビジネス・エンティティに対して Generate および Receive イベント・メッセージ・ハンドラを提供できます。たとえば、メッセージ・ハンドラを使用して、マスター/コピーのレプリケーションを分散アプリケーションに実装できます。
- **メッセージ・ベースのシステム統合**： ビジネス・イベントの発生時に、システム間にメッセージが送信されるようにサブスクリプションを設定できます。このように、イベント・マネージャを使用してポイントツーポイントのメッセージ統合を実現できます。
- **ビジネス・イベント・ベースのワークフロー・プロセス**： ビジネス・イベントの内容に基づいた拡張ルーティングまたはプロセスを組み込んで、高度なワークフロー・プロセスを開発できます。

- **パッケージ・アプリケーションの非侵入性カスタマイズ**：分析者は、インターネットまたはイントラネット・アプリケーションに対して、興味あるビジネス・イベントを登録できます。アプリケーションのユーザーは、それらのイベントへのサブスクリプションを登録して、カスタム・コードまたはワークフロー・プロセスをトリガーできます。

イベント・マネージャ

Oracle Workflow のイベント・マネージャには、アプリケーションで発生する興味あるビジネス・イベント、イベントを交換するシステム、それらのシステム内の名前付き通信エージェント、およびイベントと特定のシステムの関連付けを示すサブスクリプションを登録できます。「イベント・マネージャ」Web ページを使用して、これらのイベント、システム、エージェントおよびサブスクリプションを定義およびメンテナンスできます。

注意：「イベント・マネージャ」Web ページにアクセスするには、ワークフロー管理者権限が必要です。

Workflow XML Loader を使用して、フラット・ファイルとデータベース間でビジネス・イベント・システムオブジェクトの XML 定義をアップロードおよびダウンロードすることができます。『Oracle Workflow 管理者ガイド』の「Workflow XML Loader の使用」を参照してください。

ローカル・アプリケーションからイベントが呼び出されたとき、またはローカル・システムあるいは外部システムからイベントが着信したときは、イベント・マネージャがそのイベントへのサブスクリプションを実行します。イベント・マネージャは、サブスクリプションに定義されているアクションに応じて、カスタム・コードのコール、ワークフロー・プロセスへのイベント情報の送信、またはエージェントへのイベント情報の送信を行います。

ビジネス・イベント・システムの設定が完了したら、イベント・マネージャを使用して、イベントの手動呼出し、ビジネス・イベントを交換するためのシステムのサインアップ、システム間の同期およびローカル・キューの確認を行うことができます。「ワークフロー・エージェントの Ping/ 確認」を使用して、設定をテストできます。

注意：現在、Oracle Applications に組み込まれている Workflow で使用できる Oracle Applications Framework ユーザー・インタフェース形式のイベント・マネージャの詳細については、8-72 ページの「[Oracle Applications のためのイベント・マネージャ](#)」を参照してください。

イベント

ビジネス・イベントは、システム内の他のオブジェクトまたは外部エージェントに関連付けられた、インターネットまたはイントラネット内の状態変化です。たとえば、発注書の作成は、購買アプリケーションにおけるビジネス・イベントの例です。イベントの関連付けは、イベント・マネージャで定義できます。

Oracle Workflow には、ビジネス・イベント・システム内で関連付けられた状態変化に対して、いくつかの事前定義済イベントが用意されています。9-2 ページの「[事前定義済ワークフロー・イベント](#)」を参照してください。

ローカル・システム上のアプリケーションでイベントが発生する場合は、イベントの特定のインスタンスを一意に識別するイベント・キーを割り当ててから、イベント・マネージャに送信する必要があります。

イベントは次の方法で呼び出すことができます。

- イベントを呼び出すアプリケーションから `WF_EVENT.Raise()` API を使用して、イベントを呼び出します。『Oracle Workflow API リファレンス』の「[Raise](#)」を参照してください。
- ワークフロー・プロセスから `Raise` イベント・アクティビティを使用して、イベントを呼び出します。3-56 ページの「[イベント・アクティビティ](#)」を参照してください。
- 「[イベントの呼び出し](#)」ページから、イベントを手動で呼び出します。8-60 ページの「[イベントの呼出し](#)」を参照してください。

また、イベント・マネージャでは、ローカル・システムまたはリモート・システムから送信されたイベントを受信できます。

イベント・マネージャにイベントを定義する際に、一意の内部名をイベントに割り当てる必要があります。この名前は、大文字 / 小文字が区別されます。イベントの内部名を記述するときは、次のように複数の識別子をピリオドで区切った書式を使用することをお勧めします。

```
<company>.<family>.<product>.<component>.<object>.<event>
```

この書式では、定義するイベントを分類階層に体系化することができます。

また、イベント・グループを定義して、イベントを相互に関連付け、それらのイベントをイベント・サブスクリプションのグループとして参照することもできます。イベント・グループはイベントの一種で、個別のメンバー・イベントの集合で構成されます。イベント・グループの内部名には、個別のイベント名と同じ書式を使用する必要があります。イベント・グループを定義すると、グループに対してサブスクリプションを登録できます。グループ内のイベントに対して個別にサブスクリプションを作成する必要はありません。サブスクリプションは、グループのいずれかのメンバー・イベントが発生したときに実行されます。

注意： イベント・グループを使用して、イベントを呼び出すことはできません。各イベントは個別に呼び出す必要があります。

イベントの内容を説明するために必要な詳細情報は、イベント名およびイベント・キーとともに、イベント・データと呼ばれます。たとえば、発注イベントのイベント・データには、項目番号、説明およびコストなどが含まれます。イベント・データは XML 文書として作成できます。

イベントが発生するアプリケーションには、イベントをイベント・マネージャに呼び出す際に、イベント・データを組み込むことができます。アプリケーションからイベント・データが渡されない場合は、イベントにジェネレート関数を指定して、イベント名、イベント・キーおよびパラメータ・リスト（オプション）から完全なイベント・データを生成する必要があります。ジェネレート関数は、標準 API に従ってください。『Oracle Workflow API リファレンス』の「Raise」および 6-24 ページの「[イベント・データ・ジェネレート関数の標準 API](#)」を参照してください。

イベント・マネージャはサブスクリプションを実行する前に、各サブスクリプションにイベント・データが必要かどうかをチェックします。必要なイベント・データが渡されていない場合、イベント・マネージャはそのイベントに対してジェネレート関数をコールし、イベント・データを生成します。イベント・データを必要とするイベントにジェネレート関数が定義されていない場合、イベント名およびイベント・キーを使用してデフォルトのイベント・データが作成されます。

注意： ジェネレート関数の負荷が大きい場合、イベントの呼び出し後に制御をできるだけ早くコール側アプリケーションに戻したい場合は、完全なイベント・データを必要とするすべてのサブスクリプションを遅延させることができます。後でそれらのサブスクリプションを実行するまで、イベント・マネージャはジェネレート関数を実行しません。8-49 ページの「[遅延サブスクリプション処理](#)」を参照してください。

イベント定義を自動的に作成するプログラムを使用する場合は、プログラム独自の名前と簡単な識別子をイベントの所有者名および所有者タグとして設定できます。この識別情報を使用して、所有するイベントをプログラムから検索することができます。所有者名および所有者タグは、「イベントの編集」および「グループの編集」ページから必要に応じて手動で更新できます。

各イベントには、イベント定義を更新できるかどうかを判断するためのカスタマイズ・レベルが割り当てられています。このカスタマイズ・レベルは、Oracle Applications のシード・データを保護し、アップグレード中にユーザーのカスタマイズを保存するために使用されます。各イベントには、次のカスタマイズ・レベルの 1 つが割り当てられています。

- **コア：** イベント定義を変更することはできません。このレベルは、Oracle Applications にシードされたイベントに対してのみ使用されます。
- **制限：** イベントのステータスを「使用可能」または「使用不能」に更新できますが、イベント定義を変更することはできません。このレベルは、Oracle Applications にシードされたイベントに対してのみ使用されます。

- ユーザー： イベント定義内のすべてのプロパティを変更することができます。このレベルは、ユーザーが定義したイベントに自動的に設定されます。

▶ イベントの定義

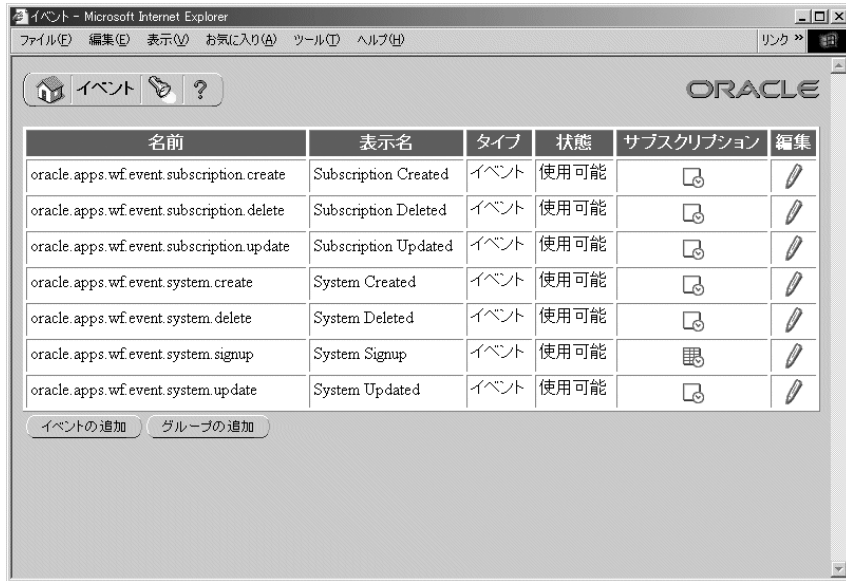
1. Web ブラウザを使用して、次の URL に接続します。

```
<webagent>/wf_event_html.listevents
```

<webagent> は、Web サーバーで Oracle Workflow 用に構成された Web エージェントのベース URL に置き換えてください。『Oracle Workflow 管理者ガイド』の「グローバル・ユーザー設定の設定」を参照してください。

注意： この URL はセキュリティ画面にアクセスするため、現行 Web セッションで有効なユーザーとしてログオンしていない場合は、ページが表示される前に有効なユーザーとしてのログオンを求めるプロンプトが表示されます。「イベント・マネージャ」Web ページにアクセスするには、ワークフロー管理者権限が必要です。

注意： 「イベント」Web ページには、Oracle Workflow ホーム・ページからもアクセスできます。『Oracle Workflow 管理者ガイド』の「Oracle Workflow ホーム・ページへのアクセス」を参照してください。



- 「イベント」ページに、既存のイベントのリストが表示されます。「イベント」ページには、各イベントの内部名、表示名、タイプおよびステータスの一覧が表示されます。「イベントの追加」ボタンを選択して「イベントの編集」ページを開きます。

注意： カスタマイズ・レベルが「制限」のシードされたイベントの場合は、イベントのステータスのみを更新できます。カスタマイズ・レベルが「コア」のシードされたイベントの場合、イベント定義は表示できますが、プロパティは更新できません。

The screenshot shows a web browser window titled 'イベントの編集 - Microsoft Internet Explorer'. The page header includes 'イベントの編集' and the Oracle logo. The form contains the following fields and controls:

- 名前 (Name): Text input field.
- 表示名 (Display Name): Text input field.
- 説明 (Description): Text area with a vertical scrollbar.
- 状態 (Status): Dropdown menu with '使用可能' (Available) selected.
- 機能の生成 (Function Generation): Text input field.
- 所有者名 (Owner Name): Text input field.
- 所有者タグ (Owner Tag): Text input field.

At the bottom of the form, there are two buttons: '送信' (Submit) and '取消' (Cancel).

3. イベントの内部名を「名前」フィールドに入力します。すべての Oracle Workflow API、SQL スクリプトおよび PL/SQL プロシージャでは、イベントの識別時に内部名が参照されます。内部名は大文字 / 小文字が区別されます。次のように、複数の識別子をピリオドで区切った書式で記述することをお勧めします。

`<company>.<family>.<product>.<component>.<object>.<event>`

4. イベントの表示名を入力します。この名前は、「イベント」リストに表示されます。
5. オプションで、イベントの説明を入力します。
6. 「ステータス」フィールドで、イベントのステータスとして「使用可能」または「使用不能」を選択します。イベントを使用不能にしても、そのイベントは「イベント」リストに表示され、参照できます。ただし、有効なサブスクリプションでは使用できません。
7. ローカル・システムで発生するイベントを定義する場合は、イベントのジェネレート関数を入力します。ジェネレート関数は、イベント名、イベント・キーおよびパラメータ・リスト（オプション）から完全なイベント・データを生成する PL/SQL プロシージャです。6-24 ページの「イベント・データ・ジェネレート関数の標準 API」を参照してください。
8. オプションで、イベントを所有するプログラムまたはアプリケーションを識別することもできます。プログラム名を「所有者名」フィールドに、プログラム ID を「所有者タグ」フィールドに入力してください。「イベントの編集」ページでイベントを手動で定義する場合は、「所有者名」および「所有者タグ」を入力する必要はありません。ただし、イベント定義を自動的に作成するプログラムを使用する場合、これらのフィールド

にはそのプログラムで設定された所有者情報が表示されます。この情報は、「イベントの編集」ページから必要に応じて手動で更新できます。

9. イベントのカスタマイズ・レベルを確認することができます。
 - コア： イベント定義を変更することはできません。このレベルは、Oracle Applications にシードされたイベントに対してのみ使用されます。
 - 制限： イベント・ステータスを「使用可能」または「使用不能」に更新できますが、イベント定義を変更することはできません。このレベルは、Oracle Applications にシードされたイベントに対してのみ使用されます。
 - ユーザー： イベント定義内のすべてのプロパティを変更することができます。このレベルは、ユーザーが定義したイベントに自動的に設定されます。
10. 「送信」ボタンを選択してイベントを保存し、「イベント」ページに戻ります。「イベント」ページには、更新されたイベントのリストが表示されます。

「取消」ボタンを選択し、イベントを保存しないで「イベント」ページに戻ることもできます。

▶ **イベント・グループの定義**

1. Web ブラウザを使用して、次の URL に接続します。

```
<webagent>/wf_event_html.listevents
```

<webagent> は、Web サーバーで Oracle Workflow 用に構成された Web エージェントのベース URL に置き換えてください。『Oracle Workflow 管理者ガイド』の「グローバル・ユーザー設定の設定」を参照してください。

注意： この URL はセキュリティ画面にアクセスするため、現行 Web セッションで有効なユーザーとしてログオンしていない場合は、ページが表示される前に有効なユーザーとしてのログオンを求めるプロンプトが表示されます。「イベント・マネージャ」Web ページにアクセスするには、ワークフロー管理者権限が必要です。

注意： 「イベント」Web ページには、Oracle Workflow ホーム・ページからもアクセスできます。『Oracle Workflow 管理者ガイド』の「Oracle Workflow ホーム・ページへのアクセス」を参照してください。

2. 「イベント」ページに、既存のイベントのリストが表示されます。「グループの追加」ボタンを選択して「イベントの編集」ページを開きます。

注意： カスタマイズ・レベルが「制限」のシードされたイベント・グループの場合は、イベント・グループのステータスのみを更新できます。カスタマイズ・レベルが「コア」のシードされたイベントの場合、イベント・グループ定義は表示できますが、プロパティは更新できません。

グループの編集 - Microsoft Internet Explorer

ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H) リンク 印刷

グループの編集 ?

ORACLE

名前

表示名

説明

状態

所有者名

所有者タグ

3. イベント・グループの内部名を「名前」フィールドに入力します。すべての Oracle Workflow API、SQL スクリプトおよび PL/SQL プロシージャでは、イベント・グループの識別時に内部名が参照されます。内部名は大文字 / 小文字が区別されます。内部名を記述するときは、次のように複数の識別子をピリオドで区切った書式を使用してください。
`<company>.<family>.<product>.<component>.<object>.<event>`
4. イベント・グループの表示名を入力します。この名前は、「イベント」リストに表示されます。
5. オプションで、イベント・グループの説明を入力します。
6. 「ステータス」フィールドで、イベント・グループのステータスとして「使用可能」または「使用不能」を選択します。イベント・グループを使用不能にしても、そのイベント・グループは「イベント」リストに表示され、参照できます。ただし、有効なサブスクリプションでは使用できません。
7. オプションで、イベント・グループを所有するプログラムまたはアプリケーションを識別することもできます。プログラム名を「所有者名」フィールドに、プログラム ID を「所有者タグ」フィールドに入力してください。「グループの編集」ページでイベント・

グループを手動で定義する場合は、「所有者名」および「所有者タグ」を入力する必要はありません。ただし、イベント・グループ定義を自動的に作成するプログラムを使用する場合、イベント・マネージャのこれらのフィールドにはそのプログラムで設定された所有者情報が表示されます。この情報は、「グループの編集」ページから必要に応じて手動で更新できます。

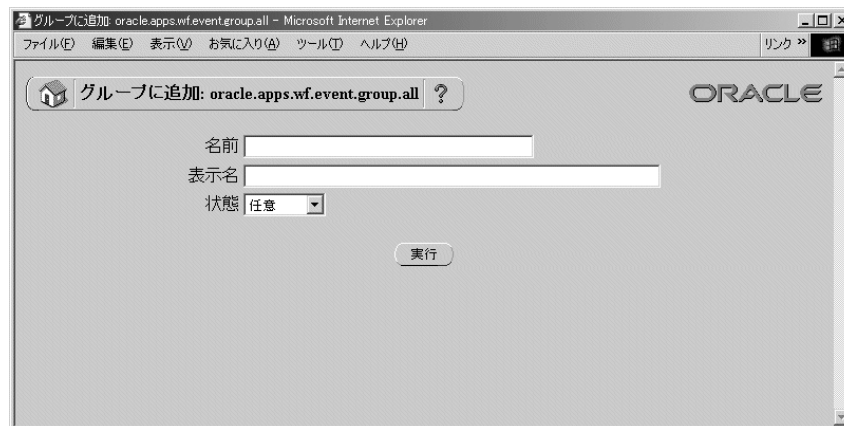
8. イベント・グループのカスタマイズ・レベルを確認することができます。
 - コア： イベント・グループ定義を変更することはできません。このレベルは、Oracle Applications にシードされたイベント・グループに対してのみ使用されます。
 - 制限： イベント・グループ・ステータスを「使用可能」または「使用不能」に更新できますが、イベント・グループ定義を変更することはできません。このレベルは、Oracle Applications にシードされたイベント・グループに対してのみ使用されます。
 - ユーザー： イベント・グループ定義内のすべてのプロパティを変更することができます。このレベルは、ユーザーが定義したイベント・グループに自動的に設定されます。
9. 「送信」ボタンを選択してイベント・グループを保存します。

注意：「取消」ボタンを選択し、イベント・グループを保存しないで「イベント」ページに戻ることもできます。

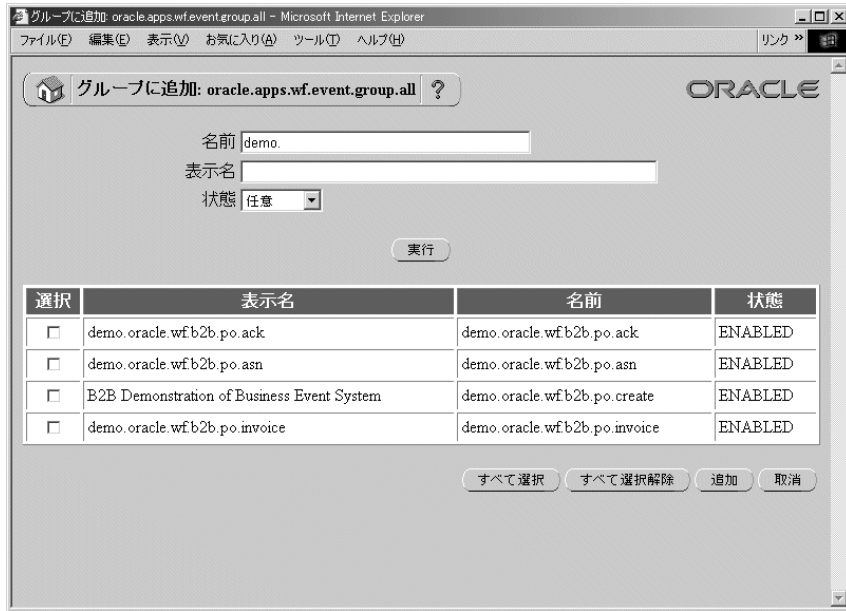
イベント・グループ定義を保存すると、「グループの編集」ページにそのグループのメンバー・イベントのリストが表示され、各イベントの名前、表示名およびステータスが示されます。

10. メンバー・イベントをグループに追加するには、「イベントの追加」ボタンを選択します。

注意： イベント・グループには、個別のイベントのみを登録できます。イベント・グループには、別のグループを登録できません。



11. 表示された「グループに追加」ページで、検索基準を入力し、追加するイベントを検索します。検索基準として、次の項目を使用できます。
 - 名前： 追加するイベントの内部名を入力します。
 - 表示名： 追加するイベントの表示名を入力します。
 - ステータス： 追加するイベントのステータスとして「使用可能」または「使用不能」を選択します。すべてのステータスからイベントを検索するには、「任意」を選択します。
12. 「実行」を選択します。「グループに追加」ページに、検索基準と一致したイベントのリストが表示されます。




13. イベント・グループに追加するイベントを選択します。リスト内のイベントをすべて選択するときは、「すべて選択」ボタンを選択します。リスト内のイベントをすべて選択解除するときは、「すべて選択解除」ボタンを選択します。

別のイベントを検索する場合は、新しい検索基準を入力して「実行」ボタンを選択します。「グループに追加」ページに、新しい検索基準と一致したイベントのリストが表示されます。

「取消」ボタンを選択し、現行の検索を取り消して前の検索結果に戻することもできます。

14. 追加するイベントの選択が終了したら、「追加」ボタンを選択して、選択したイベントをイベント・グループに追加します。「グループの編集」ページに、更新されたイベント・グループ・メンバーのリストが表示されます。



名前 oracle.group.new
 表示名 New Group
 説明
 状態 使用可能
 所有者名
 所有者タグ

グループ内のイベント

選択	名前	表示名	状態	サブスクリプション	編集
<input type="checkbox"/>	oracle.apps.wf.event.all.sync	Synchronize Event Systems	ENABLED	⏏	✎
<input type="checkbox"/>	oracle.apps.wf.event.event.create	Event Created	ENABLED	⏏	✎
<input type="checkbox"/>	oracle.apps.wf.event.event.delete	Event Deleted	ENABLED	⏏	✎
<input type="checkbox"/>	oracle.apps.wf.event.event.update	Event Updated	ENABLED	⏏	✎
<input type="checkbox"/>	oracle.apps.wf.event.group.create	Event Group Creation	ENABLED	⏏	✎
<input type="checkbox"/>	oracle.apps.wf.event.group.delete	Event Group Deleted	ENABLED	⏏	✎

すべて選択 すべて選択解除 削除 イベントの追加
 送信 取消

15. 「送信」 ボタンを選択してイベント・グループ定義を保存します。

注意：「取消」 ボタンを選択して、最新の検索結果が表示された「グループに追加」 ページに戻ることもできます。

16. グループからメンバー・イベントを削除する場合は、「グループの編集」 ページで、削除するイベントを選択します。リスト内のイベントをすべて選択するときは、「すべて選択」を選択します。リスト内のイベントをすべて選択解除するときは、「すべて選択解除」を選択します。
17. 「削除」 ボタンを選択して、選択したイベントをイベント・グループから削除します。「グループの編集」 ページに、更新されたイベント・グループ・メンバーのリストが表示されます。

注意：メンバー・イベントをイベント・グループから個別に削除しても、各イベントのイベント定義は削除されません。各イベントは「イベント」リストに残ります。

18. イベントを参照するサブスクリプションを表示するには、そのイベントの「サブスクリプション」列にあるスケジュール・アイコンを選択します。「イベント・サブスクリプション」ページに、イベントのサブスクリプションのリストが表示されます。

注意： イベントにサブスクリプションを指定していない場合は、空のスケジュール・アイコンが表示されます。サブスクリプションから参照されているイベントの場合は、スケジュール・アイコンが正式に表示されません。

「イベント・サブスクリプション」ページの「サブスクリプションの追加」ボタンを選択して、イベントに対して新しいサブスクリプションの定義を開始できます。「サブスクリプションの編集」ページが表示され、イベント名が「イベント・フィルタ」フィールドに自動的に入力されます。8-52 ページの「[イベント・サブスクリプションの定義](#)」を参照してください。

19. イベントを更新するには、そのイベントの「編集」列にある鉛筆アイコンを選択します。「イベントの編集」ページが表示されます。イベント定義に変更を加え、変更内容を保存します。8-7 ページの「[イベントの定義](#)」を参照してください。

▶ イベントの検索

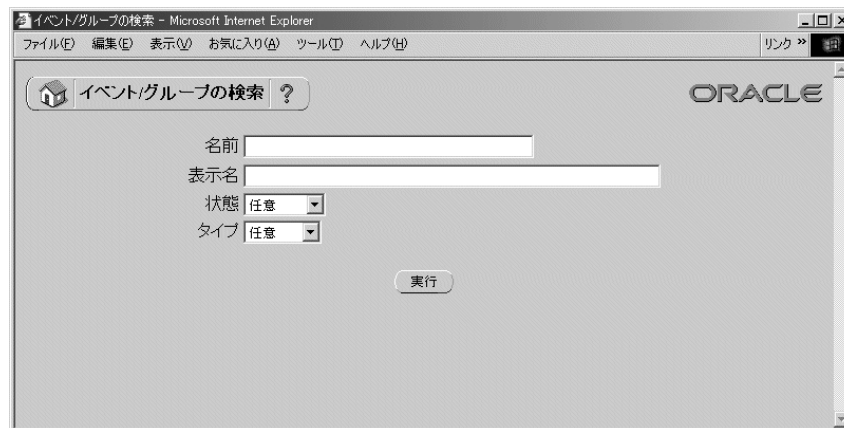
1. Web ブラウザを使用して、次の URL に接続します。

```
<webagent>/wf_event_html.findevent
```

<webagent> は、Web サーバーで Oracle Workflow 用に構成された Web エージェントのベース URL に置き換えてください。『Oracle Workflow 管理者ガイド』の「グローバル・ユーザー設定の設定」を参照してください。

注意： この URL はセキュリティが適用されるページにアクセスするため、現行 Web セッションで有効なユーザーとしてログオンしていない場合は、ページが表示される前に有効なユーザーとしてのログオンを求めるプロンプトが表示されます。「イベント・マネージャ」Web ページにアクセスするには、ワークフロー管理者権限が必要です。

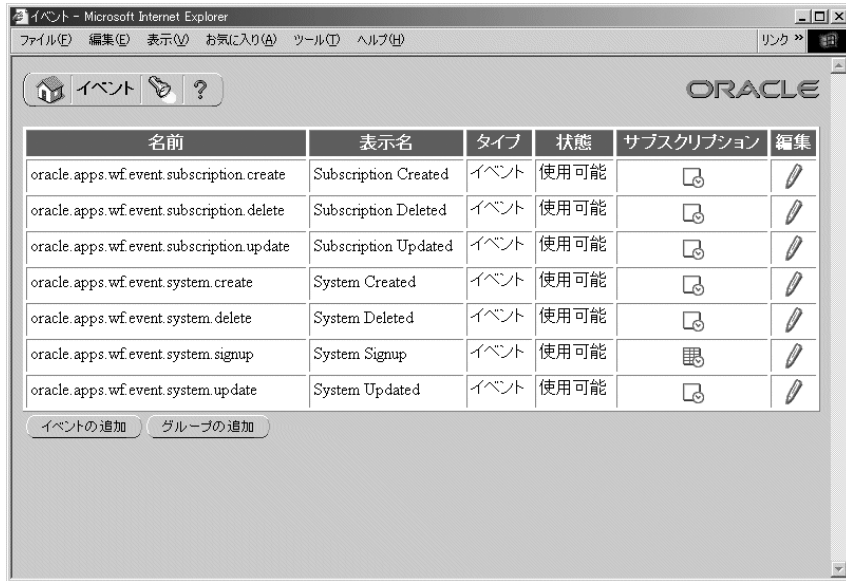
注意： 「イベント / グループの検索」Web ページには、Oracle Workflow ホーム・ページからもアクセスできます。『Oracle Workflow 管理者ガイド』の「Oracle Workflow ホーム・ページへのアクセス」を参照してください。



2. 「イベント / グループの検索」ページが表示されます。「イベント / グループの検索」ページでは、検索基準を入力して特定のイベントを検索できます。検索基準として、次の項目を使用できます。
 - 名前： 表示するイベントの内部名を入力します。
 - 表示名： 表示するイベントの表示名を入力します。
 - ステータス： 表示するイベントのステータスとして「使用可能」または「使用不能」を選択するか、「任意」を選択してすべてのステータスのイベントを表示します。
 - タイプ： 表示するイベントのタイプとして「イベント」または「グループ」を選択するか、「任意」を選択してすべてのタイプのイベントを表示します。
3. 「実行」を選択します。「イベント」ページに、検索基準と一致したイベントのリストが表示されます。

▶ **イベントの更新または削除**

1. 「イベント」ページで、更新または削除するイベントを選択します。「イベント / グループの検索」ページを使用して、必要なイベントを検索し、「イベント」ページに表示することができます。8-16 ページの「[イベントの検索](#)」を参照してください。



2. イベントを参照するサブスクリプションを表示するには、そのイベントの「サブスクリプション」列にあるスケジュール・アイコンを選択します。「イベント・サブスクリプション」ページに、イベントのサブスクリプションのリストが表示されます。

注意： イベントにサブスクリプションを指定していない場合は、空のスケジュール・アイコンが表示されます。サブスクリプションから参照されているイベントの場合は、スケジュール・アイコンが正式に表示されます。

「イベント・サブスクリプション」ページの「サブスクリプションの追加」ボタンを選択して、イベントに対して新しいサブスクリプションの定義を開始できます。「サブスクリプションの編集」ページが表示され、イベント名が「イベント・フィルタ」フィールドに自動的に入力されます。8-52 ページの「[イベント・サブスクリプションの定義](#)」を参照してください。

3. イベントを更新するには、そのイベントの「編集」列にある鉛筆アイコンを選択します。「イベントの編集」ページが表示されます。イベント定義に変更を加え、変更内容を保存します。8-7 ページの「[イベントの定義](#)」を参照してください。

注意： カスタマイズ・レベルが「制限」のシードされたイベントの場合は、イベントのステータスのみを更新できます。カスタマイズ・レベルが「コア」のシードされたイベントの場合、イベント定義は表示できますが、プロパティは更新できません。

4. イベントを削除するには、そのイベントの「削除」列にあるごみ箱アイコンを選択し、表示された確認ウィンドウで「OK」を選択します。確認ウィンドウで「取消」ボタンを選択し、イベントを削除しないで「イベント」ページに戻ることもできます。

注意： 削除できるイベントは、サブスクリプションから参照されていないイベントおよびイベント・グループに属していないイベントのみです。

また、カスタマイズ・レベルが「コア」または「制限」の Oracle Applications にシードされたイベントは、削除することができません。

システム

システムとは、ホスト・マシンやデータベース・インスタンスなど、論理的に独立したソフトウェア環境のことです。イベント・マネージャ内でイベントを交換するシステムを定義する必要があります。

システムを定義する際に、イベント・マネージャのオブジェクト定義の更新の送信元となるマスター・システムに関連付けるかどうかを指定できます。

各システムでは、アドレス指定可能な通信ポイント（エージェント）を1つ以上公開できます。システムを定義したら、ビジネス・イベントの交換に使用するエージェントをシステム内に定義する必要があります。8-25 ページの「エージェント」を参照してください。

ローカル・システム

Oracle Workflow をデータベース内にインストールすると、そのデータベースはイベント・マネージャのシステムとして自動的に定義され、「グローバル・ワークフロー設定」ページのローカル・システムとして設定されます。次の表は、ローカル・システム定義のデフォルト・プロパティの一覧です。

表 8-1

システム・プロパティ	値
名前	<database global name>
表示名	<database global name>
説明	Oracle Workflow Configuration Assistant によって作成されたローカル・システム
マスター	(空白)

ローカル・システム定義は、必要に応じて更新できます。

ローカル・システムのステータスは、デフォルトでは「使用可能」に設定されます。ビジネス・イベント・システムの設定が終了したら、「グローバル・ワークフロー設定」ページを使用して、イベントの処理に必要なシステム・ステータスを設定できます。『Oracle Workflow 管理者ガイド』の「グローバル・ユーザー設定の設定」を参照してください。

▶ システムの定義

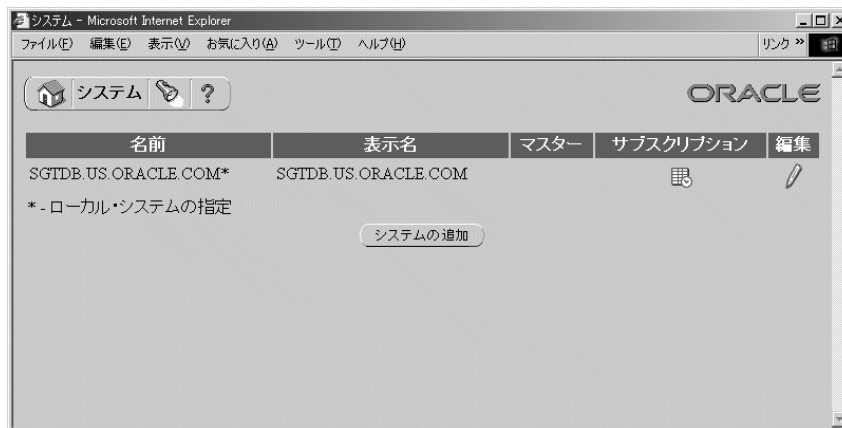
1. Web ブラウザを使用して、次の URL に接続します。

```
<webagent>/wf_event_html.listsystems
```

<webagent> は、Web サーバーで Oracle Workflow 用に構成された Web エージェントのベース URL に置き換えてください。『Oracle Workflow 管理者ガイド』の「グローバル・ユーザー設定の設定」を参照してください。

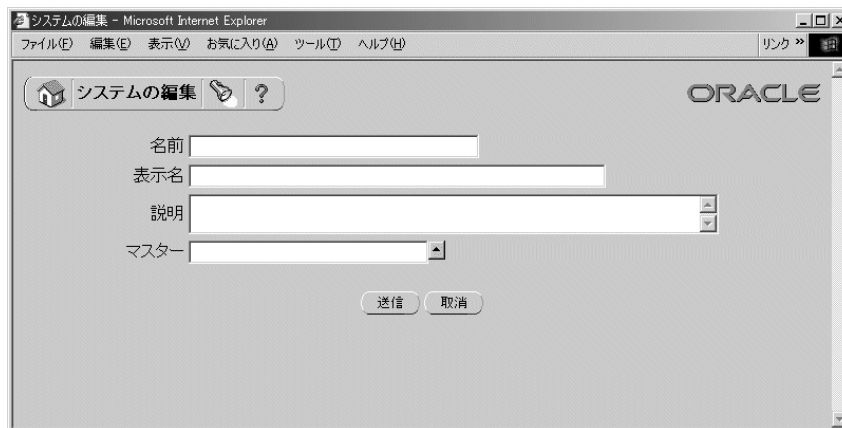
注意： この URL はセキュリティが適用されるページにアクセスするため、現行 Web セッションで有効なユーザーとしてログオンしていない場合は、ページが表示される前に有効なユーザーとしてのログオンを求めるプロンプトが表示されます。「イベント・マネージャ」Web ページにアクセスするには、ワークフロー管理者権限が必要です。

注意： 「システム」Web ページには、Oracle Workflow ホーム・ページからもアクセスできます。『Oracle Workflow 管理者ガイド』の「Oracle Workflow ホーム・ページへのアクセス」を参照してください。



- 「システム」ページに、既存のシステムのリストが表示されます。「システム」ページには、各システムの内部名、表示名およびマスター・システムの一覧が表示されます。アスタリスク (*) はローカル・システムを示します。

「システムの追加」ボタンを選択して「システムの編集」ページを開きます。



- システムの内部名を「名前」フィールドに入力します。すべての Oracle Workflow API、SQL スクリプトおよび PL/SQL プロシージャでは、システムの識別時に内部名が参照されます。

注意： 内部名には大文字のみを使用する必要があります。一重引用符 (')、二重引用符 (") またはスペースを含めないでください。

4. システムの表示名を入力します。この名前は、「システム」リストに表示されます。
5. オプションで、システムの説明を入力します。
6. オプションで、イベント・マネージャのオブジェクト定義の更新の送信元となる、マスター・システムを入力します。「マスター」フィールドの上矢印のアイコンをクリックすると、選択可能なシステムのリストが表示されます。8-25 ページの「[値リストの使用](#)」を参照してください。
7. 「送信」ボタンを選択してシステムを保存し、「システム」ページに戻ります。「システム」ページに、更新されたシステムのリストが表示されます。

「取消」ボタンを選択し、システムを保存しないで「システム」ページに戻ることもできます。

▶ **システムの検索**

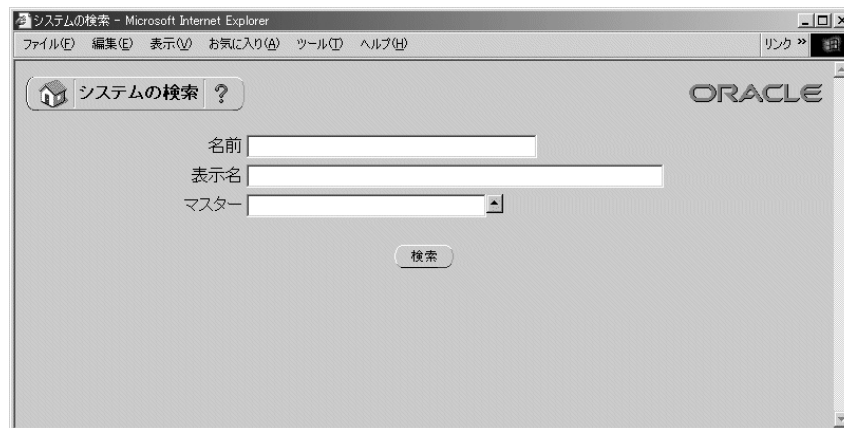
1. Web ブラウザを使用して、次の URL に接続します。

```
<webagent>/wf_event_html.findsystem
```

<webagent> は、Web サーバーで Oracle Workflow 用に構成された Web エージェントのベース URL に置き換えてください。『Oracle Workflow 管理者ガイド』の「グローバル・ユーザー設定の設定」を参照してください。

注意： この URL はセキュリティが適用されるページにアクセスするため、現行 Web セッションで有効なユーザーとしてログオンしていない場合は、ページが表示される前に有効なユーザーとしてのログオンを求めるプロンプトが表示されます。「イベント・マネージャ」Web ページにアクセスするには、ワークフロー管理者権限が必要です。

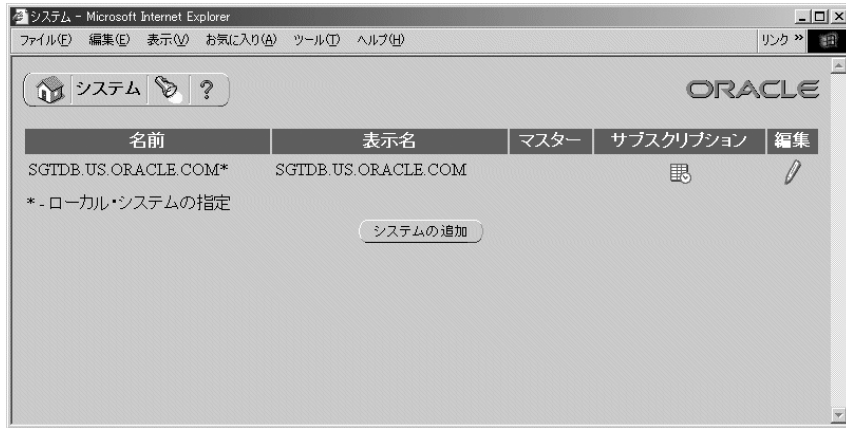
注意： 「システムの検索」Web ページには、Oracle Workflow ホーム・ページからもアクセスできます。『Oracle Workflow 管理者ガイド』の「Oracle Workflow ホーム・ページへのアクセス」を参照してください。



2. 「システムの検索」ページが表示されます。「システムの検索」ページでは、検索基準を入力して特定のシステムを検索できます。検索基準として、次の項目を使用できます。
 - 名前： 表示するシステムの内部名を入力します。
 - 表示名： 表示するシステムの表示名を入力します。
 - マスター： 表示するシステムのマスター・システムを入力します。このフィールドの上矢印のアイコンをクリックすると、選択可能なシステムのリストが表示されます。8-25 ページの「[値リストの使用](#)」を参照してください。
3. 「検索」ボタンを選択します。「システム」ページに、検索基準と一致したシステムのリストが表示されます。アスタリスク (*) はローカル・システムを示します。

▶ システムの更新または削除

1. 「システム」ページで、更新または削除するシステムを選択します。「システムの検索」ページを使用して、必要なシステムを検索し、「システム」ページに表示することができます。8-22 ページの「[システムの検索](#)」を参照してください。



2. 特定のシステムのサブスクリプションを表示するには、そのシステムの「サブスクリプション」列にあるスケジュール・アイコンを選択します。「イベント・サブスクリプション」ページに、サブスクリプションのリストがシステム単位に表示されます。

注意： システムにサブスクリプションを指定していない場合は、空のスケジュール・アイコンが表示されます。サブスクリプションが指定されているイベントの場合は、スケジュール・アイコンが正式に表示されます。

「イベント・サブスクリプション」ページの「サブスクリプションの追加」ボタンを選択して、システムに対して新しいサブスクリプションの定義を開始できます。「サブスクリプションの編集」ページが表示され、システム名が「システム」フィールドに自動的に入力されます。8-52 ページの「[イベント・サブスクリプションの定義](#)」を参照してください。

3. システムを更新するには、そのシステムの「編集」列にある鉛筆アイコンを選択します。「システムの編集」ページが表示されます。システム定義に変更を加え、変更内容を保存します。8-20 ページの「[システムの定義](#)」を参照してください。
4. システムを削除するには、そのシステムの「削除」列にあるごみ箱アイコンを選択して、表示される確認ウィンドウで「OK」を選択します。確認ウィンドウで「取消」ボタンを選択し、システムを削除しないで「システム」ページに戻ることもできます。

注意： 削除できるシステムは、エージェントが定義されていないシステムまたはサブスクリプションから参照されていないシステムのみです。

▶ 値リストの使用

1. 値リストをサポートしているフィールドには、フィールドの上矢印のアイコンをクリックすると、値リスト・ウィンドウが表示されます。
2. 「検索」フィールドに検索基準を入力し、「検索」ボタンを選択して、基準と一致する値を検索します。「消去」ボタンを選択して、「検索」フィールドの内容を消去することもできます。検索基準を指定せずに「検索」のみを選択すると、完全な値リストが表示されます。
3. リストの値をクリックして選択し、値リスト・ウィンドウをクローズします。選択した値が元のフィールドに挿入されます。

エージェント

エージェントは、システム内の名前付き通信ポイントです。システム内およびシステム間の通信は、エージェント間のメッセージの送信によって実行されます。1つのシステムに対して複数のエージェントを設定し、それぞれ異なる通信代替を指定できます。たとえば、1つのシステムに対して複数のエージェントを定義し、インバウンド通信とアウトバウンド通信、異なるプロトコルによる通信、異なる伝播頻度など、様々な通信代替に対応しています。

イベント・マネージャ内でイベントの交換に使用するエージェントを定義する必要があります。各エージェントの名前は、システム内で一意である必要があります。エージェントは、次のような書式の複合名を使用して、Oracle Workflow 内のコードから参照できます。

```
<agent_name>@<system_name>
```

たとえば、システム HUB 内のエージェント WF_IN は、WF_IN@HUB として参照できます。

ローカル・システムにエージェントを定義したら、ローカル・インバウンド・エージェントのリスナーおよびローカル・アウトバウンド・エージェントの伝播をスケジュールして、エージェントをイベント・メッセージの伝播用に設定する必要があります。『Oracle Workflow 管理者ガイド』の「ローカル・インバウンド・エージェントのリスナーのスケジュール」および「ローカル・アウトバウンド・エージェントの伝播のスケジュール」を参照してください。

エージェントへの方向の割当て

イベント・マネージャでエージェントを定義する際に、エージェントがローカル・システム内でサポートする通信方向を指定する必要があります。

- 受信： システムへのインバウンド通信。エージェントはメッセージを特定のプロトコルで受信して、標準の書式でシステムに渡します。
- 送信： システムからのアウトバウンド通信。エージェントはシステムから標準の書式でメッセージを受け取り、指定されたプロトコルを使用して送信します。

エージェントへのプロトコルの割当て

各エージェントは、メッセージの通信に使用するプロトコルに関連付ける必要があります。プロトコルには、メッセージのエンコードおよび送信方法が指定されています。メッセージを正しく通信するには、エージェントの送受信で同じプロトコルを使用する必要があります。

プロトコルは、SQLNET などのネットワーク標準を表します。また、ビジネス間標準も表します。つまり、システム間の上位レベルのメッセージ書式およびハンドシェイク手順を定義します。

ビジネス・イベント・システムは、AQ キューを介してエージェントと対話します。AQ を使用すれば、AQ でサポートされている SQLNET プロトコルを介してメッセージを伝播できます。Oracle9i Database 以降の AQ には、インターネット・アクセス機能も組み込まれています。AQ の Internet Data Access Presentation (IDAP) を使用してメッセージを作成し、それらの IDAP メッセージを HTTP や HTTPS などのトランスポート・プロトコルを使用してインターネットを介して送信すれば、AQ 操作をインターネットを介して行うことができます。また、Oracle9i Database 以降では、AQ のメッセージ・ゲートウェイ機能を使用し、Oracle 以外のメッセージ・システムを使用するアプリケーションと AQ との間で通信することができ、サードパーティのメッセージ・ソリューションとの統合が可能になります。さらに、他のサービスでサポートされているプロトコルを使用して、メッセージを伝播することもできます。次の表は、様々なプロトコルに使用できるサービスを、データベースのバージョン別に示しています。

表 8-2

データベースのバージョン	SQLNET プロトコル	HTTP/HTTPS プロトコル	サードパーティーのメッセージ・ソリューションとの統合
Oracle8i Database	Oracle Advanced Queuing	Oracle Message Broker	Oracle Message Broker
Oracle9i Database 以降	Oracle Advanced Queuing	Oracle Advanced Queuing	Oracle Advanced Queuing メッセージ・ゲートウェイ機能

カスタム・プロトコルを実装するには、次の手順を実行する必要があります。

1. 保留中のインバウンドおよびアウトバウンド・メッセージを保持するように、AQ キューを定義します。
2. AQ キューとの間でメッセージを伝播するコードを指定します。
3. 「イベント・プロトコル・タイプ (WF_AQ_PROTOCOLS)」 選択枝タイプの新しいプロトコルに対して、選択枝コードを定義します。この選択枝コードは、「標準」項目タイプに格納されます。イベント・マネージャでは、エージェントのプロトコルを検証するときに、「イベント・プロトコル・タイプ」 選択枝タイプが使用されます。3-22 ページの「[選択枝タイプの選択枝コードの作成](#)」を参照してください。
4. 新しいプロトコルに基づいてエージェントを定義します。8-34 ページの「[エージェントの定義](#)」を参照してください。

エージェントがインバウンド通信を使用する場合は、システムがエージェントとの通信に使用するアドレスを指定する必要があります。アドレスの書式は、エージェントのプロトコルによって異なります。SQLNET プロトコルを使用するエージェントの場合、AQ 伝播を有効にするには、次の書式でアドレスを指定する必要があります。

```
<schema>.<queue>@<database link>
```

この書式の `<schema>` はキューを所有するスキーマ、`<queue>` はキューの名前、`<database link>` はキューが配置されているインスタンスへのデータベース・リンクの名前を表します。

注意： データベース・リンク名には、データベース・リンクの作成時に指定された名前を正確に入力する必要があります。たとえば、データベース・リンク名が `ORA816.US.ORACLE.COM` の場合は、データベース上のエージェントのアドレスに対してその名前を正確に入力する必要があります。ORA816 のように名前を省略することはできません。

ビジネス・イベント・システムで使用するデータベース・リンクの名前は、ドメイン名を含む完全修飾名である必要があります。データベース・リンクの名前を確認するには、次の構文を使用します。

```
SELECT db_link FROM all_db_links
```

『Oracle Workflow 管理者ガイド』の「[データベース・リンクの作成](#)」を参照してください。

エージェントへのキューの割当て

ワークフロー対応システムの各エージェントは、AQ キューに関連付ける必要があります。ローカル・システムは、このキューを使用してエージェントと対話します。メッセージを送信する場合、システムはキューにあるメッセージをエンキューして、宛先アドレスを設定します。メッセージを受信する場合、システムはキューにあるキュー・リスナーを実行します。『Oracle Workflow 管理者ガイド』の「キューの設定」を参照してください。

Oracle Workflow のビジネス・イベント・システムのイベント・メッセージは、データ型 WF_EVENT_T で定義された標準の書式でエンコードされます。各エージェントは、キュー・ハンドラと呼ばれる PL/SQL パッケージに割り当て、Workflow 標準書式とエージェントのキューに必要な書式との間で変換を行う必要があります。『Oracle Workflow API リファレンス』の「イベント・メッセージ構造」を参照してください。

注意： エージェントのキューのペイロード・タイプが WF_EVENT_T の場合でも、システム固有の AQ メッセージ・プロパティを設定するために、キュー・ハンドラに割り当てる必要があります。

Oracle Workflow には、WF_EVENT_QH および WF_ERROR_QH という 2 つの標準キュー・ハンドラが用意されています。これらのキュー・ハンドラは、SQLNET 伝播を使用し、WF_EVENT_T データ型をペイロードとして使用するキューに対応しています。WF_EVENT_QH は、通常のビジネス・イベント・システム処理を扱うキューで使用します。WF_ERROR_QH は、エラー・キューでのみ使用します。

Oracle Workflow には、WF_EVENT_OJMSTEXT_QH という標準キュー・ハンドラも用意されています。このキュー・ハンドラでは、ペイロード・タイプとして SYS.AQ\$_JMS_TEXT_MESSAGE データ型が使用されます。このキュー・ハンドラによって、ビジネス・イベント・システム経由で JMS テキスト・メッセージと通信できるようになります。『Oracle Workflow API リファレンス』の「WF_EVENT_T および SYS.AQ\$_JMS_TEXT_MESSAGE 間のマッピング」を参照してください。

また、Oracle Workflow には、WF_EVENT_OMB_QH というキュー・ハンドラも用意されています。Oracle8i Database に Oracle Message Broker を実装して、HTTP などの別のプロトコルを使用してシステム間でメッセージを伝播するときは、このキュー・ハンドラを使用します。『Oracle Workflow 管理者ガイド』の「WF_EVENT_OMB_QH キュー・ハンドラの設定」および『Oracle Workflow API リファレンス』の「WF_EVENT_T および OMBAQ_TEXT_MSG 間のマッピング」を参照してください。

他の書式を必要とするキューの場合は、その書式用のカスタム・キュー・ハンドラを作成します。カスタム・キュー・ハンドラには、カスタム書式のメッセージをエンキューおよびデキューするための標準 API を組み込む必要があります。6-26 ページの「[キュー・ハンドラの標準 API](#)」を参照してください。

標準エージェント

Oracle Workflow をインストールすると、ビジネス・イベント・システムに対して複数の標準エージェントが自動的に定義されます。

- WF_CONTROL: ユーザー向けではない、Oracle Workflow の内部エージェント
- WF_DEFERRED: 遅延サブスクリプション処理用標準エージェント
- WF_ERROR: エラー処理用標準エージェント
- WF_IN: デフォルトのインバウンド・エージェント
- WF_JMS_IN: JMS テキスト・メッセージ用のデフォルトのインバウンド・エージェント
- WF_JMS_OUT: JMS テキスト・メッセージ用のデフォルトのアウトバウンド・エージェント
- WF_NOTIFICATION_IN: 電子メール通知の応答用の標準インバウンド・エージェント
- WF_NOTIFICATION_OUT: 電子メール通知の応答用の標準アウトバウンド・エージェント
- WF_OUT: デフォルトのアウトバウンド・エージェント

これらのエージェントでは、Oracle Workflow のインストール時に自動的に定義される標準のキューが使用されます。『Oracle Workflow 管理者ガイド』の「キューの設定」を参照してください。

WF_IN、WF_JMS_IN、WF_OUT および WF_JMS_OUT エージェントは、使用可能 / 使用不可を切り替えることができます。ただし、エージェントの定義は変更しないでください。その他のエージェントの定義も変更しないでください。

Oracle Workflow では、ビジネス・イベント・システムの遅延サブスクリプション処理、エラー処理および通知メーラーのインバウンド電子メール処理を実行するために、それぞれ標準 WF_DEFERRED エージェント、標準 WF_ERROR エージェントおよび標準 WF_NOTIFICATION_IN エージェントに対応する、エージェント・リスナー・サービス・コンポーネントが自動的に実行されます。WF_IN、WF_JMS_IN、WF_OUT および WF_JMS_OUT エージェントをイベント・メッセージの伝播に使用する場合は、WF_IN と WF_JMS_IN のリスナーおよび WF_OUT と WF_JMS_OUT の伝播もスケジュールしてください。ただし、WF_CONTROL を使用する中間層のプロセスは直接キューからメッセージをデキューし、通知メーラーは WF_NOTIFICATION_OUT キューにあるメッセージを送信するため、WF_CONTROL および WF_NOTIFICATION_OUT エージェントの伝播はスケジュールする必要がありません。『Oracle Workflow 管理者ガイド』の「ローカル・インバウンド・エージェントのリスナーのスケジュール」および「ローカル・アウトバウンド・エージェントの伝播のスケジュール」を参照してください。

注意： Oracle Workflow には、WF_REPLAY_IN、WF_REPLAY_OUT および WF_SMTP_O_1_QUEUE という名前の、現在は使用されていない3つのエージェントもあります。

次の表は、標準 WF_CONTROL エージェントのデフォルトのプロパティを示しています。『Oracle Workflow 管理者ガイド』の「Workflow 制御キューのクリーン・アップ」を参照してください。

表 8-3

エージェント・プロパティ	値
名前	WF_CONTROL
表示名	Workflow Control Out Queue
説明	中間層プロセスへメッセージを知らせる Workflow JMS テキスト・メッセージ・キュー
プロトコル	SQLNET
アドレス	<workflow schema>.WF_CONTROL@<local database>
システム	<local system>
キュー・ハンドラ	WF_EVENT_OJMSTEXT_QH
キュー名	<workflow schema>.WF_CONTROL
方向	送信
ステータス	使用可能

次の表は、標準 WF_DEFERRED エージェントのデフォルトのプロパティを示しています。

表 8-4

エージェント・プロパティ	値
名前	WF_DEFERRED
表示名	WF_DEFERRED
説明	WF_DEFERRED
プロトコル	SQLNET
アドレス	<workflow schema>.WF_DEFERRED@<local database>
システム	<local system>
キュー・ハンドラ	WF_EVENT_QH

表 8-4 (続き)

エージェント・プロパティ	値
キュー名	<workflow schema>.WF_DEFERRED
方向	受信
ステータス	使用可能

次の表は、標準 WF_ERROR エージェントのデフォルトのプロパティを示しています。

表 8-5

エージェント・プロパティ	値
名前	WF_ERROR
表示名	WF_ERROR
説明	WF_ERROR
プロトコル	SQLNET
アドレス	<workflow schema>.WF_ERROR@<local database>
システム	<local system>
キュー・ハンドラ	WF_ERROR_QH
キュー名	<workflow schema>.WF_ERROR
方向	受信
ステータス	使用可能

次の表は、標準 WF_IN エージェントのデフォルトのプロパティを示しています。

表 8-6

エージェント・プロパティ	値
名前	WF_IN
表示名	WF_IN
説明	WF_IN
プロトコル	SQLNET
アドレス	<workflow schema>.WF_IN@<local database>
システム	<local system>

表 8-6 (続き)

エージェント・プロパティ	値
キュー・ハンドラ	WF_EVENT_QH
キュー名	<workflow schema>.WF_IN
方向	受信
ステータス	使用可能

次の表は、標準 WF_JMS_IN エージェントのデフォルトのプロパティを示しています。

表 8-7

エージェント・プロパティ	値
名前	WF_JMS_IN
表示名	Workflow JMS In Queue
説明	Workflow JMS テキスト・メッセージ・キュー
プロトコル	SQLNET
アドレス	<workflow schema>.WF_JMS_IN@<local database>
システム	<local system>
キュー・ハンドラ	WF_EVENT_OJMSTEXT_QH
キュー名	<workflow schema>.WF_JMS_IN
方向	受信
ステータス	使用可能

次の表は、標準 WF_JMS_OUT エージェントのデフォルトのプロパティを示しています。

表 8-8

エージェント・プロパティ	値
名前	WF_JMS_OUT
表示名	Workflow JMS Out Queue
説明	Workflow JMS テキスト・メッセージ・キュー
プロトコル	SQLNET
アドレス	<workflow schema>.WF_JMS_OUT@<local database>
システム	<local system>

表 8-8 (続き)

エージェント・プロパティ	値
キュー・ハンドラ	WF_EVENT_OJMSTEXT_QH
キュー名	<workflow schema>.WF_JMS_OUT
方向	送信
ステータス	使用可能

次の表は、標準 WF_NOTIFICATION_IN エージェントのデフォルトのプロパティを示しています。『Oracle Workflow 管理者ガイド』の「通知メーラーの導入」を参照してください。

表 8-9

エージェント・プロパティ	値
名前	WF_NOTIFICATION_IN
表示名	Workflow Notification In Queue
説明	Workflow 受信通知応答キュー
プロトコル	SQLNET
アドレス	<workflow schema>.WF_NOTIFICATION_IN@<local database>
システム	<local system>
キュー・ハンドラ	WF_EVENT_OJMSTEXT_QH
キュー名	<workflow schema>.WF_NOTIFICATION_IN
方向	受信
ステータス	使用可能

次の表は、標準 WF_NOTIFICATION_OUT エージェントのデフォルトのプロパティを示しています。『Oracle Workflow 管理者ガイド』の「通知メーラーの導入」を参照してください。

表 8-10

エージェント・プロパティ	値
名前	WF_NOTIFICATION_OUT
表示名	Workflow Notification Out Queue
説明	Workflow 通知送信キュー

表 8-10 (続き)

エージェント・プロパティ	値
プロトコル	SQLNET
アドレス	<workflow schema>.WF_NOTIFICATION_OUT@<local database>
システム	<local system>
キュー・ハンドラ	WF_EVENT_OJMSTEXT_QH
キュー名	<workflow schema>.WF_NOTIFICATION_OUT
方向	送信
ステータス	使用可能

次の表は、標準 WF_OUT エージェントのデフォルトのプロパティを示しています。

表 8-11

エージェント・プロパティ	値
名前	WF_OUT
表示名	WF_OUT
説明	WF_OUT
プロトコル	SQLNET
アドレス	<workflow schema>.WF_OUT@<local database>
システム	<local system>
キュー・ハンドラ	WF_EVENT_QH
キュー名	<workflow schema>.WF_OUT
方向	送信
ステータス	使用可能

▶ エージェントの定義

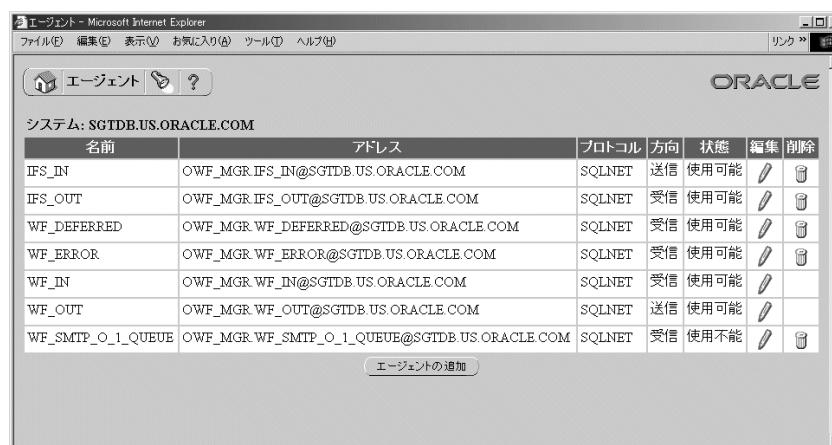
1. Web ブラウザを使用して、次の URL に接続します。

```
<webagent>/wf_event_html.listagents
```

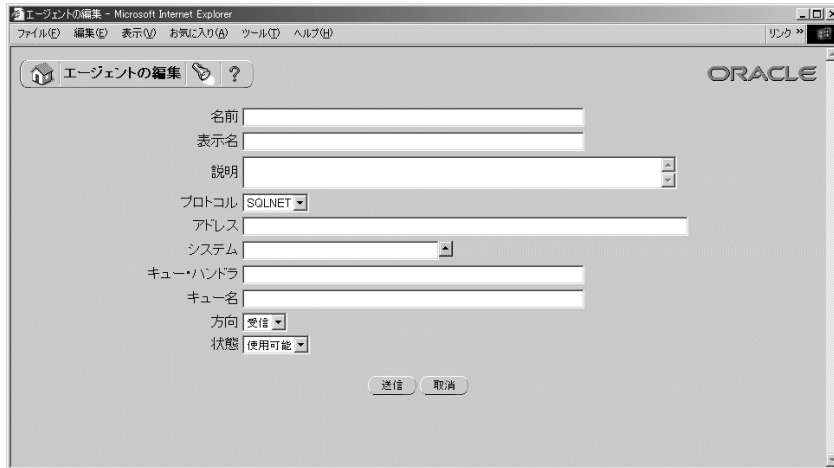
<webagent> は、Web サーバーで Oracle Workflow 用に構成された Web エージェントのベース URL に置き換えてください。『Oracle Workflow 管理者ガイド』の「グローバル・ユーザー設定の設定」を参照してください。

注意： この URL はセキュリティ画面にアクセスするため、現行 Web セッションで有効なユーザーとしてログオンしていない場合は、ページが表示される前に有効なユーザーとしてのログオンを求めるプロンプトが表示されます。「イベント・マネージャ」 Web ページにアクセスするには、ワークフロー管理者権限が必要です。

注意： 「エージェント」 Web ページには、Oracle Workflow ホーム・ページからもアクセスできます。『Oracle Workflow 管理者ガイド』の「Oracle Workflow ホーム・ページへのアクセス」を参照してください。



- 「エージェント」ページに既存のエージェントのリストが表示されます。配置されているシステム別にグループ化されています。「エージェント」ページには、各エージェントの内部名、アドレス、プロトコル、方向およびステータスの一覧が表示されます。「エージェントの追加」ボタンを選択して「エージェントの編集」ページを開きます。



3. エージェントの内部名を「名前」フィールドに入力します。エージェントの内部名は、エージェントのシステム内で一意である必要があります。すべての Oracle Workflow API、SQL スクリプトおよび PL/SQL プロシージャでは、エージェントの識別時に内部名が参照されます。

注意： 内部名には大文字のみを使用する必要があります。一重引用符 (')、二重引用符 (") またはスペースを含めないでください。

4. エージェントの表示名を入力します。
5. オプションで、エージェントの説明を入力します。
6. エージェントが使用するメッセージ通信プロトコルを選択します。
7. エージェントがシステムへのインバウンド通信を利用する場合は、エージェントのアドレスを入力します。アドレスの書式は、選択するプロトコルによって異なります。

SQLNET プロトコルを使用するエージェントの場合、AQ 伝播を有効にするには、次の書式でアドレスを指定する必要があります。

```
<schema>.<queue>@<database link>
```

<schema> はキューを所有するスキーマ、<queue> はキューの名前、<database link> はキューが配置されているインスタンスへのデータベース・リンクを表します。

注意： データベース・リンク名には、データベース・リンクの作成時に指定された名前を正確に入力する必要があります。『Oracle Workflow 管理者ガイド』の「データベース・リンクの作成」を参照してください。

8. エージェントが定義されているシステムを入力します。「システム」フィールドの上矢印のアイコンをクリックすると、選択可能なシステムのリストが表示されます。8-25 ページの「[値リストの使用](#)」を参照してください。
9. エージェントのキュー・ハンドラを入力します。キュー・ハンドラは PL/SQL パッケージの 1 つで、Workflow イベント・メッセージ書式 (WF_EVENT_T) とエージェントに関連付けられているキューに必要なメッセージ書式との間で変換を行います。6-26 ページの「[キュー・ハンドラの標準 API](#)」を参照してください。

注意： キュー・ハンドラ名は、すべて大文字で入力する必要があります。

10. ローカル・システムがエージェントとの対話に使用するキューの名前を入力します。このキュー名は、ローカル・システムのみが参照するため、このシステムの有効範囲に対応し、データベース・リンクを必要としない名前である必要があります。次の書式を使用してキュー名を指定します。

<schema>.<queue>

<schema> はキューを所有するスキーマ、<queue> はキュー名を表します。

注意： キュー名は、すべて大文字で入力する必要があります。

11. 「方向」フィールドで、システムへのインバウンド通信をサポートするエージェントの場合は「受信」を選択し、システムからのアウトバウンド通信をサポートするエージェントの場合は「送信」を選択します。
12. 「ステータス」フィールドで、エージェントのステータスとして「使用可能」または「使用不能」を選択します。エージェントを使用不能にしても、そのエージェントは「エージェント」リストに表示され、参照できます。ただし、有効なサブスクリプションでは使用できません。
13. 「送信」ボタンを選択してエージェントを保存し、「エージェント」ページに戻ります。「エージェント」ページに、更新されたエージェントのリストが表示されます。

「取消」ボタンを選択し、エージェントを保存しないで「エージェント」ページに戻ることもできます。

▶ エージェントの検索

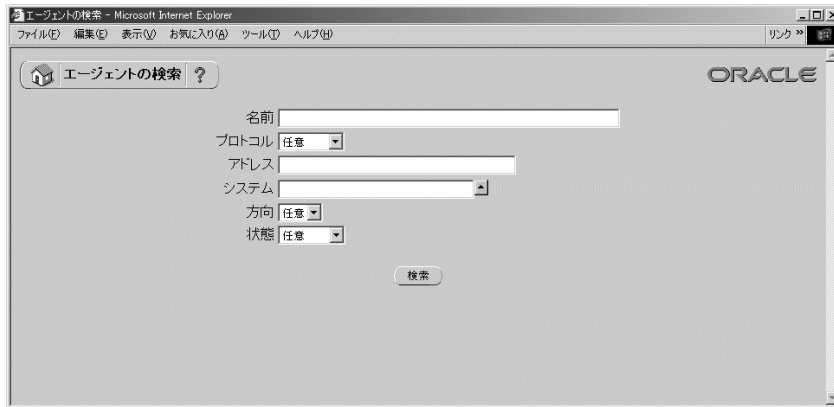
1. Web ブラウザを使用して、次の URL に接続します。

```
<webagent>/wf_event_html.findagent
```

<webagent> は、Web サーバーで Oracle Workflow 用に構成された Web エージェントのベース URL に置き換えてください。『Oracle Workflow 管理者ガイド』の「グローバル・ユーザー設定の設定」を参照してください。

注意： この URL はセキュリティ画面にアクセスするため、現行 Web セッションで有効なユーザーとしてログオンしていない場合は、ページが表示される前に有効なユーザーとしてのログオンを求めるプロンプトが表示されます。「イベント・マネージャ」Web ページにアクセスするには、ワークフロー管理者権限が必要です。

注意： 「エージェントの検索」Web ページには、Oracle Workflow ホーム・ページからもアクセスできます。『Oracle Workflow 管理者ガイド』の「Oracle Workflow ホーム・ページへのアクセス」を参照してください。

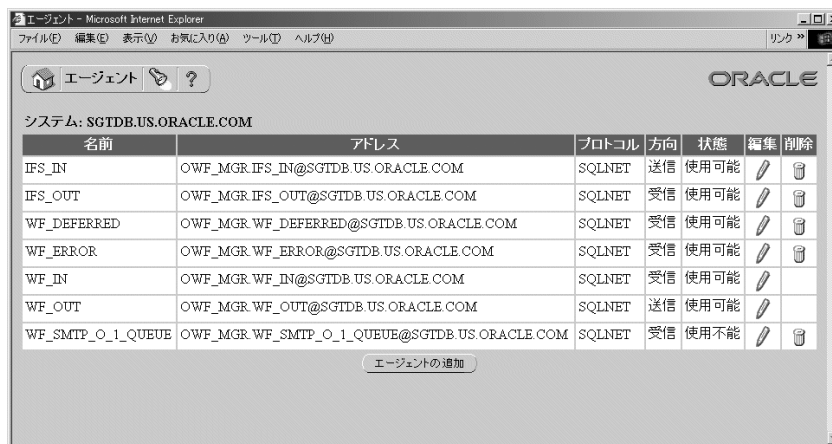


2. 「エージェントの検索」ページが表示されます。「エージェントの検索」ページでは、検索基準を入力して特定のエージェントを検索できます。検索基準として、次の項目を使用できます。
 - 名前： 表示するエージェントの内部名を入力します。
 - プロトコル： 表示するエージェントのプロトコルを選択するか、「任意」を選択してすべてのプロトコルのエージェントを表示します。
 - アドレス： 表示するエージェントのアドレスを入力します。

- システム： 表示するエージェントのシステムを入力します。このフィールドの上矢印のアイコンをクリックすると、選択可能なシステムのリストが表示されます。8-25 ページの「[値リストの使用](#)」を参照してください。
 - 方向： 表示するエージェントの方向として、「受信」または「送信」を選択します。すべての方向のエージェントを表示するときは、「任意」を選択します。
 - ステータス： 表示するエージェントのステータスとして「使用可能」または「使用不能」を選択します。すべてのステータスのエージェントを表示するときは、「任意」を選択します。
3. 「検索」ボタンを選択します。「エージェント」ページに、検索基準と一致したエージェントのリストが表示されます。

▶ エージェントの更新または削除

1. 「エージェント」ページで、更新または削除するエージェントを選択します。「エージェントの検索」ページを使用して目的のエージェントを検索し、「エージェント」ページに表示することができます。8-37 ページの「[エージェントの検索](#)」を参照してください。



The screenshot shows a web browser window titled "エージェント - Microsoft Internet Explorer" displaying the Oracle Event Manager interface. The page title is "システム: SGTDB.US.ORACLE.COM". Below the title is a table with columns: 名前 (Name), アドレス (Address), プロトコル (Protocol), 方向 (Direction), 状態 (Status), 編集 (Edit), and 削除 (Delete). The table lists several agents with their respective details.

名前	アドレス	プロトコル	方向	状態	編集	削除
IFS_IN	OWF_MGR.IFS_IN@SGTDB.US.ORACLE.COM	SQLNET	送信	使用可能		
IFS_OUT	OWF_MGR.IFS_OUT@SGTDB.US.ORACLE.COM	SQLNET	受信	使用可能		
WF_DEFERRED	OWF_MGR.WF_DEFERRED@SGTDB.US.ORACLE.COM	SQLNET	受信	使用可能		
WF_ERROR	OWF_MGR.WF_ERROR@SGTDB.US.ORACLE.COM	SQLNET	受信	使用可能		
WF_IN	OWF_MGR.WF_IN@SGTDB.US.ORACLE.COM	SQLNET	受信	使用可能		
WF_OUT	OWF_MGR.WF_OUT@SGTDB.US.ORACLE.COM	SQLNET	送信	使用可能		
WF_SMTP_O_1_QUEUE	OWF_MGR.WF_SMTP_O_1_QUEUE@SGTDB.US.ORACLE.COM	SQLNET	受信	使用不能		

Below the table is a button labeled "エージェントの追加".

2. エージェントを更新するには、そのエージェントの「編集」列にある鉛筆アイコンを選択します。「エージェントの編集」ページが表示されます。エージェント定義に変更を加え、変更内容を保存します。8-34 ページの「[エージェントの定義](#)」を参照してください。
3. エージェントを削除するには、そのエージェントの「削除」列にあるごみ箱アイコンを選択して、表示される確認ウィンドウで「OK」を選択します。確認ウィンドウで「取消」ボタンを選択し、エージェントを削除しないで「エージェント」ページに戻ることができます。

注意： 削除できるエージェントは、サブスクリプションから参照されていないエージェントのみです。

注意： エージェントに加えた変更がメッセージ伝播に必要な物理実装に影響する場合は、伝播設定を再度チェックする必要があります。『Oracle Workflow 管理者ガイド』の「ビジネス・イベント・システムの設定」を参照してください。

イベント・サブスクリプション

イベント・サブスクリプションには、特定のイベントと特定のシステムの間、およびトリガー・イベントの発生時に実行する処理を登録します。イベント・サブスクリプションは、イベント・マネージャで定義できます。

Oracle Workflow をインストールすると、事前定義済みの Workflow イベントに対していくつかのデフォルトのサブスクリプションが自動的に作成されます。これらのサブスクリプションを使用可能 / 使用不可を切り替えたり、コピーしたりすることにより、必要なイベント処理を実行できます。9-2 ページの「事前定義済 Workflow イベント」を参照してください。

イベントがローカルで呼び出されるか外部ソースから着信すると、イベント・マネージャは、そのイベントまたは Any イベントへの有効なサブスクリプションをローカル・システム単位に検索および実行します。発生したイベントへの有効なサブスクリプションが存在しない場合 (Any イベントへのサブスクリプションを除く)、Oracle Workflow は Unexpected イベントへの有効なサブスクリプションを実行します。9-15 ページの「Any イベント」および 9-18 ページの「Unexpected イベント」を参照してください。

Oracle Workflow には、「システム:エラー」項目タイプ内の Unexpected イベントおよびデフォルト・イベント・エラー・プロセスに対する事前定義済サブスクリプションのデフォルトのエラー処理が用意されています。また、独自のイベント用にカスタムのエラー処理を定義することもできます。11-5 ページの「イベント・サブスクリプション処理のエラー処理」を参照してください。

サブスクライバの定義

サブスクリプションの定義を開始するには、サブスクライバとなるシステムを指定します。サブスクライバは、サブスクリプションを実行するシステムです。

各サブスクリプションには、特定のシステムに対するアクションを定義します。つまり、サブスクリプションは、実行する処理に関係するシステム単位に、個別に定義する必要があります。たとえば、システム間でデータを伝播する場合は、送信システムに対して 1 つのサブスクリプションを定義し、受信システムに対して別のサブスクリプションを定義します。

サブスクリプションのトリガーの定義

サブスクリプションを適用するイベントのソース・タイプを指定する必要があります。イベントには次のソース・タイプがあります。

- ローカル：サブスクリプションは、サブスクライバ・システムで発生したイベントにのみ適用されます。
- 外部：サブスクリプションは、サブスクライバ・システム上のインバウンド・エージェントが受信したイベントにのみ適用されます。

注意：サブスクライバ・システム上のインバウンド・エージェントが受信したイベント・メッセージは、送信元エージェントがリモート・システム上に存在するかローカル・システム上に存在するかにかかわらず、すべて外部ソースと見なされます。

- エラー：サブスクリプションは、WF_ERROR キューからデキューされたエラー・イベントにのみ適用されます。

次に、サブスクリプションをトリガーするイベントを選択します。個々のイベントまたはイベント・グループを選択できます。イベント・グループを選択した場合、サブスクリプションは、グループのいずれかのメンバー・イベントが発生したときにトリガーされます。

また、特定のソース・エージェントから受信したイベントのみがトリガーするように、サブスクリプションを制限することもできます。ただし、ほとんどの場合、特定のソース・エージェントを指定する必要はありません。

サブスクリプションの実行の制御

複数のサブスクリプションを同じイベントに対して定義する場合は、各サブスクリプションにフェーズ番号を指定することで、イベント・マネージャがサブスクリプションを実行する順序を制御できます。サブスクリプションは、フェーズの昇順で実行されます。たとえば、イベントの発生時に最初に実行するサブスクリプションに対して 10、次に実行するサブスクリプションに対して 20 を指定します。検証を行うサブスクリプションを実行してから他のタイプの処理を行うサブスクリプションを実行するなど、フェーズを使用すれば、異なるタイプのアクションを適切な順序で実行することができます。

注意： 同じフェーズ番号を複数のサブスクリプションに対して指定した場合は、関連するサブスクリプションが無作為に実行されます。ただし、そのサブスクリプション・グループは、他のフェーズ番号のサブスクリプションに基づいて、フェーズ番号で指定された順序で実行されます。

サブスクリプションのフェーズ番号を使用して、サブスクリプションがすぐに実行されるか遅延されるかを制御することもできます。イベント・マネージャは、フェーズ番号が 100 以降のサブスクリプションを遅延サブスクリプションとして処理します。1～99 のフェーズ番号付きのサブスクリプションは、イベントの処理が他の方法で延期されなければ、ただちに実行されます。8-94 ページの「[遅延サブスクリプション処理](#)」を参照してください。

フェーズ番号 0 (ゼロ) は、Oracle Workflow にシードされたサブスクリプションのために予約されています。

サブスクリプションには、実行される処理に応じて、イベント・データに必要なイベント情報が異なります。すべてのイベント情報が必要な場合や、イベントのインスタンスを識別するイベント・キーのみが必要な場合があります。すべてのイベント・データを必要としないサブスクリプションの場合は、イベント・キーをルール・データとして指定すると、パフォーマンスを向上させることができます。ローカルで呼び出されるイベントの場合、イベント・マネージャはサブスクリプションを実行する前に、各サブスクリプションにすべてのイベント・データが必要かどうかをチェックします。必要なイベント・データが渡されていない場合、イベント・マネージャはそのイベントに対してジェネレート関数を実行し、イベント・データを生成します。ただし、イベントへのサブスクリプションがイベント・データを必要としない場合は、サブスクリプションの実行に必要なリソースを最小限に抑えるため、イベント・マネージャはジェネレート関数を実行しません。

注意： すべてのイベント・データを必要とするサブスクリプションの場合でも、イベントを呼び出した後にできるだけ早く制御をコール側アプリケーションに戻すには、すべてのサブスクリプションを遅延させます。後でそれらのサブスクリプションを実行するまで、イベント・マネージャはジェネレート関数を実行しません。

サブスクリプションに対するアクションの定義

サブスクリプションには、次のタイプの処理を指定できます。

- イベント・メッセージに対する関数の実行
- ワークフロー・プロセスへのイベント・メッセージの送信
- エージェントへのイベント・メッセージの送信

ルール関数の実行

イベント・メッセージに対して関数を実行するには、実行するルール関数を指定します。また、関数に渡す追加のパラメータを指定することもできます。

Oracle Workflow には、基本的なサブスクリプション処理を実行するための標準のデフォルト・ルール関数が用意されています。この関数は、サブスクリプションに対してルール関数が指定されていない場合に、デフォルトで実行されます。デフォルトのルール関数では、次の処理が行われます。

- ワークフロー・プロセスへのイベント・メッセージの送信（サブスクリプション定義で指定されている場合）
- エージェントへのイベント・メッセージの送信（サブスクリプション定義で指定されている場合）

『Oracle Workflow API リファレンス』の「Default_Rule」を参照してください。

Oracle Workflow には、テストおよびデバッグ、またはその他の目的で使用可能な標準のルール関数もいくつか用意されています。『Oracle Workflow API リファレンス』の「イベント・サブスクリプションのルール関数の API」を参照してください。

サブスクリプション処理は、カスタム・ルール関数を作成することで拡張できます。カスタム・ルール関数は、標準 API に従って定義する必要があります。6-29 ページの「イベント・サブスクリプションのルール関数の標準 API」を参照してください。

ルール関数は、次のような様々な目的で使用できます。

- 検証を実行する。
- インバウンド・メッセージをアプリケーションの「Receive」メッセージ・ハンドラとして処理する。
- アウトバウンド・メッセージに変更を加える。他のメッセージに関連付ける関連 ID を追加するなど。

注意： カスタム・ルール関数から

WF_EVENT_FUNCTIONS_PKG.AddCorrelation() をコールすると、カスタム処理中に関連 ID を追加できます。『Oracle Workflow API リファレンス』の「AddCorrelation」を参照してください。

ルール関数は、イベント・メッセージに対して読取りまたは書込みを行ったり、任意のデータベース処理を実行したりします。ただし、ルール関数内ではコミットしないでください。コミットは、コール側のアプリケーションが行うため、イベント・マネージャではコミットを発行しません。また、ルール関数を使用して、セキュリティやNLSの設定などの接続コンテキストを変更しないでください。ルール関数がエラーを返した場合、サブスクリプション処理は中止されます。

特定のイベントに対してサブスクリプション処理を実行するときに、連続したステップがいくつか含まれている場合は、再利用できない複雑で特別なルール関数を作成するのではなく、再利用できる単純なルール関数を作成して複数のサブスクリプションをイベントに定義することをお勧めします。複数のサブスクリプションの実行順序を指定するには、サブスクリプションに対してフェーズ値を入力します。

注意： デフォルト以外のルール関数を入力した場合は、参照する関数に対してワークフロー情報およびエージェント情報を入力できますが、指定されたワークフローおよびエージェントに対してイベント・メッセージは自動的に送信されません。この場合、送信処理をルール関数に明示的に組み込むか、デフォルトのルール関数を使用せずに送信処理を実行する別のサブスクリプションを定義する必要があります。

ワークフロー・プロセスへのイベントの送信

イベントをワークフロー・プロセスに送信するには、送信先のプロセスの項目タイプとプロセス名を指定する必要があります。プロセスの項目キーは、イベント・メッセージに指定されている相関 ID によって決定されます。相関 ID が指定されていない場合は、イベント・キーによって決定されます。

注意： プロセス・インスタンスの項目キーには、シングルバイト・キャラクターのみ使用できます。マルチバイトの値を含めることはできません。

注意： サブスクリプションの処理中に `WF_EVENT_FUNCTIONS_PKG.AddCorrelation()` をコールすると、イベント・メッセージに相関 ID を追加できます。 `AddCorrelation()` を使用するには、 `ITEMKEY` というサブスクリプション・パラメータを入力して、相関 ID を生成する関数を指定する必要があります。関数は次の書式で指定する必要があります。

`ITEMKEY=<package_name.function_name>`

『Oracle Workflow API リファレンス』の「AddCorrelation」を参照してください。

イベントをワークフロー・プロセスに送信することにより、事前定義済の関数を直接実行したり、事前定義済の受信者にイベントを送信するだけでなく、複雑な処理またはルーティング・ロジックをモデル化できます。たとえば、イベント・メッセージの内容に基づいて、別の関数に分岐したり、サブプロセスを開始したり、通知を送信したり、宛先エージェントを選択したりできます。また、イベント・メッセージ自体を変更することもできます。

ワークフロー・プロセス内では、イベントはイベント・アクティビティ単位に表されます。3-56 ページの「[イベント・アクティビティ](#)」を参照してください。

プロセスがイベントを受信すると、ワークフロー・エンジンは「受信」イベント・アクティビティ・ノードのイベントの詳細の指定に従って、イベント名、イベント・キーおよびイベント・メッセージを項目タイプ属性に格納します。さらに、ワークフロー・エンジンは、イベント・メッセージのパラメータ・リストのパラメータをプロセスの項目タイプとして設定します。パラメータに対応する属性がまだ存在しない場合は、新しい項目タイプ属性を作成します。また、サブスクリプションの GUID (グローバル意識別子) が動的な項目属性として設定されます。このため、ワークフロー・プロセスから、サブスクリプション定義に含まれる他の情報を参照することができます。

イベントが別のワークフロー・プロセスの呼出しイベント・アクティビティによって発生した場合は、そのプロセスの項目タイプと項目キーがイベント・メッセージ内のパラメータ・リストに追加されます。ワークフロー・エンジンは、イベントを受信するプロセスの親として、指定されたプロセスを自動的に設定し、既存の親の設定を上書きします。『Oracle Workflow API リファレンス』の「[SetItemParent](#)」を参照してください。

追加パラメータを指定すると、ワークフロー・プロセスの項目属性として設定されます。追加パラメータをサブスクリプションの「パラメータ」フィールドに入力し、サブスクリプションのルール関数で `WF_RULE.SetParametersIntoParameterList()` を使用すれば、指定したサブスクリプション・パラメータをイベント・メッセージのパラメータ・リストに設定することができます。ワークフロー・プロセスにイベントが着信すると、それらのイベント・パラメータはワークフロー・プロセスの項目属性として設定されます。『Oracle Workflow API リファレンス』の「[SetParametersIntoParameterList](#)」を参照してください。

注意： イベントをワークフロー・プロセスに送信するには、Oracle Workflow から提供されるデフォルトのルール関数を使用するか、送信処理をカスタム・ルール関数に組み込む必要があります。6-29 ページの「[イベント・サブスクリプションのルール関数の標準 API](#)」を参照してください。

エージェントへのイベントの送信

イベントをエージェントに送信するには、アウトバウンド・メッセージを送信する送信エージェントまたはインバウンド・メッセージを受信する宛先エージェント、あるいはその両方のエージェントを指定します。

- 宛先エージェントと送信エージェントの両方を指定した場合、イベント・メッセージは送信エージェントのキューに格納され、宛先エージェントに伝播されます。

- 送信エージェントを指定しないで宛先エージェントを指定した場合は、サブスクライバ・システム上のアウトバウンド・エージェントのうち、キュー・タイプが宛先エージェントと一致するものが選択されます。イベント・メッセージは、このアウトバウンド・エージェントのキューに格納され、宛先エージェントに伝播されます。
- 宛先エージェントを指定しないで送信エージェントを指定した場合、イベント・メッセージは送信エージェントのキューに格納されますが、受信者は指定されていません。
 - 送信エージェントがサブスクライバ・リストで複数コンシューマ・キューを使用する場合は、宛先エージェントを省略できます。（標準の Workflow キュー・ハンドラは、複数コンシューマ・キューでのみ機能します。）この場合、キューのサブスクライバ・リストによって、メッセージをデキューできるコンシューマが決定されます。キューに対してサブスクライバ・リストが定義されていない場合、イベント・メッセージはエラー処理のために WF_ERROR キューに格納されます。

注意： Oracle Advanced Queuing が使用する複数コンシューマ・キューのサブスクライバ・リストは、Oracle Workflow Business Event System のイベント・サブスクリプションとは異なります。詳細は、『Oracle9i アプリケーション開発者ガイド-アドバンスト・キューイング』の「サブスクリプションおよび受信者リスト」または『Oracle Streams アドバンスト・キューイング・ユーザーズ・ガイドおよびリファレンス』を参照してください。

- 送信エージェントが単一コンシューマ・キューを使用するときに、カスタム・キュー・ハンドラが定義されている場合は、宛先エージェントを省略できます。単一コンシューマ・キューの場合、コンシューマを指定する必要はありません。

受信者がメッセージをデキューするときの、優先順位を指定することができます。メッセージは、優先順位の昇順でデキューされます。

注意： イベントをエージェントに送信するには、Oracle Workflow から提供されるデフォルトのルール関数を使用するか、送信処理をカスタム・ルール関数に組み込む必要があります。6-29 ページの「[イベント・サブスクリプションのルール関数の標準 API](#)」を参照してください。

伝播したイベント・メッセージを受信者にすぐに送信せずに後で送信する場合は、イベント・メッセージの SEND_DATE 属性を未来日付に設定します。設定する送信日は、サブスクリプションの処理期間中で、イベントが送信される前である必要があります。つまり、送信処理が開始される前に、直前のサブスクリプションまたはルール関数の初期処理が実行されている期間です。イベント・メッセージは宛先エージェントに伝播されますが、指定した日付までデキューされません。

サブスクリプションの識別情報の定義

サブスクリプション定義を自動的に作成するプログラムを使用する場合は、プログラム独自の名前と簡単な識別子をサブスクリプションの所有者名および所有者タグとして設定できます。この識別情報を使用して、所有するサブスクリプションをプログラムから検索することができます。所有者名および所有者タグは、「サブスクリプションの編集」ページから必要に応じて手動で更新できます。

各サブスクリプションには、サブスクリプション定義を更新できるかどうかを判断するためのカスタマイズ・レベルが割り当てられています。このカスタマイズ・レベルは、**Oracle Applications** のシード・データを保護し、アップグレード中にユーザーのカスタマイズを保存するために使用されます。各サブスクリプションには、次のカスタマイズ・レベルの1つが割り当てられています。

- **コア**： サブスクリプション定義を変更することはできません。このレベルは、**Oracle Applications** にシードされたサブスクリプションに対してのみ使用されます。
- **制限**： サブスクリプションのステータスを「使用可能」または「使用不能」に更新できますが、サブスクリプション定義を変更することはできません。このレベルは、**Oracle Applications** にシードされたサブスクリプションに対してのみ使用されます。
- **ユーザー**： サブスクリプション定義内のすべてのプロパティを変更することができます。このレベルは、ユーザーが定義したサブスクリプションに自動的に設定されます。

遅延サブスクリプション処理

イベントが発生したときにイベントのサブスクリプションをすぐに実行しない場合は、サブスクリプションを遅延できます。サブスクリプションを遅延すると、コール側アプリケーションに制御を早く戻すことができ、イベント・マネージャは負荷の大きいサブスクリプション処理を後で実行できます。

サブスクリプションは、次の3つの方法で遅延できます。

- **SEND_DATE** 属性に未来日付を指定して、イベントを呼び出します。この方法は、ローカルで呼び出されたイベントに対するすべてのサブスクリプション処理を特定の有効日まで遅延するときに使用します。
- イベントに対するサブスクリプションのフェーズ番号を 100 以降に定義します。この方法は、ローカル・イベントまたは外部イベントの特定のサブスクリプション処理を遅延するときに使用します。
- イベントを呼び出す前に、イベント・マネージャのディスパッチ・モードを遅延処理に設定します。この方法は、ローカルで呼び出されるイベントに対するサブスクリプション処理をすべて遅延するときに使用できます。ただし、この方法は特別な状況でのみ使用してください。

これらのいずれかの方法によってイベントのサブスクリプション処理が遅延されると、イベント・メッセージは **WF_DEFERRED** エージェントに関連付けられた標準 **WF_DEFERRED** キューに格納されます。**WF_DEFERRED** エージェントを監視するには、リスナーをスケジュールする必要があります。『Oracle Workflow 管理者ガイド』の「ローカル・インバウンド・エージェントのリスナーのスケジュール」を参照してください。

リスナーは、優先順位に従って **WF_DEFERRED** エージェントからイベント・メッセージをデキューします。イベント・メッセージのソース・タイプ（「ローカル」または「外部」）は保持されます。これらのイベントに対するサブスクリプション処理の遅延時間は、リスナーに定義されているスケジュールおよび指定した有効日（未来日付のイベントの場合）によって決まります。

未来日付の送信日を使用したサブスクリプション処理の遅延

特定の未来有効日までローカル・イベントのサブスクリプション処理を遅延するには、**SEND_DATE** 属性に特定の未来有効日を設定してイベントを呼び出します。たとえば、新しい従業員が採用されたらすぐにその従業員の情報を人事管理アプリケーションに入力し、給与計算の処理は従業員の勤務開始日まで遅延します。

イベントが未来送信日で呼び出されると、イベント・マネージャはそのイベント・メッセージをすぐに **WF_DEFERRED** キューに格納し、イベントのサブスクリプションは実行しません。そのイベントに対するすべてのサブスクリプションは、フェーズ番号にかかわらずすべて遅延されます。イベントはその送信日まで **WAIT** 状態になります。送信日になると、イベント・メッセージはデキュー可能になり、エージェント・リスナーが **WF_DEFERRED** キュー上で実行されたときにデキューされます。サブスクリプション処理の遅延時間は、指定した送信日およびリスナーに定義されているスケジュールによって決まります。

リスナーがイベント・メッセージをデキューすると、イベント・マネージャが `ERROR_SUBSCRIPTION` 属性のサブスクリプション ID をチェックします。イベント・メッセージにサブスクリプション ID が含まれている場合、つまり、イベントが呼び出された直後にそのイベントに対するすべてのサブスクリプション処理が遅延された場合、イベント・マネージャはイベントに対するすべてのサブスクリプションをフェーズの昇順で実行します。

注意： イベントが呼び出されたときにイベントが遅延されていなかった場合は、そのイベントが `WF_DEFERRED` キューからデキューされるときに、イベント・マネージャはそのイベントに対する適切なサブスクリプションをすべて実行します。サブスクリプションのフェーズ番号は無視されます。100 以降のフェーズ番号が付いている場合でも、サブスクリプションは遅延されません。

関連項目：

『Oracle Workflow API リファレンス』の「Raise」

サブスクリプションのフェーズ番号を使用したサブスクリプション処理の遅延

サブスクリプションのフェーズ番号を使用して、サブスクリプションがすぐに実行されるか遅延されるかを制御することもできます。イベント・マネージャは、フェーズ番号が 100 以降のサブスクリプションを遅延サブスクリプションとして処理します。「ローカル」および「外部」のサブスクリプションをこの方法で遅延できます。

注意： イベントがローカルで呼び出されるときにこの遅延方法を使用するには、そのイベントが未来日付の送信日で呼び出され、イベント・マネージャのサブスクリプション処理が通常の同期モードに設定されている必要があります。それ以外の場合、イベント・メッセージは `WF_DEFERRED` キューに格納され、イベントがデキューされるまでイベント・マネージャはサブスクリプションを実行しません。

トリガー・イベントが呼び出されるか着信すると、100 以降のフェーズ番号のサブスクリプションを検出するまで、イベント・マネージャはそのイベントに対するサブスクリプションをフェーズ番号順に実行します。イベント・マネージャは、そのサブスクリプションをイベント・メッセージ内の `ERROR_SUBSCRIPTION` 属性に設定し、サブスクリプション・プロパティに指定されている優先度を `PRIORITY` 属性に設定します。イベント・メッセージは、標準 `WF_DEFERRED` キューに格納されます。

サブスクリプション処理の遅延時間は、`WF_DEFERRED` エージェントを監視するエージェント・リスナーに定義されているスケジュールによって決まります。リスナーがイベント・メッセージをデキューすると、イベント・マネージャは `ERROR_SUBSCRIPTION` 属性のサブスクリプション ID をチェックします。サブスクリプション ID が存在する場合、つまり、そのサブスクリプション以降のサブスクリプション処理が遅延されている場合、イベント・

マネージャは最初にそのサブスクリプションを実行してから、そのサブスクリプション以降のフェーズ番号を持つイベントについてすべてのサブスクリプションを実行します。

注意： イベント・マネージャは、イベント・メッセージに設定されているサブスクリプションのフェーズ番号で、サブスクリプション処理を再開します。そのイベントが最初に処理されたときにフェーズ番号が 100 のサブスクリプションが存在しなかった場合は、フェーズ番号 100 から開始されない場合があります。

イベント・マネージャのディスパッチ・モードを使用したサブスクリプション処理の遅延

イベントをローカル・アプリケーションから呼び出す場合、呼び出すときにそのイベントのサブスクリプション処理をすべて遅延することもできます。この方法を実行するには、Raise() API をコールする直前に、遅延（非同期）処理を示す「ASYNC」モードで SetDispatchMode() API をコールします。『Oracle Workflow API リファレンス』の「SetDispatchMode」を参照してください。

このメソッドは、アプリケーション内で遅延をハードコーディングする必要があるため、特別な状況でのみ使用してください。柔軟性を維持しながら、アプリケーションに影響を与えずにサブスクリプション処理を変更するには、未来日付の送信日でイベントを呼び出すか、サブスクリプションのフェーズ番号を使用して一部またはすべてのサブスクリプションを遅延として指定します。

ディスパッチ・モードが遅延処理に設定された後にイベントが呼び出されると、イベント・マネージャはイベント・メッセージをすぐに WF_DEFERRED キューに格納し、イベントのサブスクリプションは実行しません。そのイベントに対するすべてのサブスクリプションは、フェーズ番号にかかわらずすべて遅延されます。

サブスクリプション処理の遅延時間は、WF_DEFERRED エージェントを監視するエージェント・リスナーに定義されているスケジュールによって決まります。リスナーがイベント・メッセージをデキューすると、イベント・マネージャが ERROR_SUBSCRIPTION 属性のサブスクリプション ID をチェックします。イベント・メッセージにサブスクリプション ID が含まれている場合、つまり、イベントが呼び出された直後にそのイベントに対するすべてのサブスクリプション処理が遅延された場合、イベント・マネージャはイベントに対するすべてのサブスクリプションをフェーズの昇順で実行します。

注意： イベントが呼び出されたときにイベントが遅延されていなかった場合は、そのイベントが WF_DEFERRED キューからデキューされるときに、イベント・マネージャはそのイベントに対する適格なサブスクリプションをすべて実行します。サブスクリプションのフェーズ番号は無視されます。100 以降のフェーズ番号が付いている場合でも、サブスクリプションは遅延されません。

関連項目：

6-29 ページ「[イベント・サブスクリプションのルール関数の標準 API](#)」

『Oracle Workflow 管理者ガイド』の「ローカル・インバウンド・エージェントのリスナーのスケジュール」

▶ **イベント・サブスクリプションの定義**

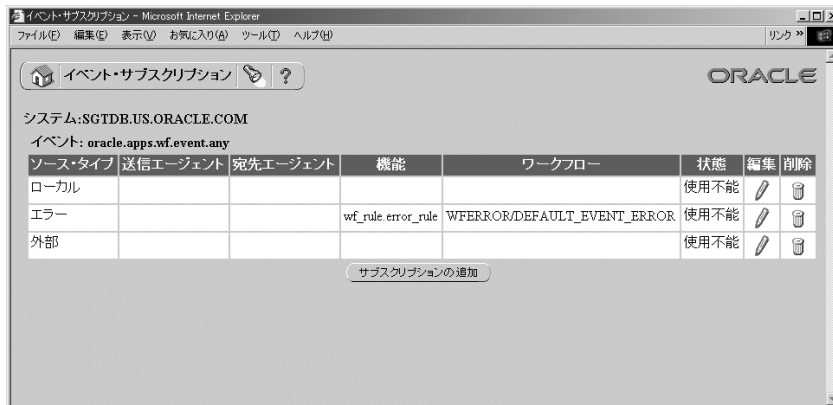
1. Web ブラウザを使用して、次の URL に接続します。

```
<webagent>/wf_event_html.listsubscriptions
```

<webagent> は、Web サーバーで Oracle Workflow 用に構成された Web エージェントのベース URL に置き換えてください。『Oracle Workflow 管理者ガイド』の「グローバル・ユーザー設定の設定」を参照してください。

注意： この URL はセキュリティ画面にアクセスするため、現行 Web セッションで有効なユーザーとしてログオンしていない場合は、ページが表示される前に有効なユーザーとしてのログオンを求めるプロンプトが表示されます。「イベント・マネージャ」Web ページにアクセスするには、ワークフロー管理者権限が必要です。

注意： 「イベント・サブスクリプション」Web ページには、Oracle Workflow ホーム・ページからもアクセスできます。『Oracle Workflow 管理者ガイド』の「Oracle Workflow ホーム・ページへのアクセス」を参照してください。



- 「イベント・サブスクリプション」ページに既存のサブスクリプションのリストが表示されます。サブスクライバ・システムおよびトリガー・イベント別にグループ化されています。「イベント・サブスクリプション」ページには、各サブスクリプションのソース・タイプ、送信エージェント、宛先エージェント、関数、ワークフローおよびステータスの一覧が表示されます。

「サブスクリプションの追加」ボタンを選択して「サブスクリプションの編集」ページを開きます。

注意： カスタマイズ・レベルが「制限」のシードされたサブスクリプションの場合は、サブスクリプションのステータスのみを更新できます。カスタマイズ・レベルが「コア」のシードされたサブスクリプションの場合、サブスクリプション定義は表示できますが、プロパティは更新できません。

サブスクリプションの編集 - Microsoft Internet Explorer

ファイル(F) 編集(E) 表示(V) お気に入りに追加(A) ツール(T) ヘルプ(H) リンク(R) 印刷(P)

サブスクリプションの編集 ?

ORACLE

サブスクライバ
システム

トリガー条件
ソース・タイプ ローカル
イベント・フィルタ
ソース・エージェント

実行制御
フェーズ
状態 使用可能
ルール・データ キー

処理
ルール機能
ワークフロー項目タイプ
ワークフロー・プロセス名
送信エージェント
宛先エージェント
優先度 標準
パラメータ

文書
所有者名
所有者タグ

- 「サブスクライバ」領域に、サブスクリプションを実行するシステムを入力します。「システム」フィールドの上矢印のアイコンをクリックすると、選択可能なシステムのリストが表示されます。8-25 ページの「値リストの使用」を参照してください。
- 「トリガー条件」領域の「ソース・タイプ」フィールドに、サブスクリプションを適用するソース・システムのタイプを指定します。

- ローカル：サブスクリプションは、サブスクライバ・システムで発生したイベントにのみ適用されます。
- 外部：サブスクリプションは、サブスクライバ・システム上のインバウンド・エージェントが受信したイベントにのみ適用されます。

注意： サブスクライバ・システム上のインバウンド・エージェントが受信したイベント・メッセージは、送信元エージェントがリモート・システム上に存在するかローカル・システム上に存在するかにかかわらず、すべて外部ソースと見なされます。

- エラー：サブスクリプションは、WF_ERROR キューからデキューされたエラー・イベントにのみ適用されます。
5. サブスクリプションを適用するイベントを「イベント・フィルタ」フィールドに入力します。個々のイベントまたはイベント・グループを指定できます。「イベント・フィルタ」フィールドの上矢印のアイコンをクリックすると、選択可能なイベントのリストが表示されます。8-25 ページの「[値リストの使用](#)」を参照してください。
 6. オプションで、サブスクリプションを適用するソース・エージェントを入力します。ソース・エージェントを指定すると、サブスクリプションは、トリガー・イベントがそのエージェントから着信した場合にのみ実行されます。「ソース・エージェント」フィールドの上矢印のアイコンをクリックすると、選択可能なエージェントのリストが表示されます。8-25 ページの「[値リストの使用](#)」を参照してください。

注意： ほとんどの場合、「ソース・エージェント」フィールドは空白のままにします。

7. 同じイベントに複数のサブスクリプションを適用する場合、その実行順序を指定するときは、「実行制御」領域にサブスクリプションのフェーズ番号を入力します。フェーズ番号を使用して、サブスクリプションがすぐに実行されるか遅延されるかを制御することもできます。
8. サブスクリプションのステータスとして「使用可能」または「使用不能」を選択します。サブスクリプションを使用不能にしても、そのサブスクリプションは「イベント・サブスクリプション」リストに表示され、参照できます。ただし、このサブスクリプションを使用して、イベントに応答することはできません。
9. 「ルール・データ」フィールドで、サブスクリプションに必要なイベント情報を指定します。
 - キー：サブスクリプションにはイベント・キーのみが必要です。
 - メッセージ：サブスクリプションにはすべてのイベント・データが必要です。

10. 「処理」領域で、トリガー・イベントの発生時に実行するサブスクリプション処理を定義します。サブスクリプションには、次の処理を指定できます。
- イベント・メッセージに対する関数の実行
 - ワークフロー・プロセスへのイベント・メッセージの送信
 - エージェントへのイベント・メッセージの送信
11. イベント・メッセージに対して関数を実行する場合は、実行するルール関数を入力します。ルール関数は、標準 API に従って定義する必要があります。6-29 ページの「[イベント・サブスクリプションのルール関数の標準 API](#)」を参照してください。

ルール関数を指定しない場合は、Oracle Workflow によってデフォルトのルール関数が実行され、イベント・メッセージは指定したワークフロー・プロセスおよびエージェントに送信されます。

注意： デフォルト以外のルール関数を入力した場合、指定されたワークフローおよびエージェントにイベント・メッセージは自動的に送信されません。この場合、送信処理をカスタム・ルール関数に明示的に組み込む必要があります。ただし、「処理」領域には、参照する関数に対するワークフロー情報およびエージェント情報を入力できます。

12. イベント・メッセージをワークフロー・プロセスに送信する場合は、プロセスの項目タイプを「ワークフロー項目タイプ」フィールドに、プロセスの名前を「ワークフロー・プロセス名」フィールドに入力します。各フィールドの上矢印のアイコンをクリックすると、選択可能な値のリストが表示されます。8-25 ページの「[値リストの使用](#)」を参照してください。

注意： 「ワークフロー・プロセス名」フィールドの値リストには、指定した項目タイプの実行可能プロセスのみが表示されます。

13. イベントをエージェントに送信する場合は、アウトバウンド・メッセージを送信する送信エージェントまたはインバウンド・メッセージを受信する宛先エージェント、あるいはその両方のエージェントを入力します。各フィールドの上矢印のアイコンをクリックすると、選択可能な値のリストが表示されます。8-25 ページの「[値リストの使用](#)」を参照してください。
- 宛先エージェントと送信エージェントの両方を指定した場合、イベント・メッセージは送信エージェントのキューに格納され、宛先エージェントに伝播されます。
 - 送信エージェントを指定しないで宛先エージェントを指定した場合は、サブスクライバ・システム上のアウトバウンド・エージェントのうち、キュー・タイプが宛先エージェントと一致するものが選択されます。イベント・メッセージは、このアウ

トバウンド・エージェントのキューに格納され、宛先エージェントに伝播されま
す。

- 宛先エージェントを指定しないで送信エージェントを指定した場合、イベント・メッセージは送信エージェントのキューに格納されますが、受信者は指定されていません。送信エージェントは、サブスクライバ・リストが定義された複数コンシューマ・キューまたは単一コンシューマ・キューを使用する必要があります。

注意： 送信エージェントは、サブスクライバ・システム上に存在する必要があります。「送信エージェント」フィールドの値リストには、送信方向のエージェントのみが表示されます。

「宛先エージェント」フィールドの値リストには、インバウンド方向のエージェントのみが表示されます。

14. イベント・メッセージをエージェントに送信する場合は、受信者がメッセージをデキューするときの優先順位として、「標準」、「高」または「低」を選択します。
15. オプションで、「パラメータ」フィールドにルール関数の追加パラメータを入力します。複数のパラメータを区切るには、スペースを使用します。各パラメータの名前および値は、次の書式で指定します。

```
<name1>=<value1> <name2>=<value2> ... <nameN>=<valueN>
```

注意： イベント・メッセージをワークフロー・プロセスに送信するときに、追加パラメータを指定すると、ワークフロー・プロセスの項目属性として設定されます。追加パラメータをサブスクリプションの「パラメータ」フィールドに入力し、サブスクリプションのルール関数で `WF_RULE.SetParametersIntoParameterList()` を使用すれば、指定したサブスクリプション・パラメータをイベント・メッセージのパラメータ・リストに設定することができます。ワークフロー・プロセスにイベントが着信すると、それらのイベント・パラメータはワークフロー・プロセスの項目属性として設定されます。『Oracle Workflow API リファレンス』の「SetParametersIntoParameterList」を参照してください。

16. 「文書」領域を利用して、サブスクリプションを所有するプログラムまたはアプリケーションを識別することができます。プログラム名を「所有者名」フィールドに、プログラム ID を「所有者タグ」フィールドに入力してください。「サブスクリプションの編集」ページでサブスクリプションを手動で定義する場合は、「所有者名」および「所有者タグ」を入力する必要はありません。ただし、サブスクリプション定義を自動的に作成するプログラムを使用する場合、これらのフィールドにはそのプログラムで設定された所有者情報が表示されます。この情報は、「サブスクリプションの編集」ページから必要に応じて手動で更新できます。

17. サブスクリプションのカスタマイズ・レベルを確認することができます。
 - コア： サブスクリプション定義を変更することはできません。このレベルは、Oracle Applications にシードされたサブスクリプションに対してのみ使用されます。
 - 制限： サブスクリプションのステータスを「使用可能」または「使用不能」に更新できますが、サブスクリプション定義を変更することはできません。このレベルは、Oracle Applications にシードされたサブスクリプションに対してのみ使用されます。
 - ユーザー： サブスクリプション定義内のすべてのプロパティを変更することができます。このレベルは、ユーザーが定義したサブスクリプションに自動的に設定されます。
18. オプションで、サブスクリプションの説明を入力します。
19. 「送信」 ボタンを選択してサブスクリプションを保存し、「イベント・サブスクリプション」 ページに戻ります。「イベント・サブスクリプション」 ページに、更新されたサブスクリプションのリストが表示されます。

「取消」 ボタンを選択し、サブスクリプションを保存しないで「イベント・サブスクリプション」 ページに戻ることもできます。

▶ イベント・サブスクリプションの検索

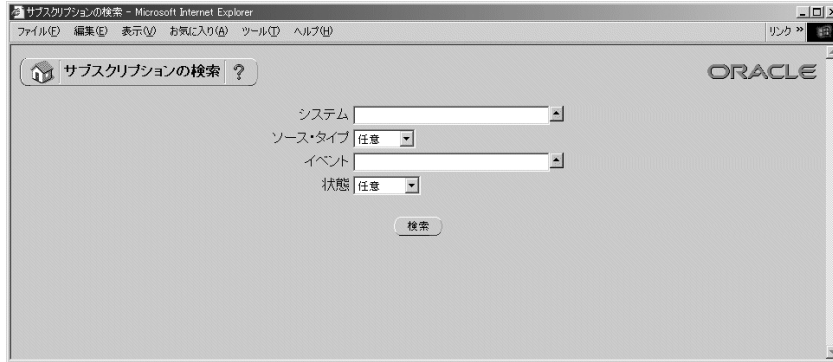
1. Web ブラウザを使用して、次の URL に接続します。

```
<webagent>/wf_event_html.findsubscription
```

<webagent> は、Web サーバーで Oracle Workflow 用に構成された Web エージェントのベース URL に置き換えてください。『Oracle Workflow 管理者ガイド』の「グローバル・ユーザー設定の設定」を参照してください。

注意： この URL はセキュリティ画面にアクセスするため、現行 Web セッションで有効なユーザーとしてログオンしていない場合は、ページが表示される前に有効なユーザーとしてのログオンを求めるプロンプトが表示されます。「イベント・マネージャ」 Web ページにアクセスするには、ワークフロー管理者権限が必要です。

注意： 「サブスクリプションの検索」 Web ページには、Oracle Workflow ホーム・ページからもアクセスできます。『Oracle Workflow 管理者ガイド』の「Oracle Workflow ホーム・ページへのアクセス」を参照してください。



2. 「サブスクリプションの検索」ページが表示されます。「サブスクリプションの検索」ページでは、検索基準を入力して特定のイベント・サブスクリプションを検索できます。検索基準として、次の項目を使用できます。
 - システム：サブスクリプションを表示するシステムを選択します。このフィールドの上矢印のアイコンをクリックすると、選択可能なシステムのリストが表示されます。8-25 ページの「[値リストの使用](#)」を参照してください。
 - ソース・タイプ：サブスクリプションを表示するソース・システムのタイプとして、「ローカル」、「外部」または「エラー」を選択します。すべてのタイプのソース・システムについてサブスクリプションを表示する場合は、「任意」を選択します。
 - イベント：サブスクリプションを表示するイベントを選択します。このフィールドの上矢印のアイコンをクリックすると、選択可能なイベントのリストが表示されます。8-25 ページの「[値リストの使用](#)」を参照してください。
 - ステータス：表示するサブスクリプションのステータスとして、「使用可能」または「使用不能」を選択します。すべてのステータスのサブスクリプションを表示する場合は、「任意」を選択します。
3. 「検索」ボタンを選択します。「イベント・サブスクリプション」ページに、検索基準と一致したサブスクリプションのリストが表示されます。

検索基準にシステムまたはイベントを含めた場合は、「サブスクリプションの追加」ボタンを選択すると、「サブスクリプションの編集」ページが表示されます。指定したシステム情報およびイベント情報が、「システム」および「イベント・フィルタ」フィールドにそれぞれ自動的に入力されています。検索基準と一致する既存のサブスクリプションが見つからなかった場合は、この方法で新しいサブスクリプションの定義を開始できます。

▶ イベント・サブスクリプションの更新または削除

1. 「イベント・サブスクリプション」ページで、更新または削除するイベントを選択します。「サブスクリプションの検索」ページを使用して目的のサブスクリプションを検索し、「イベント・サブスクリプション」ページに表示することができます。8-57 ページの「イベント・サブスクリプションの検索」を参照してください。



2. サブスクリプションを更新するには、そのサブスクリプションの「編集」列にある鉛筆アイコンを選択します。「サブスクリプションの編集」ページが表示されます。サブスクリプション定義に変更を加え、変更内容を保存します。8-52 ページの「イベント・サブスクリプションの定義」を参照してください。

注意： カスタマイズ・レベルが「制限」のシードされたサブスクリプションの場合は、サブスクリプション・ステータスのみを更新できます。カスタマイズ・レベルが「コア」のシードされたサブスクリプションの場合、サブスクリプション定義は表示できますが、プロパティは更新できません。

3. サブスクリプションを削除するには、そのサブスクリプションの「削除」列にあるごみ箱アイコンを選択して、表示される確認ウィンドウで「OK」を選択します。確認ウィンドウで「取消」ボタンを選択し、サブスクリプションを削除しないで「イベント・サブスクリプション」ページに戻ることもできます。

注意： カスタマイズ・レベルが「コア」または「制限」の Oracle Applications にシードされたサブスクリプションは、削除することができません。

イベントの呼出し

イベントの呼出しは、アプリケーションまたはワークフローを介して行う以外に、「イベントの呼出し」 Web ページを使用してテストのために手動で呼び出すことができます。後者の場合は、パラメータを追加することはできません。イベントを呼び出すと、イベント・マネージャは、ソース・タイプを「ローカル」にして、そのイベントの有効なサブスクリプションをローカル・システム単位に検索および実行します。また、ソース・タイプを「ローカル」にして、Any イベントの有効なサブスクリプションをローカル・システム単位に検索および実行します。

▶ イベントの呼出し

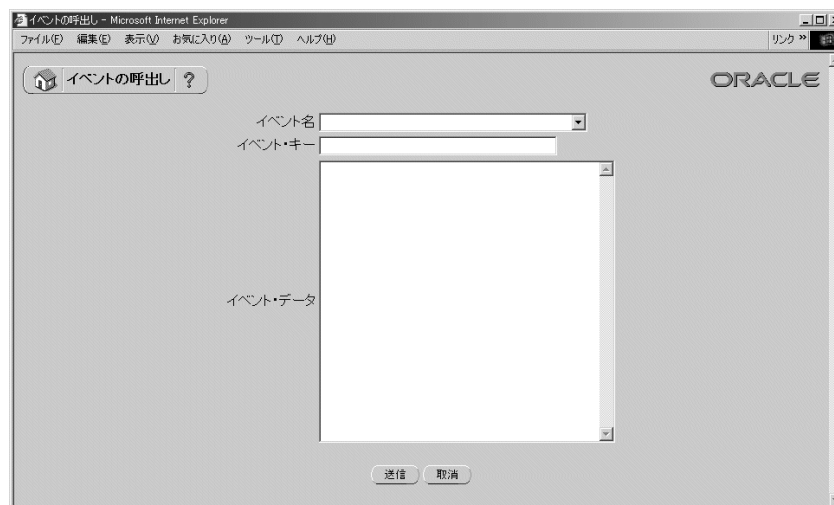
1. Web ブラウザを使用して、次の URL に接続します。

```
<webagent>/wf_event_html.entereventdetails
```

<webagent> は、Web サーバーで Oracle Workflow 用に構成された Web エージェントのベース URL に置き換えてください。『Oracle Workflow 管理者ガイド』の「グローバル・ユーザー設定の設定」を参照してください。

注意： この URL はセキュリティ画面にアクセスするため、現行 Web セッションで有効なユーザーとしてログオンしていない場合は、ページが表示される前に有効なユーザーとしてのログオンを求めるプロンプトが表示されます。「イベント・マネージャ」 Web ページにアクセスするには、ワークフロー管理者権限が必要です。

注意： 「イベントの呼出し」 Web ページには、Oracle Workflow ホーム・ページからもアクセスできます。『Oracle Workflow 管理者ガイド』の「Oracle Workflow ホーム・ページへのアクセス」を参照してください。



2. 「イベントの呼出し」ページが表示されます。
3. 「イベント名」フィールドで、呼び出すイベントを選択します。
4. イベントのインスタンスを一意に識別するイベント・キーを入力します。
5. オプションで、イベントの内容を説明するイベント・データを入力します。

注意：「イベント・データ」フィールドに入力可能な最大データ長は、32KBです。イベント・データが32KBを超える場合は、「イベント・データ」フィールドにデータを直接入力せず、イベント定義にジェネレート関数を割り当ててイベント・データを生成する必要があります。8-7ページの「[イベントの定義](#)」を参照してください。

WF_EVENT.Raise API を使用してイベントを呼び出すこともできます。この方法を使用すると、最大 4GB のデータを格納する CLOB としてイベント・データを渡すことができます。『Oracle Workflow API リファレンス』の「Raise」を参照してください。

6. 「送信」ボタンを選択して、イベントをイベント・マネージャに呼び出します。「取消」ボタンを選択し、イベントを呼び出さずに Oracle Workflow ホーム・ページに戻ることができます。

「送信」ボタンを選択すると、イベントが呼び出され、確認メッセージにイベント名およびイベント・キーが表示されます。「OK」ボタンを選択して「イベントの呼出し」ページに戻ります。

システムのサインアップ

システム間でビジネス・イベントを送信するには、イベント・メッセージの受信者として宛先システムをソース・システムにサインアップする必要があります。システムのサインアップとは、宛先システムおよびそのインバウンド・エージェントをソース・システムのイベント・マネージャに定義して、ソース・システムのイベント・メッセージを宛先エージェントに送信できるようにすることです。

通常、宛先システムおよびソース・システムを相互にサインアップして、システム間でメッセージを送受信できるようにする必要があります。

宛先システムをサインアップして、ソース・システムのイベント・メッセージを受信するには、次の手順を実行します。

1. ローカル・システムおよびインバウンド・エージェントの定義を宛先システムから取得します。これらの情報は、システム識別子情報となります。宛先システムの「システム識別子」Web ページを使用して、システム識別子情報で構成される XML 文書を作成できます。8-63 ページの「[システム識別子情報の取得](#)」を参照してください。

注意： 宛先システムにインストールされている Oracle Workflow へのアクセス権がない場合は、宛先システムのワークフロー管理者にこの手順の実行を依頼してください。

2. 宛先システムの識別子情報をソース・システムのイベント・マネージャに追加します。ソース・システムの「システムのサインアップ」Web ページを使用して情報を追加することができます。システムのサインアップ・イベントを XML 文書とともに宛先システムからイベント・データとして呼び出します。ソース・システムでシステムのサインアップ・イベントを呼び出すと、事前定義済みのサブスクリプションが Oracle Workflow によって実行され、システム識別子情報がソース・システムのイベント・マネージャに追加されます。8-63 ページの「[システムのサインアップ](#)」を参照してください。

注意： ソース・システムにインストールされている Oracle Workflow へのアクセス権がない場合は、ソース・システムのワークフロー管理者にこの手順を実行するよう依頼してください。

▶ システム識別子情報の取得

1. Web ブラウザを使用して、宛先システムとしてサインアップするシステムの次の URL に接続します。

```
<webagent>/wf_event_html.getsystemidentifier
```

<webagent> は、Web サーバーで Oracle Workflow 用に構成された Web エージェントのベース URL に置き換えてください。『Oracle Workflow 管理者ガイド』の「グローバル・ユーザー設定の設定」を参照してください。

注意： この URL はセキュリティが適用されるページにアクセスするため、現行 Web セッションで有効なユーザーとしてログオンしていない場合は、ページが表示される前に有効なユーザーとしてのログオンを求めるプロンプトが表示されます。「イベント・マネージャ」Web ページにアクセスするには、ワークフロー管理者権限が必要です。

宛先システムにインストールされている Oracle Workflow へのアクセス権がない場合は、宛先システムのワークフロー管理者にこの手順の実行を依頼してください。

注意： 「システム識別子」Web ページには、Oracle Workflow ホーム・ページからもアクセスできます。『Oracle Workflow 管理者ガイド』の「Oracle Workflow ホーム・ページへのアクセス」を参照してください。

2. Oracle Workflow によって、ローカル・システムおよびそのインバウンド・エージェントの定義を含む、システム識別子 XML 文書が作成されます。この文書をテキスト・ファイルとして保存します。このシステムをソース・システムにサインアップするとき、保存した文書をコピーすれば、システムのサインアップ・イベントのイベント・データとして入力できます。8-63 ページの「[システムのサインアップ](#)」を参照してください。

▶ システムのサインアップ

1. Web ブラウザを使用して、宛先システムをサインアップするソース・システムの次の URL に接続します。

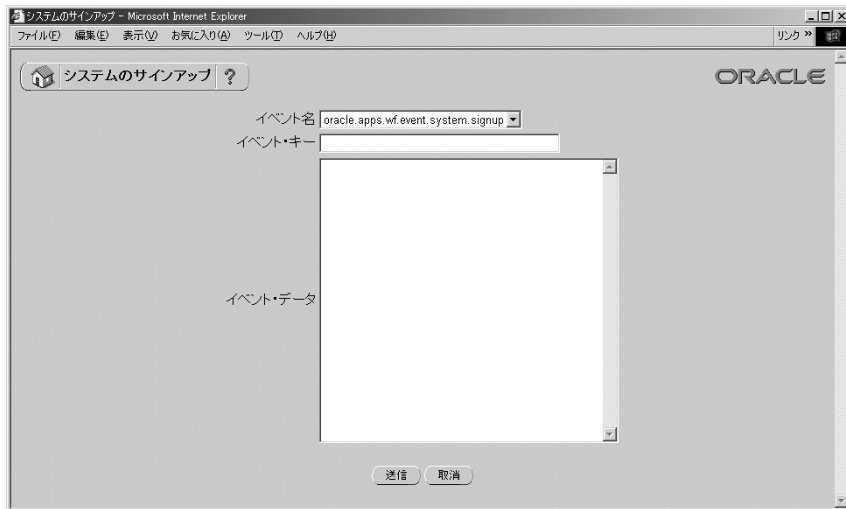
```
<webagent>/wf_event_html.entereventdetails?p_event_name=  
oracle.apps.wf.event.system.signup
```

<webagent> は、Web サーバーで Oracle Workflow 用に構成された Web エージェントのベース URL に置き換えてください。『Oracle Workflow 管理者ガイド』の「グローバル・ユーザー設定の設定」を参照してください。

注意： この URL はセキュリティが適用されるページにアクセスするため、現行 Web セッションで有効なユーザーとしてログオンしていない場合は、ページが表示される前に有効なユーザーとしてのログオンを求めるプロンプトが表示されます。「イベント・マネージャ」 Web ページにアクセスするには、ワークフロー管理者権限が必要です。

ソース・システムにインストールされている Oracle Workflow へのアクセス権がない場合は、ソース・システムのワークフロー管理者にこの手順を実行するよう依頼してください。

注意： 「システムのサインアップ」 Web ページには、Oracle Workflow ホーム・ページからもアクセスできます。『Oracle Workflow 管理者ガイド』の「Oracle Workflow ホーム・ページへのアクセス」を参照してください。



2. 「システムのサインアップ」ページに、システムのサインアップ・イベントの内部名が表示されます。
3. イベントのインスタンスを一意に識別するイベント・キーを入力します。
4. 宛先システムの識別子情報を含む XML 文書を、「イベント・データ」フィールドにコピーします。8-63 ページの「システム識別子情報の取得」を参照してください。

5. 「送信」ボタンを選択して、システムのサインアップ・イベントをイベント・マネージャに呼び出します。確認メッセージが表示されます。システムのサインアップ・イベントが呼び出されると、事前定義済のサブスクリプションが Oracle Workflow によって実行され、システム識別子情報がイベント・データからイベント・マネージャに追加されます。9-14 ページの「システムのサインアップ・イベント」を参照してください。

「取消」ボタンを選択し、システムのサインアップ・イベントを呼び出さないで Oracle Workflow ホーム・ページに戻ることもできます。

システムの同期

システムの同期とは、ソース・システムに定義されているすべてのイベント・マネージャ・オブジェクトを、ターゲット・システムにレプリケートすることです。イベント・システムの同期イベントを使用して、ソース・システムおよびターゲット・システムを同期させることができます。

▶ システムの同期

1. ソース・システムおよびターゲット・システムを相互にサインアップします。8-63 ページの「システムのサインアップ」を参照してください。
2. ソース・システム上で、「ローカル」ソース・タイプを持つ「シード・イベント・グループ」への事前定義済サブスクリプションを、次のように変更します。
 - イベント・メッセージを受信するターゲット・システムに対してインバウンド・エージェントを指定するか、イベント・メッセージをターゲット・システムに送信するワークフロー・プロセスを指定します。

注意： イベント・メッセージを複数のターゲット・システムに送信する場合は、追加のサブスクリプションを定義するか、イベント・メッセージを複数のシステムに送信するワークフロー・プロセスを指定します。

- サブスクリプションを使用可能にします。

注意： ソース・システムにインストールされている Oracle Workflow へのアクセス権がない場合は、ソース・システムのワークフロー管理者にこの手順を実行するよう依頼してください。

3. ターゲット・システム上で、「外部」ソース・タイプを持つ「シード・イベント・グループ」への事前定義済サブスクリプションを、使用可能にします。

注意： ターゲット・システムにインストールされている Oracle Workflow へのアクセス権がない場合は、ターゲット・システムのワークフロー管理者にこの手順の実行を依頼してください。

4. ソース・システム上で、「イベントの呼出し」ページを使用してイベント・システムの同期イベント (oracle.apps.wf.event.all.sync) を呼び出します。一意のイベント・キーを入力し、「イベント・データ」フィールドは空白のままにします。8-60 ページの「[イベントの呼出し](#)」を参照してください。

注意： ソース・システムにインストールされている Oracle Workflow へのアクセス権がない場合は、ソース・システムのワークフロー管理者にこの手順を実行するよう依頼してください。

ソース・システム上でイベント・システムの同期イベントが呼び出されると、「ローカル」ソース・タイプを持つ「シード・イベント・グループ」へのサブスクリプションがトリガーされます。イベント・マネージャがイベント・メッセージを生成します。このメッセージには、ローカル・システム上のイベント、イベント・グループ、システムおよびサブスクリプションを含む、すべてのイベント・マネージャ・オブジェクトの定義が含まれます。イベント・メッセージは、ターゲット・システム上の特定のインバウンド・エージェント、またはイベント・メッセージをターゲット・システムに送信する特定のワークフロー・プロセスに送信されます。

ターゲット・システムにイベント・システムの同期イベントが着信すると、「外部」ソース・タイプを持つ「シード・イベント・グループ」へのサブスクリプションがトリガーされます。イベント・メッセージのオブジェクト定義がターゲット・システムのイベント・マネージャにロードされます。必要に応じて、新しい定義が作成されたり、既存の定義が更新されます。

自動レプリケーション

手順 2 および 3 で事前定義済のサブスクリプションを使用可能にすると、これらのサブスクリプションは、ソース・システムのイベント・マネージャのオブジェクト定義に加えられた変更をレプリケートします。イベント、イベント・グループ・メンバー、システム、エージェントまたはサブスクリプションの作成、更新または削除を行うたびに、対応する事前定義済のイベントが **Oracle Workflow** によって呼び出されます。これらのイベントにより、ソース・システム上の「シード・イベント・グループ」への「ローカル」サブスクリプションがトリガーされ、オブジェクト定義データがターゲット・システムに送信されます。ターゲット・システム上の「シード・イベント・グループ」への「外部」サブスクリプションは、そのデータを受信して、イベント・マネージャのオブジェクト定義を追加、更新または削除します。

ソース・システムの変更をターゲット・システムに自動レプリケートする処理を停止する場合は、システムの同期の完了後にサブスクリプションを使用不可にするか、変更に対応する事前定義済のイベントを使用不可にします。

マスター/コピー・システム

必要に応じて、特定のシステムをマスター・システムとして設定できます。このようにすると、マスター・システムのイベント・マネージャのオブジェクト定義を関連付けられたコピー・システムにレプリケートすることができます。ただし、コピー・システムのオブジェクト定義への変更は、マスター・システムにレプリケートされません。マスター/コピーのレプリケーションをセットアップするには、ターゲット・コピー・システムをソース・マスター・システムに同期させる手順を通常どおりに実行します。次に、オブジェクト定義がコピー・システムから送信されないようにするために、コピー・システム上の「シード・イベント・グループ」への「ローカル」サブスクリプションを使用不可にします。オブジェクト定義がマスター・システムに着信しないようにするために、マスター・システム上の「シード・イベント・グループ」への「外部」サブスクリプションも使用不可にします。

関連項目：

9-2 ページ [「事前定義済ワークフロー・イベント」](#)

9-9 ページ [「イベント・システムの同期イベント」](#)

9-10 ページ [「シード・イベント・グループ」](#)

8-52 ページ [「イベント・サブスクリプションの定義」](#)

ローカル・キューの確認

「イベント・システム・ローカル・キュー」ページを使用して、ビジネス・イベント・システムで使用されているローカル・キューおよびそれらのキューに現在保持されているメッセージを確認できます。

注意： Oracle Workflow のスタンドアロン版では、「セットアップのチェック」ページを使用して、ローカル・キューの設定を確認することもできます。『Oracle Workflow 管理者ガイド』の「ビジネス・イベント・システムのセットアップのチェック」を参照してください。

注意： Oracle Applications Manager を実装している環境で Oracle Applications に組み込まれている Workflow を使用している場合は、Oracle Workflow Manager を使用してローカル・キューに保持されているメッセージのステータスを確認することができます。詳細は、Oracle Applications Manager のオンライン・ヘルプを参照してください。

スタンドアロン版の Oracle Workflow を使用している場合は、Oracle Enterprise Manager から利用可能なスタンドアロン版の Oracle Workflow Manager コンポーネントを使用して、ローカル・キューに保持されているメッセージのステータスを確認できます。詳細は、Oracle Workflow Manager のオンライン・ヘルプを参照してください。

▶ ローカル・キューの確認

1. Web ブラウザを使用して、次の URL に接続します。

```
<webagent>/wf_event_html.eventqueuedisplay
```

<webagent> は、Web サーバーで Oracle Workflow 用に構成された Web エージェントのベース URL に置き換えてください。『Oracle Workflow 管理者ガイド』の「グローバル・ユーザー設定の設定」を参照してください。

注意： この URL はセキュリティが適用されるページにアクセスするため、現行 Web セッションで有効なユーザーとしてログオンしていない場合は、ページが表示される前に有効なユーザーとしてのログオンを求めるプロンプトが表示されます。「イベント・マネージャ」Web ページにアクセスするには、ワークフロー管理者権限が必要です。

注意： 「イベント・システム・ローカル・キュー」Web ページには、Oracle Workflow ホーム・ページからもアクセスできます。『Oracle Workflow 管理者ガイド』の「Oracle Workflow ホーム・ページへのアクセス」を参照してください。

Protocol	Agent Name	Direction	Message Count	View Detail
SQLNET	WF_DEFERRED	IN	0	
SQLNET	WF_ERROR	IN	0	
SQLNET	WF_IN	IN	0	
SQLNET	WF_OUT	OUT	0	
SQLNET	WF_SMTP_O_1_QUEUE	IN	6	Not Available

2. 「イベント・システム・ローカル・キュー」ページに、ビジネス・イベント・システムで使用されているローカル・キューのリストが表示されます。「イベント・システム・ローカル・キュー」ページには、キューに関連付けられているエージェントのプロトコル

ルおよび名前、エージェントが使用されている通信（インバウンドまたはアウトバウンド）、キューに現在保持されているメッセージの数の一覧がキューごとに表示されます。

3. キューに格納されているメッセージの詳細のうち、標準 WF_EVENT_T データ型をペイロード・タイプとして使用するものを確認できます。キューに格納されているメッセージの詳細を確認するには、そのキューの「詳細の表示」列にある懐中電灯アイコンを選択します。

注意： 標準 WF_EVENT_T データ型をペイロード・タイプとして使用しないキューに関するメッセージの詳細は確認できません。



4. 「標準イベント・キュー・メッセージの検索」ページに、選択したキューの名前が表示されます。「標準イベント・キュー・メッセージの検索」ページでは、検索基準を入力して特定のイベントを検索できます。検索基準として、次の項目を使用できます。
 - イベント名： 表示するイベント・メッセージの内部イベント名を入力します。
 - イベント・キー： 表示するイベント・メッセージのイベント・キーを入力します。
 - ステータス： 表示するイベント・メッセージのステータスとして、「準備完了」、「待機」、「処理済」または「期限満了」を選択します。すべてのステータスのメッセージを表示するときは、「任意」を選択します。
5. 「実行」を選択します。「ローカル・キュー・メッセージ」ページに、選択したキュー内で検索基準と一致したイベント・メッセージの一覧が表示されます。「ローカル・キュー・メッセージ」ページには、イベント名、イベント・キー、関連 ID、イベント・パラメータ、メッセージを送信した「送信元システム」、メッセージを受信した「宛先システム」、送信日、エラー・メッセージ、エラー・スタックおよびメッセージのステータスの一覧が、メッセージごとに表示されます。

「ローカル・キュー・メッセージ」ページに表示されるリストには、選択したキューに関連付けられている例外キューのメッセージもすべて含まれます。AQ がなんらかの理由でメッセージを取得または処理できない場合は、ユーザー・キューから関連付けられた例外キューへメッセージが転送されます。詳細は、『Oracle9i アプリケーション開発者ガイド-アドバンスト・キューイング』の「例外処理」または『Oracle Streams アドバンスト・キューイング・ユーザーズ・ガイドおよびリファレンス』の「Oracle Streams AQ 例外処理」を参照してください。

注意： 各キュー表には、そのキュー表内のすべてのユーザー・キューに共有されるデフォルトの例外キューが 1 つ含まれています。特定のキューにあるメッセージを検索する場合、「ローカル・キュー・メッセージ」ページには、元のユーザー・キューとは関係なく、関連付けられた例外キューにあるすべてのメッセージが含まれます。したがって、同じキュー表内に複数のユーザー・キューを作成すると、「ローカル・キュー・メッセージ」ページには、選択したキュー以外のキューからの例外メッセージも表示される可能性があります。

Event Name	Event Key	Correlation ID	Parameters	From System	To System	Send Date	Error Message	Error Stack	Message State	Event Data (XML Format)
oracle.apps.ecx.bes.qa.t5	5734	ecx_generate_5a		WF_OUT@IV51	WF_IN@IV51	09-OCT-2001 11:59:56			Ready	
oracle.apps.ecx.bes.qa.t5	5734	ecx_gena2a_5c		WF_OUT@IV51	WF_IN@IV51	09-OCT-2001 12:00:02			Ready	
oracle.apps.ecx.bes.qa.t5	5734	ecx_gena2a_5d		WF_OUT@IV51	WF_IN@IV51	09-OCT-2001 12:00:05			Ready	
oracle.apps.ecx.bes.qa.t5	5734	ecx_gena2a_5f		WF_OUT@IV51	WF_IN@IV51	09-OCT-2001 12:00:11			Ready	
oracle.apps.ecx.bes.qa.t5	5734	ecx_gena2a_5g		WF_OUT@IV51	WF_IN@IV51	09-OCT-2001 12:00:14			Ready	

- メッセージのイベント・データを XML 文書で確認するには、そのメッセージの「イベント・データ (XML 形式)」列にある懐中電灯アイコンを選択します。
- メッセージのイベント・データをテキスト文書で確認するには、そのメッセージの「イベント・データ (Text 形式)」列にある懐中電灯アイコンを選択します。

関連項目：

8-29 ページ「標準エージェント」

『Oracle Workflow 管理者ガイド』の「キューの設定」

Oracle Applications のためのイベント・マネージャ

注意： Oracle Workflow の Web ページは、Oracle Applications Framework のユーザー・インタフェース形式に変換されます。Oracle Workflow のバージョンおよび適用したパッチに応じて、新しい形式または以前の形式でも Oracle Workflow の Web ページを表示できます。新しい「イベント・マネージャ」ページは、現在、Oracle Applications に組み込まれている Workflow で利用できます。

Oracle Workflow のイベント・マネージャには、アプリケーションで発生する興味あるビジネス・イベント、イベントを交換するシステム、それらのシステム内の名前付き通信エージェント、およびイベントと特定のシステムの関連付けを示すサブスクリプションを登録できます。またイベント・マネージャは、イベント発生時にサブスクリプションを処理します。

「イベント・マネージャ」ページでビジネス・イベント・システムのオブジェクトを管理するには、ワークフロー管理者権限が必要です。ワークフロー管理者権限を持っている場合は、イベント、サブスクリプション、システムおよびエージェントを、作成、更新および削除することができます。また、テスト・イベントを手動で呼び出して、ビジネス・イベントをテストすることもできます。管理者権限を持っていない場合は、「イベント・マネージャ」ページでビジネス・イベント・システムのオブジェクトを表示することはできますが、変更することはできません。ワークフロー管理者権限は、「グローバル・ワークフロー設定」ページで割り当てられます。『Oracle Workflow 管理者ガイド』の「グローバル・ユーザー設定の設定」を参照してください。

Workflow XML Loader を使用して、フラット・ファイルとデータベース間でビジネス・イベント・システムオブジェクトの XML 定義をアップロードおよびダウンロードすることができます。『Oracle Workflow 管理者ガイド』の「Workflow XML Loader の使用」を参照してください。

ローカル・システム上のアプリケーションでイベントが発生する場合は、イベントの特定のインスタンスを一意に識別するイベント・キーを割り当ててから、イベント・マネージャに送信する必要があります。

イベントは次の方法で呼び出すことができます。

- イベントを呼び出すアプリケーションから WF_EVENT.Raise() API を使用して、イベントを呼び出します。『Oracle Workflow API リファレンス』の「Raise」を参照してください。
- ワークフロー・プロセスから Raise イベント・アクティビティを使用して、イベントを呼び出します。3-56 ページの「イベント・アクティビティ」を参照してください。
- 「ビジネス・イベントのテスト」ページから、イベントを手動で呼び出します。8-84 ページの「テスト・イベントの呼出し」を参照してください。

また、イベント・マネージャでは、ローカル・システムまたはリモート・システムから送信されたイベントを受信できます。

ローカル・アプリケーションからイベントが呼び出されたとき、またはローカル・システムあるいは外部システムからイベントが着信したときは、イベント・マネージャがそのイベントへのサブスクリプションを実行します。イベント・マネージャは、サブスクリプションに定義されているアクションに応じて、カスタム・コードのコール、ワークフロー・プロセスへのイベント情報の送信、またはエージェントへのイベント情報の送信を行います。

システム間でイベント・メッセージを通信する場合は、アウトバウンド・メッセージの伝播とインバウンド・メッセージのエージェント・リスナーをスケジュールする必要があります。スケジュールの伝播には Oracle Enterprise Manager を、エージェント・リスナーの実行には Oracle Applications Manager の Workflow Manager コンポーネントを使用することができます。Workflow Manager では、様々なエージェントのキューの詳細情報を確認するだけでなく、各エージェントのイベント・メッセージの配布状況を調べることもできます。詳細は、Oracle Applications Manager のオンライン・ヘルプ、『Oracle9i アプリケーション開発者ガイド-アドバンスト・キューイング』の「Oracle Enterprise Manager のサポート」または『Oracle Streams アドバンスト・キューイング・ユーザーズ・ガイドおよびリファレンス』、『Oracle Enterprise Manager 管理者ガイド』の「分散管理」および『Oracle Workflow 管理者ガイド』の「ビジネス・イベント・システムの設定」を参照してください。

ビジネス・イベント・システムの設定をテストする場合は、「ワークフロー・エージェントの Ping/ 確認」ワークフローを実行します。8-137 ページの「ワークフロー・エージェントの Ping/ 確認」を参照してください。

イベント

ビジネス・イベントは、システム内の他のオブジェクトまたは外部エージェントに関連付けられた、インターネットまたはイントラネット内の状態変化です。たとえば、発注書の作成は、購買アプリケーションにおけるビジネス・イベントの例です。イベントの関連付けは、イベント・マネージャで定義できます。

Oracle Workflow には、ビジネス・イベント・システム内で関連付けられた状態変化に対して、いくつかの事前定義済イベントが用意されています。9-2 ページの「事前定義済ワークフロー・イベント」を参照してください。

イベント・マネージャにイベントを定義する際に、一意の内部名をイベントに割り当てる必要があります。この名前は大文字 / 小文字が区別されます。イベントの内部名を記述するときは、次のように複数の識別子をピリオドで区切った書式を使用することをお勧めします。

```
<company>.<family>.<product>.<component>.<object>.<event>
```

この書式では、定義するイベントを分類階層に体系化することができます。

イベントの内容を説明するために必要な詳細情報は、イベント名およびイベント・キーとともに、イベント・データと呼ばれます。たとえば、発注イベントのイベント・データには、項目番号、説明およびコストなどが含まれます。イベント・データは XML 文書として作成できます。

イベントが発生するアプリケーションには、イベントをイベント・マネージャに呼び出す際に、イベント・データを組み込むことができます。アプリケーションからイベント・データが渡されない場合は、イベントにジェネレート関数を指定して、イベント名、イベント・キーおよびパラメータ・リスト（オプション）から完全なイベント・データを生成する必要があります。ジェネレート関数は、標準 API に従ってください。『Oracle Workflow API リファレンス』の「Raise」および 6-24 ページの「イベント・データ・ジェネレート関数の標準 API」を参照してください。

イベント・マネージャはサブスクリプションを実行する前に、各サブスクリプションにイベント・データが必要かどうかをチェックします。必要なイベント・データが渡されていない場合、イベント・マネージャはそのイベントに対してジェネレート関数をコールし、イベント・データを生成します。イベント・データを必要とするイベントにジェネレート関数が定義されていない場合、イベント名およびイベント・キーを使用してデフォルトのイベント・データが作成されます。

注意： ジェネレート関数の負荷が大きいため、イベントの呼び出し後に制御をできるだけ早くコール側アプリケーションに戻したい場合は、完全なイベント・データを必要とするすべてのサブスクリプションを遅延させることができます。後でそれらのサブスクリプションを実行するまで、イベント・マネージャはジェネレート関数を実行しません。8-94 ページの「遅延サブスクリプション処理」を参照してください。

プログラム名と簡単な識別子をイベントの所有者名および所有者タグとして設定することにより、各イベントを、イベントが属するプログラムまたはアプリケーションと関連付けることができます。たとえば、発注イベントでは所有者は購買アプリケーションです。この識別情報を使用して、所有するイベントをプログラムから検索することができます。

各イベントには、イベント定義を更新できるかどうかを判断するためのカスタマイズ・レベルが割り当てられています。このカスタマイズ・レベルは、Oracle Applications のシード・データを保護し、アップグレード中にユーザーのカスタマイズを保存するために使用されます。各イベントには、次のカスタマイズ・レベルの 1 つが割り当てられています。

- **コア：** イベント定義を変更することはできません。このレベルは、Oracle Applications にシードされたイベントに対してのみ使用されます。
- **制限：** イベント・ステータスを「使用可能」または「使用不能」に更新できますが、イベント定義を変更することはできません。このレベルは、Oracle Applications にシードされたイベントに対してのみ使用されます。
- **ユーザー：** イベント定義内のすべてのプロパティを変更することができます。このレベルは、ユーザーが定義したイベントに自動的に設定されます。

Oracle Applications 製品には、シードされたイベントとサブスクリプションが用意されているものがあります。「インストール済」または「共有」ステータスでライセンスを受けている製品でトリガー・イベントとサブスクリプションの両方が所有されている場合にのみ、サブスクリプションが実行されます。

Oracle Applications のインストールには、ライセンスを受けていない Oracle Applications 製品が所有するシードされたイベントが含まれている可能性があります。そのようなイベントに対しては、「イベントの更新」ページに、そのイベントがライセンスを受けていないことを表す警告が表示されます。Oracle Workflow では、このようなイベントに対してはサブスクリプションが実行されません。また、ライセンスを受けていない製品に所有されるサブスクリプションは、そのようなサブスクリプションのイベントのトリガーに対するライセンスを受けていたとしても、実行されません。

現在ライセンスを受けている製品を確認するために、License Manager AD ユーティリティを使用することができます。製品ライセンスと Workflow BES ライセンスの同期コンカレント・プログラムを使用して、現在ライセンスを受けている製品のステータスに従ってビジネス・イベント・システム内のシードされたイベントおよびサブスクリプションのライセンス・ステータスが確実に更新されるようにすることができます。『Oracle Workflow 管理者ガイド』の「ライセンス・ステータスの同期」を参照してください。

注意： カスタマイズ・レベルを「ユーザー」と定義したイベントは常に、ライセンスを保有しているとみなされます。

イベント・グループ

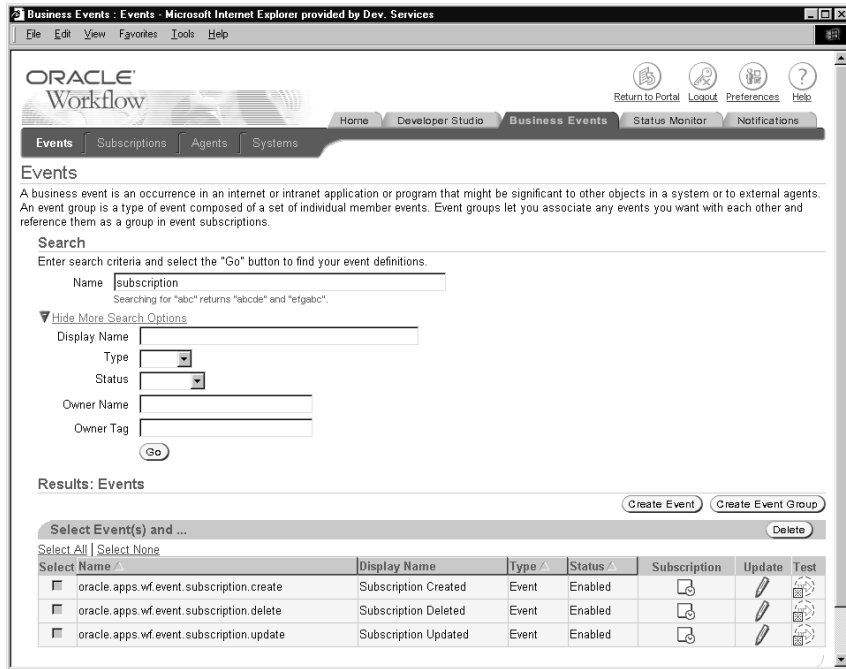
また、イベント・グループを定義して、イベントを相互に関連付け、それらのイベントをイベント・サブスクリプションのグループとして参照することもできます。イベント・グループはイベントの一種で、個別のメンバー・イベントの集合で構成されます。イベント・グループの内部名には、個別のイベント名と同じ書式を使用する必要があります。イベント・グループを定義すると、グループに対してサブスクリプションを登録できます。グループ内のイベントに対して個別にサブスクリプションを作成する必要はありません。サブスクリプションは、グループのいずれかのメンバー・イベントが発生したときに実行されます。

注意： イベント・グループを使用して、イベントを呼び出すことはできません。各イベントは個別に呼び出す必要があります。

▶ イベントの表示と管理

1. システム管理者によって指定された職責およびナビゲーション・パスを使用して、Web ブラウザでイベント・マネージャに移動します。次に水平ナビゲーションの「イベント」を選択します。A-2 ページの「[Oracle Workflow 開発者のナビゲーション・パス](#)」を参照してください。

Oracle Workflow の管理者用 Web ページの「ビジネス・イベント」タブまたは下部にある「ビジネス・イベント」リンクを選択して、イベント・マネージャへ移動することができます。



2. 特定のイベントを検索する場合は、「検索」領域に検索基準を入力します。主要な検索オプションとして、次の項目を使用できます。
 - 名前： 表示するイベントの内部名を入力します。内部名に含まれる数文字を入力しても、イベントを検索することができます。
3. イベントを検索するための追加検索基準として、他の特性を入力することもできます。追加オプションが非表示になっている場合に表示するには、「詳細検索の表示オプション」リンクを選択します。追加オプションが表示されている場合に非表示にする場合は、「詳細検索の非表示オプション」リンクを選択します。追加検索基準として、次の項目を使用できます。
 - 表示名： 表示するイベントの表示名を入力します。
 - タイプ： 表示するイベントのタイプとして「イベント」または「グループ」を選択します。
 - ステータス： 表示するイベントのステータスとして「使用可能」または「使用不能」を選択します。
 - 所有者名： 表示するイベントを所有するプログラムまたはアプリケーションの名前を入力します。

- 所有者タグ：表示するイベントを所有するプログラムまたはアプリケーションのプログラム ID を入力します。
4. 「実行」 ボタンを選択して、検索を実行します。
 5. 「結果」 領域に、検索基準と一致したイベントが表示されます。イベントごとに、リストに表示名、内部名、タイプおよびステータスが表示されます。「名前」、「表示名」、「タイプ」または「ステータス」の列見出しを選択すると、その列を基準にしてリストがソートされます。
 6. イベントに対応するサブスクリプションを表示するには、そのイベントの「サブスクリプション」列にあるサブスクリプション・アイコンを選択します。「予約: イベント」ページに、イベントのサブスクリプションのリストが表示されます。

注意： イベントにサブスクリプションが指定されていない場合、「サブスクリプション」列には空のサブスクリプション・アイコンが表示されません。サブスクリプションが指定されているイベントの場合は、サブスクリプション・アイコンが正式に表示されます。

- ワークフロー管理者権限を持っている場合は、「予約の作成」ボタンを選択して、新しいサブスクリプションをイベントに定義することができます。「イベント予約の作成」ページが表示され、イベント名が「イベント・フィルタ」フィールドに自動的に入力されます。8-99 ページの「[イベント・サブスクリプションの作成または更新](#)」を参照してください。
- ワークフロー管理者権限を持っている場合は、そのサブスクリプションの「更新」列にある更新アイコンを選択して、既存のサブスクリプションを更新することができます。8-99 ページの「[イベント・サブスクリプションの作成または更新](#)」を参照してください。

注意： カスタマイズ・レベルが「制限」のシードされたサブスクリプションの場合は、サブスクリプションのステータスのみを更新できます。カスタマイズ・レベルが「コア」のシードされたサブスクリプションの場合、サブスクリプション定義は表示できますが、プロパティは更新できません。

リンクを選択して、前のページに戻ることもできます。

7. ワークフロー管理者権限を持っている場合は、「イベント」ページの管理アイコンやボタンを使用して、管理操作を実行できます。
 - イベントを更新するには、そのイベントの「更新」列にある更新アイコンを選択します。8-78 ページの「[イベントの作成または更新](#)」および 8-80 ページの「[イベント・グループの作成または更新](#)」を参照してください。

注意： カスタマイズ・レベルが「制限」のシードされたイベントの場合は、イベントのステータスのみを更新できます。カスタマイズ・レベルが「コア」のシードされたイベントの場合、イベント定義は表示できますが、プロパティは更新できません。

- テスト・イベントを手動で呼び出すには、そのイベントの「テスト」列にあるテスト・アイコンを選択します。8-84 ページの「[テスト・イベントの呼出し](#)」を参照してください。

注意： イベント・グループは呼び出せません。各イベントは個別に呼び出す必要があります。

- イベントを削除するには、イベントを選択して「削除」ボタンを選択します。「削除」ページで、「適用」ボタンを選択します。「取消」ボタンを選択し、イベントを削除しないで元のページに戻ることもできます。

注意： 削除できるイベントは、サブスクリプションから参照されていないイベントおよびイベント・グループに属していないイベントのみです。

また、カスタマイズ・レベルが「コア」または「制限」の Oracle Applications にシードされたイベントは、削除することができません。

- 新しい個別のイベントを作成するには、「イベントの作成」ボタンを選択します。8-78 ページの「[イベントの作成または更新](#)」を参照してください。
- 新しいイベント・グループを作成するには、「イベント・グループの作成」ボタンを選択します。8-80 ページの「[イベント・グループの作成または更新](#)」を参照してください。

▶ **イベントの作成または更新**

1. 「イベントの作成」または「イベントの更新」ページに移動します。選択したイベントに対して直前に定義した情報が「イベントの更新」ページの各フィールドに表示されることを除くと、「イベントの作成」ページと「イベントの更新」ページは同一です。

注意： カスタマイズ・レベルが「制限」のシードされたイベントの場合は、イベントのステータスのみを更新できます。カスタマイズ・レベルが「コア」のシードされたイベントの場合、イベント定義は表示できますが、プロパティは更新できません。

注意： Oracle Applications のインストールには、ライセンスを受けていない Oracle Applications 製品が所有するシードされたイベントが含まれている可能性があります。そのようなイベントに対しては、「イベントの更新」ページに、そのイベントがライセンスを受けていないことを表す警告が表示されます。Oracle Workflow では、このようなイベントに対してはサブスクリプションが実行されません。

2. イベントの内部名を「名前」フィールドに入力します。内部名は大文字 / 小文字が区別されます。次のように、複数の識別子をピリオドで区切った書式で記述することをお勧めします。

`<company>.<family>.<product>.<component>.<object>.<event>`

3. イベントの表示名を入力します。
4. オプションで、イベントの説明を入力します。
5. イベントのステータスとして「使用可能」または「使用不能」を選択します。イベントを使用不能にしても、その定義はイベント・マネージャに残り、参照できます。ただし、有効なサブスクリプションでは使用できません。
6. ローカル・システムで発生するイベントを定義する場合は、イベントのジェネレート関数を入力します。ジェネレート関数は、イベント名、イベント・キーおよびパラメータ・リスト（オプション）から完全なイベント・データを生成する PL/SQL プロシー

ジャです。6-24 ページの「[イベント・データ・ジェネレート関数の標準 API](#)」を参照してください。

次の書式で、ジェネレート関数を入力します。

```
<package_name>.<function_name>
```

7. プログラム名を「所有者名」フィールドに、プログラム ID を「所有者タグ」フィールドに入力して、イベントを所有するプログラムまたはアプリケーションを識別します。
8. イベントのカスタマイズ・レベルを確認することができます。
 - コア： イベント定義を変更することはできません。このレベルは、Oracle Applications にシードされたイベントに対してのみ使用されます。
 - 制限： イベント・ステータスを「使用可能」または「使用不能」に更新できますが、イベント定義を変更することはできません。このレベルは、Oracle Applications にシードされたイベントに対してのみ使用されます。
 - ユーザー： イベント定義内のすべてのプロパティを変更することができます。このレベルは、ユーザーが定義したイベントに自動的に設定されます。
9. 「適用」ボタンを選択して、イベントを保存します。「取消」ボタンを選択し、イベントを保存しないで元のページに戻ることもできます。

▶ **イベント・グループの作成または更新**

1. 「グループの作成」または「グループの更新」ページに移動します。選択したイベント・グループに対して直前に定義した情報が「グループの更新」ページの各フィールドに表示されることを除くと、「グループの作成」ページと「グループの更新」ページは同一です。

注意： カスタマイズ・レベルが「制限」のシードされたイベント・グループの場合は、イベント・グループのステータスのみを更新できます。カスタマイズ・レベルが「コア」のシードされたイベント・グループの場合、イベント・グループ定義は表示できますが、プロパティは更新できません。

注意： Oracle Applications のインストールには、ライセンスを受けていない Oracle Applications 製品が所有するシードされたイベント・グループが含まれている可能性があります。そのようなイベントに対しては、「グループの更新」ページに、そのイベント・グループがライセンスを受けていないことを表す警告が表示されます。Oracle Workflow では、このようなイベント・グループに対してはサブスクリプションが実行されません。

ORACLE Workflow

Return to Portal Logout Preferences Help

Home Developer Studio **Business Events** Status Monitor Notifications

Events Subscriptions Agents Systems

Business Events - Events > Create Group

Create Group

An event group is a type of event composed of a set of individual member events. Event groups let you associate any events you want with each other and reference them as a group in event subscriptions.

* Indicates required field

Cancel Apply

* Name
(Format: <company>.<product family>.<product>.<component>.<object>.<event>)

* Display Name

Description

* Status Enabled

* Owner Name

* Owner Tag

* Customization Level User

Events in Group

Add Events to Group

Select Name	Display Name	Type	Status	Owner Name	Owner Tag
No events found.					

Home | Developer Studio | **Business Events** | Status Monitor | Notifications | Return to Portal | Logout | Preferences | Help

Copyright 2003 Oracle Corporation. All rights reserved. Privacy Statement

- イベント・グループの内部名を「名前」フィールドに入力します。内部名は大文字 / 小文字が区別されます。次のように、複数の識別子をピリオドで区切った書式で記述することをお勧めします。

`<company>.<family>.<product>.<component>.<object>.<event>`

- イベント・グループの表示名を入力します。
- オプションで、イベント・グループの説明を入力します。
- イベント・グループのステータスとして「使用可能」または「使用不能」を選択します。イベント・グループを使用不能にしても、その定義はイベント・マネージャに残り、参照できます。ただし、有効なサブスクリプションでは使用できません。
- プログラム名を「所有者名」フィールドに、プログラム ID を「所有者タグ」フィールドに入力して、イベント・グループを所有するプログラムまたはアプリケーションを識別します。
- イベント・グループのカスタマイズ・レベルを確認することができます。
 - コア： イベント・グループ定義を変更することはできません。このレベルは、Oracle Applications にシードされたイベント・グループに対してのみ使用されます。

- 制限： イベント・グループ・ステータスを「使用可能」または「使用不能」に更新できますが、イベント・グループ定義を変更することはできません。このレベルは、Oracle Applications にシードされたイベント・グループに対してのみ使用されます。
 - ユーザー： イベント・グループ定義内のすべてのプロパティを変更することができます。このレベルは、ユーザーが定義したイベント・グループに自動的に設定されます。
8. 「適用」ボタンを選択して、イベント・グループを保存します。「取消」ボタンを選択し、イベント・グループを保存しないで元のページに戻ることもできます。
 9. 「グループ内イベント」領域に、グループのメンバーであるイベントが表示されます。イベントごとに、リストに内部名、表示名、タイプ、ステータス、所有者名および所有者タグが表示されます。「名前」、「表示名」、「タイプ」または「ステータス」の列見出しを選択すると、その列を基準にしてリストがソートされます。

注意： イベント・グループには、個別のイベントのみを登録できます。
イベント・グループには、別のグループを登録できません。

10. メンバー・イベントをグループに追加するには、「イベントをグループに追加」ボタンを選択します。
11. 「イベントをグループに追加」ページで、検索基準を入力し、追加するイベントを検索します。主要な検索オプションとして、次の項目を使用できます。
 - 名前： 追加するイベントの内部名を入力します。内部名に含まれる数文字を入力しても、イベントを検索することができます。
12. イベントを検索するための追加検索基準として、他の特性を入力することもできます。追加オプションが非表示になっている場合に表示するには、「詳細検索の表示オプション」リンクを選択します。追加オプションが表示されている場合に非表示にする場合は、「詳細検索の非表示オプション」リンクを選択します。追加検索基準として、次の項目を使用できます。
 - 表示名： 追加するイベントの表示名を入力します。
 - ステータス： 追加するイベントのステータスとして「使用可能」または「使用不能」を選択します。
 - 所有者名： 追加するイベントを所有するプログラムまたはアプリケーションの名前を入力します。
 - 所有者タグ： 追加するイベントを所有するプログラムまたはアプリケーションのプログラム ID を入力します。
13. 「実行」ボタンを選択して、検索を実行します。

14. 「イベントの追加」領域に、検索基準と一致したイベントが表示されます。イベントごとに、リストに内部名、表示名、タイプ、ステータス、所有者名および所有者タグが表示されます。「名前」、「表示名」、「タイプ」または「ステータス」の列見出しを選択すると、その列を基準にしてリストがソートされます。
15. イベント・グループに追加するイベントを選択し、「グループに追加」ボタンを選択します。
16. イベント・グループに追加する別のイベントを検索するための、別の検索基準を入力することもできます。
17. イベントをイベント・グループに追加した後で、「適用」ボタンを選択してイベント・グループ・メンバーを保存します。「取消」ボタンを選択し、イベント・グループ・メンバーを保存しないで元のページに戻ることもできます。
18. グループからメンバー・イベントを削除するには、「グループの作成」または「グループの更新」ページで削除するイベントを選択し、「削除」ボタンを選択します。「削除」ページで、「適用」ボタンを選択します。「取消」ボタンを選択し、イベントを削除しないで元のページに戻ることもできます。

注意： メンバー・イベントをイベント・グループから削除しても、各イベントのイベント定義は削除されません。各イベントは、イベント・マネージャに残ります。

▶ テスト・イベントの呼出し

1. 「ビジネス・イベントのテスト」ページへ移動します。

The screenshot shows a web browser window titled "Test Business Event - Microsoft Internet Explorer provided by Dev. Services". The page content includes:

- Navigation tabs: Home, Developer Studio, Business Events, Status Monitor, Notifications.
- Sub-navigation: Events, Subscriptions, Agents, Systems.
- Page Title: Test Business Event
- Instructions: "To test a business event enter an event key, a list of parameter name / value pairs (optional), and either paste in an XML Document or upload from the local file system and press submit." Buttons: Cancel, Submit.
- Section: * Indicates required field
- Section: Event Identifier
 - * Event Name: oracle.apps.wf.event.test.ping
 - * Event Key: [input field]
 - Send Date: [calendar icon]
 - Send Date must be in the Format: DD-MON-RRRR
- Section: Event Parameters
 - Select Item(s) and ... [Delete]
 - Select All | Select None
 - Table with columns: Select Label, Value
 - Add Another Row
- Section: Event Data
 - XML Content: [text area]
 - Note: If Xml content is greater than 4000 char, Press "Upload XML File" Button.
- Section: Upload Option
 - Write XML (selected)
 - Upload XML File
 - File Name: < No File Chosen to Upload >
 - Upload Xml File

2. 「イベント識別子」領域で、呼び出すイベントを指定します。

- イベント名： イベントの内部名を選択します。フィールドの検索アイコンを選択すると、選択可能な値のリストが表示されます。8-85 ページの「[値リストの使用](#)」を参照してください。

注意： イベント・グループは呼び出せません。各イベントは個別に呼び出す必要があります。

- イベント・キー： イベントのインスタンスを一意に識別するイベント・キーを入力します。
- 送信日： オプションで、イベント・メッセージがデキューで利用できる日付を指定します。イベントが呼び出されたときに送信日が未来日付に設定されている場合、イベント・メッセージは WF_DEFERRED キューに格納され、指定された日付までサブスクリプション処理は開始されません。

3. オプションで、イベント・メッセージ内のパラメータ・リストに格納する追加のパラメータ名と値のペアを「イベント・パラメータ」領域に入力します。
 - 別の空白行を表示して別のパラメータを入力するには、「他の行の追加」ボタンを選択します。最大で 100 個のパラメータを入力できます。
 - パラメータを削除するには、パラメータを選択して「削除」ボタンを選択します。
4. 「イベント・データ」領域で、イベントの内容を説明する XML 文書を入力することができます。
 - 手でイベント・データを入力するには、「XML の記述」オプションを選択し、「XML コンテンツ」フィールドに XML 文書を入力します。最大で 4000 文字を入力できます。
 - ファイル・システムから XML ファイルをアップロードするには、「XML のアップロード」オプションを選択し、「XML ファイルのアップロード」ボタンを選択します。「XML ファイルのアップロード」ページで、イベント・データが含まれた XML ファイルのフルパスとファイル名を入力し、「アップロード・ファイル」ボタンを選択します。「取消」ボタンを選択し、ファイルをアップロードしないで元のページに戻ることもできます。

注意：「ビジネス・イベントのテスト」ページを使用するかわりに、イベント定義内にジェネレート関数を割り当ててイベント・データを生成したり、WF_EVENT.Raise() API を使用してイベントを呼び出したりすることもできます。これらの方法を使用すると、最大 4GB のデータを格納する CLOB としてイベント・データを渡すことができます。8-7 ページの「[イベントの定義](#)」および『Oracle Workflow API リファレンス』の「[Raise](#)」を参照してください。

5. 「送信」ボタンを選択して、イベントをイベント・マネージャに呼び出します。「取消」ボタンを選択し、イベントを呼び出さずに元のページに戻ることもできます。

▶ 値リストの使用

1. 値リストをサポートしているフィールドには、検索アイコンを選択すると、「検索および選択」ページが表示されます。
2. 「検索」領域で、検索する値のタイプに応じて、検索のフィルタを選択します。テキスト・フィールドに検索する値を入力し、「実行」ボタンを選択します。検索する値を指定しない場合は、完全な値リストが検出されます。
3. 「結果」領域に、検索基準と一致した値のリストが表示されます。目的の値を選択し、「選択」ボタンを選択します。選択した値が元のフィールドに挿入されます。また、「検索および選択」ページで「取消」ボタンを選択し、値を選択しないで前のページに戻ることもできます。

イベント・サブスクリプション

イベント・サブスクリプションには、特定のイベントと特定のシステムの関連、およびトリガー・イベントの発生時に実行する処理を登録します。イベント・サブスクリプションは、イベント・マネージャで定義できます。

Oracle Workflow をインストールすると、事前定義済の Workflow イベントに対していくつかのデフォルトのサブスクリプションが自動的に作成されます。これらのサブスクリプションを使用可能 / 使用不可を切り替えたり、コピーしたりすることにより、必要なイベント処理を実行できます。9-2 ページの「[事前定義済ワークフロー・イベント](#)」を参照してください。

イベントがローカルで呼び出されるか外部ソースから着信すると、イベント・マネージャは、そのイベントまたは Any イベントへの有効なサブスクリプションをローカル・システム単位に検索および実行します。発生したイベントへの有効なサブスクリプションが存在しない場合（Any イベントへのサブスクリプションを除く）、Oracle Workflow は Unexpected イベントへの有効なサブスクリプションを実行します。9-15 ページの「[Any イベント](#)」および 9-18 ページの「[Unexpected イベント](#)」を参照してください。

注意： 実行予定のサブスクリプションの場合は、サブスクリプションとトリガー・イベントの両方のステータスが「使用可能」に設定され、有効とマークされている必要があります。また、Oracle Applications 製品にシードされたイベントとサブスクリプションの場合は、サブスクリプションとトリガー・イベントの両方が、「共有」または「インストール済」ステータスでライセンスを受けた製品に所有されている必要があります。

現在ライセンスを受けている製品を確認するために、License Manager AD ユーティリティを使用することができます。製品ライセンスと Workflow BES ライセンスの同期コンカレント・プログラムを使用して、現在ライセンスを受けている製品のステータスに従ってビジネス・イベント・システム内のシードされたイベントおよびサブスクリプションのライセンス・ステータスが確実に更新されるようにすることができます。『Oracle Applications AD ユーティリティ・リファレンス・ガイド』の「License Manager」および『Oracle Workflow 管理者ガイド』の「ライセンス・ステータスの同期」を参照してください。

リリース 11.5.9 以前の Oracle Applications からアップグレードする場合は、イベント・マネージャ内の既存のイベントおよびサブスクリプションのライセンス・ステータスを更新するために、アップグレード後に製品ライセンスと Workflow BES ライセンスの同期コンカレント・プログラムを 1 回実行する必要があります。その後、製品にライセンスを与えると、その製品に所有されるすべてのイベントおよびサブスクリプションのライセンス・ステータスが、Oracle Workflow によって自動的に更新されます。

Oracle Workflow には、「システム:エラー」項目タイプ内の Unexpected イベントおよびデフォルト・イベント・エラー・プロセスに対する事前定義済サブスクリプションのデフォルト

トのエラー処理が用意されています。また、独自のイベント用にカスタムのエラー処理を定義することもできます。11-5 ページの「[イベント・サブスクリプション処理のエラー処理](#)」を参照してください。

サブスクライバの定義

サブスクリプションの定義を開始するには、サブスクライバとなるシステムを指定します。サブスクライバは、サブスクリプションを実行するシステムです。

各サブスクリプションには、特定のシステムに対するアクションを定義します。つまり、サブスクリプションは、実行する処理に関するシステム単位に、個別に定義する必要があります。たとえば、システム間でデータを伝播する場合は、送信システムに対して 1 つのサブスクリプションを定義し、受信システムに対して別のサブスクリプションを定義します。

サブスクリプションのトリガーの定義

サブスクリプションを適用するイベントのソース・タイプを指定する必要があります。イベントには次のソース・タイプがあります。

- ローカル： サブスクリプションは、サブスクライバ・システムで発生したイベントにのみ適用されます。
- 外部： サブスクリプションは、サブスクライバ・システム上のインバウンド・エージェントが受信したイベントにのみ適用されます。

注意： サブスクライバ・システム上のインバウンド・エージェントが受信したイベント・メッセージは、送信元エージェントがリモート・システム上に存在するかローカル・システム上に存在するにかかわらず、すべて外部ソースと見なされます。

- エラー： サブスクリプションは、WF_ERROR キューからデキューされたエラー・イベントにのみ適用されます。

次に、サブスクリプションをトリガーするイベントを選択します。個々のイベントまたはイベント・グループを選択できます。イベント・グループを選択した場合、サブスクリプションは、グループのいずれかのメンバー・イベントが発生したときにトリガーされます。

また、特定のソース・エージェントから受信したイベントのみがトリガーするように、サブスクリプションを制限することもできます。ただし、ほとんどの場合、特定のソース・エージェントを指定する必要はありません。

サブスクリプションの実行の制御

複数のサブスクリプションを同じイベントに対して定義する場合は、各サブスクリプションにフェーズ番号を指定することで、イベント・マネージャがサブスクリプションを実行する順序を制御できます。サブスクリプションは、フェーズの昇順で実行されます。たとえば、イベントの発生時に最初に実行するサブスクリプションに対して 10、次に実行するサブスクリプションに対して 20 を指定します。検証を行うサブスクリプションを実行してから他のタイプの処理を行うサブスクリプションを実行するなど、フェーズを使用すれば、異なるタイプのアクションを適切な順序で実行することができます。

注意： 同じフェーズ番号を複数のサブスクリプションに対して指定した場合は、関連するサブスクリプションが無作為に実行されます。ただし、そのサブスクリプション・グループは、他のフェーズ番号のサブスクリプションに基づいて、フェーズ番号で指定された順序で実行されます。

サブスクリプションのフェーズ番号を使用して、サブスクリプションがすぐに実行されるか遅延されるかを制御することもできます。イベント・マネージャは、フェーズ番号が 100 以降のサブスクリプションを遅延サブスクリプションとして処理します。1～99 のフェーズ番号付きのサブスクリプションは、イベントの処理が他の方法で延期されなければ、ただちに実行されます。8-49 ページの「[遅延サブスクリプション処理](#)」を参照してください。

フェーズ番号 0 (ゼロ) は、Oracle Workflow にシードされたサブスクリプションのために予約されています。

サブスクリプションには、実行される処理に応じて、イベント・データに必要なイベント情報が異なります。すべてのイベント情報が必要な場合や、イベントのインスタンスを識別するイベント・キーのみが必要な場合があります。すべてのイベント・データを必要としないサブスクリプションの場合は、イベント・キーをルール・データとして指定すると、パフォーマンスを向上させることができます。ローカルで呼び出されるイベントの場合、イベント・マネージャはサブスクリプションを実行する前に、各サブスクリプションにすべてのイベント・データが必要かどうかをチェックします。必要なイベント・データが渡されていない場合、イベント・マネージャはそのイベントに対してジェネレート関数を実行し、イベント・データを生成します。ただし、イベントへのサブスクリプションがイベント・データを必要としない場合は、サブスクリプションの実行に必要なリソースを最小限に抑えるため、イベント・マネージャはジェネレート関数を実行しません。

注意： すべてのイベント・データを必要とするサブスクリプションの場合でも、イベントを呼び出した後にできるだけ早く制御をコール側アプリケーションに戻すには、すべてのサブスクリプションを遅延させます。後でそれらのサブスクリプションを実行するまで、イベント・マネージャはジェネレート関数を実行しません。

各サブスクリプションには、サブスクリプション定義を更新できるかどうかを判断するためのカスタマイズ・レベルが割り当てられています。このカスタマイズ・レベルは、Oracle

Applications のシード・データを保護し、アップグレード中にユーザーのカスタマイズを保存するために使用されます。各サブスクリプションには、次のカスタマイズ・レベルの1つが割り当てられています。

- コア： サブスクリプション定義を変更することはできません。このレベルは、Oracle Applications にシードされたサブスクリプションに対してのみ使用されます。
- 制限： サブスクリプションのステータスを「使用可能」または「使用不能」に更新できますが、サブスクリプション定義を変更することはできません。このレベルは、Oracle Applications にシードされたサブスクリプションに対してのみ使用されます。
- ユーザー： サブスクリプション定義内のすべてのプロパティを変更することができます。このレベルは、ユーザーが定義したサブスクリプションに自動的に設定されます。

Oracle Applications 製品には、シードされたイベントとサブスクリプションが用意されているものがあります。「インストール済」または「共有」ステータスでライセンスを受けている製品でトリガー・イベントとサブスクリプションの両方が所有されている場合にのみ、サブスクリプションが実行されます。

Oracle Applications のインストールには、ライセンスを受けていない Oracle Applications 製品が所有するシードされたサブスクリプションが含まれている可能性があります。そのようなサブスクリプションに対しては、「サブスクリプションを更新するページ」に、そのサブスクリプションがライセンスを受けていないことを表す警告が表示されます。Oracle Workflow では、このようなサブスクリプションは実行されません。また、サブスクリプション自体はライセンスを受けている製品に所有されていたとしても、ライセンスを受けていないイベントに指定されたサブスクリプションは実行されません。

現在ライセンスを受けている製品を確認するために、License Manager AD ユーティリティを使用することができます。製品ライセンスと Workflow BES ライセンスの同期コンカレント・プログラムを使用して、現在ライセンスを受けている製品のステータスに従ってビジネス・イベント・システム内のシードされたイベントおよびサブスクリプションのライセンス・ステータスが確実に更新されるようにすることができます。『Oracle Applications AD ユーティリティ・リファレンス・ガイド』の「License Manager」および『Oracle Workflow 管理者ガイド』の「ライセンス・ステータスの同期」を参照してください。

注意： カスタマイズ・レベルを「ユーザー」と定義したサブスクリプションは常に、すべてライセンスを保有しているとみなされます。

サブスクリプションに対するアクションの定義

サブスクリプションには、次のタイプの処理を指定できます。

- イベント・メッセージに対する関数の実行
- ワークフロー・プロセスへのイベント・メッセージの送信
- エージェントへのイベント・メッセージの送信

ルール関数の実行

イベント・メッセージに対して関数を実行するには、実行するルール関数を指定します。また、関数に渡す追加のパラメータを指定することもできます。

Oracle Workflow には、基本的なサブスクリプション処理を実行するための標準のデフォルト・ルール関数が用意されています。デフォルトのルール関数では、次の処理が行われません。

- ワークフロー・プロセスへのイベント・メッセージの送信（サブスクリプション定義で指定されている場合）
- エージェントへのイベント・メッセージの送信（サブスクリプション定義で指定されている場合）

『Oracle Workflow API リファレンス』の「Default_Rule」を参照してください。

Oracle Workflow には、テストおよびデバッグ、またはその他の目的で使用可能な標準のルール関数もいくつか用意されています。『Oracle Workflow API リファレンス』の「イベント・サブスクリプションのルール関数の API」を参照してください。

サブスクリプション処理は、カスタム・ルール関数を作成することで拡張できます。カスタム・ルール関数は、標準 API に従って定義する必要があります。6-29 ページの「イベント・サブスクリプションのルール関数の標準 API」を参照してください。

ルール関数は、次のような様々な目的で使用できます。

- 検証を実行する。
- インバウンド・メッセージをアプリケーションの「Receive」メッセージ・ハンドラとして処理する。
- アウトバウンド・メッセージに変更を加える。他のメッセージに関連付ける関連 ID を追加するなど。

注意： カスタム・ルール関数から WF_EVENT_FUNCTIONS_PKG.AddCorrelation() をコールすると、カスタム処理中に関連 ID を追加できます。『Oracle Workflow API リファレンス』の「AddCorrelation」を参照してください。

ルール関数は、イベント・メッセージに対して読取りまたは書込みを行ったり、任意のデータベース処理を実行したりします。ただし、ルール関数内ではコミットしないでください。コミットは、コール側のアプリケーションが行うため、イベント・マネージャではコミットを発行しません。また、ルール関数を使用して、セキュリティやNLSの設定などの接続コンテキストを変更しないでください。ルール関数がエラーを返した場合、サブスクリプション処理は中止されます。

特定のイベントに対してサブスクリプション処理を実行するときに、連続したステップがいくつか含まれている場合は、再利用できない複雑で特別なルール関数を作成するのではなく、再利用できる単純なルール関数を作成して複数のサブスクリプションをイベントに定義することをお勧めします。複数のサブスクリプションの実行順序を指定するには、サブスクリプションに対してフェーズ値を入力します。

注意： デフォルト以外のルール関数を入力した場合は、参照する関数に対してワークフロー情報およびエージェント情報を入力できますが、指定されたワークフローおよびエージェントに対してイベント・メッセージは自動的に送信されません。この場合、送信処理をルール関数に明示的に組み込むか、デフォルトのルール関数を使用せずに送信処理を実行する別のサブスクリプションを定義する必要があります。

ワークフロー・プロセスへのイベントの送信

イベントをワークフロー・プロセスに送信するには、送信先のプロセスの項目タイプとプロセス名を指定する必要があります。プロセスの項目キーは、イベント・メッセージに指定されている相関 ID によって決定されます。相関 ID が指定されていない場合は、イベント・キーによって決定されます。

注意： プロセス・インスタンスの項目キーには、シングルバイト・キャラクターのみ使用できます。マルチバイトの値を含めることはできません。

注意： サブスクリプションの処理中に `WF_EVENT_FUNCTIONS_PKG.AddCorrelation()` をコールすると、イベント・メッセージに相関 ID を追加できます。 `AddCorrelation()` を使用するには、 `ITEMKEY` というサブスクリプション・パラメータを入力して、相関 ID を生成する関数を指定する必要があります。関数は次の書式で指定する必要があります。

`ITEMKEY=<package_name.function_name>`

『Oracle Workflow API リファレンス』の「AddCorrelation」を参照してください。

イベントをワークフロー・プロセスに送信することにより、事前定義済の関数を直接実行したり、事前定義済の受信者にイベントを送信するだけでなく、複雑な処理またはルーティング・ロジックをモデル化できます。たとえば、イベント・メッセージの内容に基づいて、別の関数に分岐したり、サブプロセスを開始したり、通知を送信したり、宛先エージェントを選択したりできます。また、イベント・メッセージ自体を変更することもできます。

ワークフロー・プロセス内では、イベントはイベント・アクティビティ単位に表されます。3-56 ページの「[イベント・アクティビティ](#)」を参照してください。

プロセスがイベントを受信すると、ワークフロー・エンジンは「受信」イベント・アクティビティ・ノードのイベントの詳細の指定に従って、イベント名、イベント・キーおよびイベント・メッセージを項目タイプ属性に格納します。さらに、ワークフロー・エンジンは、イベント・メッセージのパラメータ・リストのパラメータをプロセスの項目タイプとして設定します。パラメータに対応する属性がまだ存在しない場合は、新しい項目タイプ属性を作成します。また、サブスクリプションの GUID (グローバル一意識別子) が動的な項目属性として設定されます。このため、ワークフロー・プロセスから、サブスクリプション定義に含まれる他の情報を参照することができます。

イベントが別のワークフロー・プロセスの呼出しイベント・アクティビティによって発生した場合は、そのプロセスの項目タイプと項目キーがイベント・メッセージ内のパラメータ・リストに追加されます。ワークフロー・エンジンは、イベントを受信するプロセスの親として、指定されたプロセスを自動的に設定し、既存の親の設定を上書きします。『Oracle Workflow API リファレンス』の「[SetItemParent](#)」を参照してください。

追加パラメータを指定すると、ワークフロー・プロセスの項目属性として設定されます。追加パラメータをサブスクリプションの「パラメータ」フィールドに入力し、サブスクリプションのルール関数で `WF_RULE.SetParametersIntoParameterList()` を使用すれば、指定したサブスクリプション・パラメータをイベント・メッセージのパラメータ・リストに設定することができます。ワークフロー・プロセスにイベントが着信すると、それらのイベント・パラメータはワークフロー・プロセスの項目属性として設定されます。『Oracle Workflow API リファレンス』の「[SetParametersIntoParameterList](#)」を参照してください。

注意： イベントをワークフロー・プロセスに送信するには、Oracle Workflow から提供されるデフォルトのルール関数を使用するか、送信処理をカスタム・ルール関数に組み込む必要があります。6-29 ページの「[イベント・サブスクリプションのルール関数の標準 API](#)」を参照してください。

エージェントへのイベントの送信

イベントをエージェントに送信するには、アウトバウンド・メッセージを送信する送信エージェントまたはインバウンド・メッセージを受信する宛先エージェント、あるいはその両方のエージェントを指定します。

- 宛先エージェントと送信エージェントの両方を指定した場合、イベント・メッセージは送信エージェントのキューに格納され、宛先エージェントに伝播されます。
- 送信エージェントを指定しないで宛先エージェントを指定した場合は、サブスクライバ・システム上のアウトバウンド・エージェントのうち、キュー・タイプが宛先エージェントと一致するものが選択されます。イベント・メッセージは、このアウトバウンド・エージェントのキューに格納され、宛先エージェントに伝播されます。
- 宛先エージェントを指定しないで送信エージェントを指定した場合、イベント・メッセージは送信エージェントのキューに格納されますが、受信者は指定されていません。
 - 送信エージェントがサブスクライバ・リストで複数コンシューマ・キューを使用する場合は、宛先エージェントを省略できます。(標準の **Workflow** キュー・ハンドラは、複数コンシューマ・キューでのみ機能します。) この場合、キューのサブスクライバ・リストによって、メッセージをデキューできるコンシューマが決定されます。キューに対してサブスクライバ・リストが定義されていない場合、イベント・メッセージはエラー処理のために **WF_ERROR** キューに格納されます。

注意： Oracle Advanced Queuing が使用する複数コンシューマ・キューのサブスクライバ・リストは、Oracle Workflow Business Event System のイベント・サブスクリプションとは異なります。詳細は、『Oracle9i アプリケーション開発者ガイド-アドバンスト・キューイング』の「サブスクリプションおよび受信者リスト」、または『Oracle Streams アドバンスト・キューイング・ユーザーズ・ガイドおよびリファレンス』を参照してください。

- 送信エージェントが単一コンシューマ・キューを使用するときに、カスタム・キュー・ハンドラが定義されている場合は、宛先エージェントを省略できます。単一コンシューマ・キューの場合、コンシューマを指定する必要はありません。

受信者がメッセージをデキューするときの、優先順位を指定することができます。メッセージは、優先順位の昇順でデキューされます。

注意： イベントをエージェントに送信するには、Oracle Workflow から提供されるデフォルトのルール関数を使用するか、送信処理をカスタム・ルール関数に組み込む必要があります。6-29 ページの「**イベント・サブスクリプションのルール関数の標準 API**」を参照してください。

伝播したイベント・メッセージを受信者にすぐに送信せずに後で送信する場合は、イベント・メッセージの `SEND_DATE` 属性を未来日付に設定します。設定する送信日は、サブスクリプションの処理期間中で、イベントが送信される前である必要があります。つまり、送信処理が開始される前に、直前のサブスクリプションまたはルール関数の初期処理が実行されている期間です。イベント・メッセージは宛先エージェントに伝播されますが、指定した日付までデキューされません。

サブスクリプションの識別情報の定義

プログラム名と簡単な説明をサブスクリプションの所有者名および所有者タグとして設定することにより、各サブスクリプションを、サブスクリプションが属するプログラムまたはアプリケーションと関連付けることができます。この識別情報を使用して、所有するサブスクリプションをプログラムから検索することができます。サブスクリプションの所有者は、トリガー・イベントの所有者と同じプログラムであっても、異なるプログラムであってもかまいません。

遅延サブスクリプション処理

イベントが発生したときにイベントのサブスクリプションをすぐに実行しない場合は、サブスクリプションを遅延できます。サブスクリプションを遅延すると、コール側アプリケーションに制御を早く戻すことができ、イベント・マネージャは負荷の大きいサブスクリプション処理を後で実行できます。

サブスクリプションは、次の3つの方法で遅延できます。

- `SEND_DATE` 属性に未来日付を指定して、イベントを呼び出します。この方法は、ローカルで呼び出されたイベントに対するすべてのサブスクリプション処理を特定の有効日時まで遅延するときに使用します。
- イベントに対するサブスクリプションのフェーズ番号を 100 以降に定義します。この方法は、ローカル・イベントまたは外部イベントの特定のサブスクリプション処理を遅延するときに使用します。
- イベントを呼び出す前に、イベント・マネージャのディスパッチ・モードを遅延処理に設定します。この方法は、ローカルで呼び出されるイベントに対するサブスクリプション処理をすべて遅延するときに使用できます。ただし、この方法は特別な状況でのみ使用してください。

これらのいずれかの方法によってイベントのサブスクリプション処理が遅延されると、イベント・メッセージは `WF_DEFERRED` エージェントに関連付けられた標準 `WF_DEFERRED` キューに格納されます。`WF_DEFERRED` エージェントを監視するには、リスナーをスケジュールする必要があります。『Oracle Workflow 管理者ガイド』の「ローカル・インバウンド・エージェントのリスナーのスケジュール」を参照してください。

リスナーは、優先順位に従って `WF_DEFERRED` エージェントからイベント・メッセージをデキューします。イベント・メッセージのソース・タイプ（「ローカル」または「外部」）は保持されます。これらのイベントに対するサブスクリプション処理の遅延時間は、リスナー

に定義されているスケジュールおよび指定した有効日（未来日付のイベントの場合）によって決まります。

未来日付の送信日を使用したサブスクリプション処理の遅延

特定の未来有効日までローカル・イベントのサブスクリプション処理を遅延するには、SEND_DATE 属性に特定の未来有効日を設定してイベントを呼び出します。たとえば、新しい従業員が採用されたらすぐにその従業員の情報を人事管理アプリケーションに入力し、給与計算の処理は従業員の勤務開始日まで遅延します。

イベントが未来送信日で呼び出されると、イベント・マネージャはそのイベント・メッセージをすぐに WF_DEFERRED キューに格納し、イベントのサブスクリプションは実行しません。そのイベントに対するすべてのサブスクリプションは、フェーズ番号にかかわらずすべて遅延されます。イベントはその送信日まで WAIT 状態になります。送信日になると、イベント・メッセージはデキュー可能になり、エージェント・リスナーが WF_DEFERRED キュー上で実行されたときにデキューされます。サブスクリプション処理の遅延時間は、指定した送信日およびリスナーに定義されているスケジュールによって決まります。

リスナーがイベント・メッセージをデキューすると、イベント・マネージャが ERROR_SUBSCRIPTION 属性のサブスクリプション ID をチェックします。イベント・メッセージにサブスクリプション ID が含まれている場合、つまり、イベントが呼び出された直後にそのイベントに対するすべてのサブスクリプション処理が遅延された場合、イベント・マネージャはイベントに対するすべてのサブスクリプションをフェーズの昇順で実行します。

注意： イベントが呼び出されたときにイベントが遅延されていなかった場合は、そのイベントが WF_DEFERRED キューからデキューされるときに、イベント・マネージャはそのイベントに対する適格なサブスクリプションをすべて実行します。サブスクリプションのフェーズ番号は無視されます。100 以降のフェーズ番号が付いている場合でも、サブスクリプションは遅延されません。

関連項目：

『Oracle Workflow API リファレンス』の「Raise」

サブスクリプションのフェーズ番号を使用したサブスクリプション処理の遅延

サブスクリプションのフェーズ番号を使用して、サブスクリプションがすぐに実行されるか遅延されるかを制御することもできます。イベント・マネージャは、フェーズ番号が 100 以降のサブスクリプションを遅延サブスクリプションとして処理します。「ローカル」および「外部」のサブスクリプションをこの方法で遅延できます。

注意： イベントがローカルで呼び出されるときにこの遅延方法を使用するには、そのイベントが未来日付の送信日で呼び出され、イベント・マネージャのサブスクリプション処理が通常の同期モードに設定されている必要があります。それ以外の場合、イベント・メッセージは WF_DEFERRED キューに格納され、イベントがデキューされるまでイベント・マネージャはサブスクリプションを実行しません。

トリガー・イベントが呼び出されるか着信すると、100 以降のフェーズ番号のサブスクリプションを検出するまで、イベント・マネージャはそのイベントに対するサブスクリプションをフェーズ番号順に実行します。イベント・マネージャは、そのサブスクリプションをイベント・メッセージ内の ERROR_SUBSCRIPTION 属性に設定し、サブスクリプション・プロパティに指定されている優先度を PRIORITY 属性に設定します。イベント・メッセージは、標準 WF_DEFERRED キューに格納されます。

サブスクリプション処理の遅延時間は、WF_DEFERRED エージェントを監視するエージェント・リスナーに定義されているスケジュールによって決まります。リスナーがイベント・メッセージをデキューすると、イベント・マネージャは ERROR_SUBSCRIPTION 属性のサブスクリプション ID をチェックします。サブスクリプション ID が存在する場合、つまり、そのサブスクリプション以降のサブスクリプション処理が遅延されている場合、イベント・マネージャは最初にそのサブスクリプションを実行してから、そのサブスクリプション以降のフェーズ番号を持つイベントについてすべてのサブスクリプションを実行します。

注意： イベント・マネージャは、イベント・メッセージに設定されているサブスクリプションのフェーズ番号で、サブスクリプション処理を再開します。そのイベントが最初に処理されたときにフェーズ番号が 100 のサブスクリプションが存在しなかった場合は、フェーズ番号 100 から開始されない場合があります。

イベント・マネージャのディスパッチ・モードを使用したサブスクリプション処理の遅延

イベントをローカル・アプリケーションから呼び出す場合、アプリケーションがそのイベントを呼び出すたびに、そのサブスクリプション処理をすべて遅延することもできます。この方法を実行するには、Raise() API をコールする直前に、遅延（非同期）処理を示す「ASYNC」モードで SetDispatchMode() API をコールします。『Oracle Workflow API リファレンス』の「SetDispatchMode」を参照してください。

このメソッドは、アプリケーション内で遅延をハードコーディングする必要があるため、特別な状況でのみ使用してください。柔軟性を維持しながら、アプリケーションに影響を与えずにサブスクリプション処理を変更するには、未来日付の送信日でイベントを呼び出すか、サブスクリプションのフェーズ番号を使用して一部またはすべてのサブスクリプションを遅延として指定します。

ディスパッチ・モードが遅延処理に設定された後にイベントが呼び出されると、イベント・マネージャはイベント・メッセージをすぐに WF_DEFERRED キューに格納し、イベントの

サブスクリプションは実行しません。そのイベントに対するすべてのサブスクリプションは、フェーズ番号にかかわらずすべて遅延されます。

サブスクリプション処理の遅延時間は、WF_DEFERRED エージェントを監視するエージェント・リスナーに定義されているスケジュールによって決まります。リスナーがイベント・メッセージをデキューすると、イベント・マネージャが ERROR_SUBSCRIPTION 属性のサブスクリプション ID をチェックします。イベント・メッセージにサブスクリプション ID が含まれている場合、つまり、イベントが呼び出された直後にそのイベントに対するすべてのサブスクリプション処理が遅延された場合、イベント・マネージャはイベントに対するすべてのサブスクリプションをフェーズの昇順で実行します。

注意： イベントが呼び出されたときにイベントが遅延されていなかった場合は、そのイベントが WF_DEFERRED キューからデキューされるときに、イベント・マネージャはそのイベントに対する適格なサブスクリプションをすべて実行します。サブスクリプションのフェーズ番号は無視されます。100 以降のフェーズ番号が付いている場合でも、サブスクリプションは遅延されません。

関連項目：

6-29 ページ「[イベント・サブスクリプションのルール関数の標準 API](#)」

『Oracle Workflow 管理者ガイド』の「ローカル・インバウンド・エージェントのリスナーのスケジュール」

▶ イベント・サブスクリプションの表示と管理

1. システム管理者によって指定された職責およびナビゲーション・パスを使用して、Web ブラウザでイベント・マネージャにアクセスします。次に水平ナビゲーションの「サブスクリプション」を選択します。A-2 ページの「[Oracle Workflow 開発者のナビゲーション・パス](#)」を参照してください。

Oracle Workflow の管理者用 Web ページの「ビジネス・イベント」タブまたは下部にある「ビジネス・イベント」リンクを選択して、イベント・マネージャへ移動することができます。



2. 特定のサブスクリプションを検索する場合は、「検索」領域に検索基準を入力します。主要な検索オプションとして、次の項目を使用できます。
 - システム： サブスクリプションを表示するシステムを選択します。フィールドの検索アイコンを選択すると、選択可能な値のリストが表示されます。システムの内部名または表示名を基準として検索することができます。8-85 ページの「[値リストの使用](#)」を参照してください。
3. サブスクリプションを検索するための追加検索基準として、他の特性を入力することもできます。追加オプションが非表示になっている場合に表示するには、「詳細検索の表示オプション」リンクを選択します。追加オプションが表示されている場合に非表示にする場合は、「詳細検索の非表示オプション」リンクを選択します。追加検索基準として、次の項目を使用できます。
 - ソース・タイプ： サブスクリプションを表示するソース・システムのタイプとして、「ローカル」、「外部」または「エラー」を選択します。
 - イベント： サブスクリプションを表示するイベントを選択します。フィールドの検索アイコンを選択すると、選択可能な値のリストが表示されます。イベントの内部名を基準として検索することができます。8-85 ページの「[値リストの使用](#)」を参照してください。
 - ステータス： 表示するサブスクリプションのステータスとして「使用可能」または「使用不能」を選択します。

4. 「実行」 ボタンを選択して、検索を実行します。
5. 「結果」 領域に、検索基準と一致したサブスクリプションが表示されます。各サブスクリプションについて、リストにはトリガー・イベント、サブスクライバ・システム、ソース・タイプ、ルール関数、ワークフロー・プロセスおよびステータスが表示されます。「イベント」、「システム」、「ソース・タイプ」または「関数」の列見出しを選択すると、その列を基準にリストがソートされます。
6. ワークフロー管理者権限を持っている場合は、管理アイコンやボタンを使用して、管理操作を実行できます。
 - サブスクリプションを更新するには、そのサブスクリプションの「更新」列にある更新アイコンを選択します。8-99 ページの「[イベント・サブスクリプションの作成または更新](#)」を参照してください。

注意： カスタマイズ・レベルが「制限」のシードされたサブスクリプションの場合は、サブスクリプションのステータスのみを更新できます。カスタマイズ・レベルが「コア」のシードされたサブスクリプションの場合、サブスクリプション定義は表示できますが、プロパティは更新できません。

- サブスクリプションを削除するには、サブスクリプションを選択して「削除」 ボタンを選択します。「削除」 ページで、「適用」 ボタンを選択します。「取消」 ボタンを選択し、サブスクリプションを削除しないで元のページに戻ることもできます。

注意： カスタマイズ・レベルが「コア」または「制限」の Oracle Applications にシードされたサブスクリプションは、削除することができません。

- 新しいサブスクリプションを作成するには、「予約の作成」 ボタンを選択します。8-99 ページの「[イベント・サブスクリプションの作成または更新](#)」を参照してください。

▶ **イベント・サブスクリプションの作成または更新**

1. 「イベント予約の作成」または「イベント予約の更新」 ページに移動します。選択したサブスクリプションに対して直前に定義した情報が「イベント予約の更新」 ページの各フィールドに表示されることを除くと、「イベント予約の作成」 ページと「イベント予約の更新」 ページは同一です。

注意： カスタマイズ・レベルが「制限」のシードされたサブスクリプションの場合は、サブスクリプションのステータスのみを更新できます。カスタマイズ・レベルが「コア」のシードされたサブスクリプションの場合、サブスクリプション定義は表示できますが、プロパティは更新できません。

注意： Oracle Applications のインストールには、ライセンスを受けていない Oracle Applications 製品が所有するシードされたサブスクリプションが含まれている可能性があります。そのようなサブスクリプションに対しては、「サブスクリプションを更新するページ」に、そのサブスクリプションがライセンスを受けていないことを表す警告が表示されます。Oracle Workflow では、このようなサブスクリプションは実行されません。

2. 「サブスクリバ」領域に、サブスクリプションを実行するシステムを入力します。フィールドの検索アイコンを選択すると、選択可能な値のリストが表示されます。シス

テムの内部名または表示名を基準として検索することができます。8-85 ページの「[値リストの使用](#)」を参照してください。

3. 「トリガー条件」領域の「ソース・タイプ」フィールドに、サブスクリプションを適用するソース・システムのタイプを指定します。
 - ローカル：サブスクリプションは、サブスクライバ・システムで発生したイベントにのみ適用されます。
 - 外部：サブスクリプションは、サブスクライバ・システム上のインバウンド・エージェントが受信したイベントにのみ適用されます。

注意： サブスクライバ・システム上のインバウンド・エージェントが受信したイベント・メッセージは、送信元エージェントがリモート・システム上に存在するかローカル・システム上に存在するかにかかわらず、すべて外部ソースと見なされます。

- エラー：サブスクリプションは、WF_ERROR キューからデキューされたエラー・イベントにのみ適用されます。
4. サブスクリプションをトリガーするイベントを「イベント・フィルタ」フィールドに入力します。個々のイベントまたはイベント・グループを指定できます。フィールドの検索アイコンを選択すると、選択可能な値のリストが表示されます。イベントの内部名を基準として検索することができます。8-85 ページの「[値リストの使用](#)」を参照してください。
 5. オプションで、サブスクリプションを適用するソース・エージェントを入力します。ソース・エージェントを指定すると、サブスクリプションは、トリガー・イベントがそのエージェントから着信した場合にのみ実行されます。フィールドの検索アイコンを選択すると、選択可能な値のリストが表示されます。エージェント名を基準として検索することができます。8-85 ページの「[値リストの使用](#)」を参照してください。

注意： ほとんどの場合、「ソース・エージェント」フィールドは空白のままにします。

6. 同じイベントにトリガーされるサブスクリプションの実行順序を指定するときは、「実行制御」領域にサブスクリプションのフェーズ番号を入力します。フェーズ番号を使用して、サブスクリプションがすぐに実行されるか遅延されるかを制御することもできます。
7. サブスクリプションのステータスとして「使用可能」または「使用不能」を選択します。サブスクリプションを使用不能にしても、そのサブスクリプションはイベント・マネージャに残り、参照できます。ただし、イベントが発生しても実行できなくなります。

8. 「ルール・データ」フィールドで、サブスクリプションに必要なイベント情報を指定します。
 - キー： サブスクリプションにはイベント・キーのみが必要です。
 - メッセージ： サブスクリプションにはすべてのイベント・データが必要です。
9. サブスクリプションのカスタマイズ・レベルを確認することができます。
 - コア： サブスクリプション定義を変更することはできません。このレベルは、Oracle Applications にシードされたサブスクリプションに対してのみ使用されます。
 - 制限： サブスクリプションのステータスを「使用可能」または「使用不能」に更新できますが、サブスクリプション定義を変更することはできません。このレベルは、Oracle Applications にシードされたサブスクリプションに対してのみ使用されます。
 - ユーザー： サブスクリプション定義内のすべてのプロパティを変更することができます。このレベルは、ユーザーが定義したサブスクリプションに自動的に設定されます。
10. 「処理」領域で、トリガー・イベントの発生時に実行するサブスクリプション処理を定義します。サブスクリプションには、次の処理を指定できます。
 - イベント・メッセージに対する関数の実行
 - ワークフロー・プロセスへのイベント・メッセージの送信
 - エージェントへのイベント・メッセージの送信
11. イベント・メッセージに対して実行するルール関数を入力します。ルール関数は、標準 API に従って定義する必要があります。6-29 ページの「[イベント・サブスクリプションのルール関数の標準 API](#)」を参照してください。

次の書式で、ルール関数を入力します。

```
<package_name>.<function_name>
```

注意： デフォルト以外のルール関数を入力した場合、指定されたワークフローおよびエージェントにイベント・メッセージは自動的に送信されません。この場合、送信処理をカスタム・ルール関数に明示的に組み込む必要があります。ただし、「処理」領域には、参照する関数に対するワークフロー情報およびエージェント情報を入力できます。

12. イベント・メッセージをワークフロー・プロセスに送信する場合は、プロセスが所属するワークフロー項目タイプを「Workflow タイプ」フィールドに、プロセスの名前を「ワークフロー・プロセス」フィールドに入力します。各フィールドの検索アイコンを選択すると、選択可能な値のリストが表示されます。ワークフロー・タイプを内部名、

表示名または説明を基準として、また、ワークフロー・プロセスをプロセス名を基準として検索することができます。8-85 ページの「[値リストの使用](#)」を参照してください。

注意：「ワークフロー・プロセス」フィールドの値リストには、指定した項目タイプの実行可能プロセスのみが表示されます。

13. イベントをエージェントに送信する場合は、アウトバウンド・メッセージを送信する送信エージェントまたはインバウンド・メッセージを受信する宛先エージェント、あるいはその両方のエージェントを入力します。各フィールドの検索アイコンを選択すると、選択可能な値のリストが表示されます。エージェント名を基準として検索することができます。8-85 ページの「[値リストの使用](#)」を参照してください。
- 宛先エージェントと送信エージェントの両方を指定した場合、イベント・メッセージは送信エージェントのキューに格納され、宛先エージェントに伝播されます。
 - 送信エージェントを指定しないで宛先エージェントを指定した場合は、サブスクライバ・システム上のアウトバウンド・エージェントのうち、キュー・タイプが宛先エージェントと一致するものが選択されます。イベント・メッセージは、このアウトバウンド・エージェントのキューに格納され、宛先エージェントに伝播されます。
 - 宛先エージェントを指定しないで送信エージェントを指定した場合、イベント・メッセージは送信エージェントのキューに格納されますが、受信者は指定されていません。送信エージェントは、サブスクライバ・リストが定義された複数コンシューマ・キューまたは単一コンシューマ・キューを使用する必要があります。

注意：送信エージェントは、サブスクライバ・システム上に存在する必要があります。「送信エージェント」フィールドの値リストには、送信方向のエージェントのみが表示されます。

「宛先エージェント」フィールドの値リストには、インバウンド方向のエージェントのみが表示されます。

14. イベント・メッセージをエージェントに送信する場合は、受信者がメッセージをデキューするときの優先順位として、「標準」、「高」または「低」を選択します。
15. オプションで、「パラメータ」フィールドにルール関数の追加パラメータを入力します。複数のパラメータを区切るには、スペースを使用します。各パラメータの名前および値は、次の書式で指定します。

```
<name1>=<value1> <name2>=<value2> ... <nameN>=<valueN>
```

注意： イベント・メッセージをワークフロー・プロセスに送信するときに、追加パラメータを指定すると、ワークフロー・プロセスの項目属性として設定されます。追加パラメータをサブスクリプションの「パラメータ」フィールドに入力し、サブスクリプションのルール関数で `WF_RULE.SetParametersIntoParameterList()` を使用すれば、指定したサブスクリプション・パラメータをイベント・メッセージのパラメータ・リストに設定することができます。ワークフロー・プロセスにイベントが着信すると、それらのイベント・パラメータはワークフロー・プロセスの項目属性として設定されます。『Oracle Workflow API リファレンス』の「SetParametersIntoParameterList」を参照してください。

16. 「文書」領域を利用して、サブスクリプションを所有するプログラムまたはアプリケーションを識別します。プログラム名を「所有者名」フィールドに、プログラム ID を「所有者タグ」フィールドに入力します。
17. サブスクリプションのカスタマイズ・レベルを確認することができます。
 - コア： サブスクリプション定義を変更することはできません。このレベルは、Oracle Applications にシードされたサブスクリプションに対してのみ使用されます。
 - 制限： サブスクリプションのステータスを「使用可能」または「使用不能」に更新できますが、サブスクリプション定義を変更することはできません。このレベルは、Oracle Applications にシードされたサブスクリプションに対してのみ使用されます。
 - ユーザー： サブスクリプション定義内のすべてのプロパティを変更することができます。このレベルは、ユーザーが定義したサブスクリプションに自動的に設定されます。
18. オプションで、サブスクリプションの説明を入力します。
19. 「適用」ボタンを選択して、サブスクリプションを保存します。「取消」ボタンを選択し、サブスクリプションを保存しないで元のページに戻ることもできます。

エージェント

エージェントは、システム内の名前付き通信ポイントです。システム内およびシステム間の通信は、エージェント間のメッセージの送信によって実行されます。1つのシステムに対して複数のエージェントを設定し、それぞれ異なる通信代替を指定できます。たとえば、1つのシステムに対して複数のエージェントを定義し、インバウンド通信とアウトバウンド通信、異なるプロトコルによる通信、異なる伝播頻度など、様々な通信代替に対応しています。

イベント・マネージャ内でイベントの交換に使用するエージェントを定義する必要があります。各エージェントの名前は、システム内で一意である必要があります。エージェントは、次のような書式の複合名を使用して、Oracle Workflow 内のコードから参照できます。

```
<agent_name>@<system_name>
```

たとえば、システム HUB 内のエージェント WF_IN は、WF_IN@HUB として参照できます。

ローカル・システムにエージェントを定義したら、ローカル・インバウンド・エージェントのエージェント・リスナーおよびローカル・アウトバウンド・エージェントの伝播をスケジュールして、エージェントをイベント・メッセージの通信用に設定する必要があります。スケジュールの伝播には Oracle Enterprise Manager を、エージェント・リスナーの実行には Oracle Applications Manager の Workflow Manager コンポーネントを使用することができます。Workflow Manager では、様々なエージェントのキューの詳細情報を確認するだけでなく、各エージェントのイベント・メッセージの配布状況を調べることもできます。詳細は、Oracle Applications Manager のオンライン・ヘルプ、『Oracle9i アプリケーション開発者ガイド-アドバンスト・キューイング』の「Oracle Enterprise Manager のサポート」または『Oracle Streams アドバンスト・キューイング・ユーザズ・ガイドおよびリファレンス』、『Oracle Enterprise Manager 管理者ガイド』の「分散管理」および『Oracle Workflow 管理者ガイド』の「ビジネス・イベント・システムの設定」を参照してください。

「イベント・システム・ローカル・キュー」ページを使用して、ビジネス・イベント・システムで使用されているローカル・キューおよびそれらのキューに現在保持されているメッセージを確認することもできます。8-69 ページの「ローカル・キューの確認」を参照してください。

エージェントへの方向の割当て

イベント・マネージャでエージェントを定義する際に、エージェントがローカル・システム内でサポートする通信方向を指定する必要があります。

- 受信： システムへのインバウンド通信。エージェントはメッセージを特定のプロトコルで受信して、標準の書式でシステムに渡します。
- 送信： システムからのアウトバウンド通信。エージェントはシステムから標準の書式でメッセージを受け取り、指定されたプロトコルを使用して送信します。

エージェントへのプロトコルの割当て

各エージェントは、メッセージの通信に使用するプロトコルに関連付ける必要があります。プロトコルには、メッセージのエンコードおよび送信方法が指定されています。メッセージを正しく通信するには、エージェントの送受信で同じプロトコルを使用する必要があります。

プロトコルは、SQLNET などのネットワーク標準を表します。また、ビジネス間標準も表します。つまり、システム間の上位レベルのメッセージ書式およびハンドシェイク手順を定義します。

ビジネス・イベント・システムは、AQ キューを介してエージェントと対話します。AQ を使用すれば、AQ でサポートされている SQLNET プロトコルを介してメッセージを伝播できます。Oracle9i Database 以降の AQ には、インターネット・アクセス機能も組み込まれています。AQ の Internet Data Access Presentation (IDAP) を使用してメッセージを作成し、それらの IDAP メッセージを HTTP や HTTPS などのトランスポート・プロトコルを使用してインターネットを介して送信すれば、AQ 操作をインターネットを介して行うことができます。また、Oracle9i Database 以降では、AQ のメッセージ・ゲートウェイ機能を使用して、Oracle 以外のメッセージ・システムを使用するアプリケーションと AQ との間で通信することができ、サードパーティのメッセージ・ソリューションとの統合が可能になります。さらに、他のサービスでサポートされているプロトコルを使用して、メッセージを伝播することもできます。次の表は、様々なプロトコルに使用できるサービスを、データベースのバージョン別に示しています。

表 8-12

データベースのバージョン	SQLNET プロトコル	HTTP/HTTPS プロトコル	サードパーティのメッセージ・ソリューションとの統合
Oracle8i Database	Oracle Advanced Queuing	Oracle Message Broker	Oracle Message Broker
Oracle9i Database 以降	Oracle Advanced Queuing	Oracle Advanced Queuing	Oracle Advanced Queuing メッセージ・ゲートウェイ機能

カスタム・プロトコルを実装するには、次の手順を実行する必要があります。

1. 保留中のインバウンドおよびアウトバウンド・メッセージを保持するように、AQ キューを定義します。
2. AQ キューとの間でメッセージを伝播するコードを指定します。
3. 「イベント・プロトコル・タイプ (WF_AQ_PROTOCOLS)」 選択枝タイプの新しいプロトコルに対して、選択枝コードを定義します。この選択枝コードは、「標準」項目タイプに格納されます。イベント・マネージャでは、エージェントのプロトコルを検証するときに、「イベント・プロトコル・タイプ」 選択枝タイプが使用されます。3-22 ページの「[選択枝タイプの選択枝コードの作成](#)」を参照してください。
4. 新しいプロトコルに基づいてエージェントを定義します。8-119 ページの「[エージェントの作成または更新](#)」を参照してください。

エージェントがインバウンド通信を使用する場合は、システムがエージェントとの通信に使用するアドレスを指定する必要があります。アドレスの書式は、エージェントのプロトコルによって異なります。SQLNET プロトコルを使用するエージェントの場合、AQ 伝播を有効にするには、次の書式でアドレスを指定する必要があります。

```
<schema>.<queue>@<database link>
```

この書式の <schema> はキューを所有するスキーマ、<queue> はキューの名前、<database link> はキューが配置されているインスタンスへのデータベース・リンクの名前を表します。

注意： データベース・リンク名には、データベース・リンクの作成時に指定された名前を正確に入力する必要があります。たとえば、データベース・リンク名が ORA816.US.ORACLE.COM の場合は、データベース上のエージェントのアドレスに対してその名前を正確に入力する必要があります。ORA816 のように名前を省略することはできません。

ビジネス・イベント・システムで使用するデータベース・リンクの名前は、ドメイン名を含む完全修飾名である必要があります。データベース・リンクの名前を確認するには、次の構文を使用します。

```
SELECT db_link FROM all_db_links
```

『Oracle Workflow 管理者ガイド』の「データベース・リンクの作成」を参照してください。

エージェントへのキューの割当て

ワークフロー対応システムの各エージェントは、AQ キューに関連付ける必要があります。ローカル・システムは、このキューを使用してエージェントと対話します。メッセージを送信する場合、システムはキューにあるメッセージをエンキューして、宛先アドレスを設定します。メッセージを受信する場合、システムはキューにあるリスナーを実行します。『Oracle Workflow 管理者ガイド』の「キューの設定」を参照してください。

Oracle Workflow のビジネス・イベント・システム内のイベント・メッセージは、データ型 WF_EVENT_T で定義された標準の書式でエンコードされます。各エージェントは、キュー・ハンドラと呼ばれる PL/SQL パッケージに割り当て、Workflow 標準書式とエージェントのキューに必要な書式との間で変換を行う必要があります。『Oracle Workflow API リファレンス』の「イベント・メッセージ構造」を参照してください。

注意： エージェントのキューのペイロード・タイプが WF_EVENT_T の場合でも、システム固有の AQ メッセージ・プロパティを設定するために、キュー・ハンドラに割り当てる必要があります。

Oracle Workflow には、WF_EVENT_QH および WF_ERROR_QH という 2 つの標準キュー・ハンドラが用意されています。これらのキュー・ハンドラは、SQLNET 伝播を使用し、WF_EVENT_T データ型をペイロードとして使用するキューに対応しています。WF_EVENT_QH は、通常のビジネス・イベント・システム処理を扱うキューで使用します。WF_ERROR_QH は、エラー・キューでのみ使用します。

Oracle Workflow には、WF_EVENT_OJMSTEXT_QH という標準キュー・ハンドラも用意されています。このキュー・ハンドラでは、ペイロード・タイプとして SYS.AQ\$_JMS_TEXT_MESSAGE データ型が使用されます。このキュー・ハンドラによって、ビジネス・イベント・システム経由で JMS テキスト・メッセージと通信できるようになります。『Oracle Workflow API リファレンス』の「WF_EVENT_T および SYS.AQ\$_JMS_TEXT_MESSAGE 間のマッピング」を参照してください。

また、Oracle Workflow には、WF_EVENT_OMB_QH というキュー・ハンドラも用意されています。Oracle8i Database に Oracle Message Broker を実装して、HTTP などの別のプロトコルを使用してシステム間でメッセージを伝播するときは、このキュー・ハンドラを使用します。『Oracle Workflow 管理者ガイド』の「WF_EVENT_OMB_QH キュー・ハンドラの設定」および『Oracle Workflow API リファレンス』の「WF_EVENT_T および OMBAQ_TEXT_MSG 間のマッピング」を参照してください。

他の書式を必要とするキューの場合は、その書式用のカスタム・キュー・ハンドラを作成します。カスタム・キュー・ハンドラには、カスタム書式のメッセージをエンキューおよびキューするための標準 API を組み込む必要があります。6-26 ページの「[キュー・ハンドラの標準 API](#)」を参照してください。

エージェント・グループ

エージェント・グループを定義して、複数のインバウンド・エージェントを相互に関連付け、それらをイベント・サブスクリプション内および「送信」イベント・アクティビティ内でグループとして参照することもできます。エージェント・グループはエージェントの一種で、個別のメンバー・エージェントの集合で構成されます。エージェント・グループを定義すると、グループにイベント・メッセージを送信することができます。グループ内のエージェントに個別にメッセージを送信する必要はありません。

エージェント・グループは、インバウンド通信にのみ使用できます。すべてのエージェント・グループの方向は「受信」であり、受信方向のエージェントのみがエージェント・グループのメンバーになれます。

各エージェント・グループは、各グループが所属するシステムに関連付ける必要があります。ただし、他のシステムのエージェントもグループに含めることができます。

インバウンド・メッセージを受信するために、グループ内の各エージェントに対してエージェント・リスナーを必ず実行してください。

注意： エージェント・グループに対しては、エージェント・リスナーを実行できません。エージェント・リスナーは、個々のエージェントに対してのみ実行できます。

標準エージェント

Oracle Workflow をインストールすると、ビジネス・イベント・システムに対して複数の標準エージェントが自動的に定義されます。

- WF_CONTROL: ユーザー向けではない、Oracle Workflow の内部エージェント
- WF_DEFERRED: 遅延サブスクリプション処理用標準エージェント
- WF_ERROR: エラー処理用標準エージェント
- WF_IN: デフォルトのインバウンド・エージェント
- WF_JMS_IN: JMS テキスト・メッセージ用のデフォルトのインバウンド・エージェント
- WF_JMS_OUT: JMS テキスト・メッセージ用のデフォルトのアウトバウンド・エージェント
- WF_NOTIFICATION_IN: 電子メール通知の応答用の標準インバウンド・エージェント
- WF_NOTIFICATION_OUT: 電子メール通知の応答用の標準アウトバウンド・エージェント
- WF_OUT: デフォルトのアウトバウンド・エージェント

これらのエージェントでは、Oracle Workflow のインストール時に自動的に定義される標準のキューが使用されます。『Oracle Workflow 管理者ガイド』の「キューの設定」を参照してください。

WF_IN、WF_JMS_IN、WF_OUT および WF_JMS_OUT エージェントは、使用可能 / 使用不可を切り替えることができます。ただし、エージェントの定義は変更しないでください。その他のエージェントの定義も変更しないでください。

Oracle Workflow では、ビジネス・イベント・システムの遅延サブスクリプション処理、エラー処理および通知メーラーのインバウンド電子メール処理を実行するために、それぞれ標準 WF_DEFERRED エージェント、標準 WF_ERROR エージェントおよび標準 WF_NOTIFICATION_IN エージェントに対応する、エージェント・リスナー・サービス・コンポーネントが自動的に実行されます。WF_IN、WF_JMS_IN、WF_OUT および WF_JMS_OUT エージェントをイベント・メッセージの伝播に使用する場合は、WF_IN と WF_JMS_IN のリスナーおよび WF_OUT と WF_JMS_OUT の伝播もスケジューリングしてください。ただし、WF_CONTROL を使用する中間層のプロセスは直接キューからメッセージをデキューし、通知メーラーは WF_NOTIFICATION_OUT キューにあるメッセージを送信するため、WF_CONTROL および WF_NOTIFICATION_OUT エージェントの伝播はスケジューリングする必要がありません。詳細は、Oracle Applications Manager のオンライン・ヘルプ、『Oracle9i アプリケーション開発者ガイド - アドバンスド・キューイング』の「Oracle Enterprise Manager のサポート」または『Oracle Streams アドバンスド・キューイング・ユーザズ・ガイドおよびリファレンス』、『Oracle Enterprise Manager 管理者ガイド』の「分散管理」および『Oracle Workflow 管理者ガイド』の「ビジネス・イベント・システムの設定」を参照してください。

注意： Oracle Workflow には、WF_REPLAY_IN、WF_REPLAY_OUT および WF_SMTP_O_1_QUEUE という名前の、現在は使用されていない3つのエージェントもあります。

次の表は、標準 WF_CONTROL エージェントのデフォルトのプロパティを示しています。『Oracle Workflow 管理者ガイド』の「Workflow 制御キューのクリーン・アップ」を参照してください。

表 8-13

エージェント・プロパティ	値
名前	WF_CONTROL
表示名	Workflow Control Out Queue
説明	中間層プロセスへメッセージを知らせる Workflow JMS テキスト・メッセージ・キュー
プロトコル	SQLNET
アドレス	<workflow schema>.WF_CONTROL@<local database>

表 8-13 (続き)

エージェント・プロパティ	値
システム	<local system>
キュー・ハンドラ	WF_EVENT_OJMSTEXT_QH
キュー名	<workflow schema>.WF_CONTROL
方向	送信
ステータス	使用可能

次の表は、標準 WF_DEFERRED エージェントのデフォルトのプロパティを示しています。

表 8-14

エージェント・プロパティ	値
名前	WF_DEFERRED
表示名	WF_DEFERRED
説明	WF_DEFERRED
プロトコル	SQLNET
アドレス	<workflow schema>.WF_DEFERRED@<local database>
システム	<local system>
キュー・ハンドラ	WF_EVENT_QH
キュー名	<workflow schema>.WF_DEFERRED
方向	受信
ステータス	使用可能

次の表は、標準 WF_ERROR エージェントのデフォルトのプロパティを示しています。

表 8-15

エージェント・プロパティ	値
名前	WF_ERROR
表示名	WF_ERROR
説明	WF_ERROR
プロトコル	SQLNET
アドレス	<workflow schema>.WF_ERROR@<local database>

表 8-15 (続き)

エージェント・プロパティ	値
システム	<local system>
キュー・ハンドラ	WF_ERROR_QH
キュー名	<workflow schema>.WF_ERROR
方向	受信
ステータス	使用可能

次の表は、標準 WF_IN エージェントのデフォルトのプロパティを示しています。

表 8-16

エージェント・プロパティ	値
名前	WF_IN
表示名	WF_IN
説明	WF_IN
プロトコル	SQLNET
アドレス	<workflow schema>.WF_IN@<local database>
システム	<local system>
キュー・ハンドラ	WF_EVENT_QH
キュー名	<workflow schema>.WF_IN
方向	受信
ステータス	使用可能

次の表は、標準 WF_JMS_IN エージェントのデフォルトのプロパティを示しています。

表 8-17

エージェント・プロパティ	値
名前	WF_JMS_IN
表示名	Workflow JMS In Queue
説明	Workflow JMS テキスト・メッセージ・キュー
プロトコル	SQLNET
アドレス	<workflow schema>.WF_JMS_IN@<local database>

表 8-17 (続き)

エージェント・プロパティ	値
システム	<local system>
キュー・ハンドラ	WF_EVENT_OJMSTEXT_QH
キュー名	<workflow schema>.WF_JMS_IN
方向	受信
ステータス	使用可能

次の表は、標準 WF_JMS_OUT エージェントのデフォルトのプロパティを示しています。

表 8-18

エージェント・プロパティ	値
名前	WF_JMS_OUT
表示名	Workflow JMS Out Queue
説明	Workflow JMS テキスト・メッセージ・キュー
プロトコル	SQLNET
アドレス	<workflow schema>.WF_JMS_OUT@<local database>
システム	<local system>
キュー・ハンドラ	WF_EVENT_OJMSTEXT_QH
キュー名	<workflow schema>.WF_JMS_OUT
方向	送信
ステータス	使用可能

次の表は、標準 WF_NOTIFICATION_IN エージェントのデフォルトのプロパティを示しています。『Oracle Workflow 管理者ガイド』の「通知メーカーの導入」を参照してください。

表 8-19

エージェント・プロパティ	値
名前	WF_NOTIFICATION_IN
表示名	Workflow Notification In Queue
説明	Workflow 受信通知応答キュー
プロトコル	SQLNET

表 8-19 (続き)

エージェント・プロパティ	値
アドレス	<workflow schema>.WF_NOTIFICATION_IN@<local database>
システム	<local system>
キュー・ハンドラ	WF_EVENT_OJMSTEXT_QH
キュー名	<workflow schema>.WF_NOTIFICATION_IN
方向	受信
ステータス	使用可能

次の表は、標準 WF_NOTIFICATION_OUT エージェントのデフォルトのプロパティを示しています。『Oracle Workflow 管理者ガイド』の「通知メーラーの導入」を参照してください。

表 8-20

エージェント・プロパティ	値
名前	WF_NOTIFICATION_OUT
表示名	Workflow Notification Out Queue
説明	Workflow 通知アウトバウンドキュー
プロトコル	SQLNET
アドレス	<workflow schema>.WF_NOTIFICATION_OUT@<local database>
システム	<local system>
キュー・ハンドラ	WF_EVENT_OJMSTEXT_QH
キュー名	<workflow schema>.WF_NOTIFICATION_OUT
方向	送信
ステータス	使用可能

次の表は、標準 WF_OUT エージェントのデフォルトのプロパティを示しています。

表 8-21

エージェント・プロパティ	値
名前	WF_OUT
表示名	WF_OUT
説明	WF_OUT
プロトコル	SQLNET
アドレス	<workflow schema>.WF_OUT@<local database>
システム	<local system>
キュー・ハンドラ	WF_EVENT_QH
キュー名	<workflow schema>.WF_OUT
方向	送信
ステータス	使用可能

外部システムのエージェント

イベントと相互に通信するシステムは、イベント・メッセージを相互に送信するために、それぞれのインバウンド・エージェント定義を格納する必要があります。

両方に Oracle Workflow がインストールされている 2 つのシステム間の通信において、Oracle Workflow では、他方のシステムのインバウンド・エージェント定義を自分のローカル・システムのイベント・マネージャに自動的にコピーする外部システム登録プロシージャが用意されています。8-126 ページの「[外部システムの登録](#)」を参照してください。

ローカルの Workflow 対応システムが Workflow 非対応システムと通信する場合は、Workflow 非対応システムに、ビジネス・イベント・システムのイベント・メッセージを処理するための独自の外部伝播エージェントが必要です。

- Workflow 非対応システムのインバウンド・エージェントは、ビジネス・イベント・システムのアウトバウンド・キューからイベント・メッセージをデキューし、これらメッセージの内容を処理する必要があります。インバウンド・エージェントは、システムがそのエージェントと通信できるように、アドレスを持っている必要があります。
- Workflow 非対応システムのアウトバウンド・エージェントは、ビジネス・イベント・システムのインバウンド・キューへ、イベント・メッセージを適切な形式でエンキューする必要があります。

ローカル・システムのイベント・マネージャ内で、外部 Workflow 非対応システムのインバウンド・エージェントを定義する必要があります。同様に、外部システムのアウトバウンド・エージェントを定義することもできます。

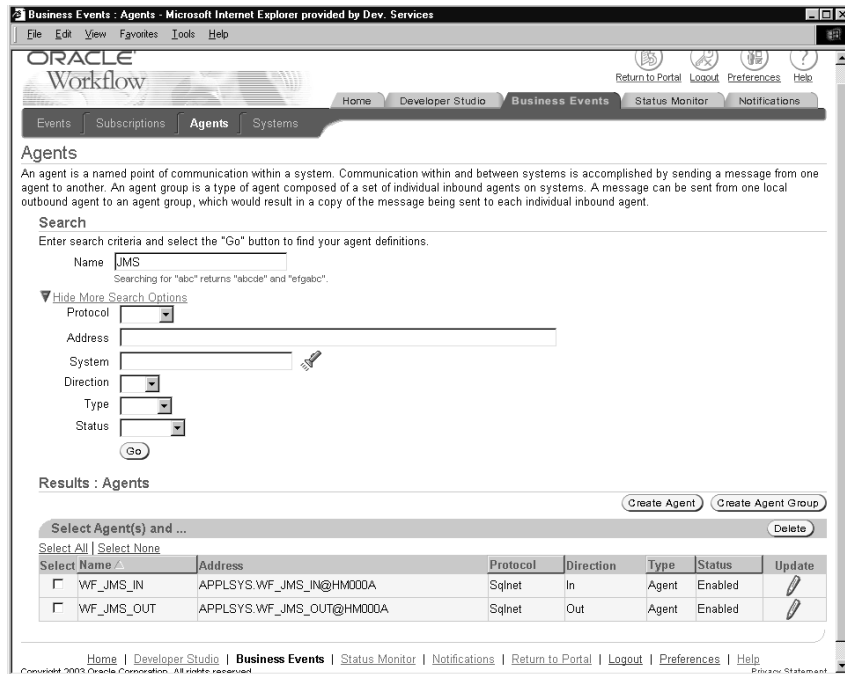
1. Workflow 非対応システムのエージェントを定義する前に、「システムの作成」ページを使用して、システム自体を定義する必要があります。8-132 ページの「[システムの作成または更新](#)」を参照してください。
2. その後、他のエージェントと同じ手順に従い、「エージェントの作成」ページを使用して Workflow 非対応システムのエージェントを定義することができます。8-119 ページの「[エージェントの作成または更新](#)」を参照してください。
 - エージェントは、そのエージェントが所属する Workflow 非対応システムに関連付ける必要があります。
 - エージェントと通信するプロトコルを指定する必要があります。
 - インバウンド・エージェントの場合は、エージェントと通信するアドレスも指定する必要があります。
 - エージェントが Oracle Advanced Queuing (AQ) キューとして実装されていない場合は、キュー名とキュー・ハンドラを空白のままにしておくことができます。

➤ エージェントの表示と管理

1. システム管理者によって指定された職責およびナビゲーション・パスを使用して、Web ブラウザでイベント・マネージャにアクセスします。次に水平ナビゲーションの「エー

エージェント」を選択します。A-2 ページの「[Oracle Workflow 開発者のナビゲーション・パス](#)」を参照してください。

Oracle Workflow の管理者用 Web ページの「ビジネス・イベント」タブまたは下部にある「ビジネス・イベント」リンクを選択して、イベント・マネージャへ移動することができます。



2. 特定のエージェントを検索する場合は、「検索」領域に検索基準を入力します。主要な検索オプションとして、次の項目を使用できます。
 - 名前： 表示するエージェントの内部名を入力します。内部名に含まれる数文字を入力しても、エージェントを検索することができます。
3. エージェントを検索するための追加検索基準として、他の特性を入力することもできます。追加オプションが非表示になっている場合に表示するには、「詳細検索の表示オプション」リンクを選択します。追加オプションが表示されている場合に非表示にする場合は、「詳細検索の非表示オプション」リンクを選択します。追加検索基準として、次の項目を使用できます。
 - プロトコル： 表示するエージェントのプロトコルを指定します。
 - アドレス： 表示するエージェントのアドレスを入力します。

- システム： 表示するエージェントのシステムを選択します。フィールドの検索アイコンを選択すると、選択可能な値のリストが表示されます。システムの内部名または表示名を基準として検索することができます。8-85 ページの「[値リストの使用](#)」を参照してください。
 - 方向： 表示するエージェントの方向として、「受信」または「送信」を選択します。
 - タイプ： 表示するエージェントのタイプとして「エージェント」または「グループ」を選択します。
 - ステータス： 表示するエージェントのステータスとして「使用可能」または「使用不能」を選択します。
4. 「実行」ボタンを選択して、検索を実行します。
 5. 「結果」領域に、検索基準と一致したエージェントが表示されます。エージェントごとに、リストに内部名、アドレス、プロトコル、方向、タイプおよびステータスが表示されます。「名前」、「プロトコル」、「タイプ」または「ステータス」の列見出しを選択すると、その列を基準にしてリストがソートされます。
 6. ワークフロー管理者権限を持っている場合は、管理アイコンやボタンを使用して、管理操作を実行できます。
 - エージェントを更新するには、そのエージェントの「更新」列にある更新アイコンを選択します。8-119 ページの「[エージェントの作成または更新](#)」および 8-122 ページの「[エージェント・グループの作成または更新](#)」を参照してください。
 - エージェントを削除するには、エージェントを選択して「削除」ボタンを選択します。「削除」ページで、「適用」ボタンを選択します。「取消」ボタンを選択し、エージェントを削除しないで元のページに戻ることもできます。

注意： 削除できるエージェントは、サブスクリプションから参照されていないエージェントおよびエージェント・グループに属していないエージェントのみです。

- 新しい個別のエージェントを作成するには、「エージェントの作成」ボタンを選択します。8-119 ページの「[エージェントの作成または更新](#)」を参照してください。
- 新しいエージェント・グループを作成するには、「エージェント・グループの作成」ボタンを選択します。8-122 ページの「[エージェント・グループの作成または更新](#)」を参照してください。

注意： エージェントに加えた変更がメッセージ伝播に必要な物理実装に影響する場合は、伝播設定を再度チェックする必要があります。詳細は、Oracle Applications Manager のオンライン・ヘルプ、『Oracle9i アプリケーション開発者ガイド-アドバンスト・キューイング』の「Oracle Enterprise Manager のサポート」または『Oracle Streams アドバンスト・キューイング・ユーザーズ・ガイドおよびリファレンス』、『Oracle Enterprise Manager 管理者ガイド』の「分散管理」および『Oracle Workflow 管理者ガイド』の「ビジネス・イベント・システムの設定」を参照してください。

▶ **エージェントの作成または更新**

1. 「エージェントの作成」または「エージェントの更新」ページに移動します。選択したエージェントに対して直前に定義した情報が「エージェントの更新」ページの各フィールドに表示されることを除くと、「エージェントの作成」ページと「エージェントの更新」ページは同一です。

注意： Oracle Workflow がインストールされた外部システムのインバウンド・エージェントに対しては、エージェント定義の作成に「エージェントの作成」ページを使用しないでください。そのかわり、「外部システム登録」ページを使用して、インバウンド・エージェント定義も含めた外部システムのシステム識別子情報を自動的に登録する必要があります。8-126 ページの「[外部システムの登録](#)」を参照してください。

- エージェントの内部名を「名前」フィールドに入力します。エージェントの内部名は、エージェントのシステム内で一意である必要があります。

注意： 内部名には大文字のみを使用する必要があります。一重引用符 (')、二重引用符 (") またはスペースを含めないでください。

- エージェントの表示名を入力します。
- オプションで、エージェントの説明を入力します。
- エージェントが使用するメッセージ通信プロトコルを選択します。
- エージェントがシステムへのインバウンド通信を利用する場合は、エージェントのアドレスを入力します。アドレスの書式は、選択するプロトコルによって異なります。

SQLNET プロトコルを使用するエージェントの場合、AQ 伝播を有効にするには、次の書式でアドレスを指定する必要があります。

```
<schema>.<queue>@<database link>
```

<schema> はキューを所有するスキーマ、<queue> はキューの名前、<database link> はキューが配置されているインスタンスへのデータベース・リンクを表します。

注意： データベース・リンク名には、データベース・リンクの作成時に指定された名前を正確に入力する必要があります。『Oracle Workflow 管理者ガイド』の「データベース・リンクの作成」を参照してください。

7. エージェントが定義されているシステムを選択します。フィールドの検索アイコンを選択すると、選択可能な値のリストが表示されます。システムの内部名または表示名を基準として検索することができます。8-85 ページの「[値リストの使用](#)」を参照してください。
8. エージェントのキュー・ハンドラを入力します。キュー・ハンドラは PL/SQL パッケージの 1 つで、Workflow イベント・メッセージ書式 (WF_EVENT_T) とエージェントに関連付けられているキューに必要なメッセージ書式との間で変換を行います。6-26 ページの「[キュー・ハンドラの標準 API](#)」を参照してください。

次の書式で、キュー・ハンドラを入力します。

`<package_name>`

注意： キュー・ハンドラ名は、すべて大文字で入力する必要があります。

注意： 非 Oracle システム上のエージェントのような、キューとして実装されていないエージェントの場合は、「キュー・ハンドラ」フィールドを空白のままにしておきます。

9. ローカル・システムがエージェントとの対話に使用するキューの名前を入力します。このキュー名は、ローカル・システムのみが参照するため、このシステムの有効範囲に対応し、データベース・リンクを必要としない名前である必要があります。次の書式を使用してキュー名を指定します。

`<schema>.<queue>`

`<schema>` はキューを所有するスキーマ、`<queue>` はキュー名を表します。

注意： キュー名は、すべて大文字で入力する必要があります。

注意： 非 Oracle システム上のエージェントのような、キューとして実装されていないエージェントの場合は、「キュー」フィールドを空白のままにしておきます。

10. 「方向」フィールドで、システムへのインバウンド通信をサポートするエージェントの場合は「受信」を選択し、システムからのアウトバウンド通信をサポートするエージェントの場合は「送信」を選択します。
11. エージェントのステータスとして「使用可能」または「使用不能」を選択します。エージェントを使用不能にしても、その定義はイベント・マネージャ内に残り、参照できます。ただし、有効なサブスクリプションでは使用できません。
12. 「適用」ボタンを選択して、エージェントを保存します。「取消」ボタンを選択し、エージェントを保存しないで元のページに戻ることもできます。

▶ エージェント・グループの作成または更新

1. 「エージェント・グループの作成」または「エージェント・グループの更新」ページに移動します。選択したエージェント・グループに対して直前に定義した情報が「エージェント・グループの更新」ページの各フィールドに表示されることを除くと、「エージェント・グループの作成」ページと「エージェント・グループの更新」ページは同一です。

Oracle Workflow
Return to Portal Logout Preferences Help

Home Developer Studio Business Events Status Monitor Notifications

Events Subscriptions Agents Systems

Business Events Agents > Create Agent Group

Create Agent Group

An agent group is a type of agent composed of a set of individual inbound agents on systems. A message can be sent from one local outbound agent to an agent group, which would result in a copy of the message being sent to each individual inbound agent. Cancel Apply

* Indicates required field

* Name

* Display Name

Description

* Protocol

* Address
(Format: schema.queueName@databaseLink)

* System

* Queue Handler
(Queue Handler is the PL/SQL package which includes enqueue & dequeue logic. Format: <package>)

Direction

* Status

Agents in Group Add Agents to Group

Select	Name	Address	Protocol	Direction	Type	Status
	No agents found.					

Home | Developer Studio | Business Events | Status Monitor | Notifications | Return to Portal | Logout | Preferences | Help
Copyright 2003 Oracle Corporation. All rights reserved. [Privacy Statement](#)

2. エージェント・グループの内部名を「名前」フィールドに入力します。エージェント・グループの内部名は、エージェント・グループのシステム内で一意である必要があります。

注意： 内部名には大文字のみを使用する必要があります。一重引用符 (')、二重引用符 (") またはスペースを含めないでください。

3. エージェント・グループの表示名を入力します。
4. オプションで、エージェント・グループの説明を入力します。
5. エージェント・グループが定義されているシステムを選択します。フィールドの検索アイコンを選択すると、選択可能な値のリストが表示されます。システムの内部名または表示名を基準として検索することができます。8-85 ページの「[値リストの使用](#)」を参照してください。

注意： エージェント・グループは特定のシステムに定義されますが、グループ内のメンバーとして、他のシステムのエージェントも含めることができます。

エージェント・グループはインバウンド通信にのみ使用されるため、エージェント・グループの方向は自動的に「受信」に設定されます。

6. エージェント・グループのステータスとして「使用可能」または「使用不能」を選択します。エージェント・グループを使用不能にしても、その定義はイベント・マネージャ内に残り、参照できます。ただし、有効なサブスクリプションでは使用できません。
7. 「適用」ボタンを選択して、エージェント・グループを保存します。「取消」ボタンを選択し、エージェント・グループを保存しないで元のページに戻ることもできます。
8. 「グループ内エージェント」領域に、グループのメンバーであるエージェントが表示されます。エージェントごとに、リストに内部名、アドレス、プロトコル、方向、タイプおよびステータスが表示されます。「名前」、「プロトコル」、「方向」、「タイプ」または「ステータス」の列見出しを選択すると、その列を基準にしてリストがソートされます。

注意： エージェント・グループには、個別のエージェントのみをメンバーとして登録できます。イベント・グループには、別のグループを登録できません。

9. メンバー・エージェントをグループに追加するには、「エージェントをグループに追加」ボタンを選択します。
10. 表示された「エージェントをグループに追加」ページで、検索基準を入力し、追加するエージェントを検索します。主要な検索オプションとして、次の項目を使用できます。
 - 名前: 追加するエージェントの内部名を入力します。内部名に含まれる数文字を入力しても、エージェントを検索することができます。

11. イベントを検索するための追加検索基準として、他の特性を入力することもできます。追加オプションが非表示になっている場合に表示するには、「詳細検索の表示オプション」リンクを選択します。追加オプションが表示されている場合に非表示にする場合は、「詳細検索の非表示オプション」リンクを選択します。追加検索基準として、次の項目を使用できます。
 - プロトコル： 追加するエージェントのプロトコルを指定します。
 - アドレス： 追加するエージェントのアドレスを入力します。
 - システム： 追加するエージェントのシステムを選択します。フィールドの検索アイコンを選択すると、選択可能な値のリストが表示されます。システムの内部名または表示名を基準として検索することができます。8-85 ページの「[値リストの使用](#)」を参照してください。
 - ステータス： 追加するエージェントのステータスとして「使用可能」または「使用不能」を選択します。
12. 「実行」ボタンを選択して、検索を実行します。
13. 「結果」領域に、検索基準と一致したエージェントが表示されます。エージェントごとに、リストに内部名、アドレス、プロトコル、方向、タイプおよびステータスが表示されます。
14. エージェント・グループに追加するエージェントを選択し、「エージェントをグループに追加」ボタンを選択します。
15. エージェント・グループに追加する別のエージェントを検索するための、別の検索基準を入力することもできます。
16. エージェントをエージェント・グループに追加した後で、「適用」ボタンを選択してエージェント・グループ・メンバーを保存します。「取消」ボタンを選択し、エージェント・グループ・メンバーを保存しないで元のページに戻ることもできます。
17. グループからメンバー・エージェントを削除するには、「エージェント・グループの作成」または「エージェント・グループの更新」ページで削除するエージェントを選択し、「削除」ボタンを選択します。「警告」ページで、「Yes」ボタンを選択します。「いいえ」ボタンを選択し、エージェントを削除しないで元のページに戻ることもできます。

注意： メンバー・エージェントをエージェント・グループから削除しても、各エージェントのエージェント定義は削除されません。各エージェントは、イベント・マネージャ内に残ります。

システム

システムとは、ホスト・マシンやデータベース・インスタンスなど、論理的に独立したソフトウェア環境のことです。イベント・マネージャ内でイベントを交換するシステムを定義する必要があります。

システムを定義する際に、イベント・マネージャのオブジェクト定義の更新の送信元となるマスター・システムに関連付けるかどうかを指定できます。

各システムでは、アドレス指定可能な通信ポイント（エージェント）を1つ以上公開できます。システムを定義したら、ビジネス・イベントの交換に使用するエージェントをシステム内に定義する必要があります。8-105 ページの「[エージェント](#)」を参照してください。

ローカル・システム

Oracle Workflow をデータベース内にインストールすると、そのデータベースはイベント・マネージャのシステムとして自動的に定義され、「グローバル・ワークフロー設定」ページのローカル・システムとして設定されます。次の表は、ローカル・システム定義のデフォルト・プロパティの一覧です。

表 8-22

システム・プロパティ	値
名前	<database global name>
表示名	<database global name>
説明	Oracle Workflow Configuration Assistant によって作成されたローカル・システム
マスター	(空白)

ローカル・システム定義は、必要に応じて更新できます。

ローカル・システムのステータスは、デフォルトでは「使用可能」に設定されます。ビジネス・イベント・システムの設定が終了したら、「グローバル・ワークフロー設定」ページを使用して、イベントの処理に必要なシステム・ステータスを設定できます。『Oracle Workflow 管理者ガイド』の「グローバル・ユーザー設定の設定」を参照してください。

外部システムの登録

システム間でビジネス・イベントを送信するには、イベント・メッセージの受信者として宛先システムをソース・システムに登録する必要があります。システムの登録とは、宛先システムおよびそのインバウンド・エージェントをソース・システムのイベント・マネージャに定義して、ソース・システムのイベント・メッセージを宛先エージェントに送信できるようにすることです。システムの登録は、システムのサインアップとも呼ばれます。

通常、両方のシステムが互いにメッセージを送受信できるように、登録し合う必要があります。

両方のシステムに Oracle Workflow がインストールされている場合、Oracle Workflow には、2つのシステム間で自動的な外部システム登録をする上で役立つ Web ページが用意されています。Workflow 対応のシステムと Workflow 非対応システム間の通信については、ソース・システム内に必要な宛先システムおよびエージェントに関する情報を手動で格納する必要があります。

宛先システムを登録してソース・システムのイベント・メッセージを受信するには、次の手順を実行します。

1. ローカル・システムおよびインバウンド・エージェントの定義を宛先システムから取得します。これらの情報は、システム識別子情報となります。

注意： システム識別子情報には、システムと個別のインバウンド・エージェントについての情報のみが含まれます。システム識別子には、エージェント・グループ定義は含まれません。

- 宛先システムに Oracle Workflow がインストールされている場合は、宛先システムの「ローカル・システム識別子」ページを使用して、システム識別子情報を含む XML 文書を生成し、保存することができます。8-133 ページの「ローカル・システムの識別子情報の取得」を参照してください。

注意： 宛先システムにインストールされている Oracle Workflow へのアクセス権がない場合は、宛先システムのワークフロー管理者にこの手順の実行を依頼してください。

- 宛先システムに Oracle Workflow がインストールされていない場合は、システムおよびインバウンド・エージェントを定義するのに必要な情報を手動で収集する必要があります。
2. 宛先システムの識別子情報をソース・システムのイベント・マネージャに登録します。
 - ソース・システムと宛先システムの両方に Oracle Workflow がインストールされている場合は、ソース・システムの「外部システム登録」Web ページで、宛先システムからのイベント・データとしてシステム識別子の XML 文書を使用し、システムのサインアップ・イベントを呼び出して情報を登録することができます。ソース・

システムでシステムのサインアップ・イベントを呼び出すと、事前定義済のサブスクリプションが Oracle Workflow によって実行され、システム識別子情報がソース・システムのイベント・マネージャにロードされます。8-134 ページの「外部システムの登録」を参照してください。

注意： ソース・システムにインストールされている Oracle Workflow へのアクセス権がない場合は、ソース・システムのワークフロー管理者にこの手順を実行するよう依頼してください。

- ソース・システムに Oracle Workflow がインストールされていて宛先システムにはインストールされていない場合、ソース・システムのイベント・マネージャ内に宛先システムのシステムおよびエージェントの定義を手動で作成する必要があります。8-132 ページの「システムの作成または更新」および 8-119 ページの「エージェントの作成または更新」を参照してください。
- ソース・システムに Oracle Workflow がインストールされていなくて宛先システムにインストールされている場合は、Workflow 非対応のソース・システムの要件に従って、ソース・システム内に宛先システムのシステムおよびインバウンド・エージェントの情報を手動で格納する必要があります。宛先システムの「ローカル・システム識別子」ページで生成されたシステム識別子の XML 文書を利用することはできますが、Workflow 非対応のソース・システムに情報を格納するために必要な、なんらかの手続きを行う必要があります。

システムの同期

システムの同期とは、ソース・システムに定義されているすべてのイベント・マネージャ・オブジェクトを、ターゲット・システムにレプリケートすることです。イベント・システムの同期イベントを使用して、2つの Workflow 対応システムを互いに同期させることができます。

▶ システムの同期

1. ソース・システムおよびターゲット・システムを相互に登録します。8-126 ページの「外部システムの登録」を参照してください。
2. ソース・システム上で、「ローカル」ソース・タイプを持つ「シード・イベント・グループ」への事前定義済サブスクリプションを、次のように変更します。
 - イベント・メッセージを受信するターゲット・システムに対してインバウンド・エージェントを指定するか、イベント・メッセージをターゲット・システムに送信するワークフロー・プロセスを指定します。

注意： イベント・メッセージを複数のターゲット・システムに送信する場合は、イベント・メッセージを受信するエージェント・グループを指定することができます。

- サブスクリプションを使用可能にします。

注意： ソース・システムにインストールされている Oracle Workflow へのアクセス権がない場合は、ソース・システムのワークフロー管理者にこの手順を実行するよう依頼してください。

3. ターゲット・システム上で、「外部」ソース・タイプを持つ「シード・イベント・グループ」への事前定義済サブスクリプションを、使用可能にします。

注意： ターゲット・システムにインストールされている Oracle Workflow へのアクセス権がない場合は、ターゲット・システムのワークフロー管理者にこの手順の実行を依頼してください。

4. ソース・システム上で、そのイベントの「ビジネス・イベントのテスト」ページを使用してイベント・システムの同期イベント (`oracle.apps.wf.event.all.sync`) を呼び出します。一意のイベント・キーを入力してください。イベント・データは入力しないでください。8-84 ページの「[テスト・イベントの呼出し](#)」を参照してください。

注意： ソース・システムにインストールされている Oracle Workflow へのアクセス権がない場合は、ソース・システムのワークフロー管理者にこの手順を実行するよう依頼してください。

ソース・システム上でイベント・システムの同期イベントが呼び出されると、「ローカル」ソース・タイプを持つ「シード・イベント・グループ」へのサブスクリプションがトリガーされます。イベント・マネージャがイベント・メッセージを生成します。このメッセージには、ローカル・システム上のイベント、イベント・グループ、システム、エージェント、エージェント・グループおよびサブスクリプションを含む、すべてのイベント・マネージャ・オブジェクトの定義が含まれます。イベント・メッセージは、ターゲット・システム上の特定のインバウンド・エージェント、またはイベント・メッセージをターゲット・システムに送信する特定のワークフロー・プロセスに送信されません。

ターゲット・システムにイベント・システムの同期イベントが着信すると、「外部」ソース・タイプを持つ「シード・イベント・グループ」へのサブスクリプションがトリガーされます。イベント・メッセージのオブジェクト定義がターゲット・システムのイベント・マネージャにロードされます。必要に応じて、新しい定義が作成されたり、既存の定義が更新されます。

自動レプリケーション

手順 2 および 3 で事前定義済みのサブスクリプションを使用可能にすると、これらのサブスクリプションは、ソース・システムのイベント・マネージャのオブジェクト定義に加えられた変更をレプリケートします。イベント、イベント・グループ・メンバー、システム、エージェント、エージェント・グループ・メンバーまたはサブスクリプションの作成、更新または削除を行うたびに、対応する事前定義済みのイベントが **Oracle Workflow** によって呼び出されます。これらのイベントにより、ソース・システム上の「シード・イベント・グループ」への「ローカル」サブスクリプションがトリガーされ、オブジェクト定義データがターゲット・システムに送信されます。ターゲット・システム上の「シード・イベント・グループ」への「外部」サブスクリプションは、そのデータを受信して、イベント・マネージャのオブジェクト定義を追加、更新または削除します。

ソース・システムの変更をターゲット・システムに自動レプリケートする処理を停止する場合は、システムの同期の完了後にサブスクリプションを使用不可にするか、変更に対応する事前定義済みのイベントを使用不可にします。

マスター/コピー・システム

必要に応じて、特定のシステムをマスター・システムとして設定できます。このようにすると、マスター・システムのイベント・マネージャのオブジェクト定義を関連付けられたコピー・システムにレプリケートすることができます。ただし、コピー・システムのオブジェクト定義への変更は、マスター・システムにレプリケートされません。マスター/コピーのレプリケーションをセットアップするには、ターゲット・コピー・システムをソース・マスター・システムに同期させる手順を通常どおりに実行します。次に、オブジェクト定義がコピー・システムから送信されないようにするために、コピー・システム上の「シード・イベント・グループ」への「ローカル」サブスクリプションを使用不可にします。オブジェクト定義がマスター・システムに着信しないようにするために、マスター・システム上の「シード・イベント・グループ」への「外部」サブスクリプションも使用不可にします。

関連項目：

9-2 ページ [「事前定義済ワークフロー・イベント」](#)

9-9 ページ [「イベント・システムの同期イベント」](#)

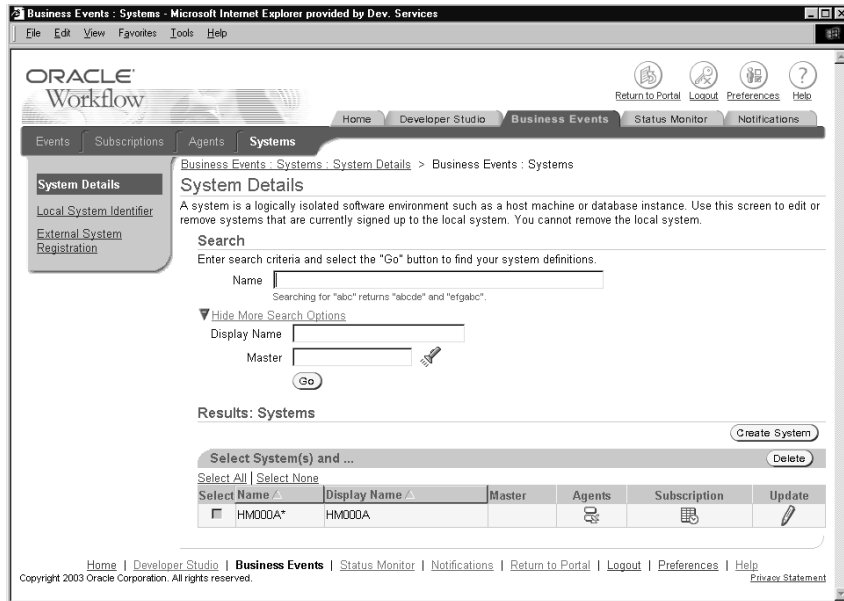
9-10 ページ [「シード・イベント・グループ」](#)

8-97 ページ [「イベント・サブスクリプションの表示と管理」](#)

▶ システムの表示と管理

1. システム管理者によって指定された職責およびナビゲーション・パスを使用して、Web ブラウザでイベント・マネージャにアクセスします。次に水平ナビゲーションで「システム」を選択し、サイド・ナビゲーションで「システム詳細」を選択します。A-2 ページの [「Oracle Workflow 開発者のナビゲーション・パス」](#) を参照してください。

Oracle Workflow の管理者用 Web ページの「ビジネス・イベント」タブまたは下部にある「ビジネス・イベント」リンクを選択して、イベント・マネージャへ移動することができます。



- 特定のシステムを検索する場合は、「検索」領域に検索基準を入力します。主要な検索オプションとして、次の項目を使用できます。
 - 名前： 表示するシステムの内部名を入力します。内部名に含まれる数文字を入力しても、システムを検索することができます。
- システムを検索するための追加検索基準として、他の特性を入力することもできます。追加オプションが非表示になっている場合に表示するには、「詳細検索の表示オプション」リンクを選択します。追加オプションが表示されている場合に非表示にする場合は、「詳細検索の非表示オプション」リンクを選択します。追加検索基準として、次の項目を使用できます。
 - 表示名： 表示するシステムの表示名を入力します。
 - マスター： 表示するシステムのマスター・システムを選択します。フィールドの検索アイコンを選択すると、選択可能な値のリストが表示されます。システムの内部名または表示名を基準として検索することができます。8-85 ページの「[値リストの使用](#)」を参照してください。
- 「実行」ボタンを選択して、検索を実行します。

5. 「結果」領域に、検索基準と一致したシステムが表示されます。システムごとに、リストに内部名、表示名およびマスター・システムが表示されます。ローカル・システムの内部名には、アスタリスク (*) のマークが付きます。「名前」または「表示名」の列見出しを選択すると、その列を基準にしてリストがソートされます。
6. システム上のエージェントを表示するには、そのシステムの「エージェント」列にあるエージェント・アイコンを選択します。「エージェントの表示:システム」ページに、そのシステムのエージェントのリストが表示されます。

ワークフロー管理者権限を持っている場合は、そのエージェントの「更新」列にある更新アイコンを選択して、エージェントを更新することができます。8-119 ページの「[エージェントの作成または更新](#)」を参照してください。
7. 特定のシステムのサブスクリプションを表示するには、そのシステムの「サブスクリプション」列にあるサブスクリプション・アイコンを選択します。「サブスクリプション:システム」ページに、システムのサブスクリプションのリストが表示されます。

注意: システムにサブスクリプションが指定されていない場合、「サブスクリプション」列には空のサブスクリプション・アイコンが表示されません。サブスクリプションが指定されているシステムの場合は、サブスクリプション・アイコンが正式に表示されます。

- ワークフロー管理者権限を持っている場合は、「予約の作成」ボタンを選択して、新しいサブスクリプションをシステムに定義することができます。「イベント予約の作成」ページが表示され、システム名がサブスクライバとして「システム」フィールドに自動的に入力されます。8-99 ページの「[イベント・サブスクリプションの作成または更新](#)」を参照してください。
- ワークフロー管理者権限を持っている場合は、そのサブスクリプションの「更新」列にある更新アイコンを選択して、既存のサブスクリプションを更新することができます。8-99 ページの「[イベント・サブスクリプションの作成または更新](#)」を参照してください。

注意: カスタマイズ・レベルが「制限」のシードされたサブスクリプションの場合は、サブスクリプションのステータスのみを更新できます。カスタマイズ・レベルが「コア」のシードされたサブスクリプションの場合、サブスクリプション定義は表示できますが、プロパティは更新できません。

リンクを選択して、前のページに戻ることもできます。

8. ワークフロー管理者権限を持っている場合は、「システム詳細」ページの管理アイコンやボタンを使用して、管理操作を実行することができます。

- システムを更新するには、そのシステムの「更新」列にある更新アイコンを選択します。8-132 ページの「[システムの作成または更新](#)」を参照してください。
- システムを削除するには、システムを選択して「削除」ボタンを選択します。「警告」ページで、「Yes」ボタンを選択します。「いいえ」ボタンを選択し、システムを削除しないで元のページに戻ることもできます。

注意： 削除できるシステムは、エージェントが定義されていないシステムまたはサブスクリプションから参照されていないシステムのみです。また、ローカル・システムは削除できません。

- 新しいシステム定義を作成するには、「システムの作成」ボタンを選択します。8-132 ページの「[システムの作成または更新](#)」を参照してください。

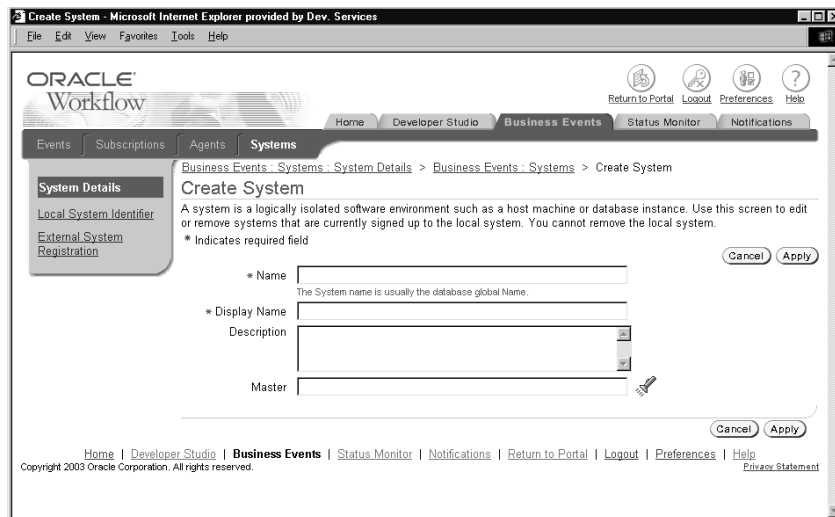
注意： Workflow 非対応の外部システムに対しては、「システムの作成」ページでのみ手動でシステム定義を行う必要があります。そのかわり、Workflow 対応のシステムの場合は、「外部システム登録」ページを使用してシステム識別子情報を自動的に登録する必要があります。8-126 ページの「[外部システムの登録](#)」を参照してください。

▶ **システムの作成または更新**

1. 「システムの作成」または「システムの更新」ページに移動します。選択したシステムに対して直前に定義した情報が「システムの更新」ページの各フィールドに表示されることを除くと、「システムの作成」ページと「システムの更新」ページは同一です。

注意： Workflow 非対応の外部システムに対しては、「システムの作成」ページでのみ手動でシステム定義を行う必要があります。そのかわり、Workflow 対応のシステムの場合は、「外部システム登録」ページを使用してシステム識別子情報を自動的に登録する必要があります。8-126 ページの「[外部システムの登録](#)」を参照してください。

「システムの更新」ページを使用して、Workflow 対応および非対応の両方のシステムを更新することができます。



2. システムの内部名を「名前」フィールドに入力します。

注意： 内部名には大文字のみを使用する必要があります。一重引用符 (')、二重引用符 (") またはスペースを含めないでください。

3. システムの表示名を入力します。
4. オプションで、システムの説明を入力します。
5. オプションで、イベント・マネージャのオブジェクト定義の更新の送信元となる、マスター・システムを入力します。フィールドの検索アイコンを選択すると、選択可能な値のリストが表示されます。システムの内部名または表示名を基準として検索することができます。8-85 ページの「[値リストの使用](#)」を参照してください。
6. 「適用」ボタンを選択して、システムを保存します。「取消」ボタンを選択し、システムを保存しないで元のページに戻ることもできます。

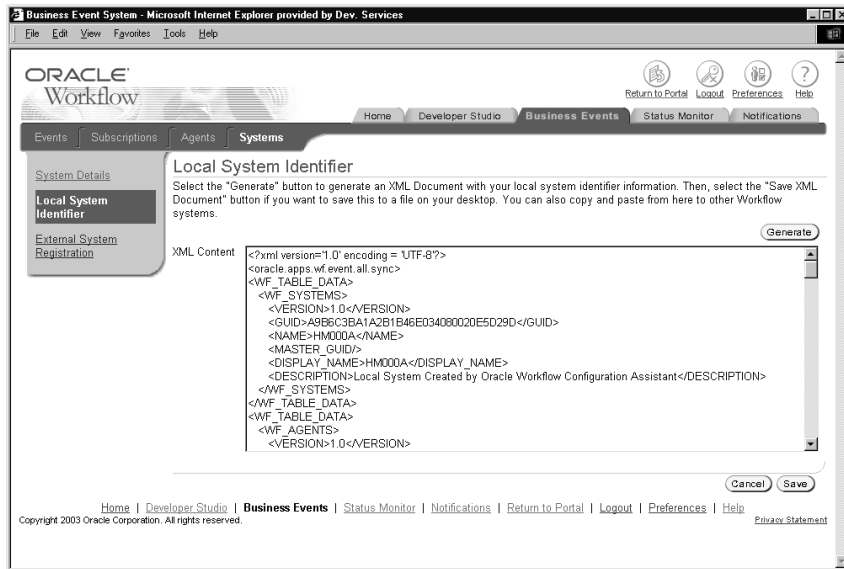
▶ ローカル・システムの識別子情報の取得

1. システム管理者によって指定された職責およびナビゲーション・パスを使用し、Web ブラウザで、宛先システムとして登録するシステムのイベント・マネージャにアクセスします。次に水平ナビゲーションで「システム」を選択し、サイド・ナビゲーションで「ローカル・システム識別子」を選択します。A-2 ページの「[Oracle Workflow 開発者のナビゲーション・パス](#)」を参照してください。

Oracle Workflow の管理者用 Web ページの「ビジネス・イベント」タブまたは下部にある「ビジネス・イベント」リンクを選択して、イベント・マネージャへ移動することができます。

注意： 宛先システムにインストールされている Oracle Workflow へのアクセス権がない場合は、宛先システムのワークフロー管理者にこの手順の実行を依頼してください。

- 「生成」ボタンを選択します。Oracle Workflow によって、ローカル・システムおよびそのインバウンド・エージェントの定義を含むローカル・システム識別子の XML 文書が作成され、「XML コンテンツ」フィールドにその文書が表示されます。



- 「適用」ボタンを選択して、XML ファイルとして XML 文書をファイル・システムに保存します。これにより、このシステムをソース・システムに登録するときに、イベント・データとして入力することができます。8-126 ページの「外部システムの登録」を参照してください。

「取消」ボタンを選択し、XML 文書を保存しないで元のページに戻ることもできます。

▶ 外部システムの登録

- システム管理者によって指定された職責およびナビゲーション・パスを使用し、Web ブラウザで、宛先システムに登録するソース・システムのイベント・マネージャに移動します。次に水平ナビゲーションで「システム」を選択し、サイド・ナビゲーションで

「外部システム登録」を選択します。A-2 ページの「Oracle Workflow 開発者のナビゲーション・パス」を参照してください。

Oracle Workflow の管理者用 Web ページの「ビジネス・イベント」タブまたは下部にある「ビジネス・イベント」リンクを選択して、イベント・マネージャへ移動することができます。

注意： ソース・システムにインストールされている Oracle Workflow へのアクセス権がない場合は、ソース・システムのワークフロー管理者にこの手順を実行するよう依頼してください。

External System Registration - Microsoft Internet Explorer provided by Dev. Services

File Edit View Favorites Tools Help

Home Developer Studio Business Events Status Monitor Notifications

Events Subscriptions Agents Systems

System Details

Local System Identifier

External System Registration

External System Registration

Enter a unique event key, and then either cut and paste in the System identified XML document , or upload it from your local file system.

* Indicates required field

Event Identifier

Event Name oracle.apps.wf.event.system.signup

* Event Key

Send Date

Send Date must be in the Format: DD-MON-RRRR

Event Parameter

Select Item(s) and ...

Select All | Select None

Select Label	Value
<input type="checkbox"/>	

Event Data

XML Content

If Xml content is greater than 4000 char , Press "Upload XML File" Button.

Upload Option Write XML Upload XML

File Name < No File Chosen to Upload >

- 「イベント識別子」領域で、システムのサインアップ・イベントの識別情報を指定します。
 - イベント名： Oracle Workflow によってシステムのサインアップ・イベントの内部名 (oracle.apps.wf.event.system.signup) が自動的に表示されます。
 - イベント・キー： イベントのインスタンスを一意に識別するイベント・キーを入力します。

- 送信日： オプションで、イベント・メッセージがデキューで利用できる日付を指定します。イベントが呼び出されたときに送信日が未来日付に設定されている場合、イベント・メッセージは WF_DEFERRED キューに格納され、指定された日付までサブスクリプション処理は開始されません。
3. オプションで、イベント・メッセージ内のパラメータ・リストに格納する追加のパラメータ名と値のペアを「イベント・パラメータ」領域に入力します。
 - 別の空白行を表示して別のパラメータを入力するには、「他の行の追加」ボタンを選択します。最大で 100 個のパラメータを入力できます。
 - パラメータを削除するには、パラメータを選択して「削除」ボタンを選択します。
 4. 「イベント・データ」領域で、登録する宛先システムのシステム識別子の XML 文書を入力します。8-133 ページの「ローカル・システムの識別子情報の取得」を参照してください。
 - 手動でシステム識別子情報を入力するには、「XML の記述」オプションを選択し、XML 文書を「XML コンテンツ」フィールドに貼り付けます。最大で 4000 文字を入力できます。
 - ファイル・システムからシステム識別子の XML ファイルをアップロードするには、「XML のアップロード」オプションを選択し、「XML ファイルのアップロード」ボタンを選択します。「XML ファイルのアップロード」ページで、システム識別子情報が含まれた XML ファイルのフルパスとファイル名を入力し、「アップロード・ファイル」ボタンを選択します。「取消」ボタンを選択し、ファイルをアップロードしないで元のページに戻ることもできます。

注意： システムを登録するにはシステム識別子情報を指定する必要があります。イベント・データの XML コンテンツを空白のままにしておくことはできません。

5. 「送信」ボタンを選択して、システムのサインアップ・イベントをイベント・マネージャに呼び出します。システムのサインアップ・イベントが呼び出されると、事前定義済みのサブスクリプションが Oracle Workflow によって実行され、システム識別子情報がイベント・データからイベント・マネージャにロードされます。9-14 ページの「システムのサインアップ・イベント」を参照してください。

「取消」ボタンを選択して、システムのサインアップ・イベントを呼び出さないで元のページに戻ることもできます。

ワークフロー・エージェントの Ping/ 確認

「ワークフロー・エージェントの Ping/ 確認」を使用して、ビジネス・イベント・システムの設定をテストできます。このワークフローは、Ping イベント・メッセージをローカル・システムまたは外部システム上の各インバウンド・エージェントに送信して、各エージェントからの受信確認イベント・メッセージを待機します。このワークフローが正常に完了すると、これらのエージェントとの通信に関するビジネス・イベント・システムの基本的な設定が完了します。

「ワークフロー・エージェントの Ping/ 確認」の機能

「ワークフロー・エージェントの Ping/ 確認」ワークフローを開始するには、「プロセスの開始」Web ページを使用します。このワークフローは、「マスター Ping」プロセスと「詳細 Ping」プロセスの2つのプロセスで構成されます。すべてのインバウンド・エージェントを ping するには、「マスター Ping」プロセスを選択して、一意の項目キーを入力します。7-2 ページの「ワークフロー定義のテスト」を参照してください。

注意： プロセス・インスタンスの項目キーには、シングル・バイト・キャラクタのみ使用できます。マルチバイトの値を含めることはできません。

「マスター Ping」プロセスを開始すると、ワークフロー・エンジンによって、ローカル・システムまたは外部システム上で定義されているすべてのインバウンド・エージェントが識別され、各エージェントの「詳細 Ping」プロセスが開始されます。マスター・プロセスは、各詳細プロセスの完了を待機します。

「詳細 Ping」プロセスは、マスター・プロセスによって識別されたインバウンド・エージェントにエージェントの Ping イベントが送信されたときに開始されます。詳細プロセスは、エージェントの Ping イベント・メッセージをローカル・システム上のアウトバウンド・エージェントに関連付けられたキューに格納されます。イベント・メッセージには、インバウンド・エージェントが宛先として指定され、所属する詳細プロセスを識別する相関 ID が含まれます。AQ 伝播により、アウトバウンド・キュー内のイベント・メッセージは、指定されたインバウンド・エージェントに関連付けられたキューに送信されます。

受信側システムのインバウンド・エージェントのリスナーは、次回動作したときに、「エージェントの Ping」メッセージをデキューします。イベント・メッセージがデキューされると、「外部」ソース・タイプを持つエージェントの Ping イベントまたは Any イベントへの有効なサブスクリプションが、イベント・マネージャによって検索および実行されます。

エージェントの Ping イベントへの事前定義済の「外部」サブスクリプションが実行されると、そのルール関数により、受信側システムのアウトバウンド・エージェントに関連付けられているキューに Ping の確認イベントが格納されます。イベント・メッセージには、元のシステムのインバウンド・エージェントが宛先として指定され、エージェントの Ping イベント・メッセージからの相関 ID が含まれます。AQ 伝播により、アウトバウンド・キュー

内のイベント・メッセージは、指定されたインバウンド・エージェントに関連付けられているキューに送信されます。

送信元システムのインバウンド・エージェントのリスナーは、次回動作したときに、「Ping の確認」メッセージをデキューします。イベント・メッセージがデキューされると、「外部」ソース・タイプを持つ Ping の確認イベントまたは Any イベントへのアクティブなサブスクリプションが、イベント・マネージャによって検索および実行されます。

Ping の確認イベントへの事前定義済の「外部」サブスクリプションが実行されると、そのルール関数（デフォルトのルール関数）により、イベント・メッセージが「詳細 Ping」プロセスに送信されます。ワークフロー・エンジンは、イベント・メッセージおよび関連 ID に関連付けられた実行中の詳細プロセスを照合します。イベント・メッセージの受信が終了すると、「詳細 Ping」プロセスが完了します。

すべての詳細プロセスが完了すると、マスター・プロセスも完了します。

Workflow Monitor を使用して、「ワークフロー・エージェントの Ping/ 確認」ワークフローの進行状況を確認できます。「イベント・システム・ローカル・キュー」ページを使用して、エージェントの Ping および Ping の確認イベント・メッセージの処理を確認することもできます。『Oracle Workflow 管理者ガイド』の「ワークフロー・モニター」および 8-69 ページの「ローカル・キューの確認」を参照してください。

「ワークフロー・エージェントの Ping/ 確認」ワークフローの完了に必要な時間は、リスナーが動作してインバウンド・エージェントのメッセージをデキューする頻度によって異なります。『Oracle Workflow 管理者ガイド』の「ローカル・インバウンド・エージェントのリスナーのスケジュール」を参照してください。

関連項目：

9-12 ページ [「エージェントの Ping イベント」](#)

「ワークフロー・エージェントの Ping/ 確認」項目タイプ

「ワークフロー・エージェントの Ping/ 確認」プロセスは、「ワークフロー・エージェントの Ping/ 確認」という項目タイプに関連付けられています。「ワークフロー・エージェントの Ping/ 確認」には現在、「マスター Ping プロセス」と「詳細 Ping プロセス」という2つのワークフロー・プロセスが関連付けられています。

「ワークフロー・エージェントの Ping/ 確認」項目タイプの詳細を Workflow Builder に表示するには、「ファイル」メニューから「オープン」を選択します。次に、データベースに接続して「ワークフロー・エージェントの Ping/ 確認」項目タイプを選択するか、`<ORACLE_HOME>/wf/Data/<language>` サブディレクトリにある `wfping.wft` というファイルに接続します。

「ワークフロー・エージェントの Ping/ 確認」のプロパティ画面を見ると、維持タイプが「一時」に、維持日数が0になっています。これは、この項目タイプの作業項目に関連付けられているランタイム・データは、終了直後に削除の対象となることを意味します。

「ワークフロー・エージェントの Ping/ 確認」項目タイプには、複数の属性が関連付けられています。これらの属性は、ワークフローのアプリケーション表にある情報を参照します。属性は、イベント・アクティビティと同様に、プロセス全体を通して関数アクティビティによって使用され保存されます。次の表は、「ワークフロー・エージェントの Ping/ 確認」項目タイプ属性を示しています。

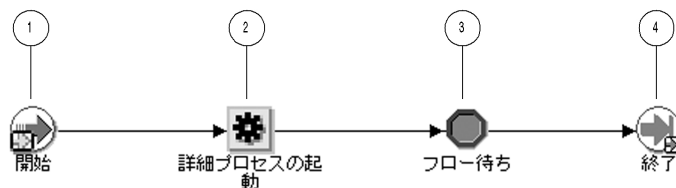
表 8-23

表示名	説明	型	長さ / 書式 / 選択肢タイプ
宛先エージェント	イベント・メッセージを受信するインバウンド・エージェント (<code><agent>@<system></code> の書式)	テキスト	
イベント名	イベントの内部名	テキスト	
送信エージェント	イベント・メッセージを送信するアウトバウンド・エージェント (<code><agent>@<system></code> の書式)	テキスト	
イベント・キー	イベントの特定のインスタンスを一意に識別するイベント・キー	テキスト	
イベント・メッセージ	イベント・メッセージ	イベント	

「マスター Ping」プロセスの概要

「マスター Ping」プロセスのプロパティを表示するには、ナビゲータ・ツリーでプロセスを選択し、「編集」メニューから「プロパティ」を選択します。このプロセス・アクティビティは実行可能です。つまり、最上位レベルのプロセスとして実行できます。

「マスター Ping」プロセスの「プロセス」ウィンドウを表示すると、プロセスが4つの異なるアクティビティから構成されていることがわかります。プロセスのアクティビティを詳しく説明するために、次のように各ノードに番号を付けて参照しやすくしてあります。番号自体は、プロセス・ダイアグラムには含まれません。



「プロセスの開始」 Web ページを使用して「マスター Ping プロセス」を開始すると、「ワークフロー・エージェントの Ping/ 確認」ワークフローが開始されます。オプションで、宛先エージェント、イベント名、送信エージェント、イベント・キーおよびイベント・メッセージを渡すことができます。7-2 ページの「ワークフロー定義のテスト」を参照してください。

ワークフローは、ノード1の「開始」アクティビティから開始します。ノード2で、マスター・プロセスはローカル・システムまたは外部システムに定義されている各インバウンド・エージェントの詳細プロセスを起動します。詳細プロセスは、エージェントの Ping イベントを送信してインバウンド・エージェントを ping し、Ping の確認イベントの形式で受信確認を待機します。

ノード3は「フロー待ち」アクティビティで、すべての詳細プロセスの終了を待機します。すべての詳細プロセスが完了すると、マスター・プロセスが終了します。

「マスター Ping」 プロセス・アクティビティ

ここでは、プロセスの各アクティビティについて、アクティビティの表示名別に説明します。

開始（ノード 1）

この「標準」関数アクティビティは、プロセスの開始をマークします。

関数	WF_STANDARD.NOOP
結果タイプ	なし
前提条件アクティビティ	なし

詳細プロセスの起動（ノード 2）

この関数アクティビティは、ローカル・システムまたは外部システム上で定義されているすべてのインバウンド・エージェントを識別し、各エージェントの「詳細 Ping」プロセスを起動します。また、エージェントの Ping イベント (oracle.apps.wf.event.test.ping) を「詳細 Ping」プロセスに送信されるように設定します。

関数	WF_EVENT_PING_PKG.LAUNCH_PROCESSES
結果タイプ	なし
前提条件アクティビティ	なし
関数によって設定される項目属性	イベント名、宛先エージェント

フロー待ち（ノード 3）

この「標準」関数アクティビティは、指定されたアクティビティを対応する詳細プロセスが完了するまで、フローを一時停止します。

関数	WF_STANDARD.WAITFORFLOW
結果タイプ	なし
前提条件アクティビティ	詳細プロセスの起動

終了（ノード 4）

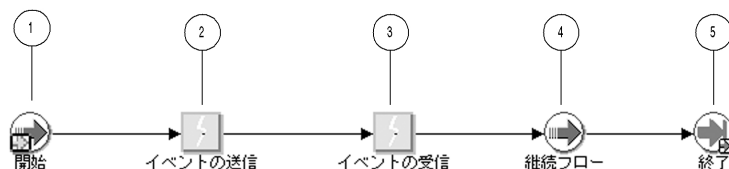
この「標準」関数アクティビティは、プロセスの終了をマークします。

関数	WF_STANDARD.NOOP
結果タイプ	なし
前提条件アクティビティ	フロー待ち

「詳細 Ping」プロセスの概要

「詳細 Ping」プロセスのプロパティを表示するには、ナビゲータ・ツリーでプロセス・アクティビティを選択し、「編集」メニューから「プロパティ」を選択します。このプロセス・アクティビティは実行可能です。つまり、最上位レベルのプロセスとして実行できます。

「詳細 Ping」プロセスの「プロセス」ウィンドウを表示すると、プロセスが5つの異なるアクティビティから構成されていることがわかります。プロセスのアクティビティを詳しく説明するために、次のように各ノードに番号を付けて参照しやすくしてあります。番号自体は、プロセス・ダイアグラムには含まれません。



「詳細 Ping」プロセスは、「マスター Ping」プロセスによって開始されます。8-140 ページの「[「マスター Ping」プロセスの概要](#)」を参照してください。

ワークフローは、ノード1の「開始」アクティビティから開始します。ノード2で、「詳細 Ping」プロセスはエージェントの Ping イベントを選択されたインバウンド・エージェントに送信します。ノード3で、「詳細 Ping」プロセスはエージェント Ping の確認イベントを待機します。受信確認を受信したマスター・プロセスは、フローを続行できます。「詳細 Ping」プロセスはこの時点で終了します。

「詳細 Ping」 プロセス・アクティビティ

ここでは、プロセスの各アクティビティについて、アクティビティの表示名別に説明します。

開始（ノード 1）

この「標準」関数アクティビティは、プロセスの開始をマークします。

関数	WF_STANDARD.NOOP
結果タイプ	なし
前提条件アクティビティ	なし

イベントの送信（ノード 2）

このイベント・アクティビティは、ローカル・システム上のアウトバウンド・エージェントのエージェントの Ping イベント（oracle.apps.wf.event.test.ping）を、マスター・プロセスによって識別されたインバウンド・エージェントに送信します。イベント・メッセージには、所属する詳細プロセスを識別する関連 ID が含まれます。

イベント・アクション	送信
前提条件アクティビティ	なし
アクティビティによって取得される項目属性	イベント・メッセージ、イベント名、イベント・キー、宛先エージェント

イベントの受信（ノード 3）

このイベント・アクティビティは、エージェントの Ping イベントを受信したシステムから発信元システムに返された、Ping の確認イベント（oracle.apps.wf.event.test.ack）を受信します。Ping の確認イベント・メッセージには、ワークフロー・エンジンがイベント・メッセージの所属する詳細プロセスを識別するために使用する、関連 ID が含まれます。

イベント・アクション	受信
イベント・フィルタ	oracle.apps.wf.event.test.ack
前提条件アクティビティ	イベントの送信
アクティビティによって設定される項目属性	イベント名、イベント・キー、イベント・メッセージ

継続フロー（ノード 4）

この「標準」関数アクティビティは、詳細プロセスの位置をマークします。詳細プロセスが完了すると、対応するマスター・プロセスの停止が解除され、処理を続行します。

関数	WF_STANDARD.CONTINUEFLOW
結果タイプ	なし
前提条件アクティビティ	イベントの受信

終了（ノード 5）

この「標準」関数アクティビティは、プロセスの終了をマークします。

関数	WF_STANDARD.NOOP
結果タイプ	なし
前提条件アクティビティ	継続フロー

事前定義済ワークフロー・イベント

この章では、Oracle Workflow の事前定義済イベントの使用方法について説明します。

事前定義済ワークフロー・イベント

Oracle Workflow には、ビジネス・イベント・システムやその他の Oracle Workflow コンポーネント内で関連付けられた状態変化に対して、いくつかの事前定義済イベントが用意されています。これらのイベントへのサブスクリプションを定義して、レプリケーションや検証などを行うことができます。

デフォルトでは、すべての事前定義済イベントが使用可能になっています。これらのイベントの多くは、必要に応じて使用不可にすることができます。

いくつかの事前定義済イベントは、Oracle Workflow のインストール時に自動的に作成されるデフォルトのサブスクリプションによって参照されます。デフォルト・サブスクリプションのサブスクリイバは、すべてローカル・システムです。これらのサブスクリプションの多くは、使用可能 / 使用不可を切り替えたり、コピーしたりすることにより、必要なイベント処理を実行できます。

注意： Unexpected イベントまたはそのイベントへの事前定義済のエラー・サブスクリプションの定義は、変更したり使用不可にしないでください。これらを変更したり使用不可にすると、イベント・マネージャはイベントおよびサブスクリプション処理に対してデフォルトのエラー処理を実行できなくなります。

注意： 事前定義済イベントおよびデフォルト・サブスクリプションは、Oracle Applications および Oracle Self-Service Web Applications にも用意されています。Oracle Applications に固有のワークフロー・イベントの詳細は、該当する Oracle Applications 製品のドキュメントまたはヘルプを参照してください。

イベント定義イベント

イベント作成

このイベントは、新しい単一イベントまたはイベント・グループ定義が作成されたときに、Oracle Workflow によって呼び出されます。

内部名	oracle.apps.wf.event.event.create
ジェネレート関数	wf_event_functions_pkg.generate
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

イベント更新

このイベントは、単一イベントまたはイベント・グループ定義が更新されたときに、Oracle Workflow によって呼び出されます。

内部名	oracle.apps.wf.event.event.update
ジェネレート関数	wf_event_functions_pkg.generate
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

イベント削除

このイベントは、単一イベントまたはイベント・グループ定義が削除されたときに、Oracle Workflow によって呼び出されます。

内部名	oracle.apps.wf.event.event.delete
ジェネレート関数	wf_event_functions_pkg.generate
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

イベント・グループ定義イベント

イベント・グループ作成

このイベントは、新しいイベント・グループ・メンバー定義が作成されたときに、Oracle Workflow によって呼び出されます。

内部名	oracle.apps.wf.event.group.create
ジェネレート関数	wf_event_functions_pkg.generate
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

イベント・グループ更新

このイベントは、イベント・グループ・メンバー定義が更新されたときに、Oracle Workflow によって呼び出されます。

内部名	oracle.apps.wf.event.group.update
ジェネレート関数	wf_event_functions_pkg.generate
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

イベント・グループ削除

このイベントは、イベント・グループ・メンバー定義が削除されたときに、Oracle Workflow によって呼び出されます。

内部名	oracle.apps.wf.event.group.delete
ジェネレート関数	wf_event_functions_pkg.generate
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

システム定義イベント

システム作成

このイベントは、新しいシステム定義が作成されたときに、Oracle Workflow によって呼び出されます。

内部名	oracle.apps.wf.event.system.create
ジェネレート関数	wf_event_functions_pkg.generate
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

システム更新

このイベントは、システム定義が更新されたときに、Oracle Workflow によって呼び出されます。

内部名	oracle.apps.wf.event.system.update
ジェネレート関数	wf_event_functions_pkg.generate
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

システム削除

このイベントは、システム定義が削除されたときに、Oracle Workflow によって呼び出されます。

内部名	oracle.apps.wf.event.system.delete
ジェネレート関数	wf_event_functions_pkg.generate
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

エージェント定義イベント

エージェント作成

このイベントは、新しいエージェント定義が作成されたときに、Oracle Workflow によって呼び出されます。

内部名	oracle.apps.wf.event.agent.create
ジェネレート関数	wf_event_functions_pkg.generate
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

エージェント更新

このイベントは、エージェント定義が更新されたときに、Oracle Workflow によって呼び出されます。

内部名	oracle.apps.wf.event.agent.update
ジェネレート関数	wf_event_functions_pkg.generate
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

エージェント削除

このイベントは、エージェント定義が削除されたときに、Oracle Workflow によって呼び出されます。

内部名	oracle.apps.wf.event.agent.delete
ジェネレート関数	wf_event_functions_pkg.generate
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

エージェント・グループ定義イベント

注意： エージェント・グループは、現在 Oracle Application に組み込まれている Workflow でのみ使用できます。

エージェント・グループ・メンバー作成

このイベントは、新しいエージェント・グループ・メンバー定義が作成されたときに、Oracle Workflow によって呼び出されます。

内部名	oracle.apps.wf.agent.group.create
ジェネレート関数	wf_event_functions_pkg.generate
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

エージェント・グループ・メンバー更新

このイベントは、エージェント・グループ・メンバー定義が更新されたときに、Oracle Workflow によって呼び出されます。

内部名	oracle.apps.wf.agent.group.update
ジェネレート関数	wf_event_functions_pkg.generate
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

エージェント・グループ・メンバー削除

このイベントは、エージェント・グループ・メンバー定義が削除されたときに、Oracle Workflow によって呼び出されます。

内部名	oracle.apps.wf.agent.group.delete
ジェネレート関数	wf_event_functions_pkg.generate
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

イベント・サブスクリプション定義イベント

サブスクリプション作成

このイベントは、新しいサブスクリプション定義が作成されたときに、Oracle Workflow によって呼び出されます。

内部名	oracle.apps.wf.event.subscription.create
ジェネレート関数	wf_event_functions_pkg.generate
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

サブスクリプション更新

このイベントは、サブスクリプション定義が更新されたときに、Oracle Workflow によって呼び出されます。

内部名	oracle.apps.wf.event.subscription.update
ジェネレート関数	wf_event_functions_pkg.generate
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

サブスクリプション削除

このイベントは、サブスクリプション定義が削除されたときに、Oracle Workflow によって呼び出されます。

内部名	oracle.apps.wf.event.subscription.delete
ジェネレート関数	wf_event_functions_pkg.generate
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

イベント・システムの同期イベント

このイベントは、ローカル・システム上のイベント・マネージャを別のシステム上のイベント・マネージャに同期させるときに呼び出すことができます。システムの同期イベントのイベント・メッセージには、ローカル・システム上のすべてのイベント・マネージャ・オブジェクトの定義が含まれます。8-65 ページの「[システムの同期](#)」を参照してください。

内部名	oracle.apps.wf.event.all.sync
ジェネレート関数	wf_event_functions_pkg.generate
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

シード・イベント・グループ

このイベント・グループは、システム間でビジネス・イベント・システム・オブジェクトを自動レプリケートするときに使用されるイベントで構成されます。シード・イベント・グループには、すべてのイベント、イベント・グループ、システム、エージェントおよびサブスクリプション定義イベント以外に、イベント・システムの同期イベントも含まれます。

内部名	oracle.apps.wf.event.group.all
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限
メンバー	oracle.apps.wf.event.event.create oracle.apps.wf.event.event.update oracle.apps.wf.event.event.delete oracle.apps.wf.event.group.create oracle.apps.wf.event.group.update oracle.apps.wf.event.group.delete oracle.apps.wf.event.system.create oracle.apps.wf.event.system.update oracle.apps.wf.event.system.delete oracle.apps.wf.event.agent.create oracle.apps.wf.event.agent.update oracle.apps.wf.event.agent.delete oracle.apps.wf.agent.group.create oracle.apps.wf.agent.group.update oracle.apps.wf.agent.group.delete oracle.apps.wf.event.subscription.create oracle.apps.wf.event.subscription.update oracle.apps.wf.event.subscription.delete oracle.apps.wf.event.all.sync

注意： エージェント・グループは、現在 Oracle Application に組み込まれている Workflow でのみ使用できます。

Oracle Workflow には、シード・イベント・グループに対して、2つのデフォルト・サブスクリプションが用意されています。一方のサブスクリプションでは、いずれかのグループ・メンバー・イベントがローカルで呼び出されたときに、イベント・マネージャ・データをエージェントまたはワークフロー・プロセスに送信することができます。このサブスクリプ

ションを使用するには、カスタマイズ・レベルを「ユーザー」に設定して、同じプロパティでこのサブスクリプションのコピーを作成し、データを送信するエージェントまたはワークフローを追加して、そのサブスクリプションを使用可能にします。次の表は、このサブスクリプションに対して定義されているプロパティを示しています。

表 9-1

サブスクリプションのプロパティ	値
システム	<local system>
ソース・タイプ	ローカル
イベント・フィルタ	oracle.apps.wf.event.group.all
フェーズ	50
ステータス	使用不能
ルール・データ	メッセージ
ルール関数	wf_rule.default_rule
優先度	標準
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

もう一方のサブスクリプションでは、いずれかのグループ・メンバー・イベントが外部ソースから着信したときに、イベント・マネージャ・データをローカル・システムにロードすることができます。このサブスクリプションを使用するには、サブスクリプションを使用可能にする必要があります。次の表は、このサブスクリプションに対して定義されているプロパティを示しています。

表 9-2

サブスクリプションのプロパティ	値
システム	<local system>
ソース・タイプ	外部
イベント・フィルタ	oracle.apps.wf.event.group.all
フェーズ	50
ステータス	使用不能
ルール・データ	キー
ルール関数	wf_event_functions_pkg.receive

表 9-2 (続き)

サブスクリプションのプロパティ	値
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

関連項目：

8-52 ページ [「イベント・サブスクリプションの定義」](#)8-65 ページ [「システムの同期」](#)

エージェントの Ping イベント

エージェントの Ping イベント

このイベントは、インバウンド・エージェントを ping するために、「ワークフロー・エージェントの Ping/ 確認」項目タイプの詳細 Ping プロセスから送信されます。「プロセスの開始」Web ページを使用してマスター Ping プロセスを開始すると、マスター Ping プロセスによって詳細 Ping プロセスが開始されます。8-139 ページの「[「ワークフロー・エージェントの Ping/ 確認」項目タイプ](#)」を参照してください。

内部名	oracle.apps.wf.event.test.ping
ジェネレート関数	なし
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

Oracle Workflow には、エージェント・イベントの Ping イベントに対して、1つのデフォルト・サブスクリプションが用意されています。このサブスクリプションでは、外部ソースからエージェント・イベントの Ping が着信したときに、Ping の確認イベントを発信元のシステムに返送することができます。このサブスクリプションは、デフォルトで使用可能になっています。次の表は、このサブスクリプションに対して定義されているプロパティを示しています。

表 9-3

サブスクリプションのプロパティ	値
システム	<local system>
ソース・タイプ	外部

表 9-3 (続き)

サブスクリプションのプロパティ	値
イベント・フィルタ	oracle.apps.wf.event.test.ping
フェーズ	50
ステータス	使用可能
ルール・データ	キー
ルール関数	wf_event_ping_pkg.acknowledge
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

Ping の確認

このサブスクリプションは、Ping の確認イベントが着信したときに、Oracle Workflow から発信元のシステムに返送されます。8-139 ページの「[ワークフロー・エージェントの Ping/ 確認](#)」項目タイプを参照してください。

内部名	oracle.apps.wf.event.test.ack
ジェネレート関数	なし
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

Oracle Workflow には、Ping の確認イベントに対して、1 つのデフォルト・サブスクリプションが用意されています。このサブスクリプションでは、外部ソースからイベントが着信したときに、Ping の確認イベントを「ワークフロー・エージェントの Ping/ 確認」項目タイプの詳細 Ping プロセスに送信することができます。このサブスクリプションは、デフォルトで使用可能になっています。次の表は、このサブスクリプションに対して定義されているプロパティを示しています。

表 9-4

サブスクリプションのプロパティ	値
システム	<local system>
ソース・タイプ	外部
イベント・フィルタ	oracle.apps.wf.event.test.ack
フェーズ	50

表 9-4 (続き)

サブスクリプションのプロパティ	値
ステータス	使用可能
ルール・データ	キー
ルール関数	wf_rule.default_rule
ワークフロー項目タイプ	WFPING
ワークフロー・プロセス名	WFDLTPNG
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

関連項目：8-52 ページ「[イベント・サブスクリプションの定義](#)」

システムのサインアップ・イベント

ソース・システムの「システムのサインアップ」Web ページからこのイベントを呼び出して、ソース・システムからイベント・メッセージを受信する宛先システムをサインアップすることができます。8-62 ページの「[システムのサインアップ](#)」を参照してください。

内部名	oracle.apps.wf.event.system.signup
ジェネレート関数	なし
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

Oracle Workflow には、システムのサインアップ・イベントに対して、1つのデフォルト・サブスクリプションが用意されています。このサブスクリプションでは、システムのサインアップ・イベントがローカルで呼び出されたときに、イベント・マネージャ・データがローカル・システムにロードされます。このサブスクリプションは、デフォルトで使用可能になっています。次の表は、このサブスクリプションに対して定義されているプロパティを示しています。

表 9-5

サブスクリプションのプロパティ	値
システム	<local system>

表 9-5 (続き)

サブスクリプションのプロパティ	値
ソース・タイプ	ローカル
イベント・フィルタ	oracle.apps.wf.event.system.signup
フェーズ	50
ステータス	使用可能
ルール・データ	キー
ルール関数	wf_event_functions_pkg.receive
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

関連項目：8-52 ページ「[イベント・サブスクリプションの定義](#)」

Any イベント

このイベントは、他のイベントがローカルで呼び出されたときまたは外部ソースから着信したときに、暗黙的に呼び出されます。**Any** イベントへのサブスクリプションを定義すれば、イベントが発生するたびに実行される処理を実装できます。

内部名	oracle.apps.wf.event.any
ジェネレート関数	なし
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

Oracle Workflow には、Any イベントに対して、3つのデフォルト・サブスクリプションが用意されています。最初のサブスクリプションは、イベントがローカルで呼び出されたときにトリガーできます。このサブスクリプションを使用するには、カスタマイズ・レベルを「ユーザー」に設定して、同じプロパティでこのサブスクリプションのコピーを作成し、コピーしたサブスクリプションのアクションを定義して、そのサブスクリプションを使用可能にします。次の表は、このサブスクリプションに対して定義されているプロパティを示しています。

表 9-6

サブスクリプションのプロパティ	値
システム	<local system>
ソース・タイプ	ローカル
イベント・フィルタ	oracle.apps.wf.event.any
フェーズ	100
ステータス	使用不能
ルール・データ	キー
優先度	標準
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

2 番目のサブスクリプションは、イベントが外部ソースから着信したときにトリガーできます。このサブスクリプションを使用するには、カスタマイズ・レベルを「ユーザー」に設定して、同じプロパティでこのサブスクリプションのコピーを作成し、コピーしたサブスクリプションのアクションを定義して、そのサブスクリプションを使用可能にします。次の表は、このサブスクリプションに対して定義されているプロパティを示しています。

表 9-7

サブスクリプションのプロパティ	値
システム	<local system>
ソース・タイプ	外部
イベント・フィルタ	oracle.apps.wf.event.any
フェーズ	100
ステータス	使用不能
ルール・データ	キー
優先度	標準
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

3 番目のサブスクリプションでは、イベントが「エラー」ソースから着信したとき（つまり、WF_ERROR キューからデキューされたとき）に、イベント・メッセージが「システム:エラー」項目タイプのデフォルト・イベント・エラー・プロセスに送信され、例外が呼び出されます。このサブスクリプションを使用するには、サブスクリプションを使用可能にする必要があります。次の表は、このサブスクリプションに対して定義されているプロパティを示しています。

表 9-8

サブスクリプションのプロパティ	値
システム	<local system>
ソース・タイプ	エラー
イベント・フィルタ	oracle.apps.wf.event.any
フェーズ	100
ステータス	使用不能
ルール・データ	キー
ルール関数	wf_rule.error_rule
ワークフロー項目タイプ	WFERROR
ワークフロー・プロセス名	DEFAULT_EVENT_ERROR
優先度	標準
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

関連項目：

8-52 ページ [「イベント・サブスクリプションの定義」](#)

Unexpected イベント

このイベントへのサブスクリプションは、イベントがローカルで呼び出されたときまたは外部ソースから着信したときに、そのイベントにサブスクリプションが存在しない場合に、Oracle Workflow によって実行されます。

注意： Unexpected イベントの定義を変更したり、使用不可にしないでください。これらを変更したり使用不可にすると、イベント・マネージャはイベントおよびサブスクリプション処理に対してエラー処理を実行できなくなります。

内部名	oracle.apps.wf.event.unexpected
ジェネレート関数	なし
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

Oracle Workflow には、Unexpected イベントに対して、2つのデフォルト・サブスクリプションが用意されています。最初のサブスクリプションでは、Unexpected イベントが外部ソースから着信したときに、イベント・メッセージが「システム:エラー」項目タイプのデフォルト・イベント・エラー・プロセスに送信されます。このサブスクリプションにより、外部システムから受信した予期しないイベント・メッセージをローカル・システムで処理できます。

デフォルト・イベント・エラー・プロセスからシステム管理者にエラーが通知されます。通知を受けたシステム管理者は、イベントへのサブスクリプション処理を再試行または中止することができます。たとえば、システム管理者はイベントを処理するサブスクリプションを定義してから、イベントを再試行することもできます。

Unexpected イベントへの「外部」サブスクリプションは、デフォルトで使用可能になっています。このサブスクリプションは、必要に応じて使用不可にすることができます。

注意： このサブスクリプションを使用不可にする場合は、外部ソースから受信した Unexpected イベントの処理に関する結果をすべて考慮してください。このサブスクリプションを使用不可にした場合、これらのイベント・メッセージは、着信したインバウンド・キューに残り、一定期間検出されないことがあります。

次の表は、このサブスクリプションに対して定義されているプロパティを示しています。

表 9-9

サブスクリプションのプロパティ	値
システム	<local system>
ソース・タイプ	外部
イベント・フィルタ	oracle.apps.wf.event.unexpected
フェーズ	50
ステータス	使用可能
ルール・データ	キー
ワークフロー項目タイプ	WFERROR
ワークフロー・プロセス名	DEFAULT_EVENT_ERROR
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

2番目のサブスクリプションでは、Unexpected イベントが「エラー」ソースから着信したとき（つまり、WF_ERROR キューからデキューされたとき）に、イベント・メッセージが「システム:エラー」項目タイプのデフォルト・イベント・エラー・プロセスに送信されます。このサブスクリプションは、デフォルトで使用可能になっています。

注意： Unexpected イベントへの事前定義済の「エラー」サブスクリプションの定義を変更したり、使用不可にしないでください。このサブスクリプションを使用不可にすると、カスタム「エラー」サブスクリプションが定義されていないイベントに対して、イベント・マネージャがエラー処理を実行できなくなります。

次の表は、このサブスクリプションに対して定義されているプロパティを示しています。

表 9-10

サブスクリプションのプロパティ	値
システム	<local system>
ソース・タイプ	エラー
イベント・フィルタ	oracle.apps.wf.event.unexpected
フェーズ	50

表 9-10 (続き)

サブスクリプションのプロパティ	値
ステータス	使用可能
ルール・データ	キー
ルール関数	wf_rule.error_rule
ワークフロー項目タイプ	WFERROR
ワークフロー・プロセス名	DEFAULT_EVENT_ERROR
優先度	標準
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

関連項目：8-52 ページ [「イベント・サブスクリプションの定義」](#)**User Entry Has Changed イベント**

このイベントは、変更されたユーザー情報が LDAP から取得されたときに、Workflow LDAP API によって呼び出されます。OID 統合を実装する場合は、Workflow LDAP API を実行して、Workflow ディレクトリ・サービスを Oracle Internet Directory に同期させることができます。変更されたユーザーごとに 1 つのイベントが呼び出されます。『Oracle Workflow 管理者ガイド』の「Workflow ディレクトリ・サービスと Oracle Internet Directory の同期」および『Oracle Workflow API リファレンス』の「Workflow LDAP API」を参照してください。

内部名	oracle.apps.global.user.change
ジェネレート関数	wf_entity_mgr.gen_xml_payload
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

Oracle Workflow のスタンドアロン版には、User Entry Has Changed イベントに対して、2 つのデフォルト・サブスクリプションが用意されています。最初のサブスクリプションでは、User Entry Has Changed イベントがローカルで呼び出されて、そのユーザーがスタンドアロン版の WF_LOCAL_USERS 表に存在しないときは、新しいユーザー・レコードが作成されます。このサブスクリプションは、デフォルトで使用可能になっています。次の表は、このサブスクリプションに対して定義されているプロパティを示しています。

表 9-11

サブスクリプションのプロパティ	値
システム	<local system>
ソース・タイプ	ローカル
イベント・フィルタ	oracle.apps.global.user.change
フェーズ	50
ステータス	使用可能
ルール・データ	キー
ルール関数	wf_sso.user_create_rf
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

2番目のサブスクリプションでは、User Entry Has Changed イベントがローカルで呼び出されたときに、変更されたユーザー・データが Oracle Workflow スタンドアロン版の WF_LOCAL_ROLES 表にロードされます。このサブスクリプションは、デフォルトで使用可能になっています。次の表は、このサブスクリプションに対して定義されているプロパティを示しています。

表 9-12

サブスクリプションのプロパティ	値
システム	<local system>
ソース・タイプ	ローカル
イベント・フィルタ	oracle.apps.global.user.change
フェーズ	60
ステータス	使用可能
ルール・データ	キー
ルール関数	wf_sso.user_change
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

Oracle Applications の Oracle Application に組み込まれている Workflow では、User Entry Has Changed イベントに対するデフォルトのサブスクリプションが1つ用意されています。そのサブスクリプションでは、User Entry Has Changed イベントがローカルで呼び出されたときに、変更されたユーザー・データが Oracle Applications の WF_LOCAL_USERS 表にロードされます。Oracle Application に組み込まれている Workflow を使用しているときに、Oracle Internet Directory との統合を実装する場合は、このサブスクリプションを使用可能にする必要があります。次の表は、このサブスクリプションに対して定義されているプロパティを示しています。

表 9-13

サブスクリプションのプロパティ	値
システム	<local system>
ソース・タイプ	ローカル
イベント・フィルタ	oracle.apps.global.user.change
フェーズ	50
ステータス	使用不能
ルール・データ	キー
ルール関数	fnd_user_pkg.user_create_rf
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

関連項目：8-52 ページ「[イベント・サブスクリプションの定義](#)」

通知イベント

通知送信のイベント

このイベントは、アウトバウンド通知メッセージを送信するために使用されます。

内部名	oracle.apps.wf.notification.send
ジェネレート関数	WF_XML.Generate
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

Oracle Workflow には、通知送信イベント用にデフォルトのサブスクリプションが1つ用意されています。これは、アウトバウンド・メッセージの生成および送信中に通知メーラーでエラーが発生したときのデフォルトのサブスクリプションです。通知送信イベントが「エラー」ソースから着信したとき（つまり、WF_ERROR キューからデキューされたとき）に、このサブスクリプションによって、イベント・メッセージが「システム:エラー」項目タイプのデフォルト・イベント・エラー・プロセスに送信されます。このサブスクリプションは、デフォルトで使用可能になっています。次の表は、このサブスクリプションに対して定義されているプロパティを示しています。

表 9-14

サブスクリプションのプロパティ	値
システム	<local system>
ソース・タイプ	エラー
イベント・フィルタ	oracle.apps.fnd.wf.notification.send
フェーズ	50
ステータス	使用可能
ルール・データ	キー
ルール関数	wf_xml.error_rule
ワークフロー項目タイプ	WFERROR
ワークフロー・プロセス名	DEFAULT_EVENT_ERROR
優先度	標準
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

関連項目：8-52 ページ [「イベント・サブスクリプションの定義」](#)**通知再割当てのイベント**

このイベントは、通知が再割当てされたときにアウトバウンド通知メッセージを新しい受信者に送信するために使用されます。

内部名	oracle.apps.wf.notification.reassign
ジェネレート関数	なし
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

通知取消しのイベント

このイベントは、以前送信された通知が取り消されるときに、アウトバウンドの取消しメッセージを送信するために使用されます。

内部名	oracle.apps.wf.notification.cancel
ジェネレート関数	なし
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

通知送信グループ

このイベント・グループには、通知メーカーによって送信される様々なタイプのアウトバウンド・メッセージ用のイベントが含まれています。

内部名	oracle.apps.wf.notification.send.group
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限
メンバー	oracle.apps.wf.notification.send. oracle.apps.wf.notification.reassign oracle.apps.wf.notification.cancel

Oracle Workflow には、通知送信グループに対して、2つのデフォルト・サブスクリプションが用意されています。最初のサブスクリプションでは、グループ・メンバーのイベントの1つがローカルで呼び出されると、通知メーラーが送信されるように、WF_NOTIFICATION_OUT エージェントにイベント・メッセージを格納します。このサブスクリプションは、デフォルトで使用可能になっています。次の表は、このサブスクリプションに対して定義されているプロパティを示しています。

表 9-15

サブスクリプションのプロパティ	値
システム	<local system>
ソース・タイプ	ローカル
イベント・フィルタ	oracle.apps.fnd.wf.notification.send.group
フェーズ	100
ステータス	使用可能
ルール・データ	メッセージ
ルール関数	wf_rule.default_rule
送信エージェント	WF_NOTIFICATION_OUT@<local system>
優先度	標準
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

2番目のサブスクリプションは、グループ・メンバー・イベントの1つがローカルで呼び出されたときに、C 言語ベースの通知メーラーで処理されるように、イベント・メッセージを WF_SMTP_O_1_QUEUE に格納するのに使用します。このサブスクリプションは、オラクル社カスタマ・サポートから指示があった場合のみ使用してください。オラクル社カスタマ・サポートから指示がないかぎり、このサブスクリプションを有効にしないでください。次の表は、このサブスクリプションに対して定義されているプロパティを示しています。

表 9-16

サブスクリプションのプロパティ	値
システム	<local system>
ソース・タイプ	ローカル
イベント・フィルタ	oracle.apps.wf.notification.send.group
フェーズ	0

表 9-16 (続き)

サブスクリプションのプロパティ	値
ステータス	使用不能
ルール・データ	キー
ルール関数	wf_xml.sendnotification
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

関連項目：8-52 ページ [「イベント・サブスクリプションの定義」](#)**要約通知送信イベント**

このイベントは、要約通知の開始イベントが呼び出されたときに、SUMMARY に設定されている通知環境設定に基づいて各ユーザーおよびロールの要約通知を生成して送信するために、Oracle Workflow によって呼び出されます。

内部名	oracle.apps.wf.notification.summary.send
ジェネレート関数	WF_XML.Generate
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

Oracle Workflow には、要約通知送信イベントに対して、1つのデフォルト・サブスクリプションが用意されています。要約通知送信イベントがローカルで呼び出されると、このサブスクリプションによって、イベント・メッセージが通知メーラーで送信されるように、WF_NOTIFICATION_OUT エージェントに格納されます。このサブスクリプションは、デフォルトで使用可能になっています。次の表は、このサブスクリプションに対して定義されているプロパティを示しています。

表 9-17

サブスクリプションのプロパティ	値
システム	<local system>
ソース・タイプ	ローカル
イベント・フィルタ	oracle.apps.fnd.wf.notification.summary.send

表 9-17 (続き)

サブスクリプションのプロパティ	値
フェーズ	0
ステータス	使用可能
ルール・データ	メッセージ
ルール関数	wf_rule.default_rule
送信エージェント	WF_NOTIFICATION_OUT@<local system>
優先度	標準
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

関連項目：8-52 ページ「[イベント・サブスクリプションの定義](#)」**通知送信エラー・イベント**

このイベントは現在使用されていません。

内部名	oracle.apps.wf.notification.send.error
ジェネレート関数	なし
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

着信応答受信イベント

このイベントは、インバウンド・メッセージが通知メーラーで受信されると、Oracle Workflow によって呼び出されます。

内部名	oracle.apps.wf.notification.receive.message
ジェネレート関数	なし
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

送信者への返送イベント

このイベントは、通知メーラーで受信されたインバウンド応答メッセージの処理中に問題が発生し、そのような問題が原因で元の通知または応答が再送信される必要がある場合に、Oracle Workflow によって呼び出されます。

内部名	oracle.apps.wf.notification.receive.sendreturn
ジェネレート関数	なし
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

通知受信者使用不可イベント

このイベントは、通知メーラーで受信されたインバウンド応答メッセージに「使用不可」タグ処理に関連付けられたタグ・パターンと一致するテキスト文字列が含まれていた場合に、Oracle Workflow によって呼び出されます。

内部名	oracle.apps.wf.notification.receive.unavail
ジェネレート関数	なし
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

通知エラー・イベント

このイベントは、通知メーラーで受信されたインバウンド応答メッセージに「エラー」タグ処理に関連するタグ・パターンと一致するテキスト文字列が含まれていた場合に、Oracle Workflow によって呼び出されます。

内部名	oracle.apps.wf.notification.receive.error
ジェネレート関数	なし
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

通知受信グループ

このイベント・グループには、通知メーラーで受信される様々なタイプの着信メッセージ用のイベントが含まれています。

内部名	oracle.apps.wf.notification.receive
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限
メンバー	oracle.apps.wf.notification.receive.message oracle.apps.wf.notification.receive.sendreturn oracle.apps.wf.notification.receive.unavail oracle.apps.wf.notification.receive.error

Oracle Workflow には、通知受信グループに対して、2つのデフォルト・サブスクリプションが用意されています。このサブスクリプションで、グループ・メンバー・イベントの1つが外部ソースから受信されると、着信メッセージが処理されます。

- **oracle.apps.wf.notification.receive.message:** 通知メーラーで、メッセージに対する通常の応答処理が行われます。『Oracle Workflow 管理者ガイド』の「インバウンド通知メーラー処理」を参照してください。
- **oracle.apps.wf.notification.receive.sendreturn:** アウトバウンド通知が送信されます。
- **oracle.apps.wf.notification.receive.unavail:** メッセージが廃棄フォルダに移されます。通知のステータスは引き続き OPEN であるため、通知メーラーは通知への返信を待機し続けます。ただし、通知のメール・ステータスは UNAVAIL に更新されます。
- **oracle.apps.wf.notification.receive.error:** メッセージが廃棄フォルダに移され、この通知アクティビティが所属するワークフロー・プロセスにエラー・プロセスが定義されていれば、Oracle Workflow によってそのエラー・プロセスが開始されます。通知のステータスは OPEN のままですが、メールとアクティビティの各ステータスは ERROR に更新されます。原則的には、ワークフロー管理者が問題点を修正し、メール・ステータスを MAIL に更新して通知を再送信します。

このサブスクリプションは、デフォルトで使用可能になっています。次の表は、このサブスクリプションに対して定義されているプロパティを示しています。

表 9-18

サブスクリプションのプロパティ	値
システム	<local system>
ソース・タイプ	外部
イベント・フィルタ	oracle.apps.wf.notification.receive

表 9-18 (続き)

サブスクリプションのプロパティ	値
フェーズ	0
ステータス	使用可能
ルール・データ	メッセージ
ルール関数	wf_xml.receive
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

関連項目：

8-52 ページ [「イベント・サブスクリプションの定義」](#)

通知応答のイベント

このイベントは、ユーザーが通知に応答するときに、Oracle Workflow によって呼び出されます。

内部名	oracle.apps.wf.notification.respond
ジェネレート関数	なし
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

通知完了のイベント

このイベントは、通知が完了したときに、Oracle Workflow によって呼び出されます。

内部名	oracle.apps.wf.notification.close
ジェネレート関数	なし
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

通知メーラー・イベント

要約通知の開始

このイベントは、通知メーラー・サービスのコンポーネントが、SUMMARY と通知環境設定されたロールに要約通知を送信させるために使用されます。要約通知には、要約が送信される時点でオープンされているロールへのすべての通知がリストとして表示されます。

内部名	oracle.apps.fnd.wf.mailer.Mailer.notification.summary
ジェネレート関数	なし
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

Oracle Workflow には、要約通知の開始イベントに対して、1つのデフォルト・サブスクリプションが用意されています。要約通知の開始イベントがローカルで呼び出されると、このサブスクリプションによって、通知環境設定が SUMMARY の各ロールに対して、oracle.apps.wf.notification.summary.send イベントが呼び出されます。このサブスクリプションは、デフォルトで使用可能になっています。次の表は、このサブスクリプションに対して定義されているプロパティを示しています。

表 9-19

サブスクリプションのプロパティ	値
システム	<local system>
ソース・タイプ	ローカル
イベント・フィルタ	oracle.apps.fnd.wf.mailer.Mailer.notification.summary
フェーズ	0
ステータス	使用可能
ルール・データ	メッセージ
ルール関数	wf_xml.summaryrule
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

関連項目：8-52 ページ「[イベント・サブスクリプションの定義](#)」

ビジネス・イベント・システム制御イベント

ビジネス・イベント・システム制御 Ping イベント

このイベントは、Workflow 制御キュー・クリーン・アップ・プログラムによって呼び出され、WF_CONTROL キューの中間層サブスクリイバへ送信されて、サブスクリイバが引き続きアクティブかどうかをチェックします。

内部名	oracle.apps.wf.bes.control.ping
ジェネレート関数	なし
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

ビジネス・イベント・システム制御グループ

このイベント・グループには、Oracle Applications またはスタンドアロン版 Oracle Workflow の内部処理のための制御イベントが含まれています。このグループ内のイベントは、中間層プロセスがデキューする WF_CONTROL キューに格納されます。

内部名	oracle.apps.wf.bes.control.group
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限
メンバー	oracle.apps.wf.bes.control.ping

Oracle Workflow には、ビジネス・イベント・システム制御グループに対して、1つのデフォルト・サブスクリプションが用意されています。グループ・メンバー・イベントの1つがローカルで呼び出されると、このサブスクリプションによって、イベント・メッセージが標準 WF_CONTROL キューに格納されます。その結果、このキューに対する中間層サブスクリイバが、イベント・メッセージをデキューできるようになります。このサブスクリプションは、デフォルトで使用可能になっています。次の表は、このサブスクリプションに対して定義されているプロパティを示しています。

表 9-20

サブスクリプションのプロパティ	値
システム	<local system>
ソース・タイプ	ローカル
イベント・フィルタ	oracle.apps.wf.bes.control.group
フェーズ	50
ステータス	使用可能
ルール・データ	キー
ルール関数	wf_rule.default_rule
送信エージェント	WF_CONTROL@<local system>
優先度	標準
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

ビジネス・イベント・システム・アプリケーション制御グループ

このイベント・グループには、Oracle Applications の内部処理のための制御イベントが含まれています。このグループ内のイベントは、Oracle Applications のコンテキスト情報で更新され、中間層プロセスがデキューする WF_CONTROL キューに格納されます。このグループには現在、シードされたメンバーはありません。

内部名	oracle.apps.fnd.bes.control.group
所有者名	Application Object Library
所有者タグ	FND
カスタマイズ・レベル	制限

Oracle Workflow には、ビジネス・イベント・システム・アプリケーション制御グループに対して、1つのデフォルト・サブスクリプションが用意されています。グループ・メンバー・イベントの1つがローカルで呼び出されると、このサブスクリプションによって、データベース ID やセキュリティ・グループ ID などの Oracle Applications のコンテキスト・データでイベント・メッセージが更新され、そのイベント・メッセージが標準 WF_CONTROL キューに格納されます。その結果、このキューに対する中間層サブスクライバが、イベント・メッセージをデキューできるようになります。このサブスクリプションは、デフォルトで使用可能になっています。次の表は、このサブスクリプションに対して定義されているプロパティを示しています。

表 9-21

サブスクリプションのプロパティ	値
システム	<local system>
ソース・タイプ	ローカル
イベント・フィルタ	oracle.apps.fnd.bes.control.group
フェーズ	50
ステータス	使用可能
ルール・データ	キー
ルール関数	fnd_bes_proc.process_event
送信エージェント	WF_CONTROL@<local system>
優先度	標準
所有者名	Application Object Library
所有者タグ	FND
カスタマイズ・レベル	制限

関連項目：

8-52 ページ [「イベント・サブスクリプションの定義」](#)

『Oracle Workflow 管理者ガイド』の「Workflow 制御キューのクリーン・アップ」

『Oracle WorkflowAPI リファレンス』の「ビジネス・イベント・システムのクリーン・アップ API」

汎用サービス・コンポーネント・フレームワーク制御イベント

開始イベント

このイベントは、汎用サービス・コンポーネント・フレームワークでサービス・コンポーネントを開始するために使用されます。

内部名	oracle.apps.fnd.cp.gsc.SvcComponent.start
ジェネレート関数	なし
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	コア

更新イベント

このイベントは、汎用サービス・コンポーネント・フレームワークで動的に更新できるパラメータに対して、実行中のサービス・コンポーネントを新しいパラメータ値で更新するために使用されます。

内部名	oracle.apps.fnd.cp.gsc.SvcComponent.refresh
ジェネレート関数	なし
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	コア

中断イベント

このイベントは、汎用サービス・コンポーネント・フレームワークで実行中のサービス・コンポーネントの処理を中断するために使用されます。

内部名	oracle.apps.fnd.cp.gsc.SvcComponent.suspend
ジェネレート関数	なし
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	コア

再開イベント

このイベントは、汎用サービス・コンポーネント・フレームワークで中断されていたサービス・コンポーネントの処理を再開するために使用されます。

内部名	oracle.apps.fnd.cp.gsc.SvcComponent.resume
ジェネレート関数	なし
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	コア

停止イベント

このイベントは、汎用サービス・コンポーネント・フレームワークでサービス・コンポーネントを停止するために使用されます。

内部名	oracle.apps.fnd.cp.gsc.SvcComponent.stop
ジェネレート関数	なし
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	コア

GSC ビジネス・イベント・システム制御グループ

このイベント・グループには、汎用サービス・コンポーネント・フレームワークの内部処理のためのイベントが含まれています。このグループ内のイベントは、中間層プロセスがデキューする WF_CONTROL キューに格納されます。

内部名	oracle.apps.fnd.cp.gsc.bes.control.group
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	コア
メンバー	oracle.apps.fnd.cp.gsc.SvcComponent.start oracle.apps.fnd.cp.gsc.SvcComponent.refresh oracle.apps.fnd.cp.gsc.SvcComponent.suspend oracle.apps.fnd.cp.gsc.SvcComponent.resume oracle.apps.fnd.cp.gsc.SvcComponent.stop

Oracle Workflow には、GSC ビジネス・イベント・システム制御グループに対して、1つのデフォルト・サブスクリプションが用意されています。グループ・メンバー・イベントの1

つがローカルで呼び出されると、このサブスクリプションによって、イベント・メッセージが標準 WF_CONTROL キューに格納されます。その結果、このキューに対する中間層サブスクリバが、イベント・メッセージをデキューできるようになります。このサブスクリプションは、デフォルトで使用可能になっています。次の表は、このサブスクリプションに対して定義されているプロパティを示しています。

表 9-22

サブスクリプションのプロパティ	値
システム	<local system>
ソース・タイプ	ローカル
イベント・フィルタ	oracle.apps.fnd.cp.gsc.bes.control.group
フェーズ	50
ステータス	使用可能
ルール・データ	キー
ルール関数	wf_rule.error_rule
送信エージェント	WF_CONTROL@<local system>
優先度	標準
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	コア

関連項目：

8-52 ページ [「イベント・サブスクリプションの定義」](#)

汎用サービス・コンポーネント・フレームワークの詳細は、Oracle Applications Manager のオンライン・ヘルプまたは Oracle Enterprise Manager のオンライン・ヘルプを参照してください。

ワークフロー送信プロトコル

ワークフロー送信プロトコル・プロセスは、イベント・メッセージの受信、送信および受信確認の例を示す、サンプル・ワークフロー・プロセスです。必要に応じて、このプロセスをコピーまたはカスタマイズし、組織固有の要件を取り込むことができます。

ワークフロー送信プロトコル・プロセスでは、サブスクリプションからイベント・メッセージを受信して、サブスクリプションに指定されているインバウンド・メッセージにイベント・メッセージを送信し、必要に応じて受信確認を待機します。また、必要に応じて受信確認を送信します。たとえば、あるシステムでこのプロセスを使用して、イベント・メッセージを別のシステムに送信できます。また、受信したシステムで同じプロセスを使用して、受信確認メッセージを送信元のシステムに返送することもできます。

「ワークフロー送信プロトコル」ワークフローは、ワークフロー・イベント・プロトコル・プロセスのみで構成されます。このプロセスは、イベント・サブスクリプションからイベント・メッセージを受信したときに開始されます。

「ワークフロー送信プロトコル」ワークフローは、次の方法で開始できます。

- 「イベントの呼出し」 ページから、ワークフロー送信プロトコル・イベントを呼び出します。一意のイベント・キーを入力し、有効な XML 文書をイベント・データとして入力します。事前定義済のサブスクリプションにより、イベント・メッセージはワークフロー・イベント・プロトコル・プロセスに送信されます。8-60 ページの「[イベントの呼出し](#)」および 9-45 ページの「[ワークフロー送信プロトコル・イベント](#)」を参照してください。
- エージェントがワークフロー送信プロトコル・イベントを外部ソースから受信したときに、プロセスを開始することもできます。事前定義済のサブスクリプションにより、イベント・メッセージはワークフロー・イベント・プロトコル・プロセスに送信されません。9-45 ページの「[ワークフロー送信プロトコル・イベント](#)」を参照してください。
- 独自のサブスクリプションを定義して、選択したイベントをワークフロー・イベント・プロトコル・プロセスに送信します。サブスクリプションのワークフローに対して、項目タイプを WFSNDPRT に、プロセス名を WFEVPRTC に指定します。イベントをワークフローに送信するために、デフォルトのルール関数を使用するか、送信処理をカスタム・ルール関数に組み込んでください。ワークフロー・イベント・プロトコル・プロセスは、サブスクリプションが実行され、プロセスがイベント・メッセージを受信したときに開始します。8-52 ページの「[イベント・サブスクリプションの定義](#)」を参照してください。

「ワークフロー送信プロトコル」項目タイプ

ワークフロー送信プロトコル・プロセスは、「ワークフロー送信プロトコル」という項目タイプに関連付けられています。現在、「ワークフロー送信プロトコル」には、ワークフロー・イベント・プロトコル・プロセスという1つのワークフローが関連付けられています。

「ワークフロー送信プロトコル」項目タイプの詳細を **Workflow Builder** に表示するには、「ファイル」メニューから「オープン」を選択します。データベースに接続して、「ワークフロー送信プロトコル」項目タイプを選択するか、ファイル・システムの `<ORACLE_HOME>/wf/Data/<language>` サブディレクトリにある `wfsndprt.wft` というファイルに接続します。

「ワークフロー送信プロトコル」のプロパティ・ページを見ると、維持タイプが「一時」に、維持日数が0になっています。これは、この項目タイプの作業項目に関連付けられているランタイム・データは、終了直後に削除の対象となることを意味します。

「ワークフロー送信プロトコル」項目タイプには、複数の属性が関連付けられています。これらの属性は、ワークフローのアプリケーション表にある情報を参照します。属性は、イベント・アクティビティと同様に、プロセス全体を通して関数アクティビティによって使用され保存されます。次の表は、「ワークフロー送信プロトコル」項目タイプの属性を示しています。

表 9-23

表示名	説明	型	長さ / 書式 / 選択肢タイプ
イベント名	イベントの内部名	テキスト	
イベント・キー	イベントの特定のインスタンスを一意に識別するイベント・キー	テキスト	
イベント・メッセージ	イベント・メッセージ	イベント	
宛先エージェント	イベント・メッセージを受信するインバウンド・エージェント (<code><agent>@<system></code> の書式)	テキスト	
送信元エージェント	イベント・メッセージを送信するアウトバウンド・エージェント (<code><agent>@<system></code> の書式)	テキスト	
受信確認が必要ですか?	送信するイベント・メッセージに、受信者からの受信確認が必要かどうかを指定するオプション	テキスト	

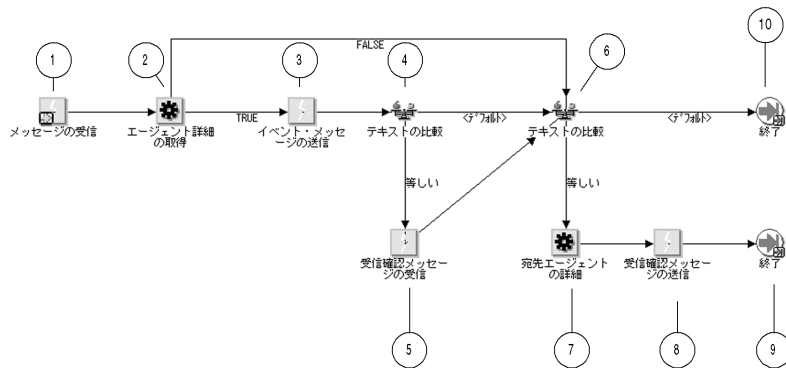
表 9-23 (続き)

表示名	説明	型	長さ / 書式 / 選択肢タイプ
受信確認を送信しますか?	受信したメッセージの受信確認を送信するかどうかを指定するオプション	テキスト	
受信確認メッセージ	送信される受信確認メッセージ	イベント	
宛先エージェントの確認	受信確認メッセージを受信するインバウンド・エージェント (書式は <agent>@<system>)	テキスト	
サブスクリプション GUID	サブスクリプションのグローバル一意識別子	テキスト	

ワークフロー・イベント・プロトコル・プロセスの概要

ワークフロー・イベント・プロトコル・プロセスのプロパティを表示するには、ナビゲータ・ツリーでプロセスを選択し、「編集」メニューから「プロパティ」を選択します。このプロセス・アクティビティは実行可能です。つまり、最上位レベルのプロセスとして実行できます。

ワークフロー・イベント・プロトコル・プロセスの「プロセス」ウィンドウを表示すると、このプロセスが 8 個の一意のアクティビティで構成されていることがわかります。そのうちのいくつかが再利用され、ワークフロー・ダイアグラムに表示される 10 個のアクティビティ・ノードを構成しています。プロセスのアクティビティを詳しく説明するために、次のように各ノードに番号を付けて参照しやすくしてあります。番号自体は、プロセス・ダイアグラムには含まれません。



「ワークフロー送信プロトコル」ワークフローは、イベント・マネージャからワークフロー・イベント・プロトコル・プロセスにイベント・メッセージが送信されたときに開始します。たとえば、ワークフロー送信プロトコル・イベントをローカルで呼び出すか、外部ソースからこのイベントを受信すると、事前定義済のサブスクリプションによって、ワークフロー・イベント・プロトコル・プロセスにイベント・メッセージが送信されます。9-45 ページの「ワークフロー送信プロトコル・イベント」を参照してください。

ワークフローは、ノード1の「受信」メッセージ・アクティビティから開始します。ノード2で、プロセスはサブスクリプションから目的のアウトバウンド・エージェントおよびインバウンド・エージェントのエージェント詳細を取得します。

インバウンド・エージェントが指定されていない場合、プロセスはノード6まで直接進み、受信確認メッセージを送信するかどうかを決定します。

サブスクリプションに「宛先エージェント」が指定されている場合、プロセスはそのエージェントにイベント・メッセージを送信します。プロセスは、サブスクリプション・パラメータに基づいて、イベント・メッセージに受信者からの受信確認が必要かどうかを決定します。受信確認が必要な場合、ワークフロー・エンジンは受信確認メッセージの受信を待機します。受信確認が必要な場合、プロセスはノード6まで直接進み、受信確認メッセージを送信するかどうかを決定します。

ノード6で、プロセスは受信した元のメッセージの受信確認を送信するかどうかを決定します。受信確認を送信する必要がない場合、この時点でプロセスは終了します。受信確認を送信する必要がある場合は、受信確認の送信先インバウンド・エージェントのエージェント詳細を取得して、受信確認をそのエージェントに送信した後で、プロセスは終了します。

ワークフロー・イベント・プロトコル・プロセスのアクティビティ

ここでは、プロセスの各アクティビティについて、アクティビティの表示名別に説明します。

メッセージの受信（ノード1）

このイベント・アクティビティは、イベント・マネージャからワークフロー・イベント・プロトコル・プロセスに送信されたイベント・メッセージを受信して、新しい項目を開始します。

イベント・アクション	受信
イベント・フィルタ	なし
前提条件アクティビティ	なし
アクティビティによって設定される項目属性	イベント名、イベント・キー、イベント・メッセージ

エージェント詳細の取得（ノード2）

この関数アクティビティは、メッセージを送信するアウトバウンド・エージェントおよびメッセージを受信するインバウンド・エージェントのサブスクリプションからエージェント詳細の取得を試みます。インバウンド・エージェントが指定されていない場合、プロセスはノード6まで直接進みます。インバウンド・エージェントは指定されているが、アウトバウンド・エージェントが指定されていない場合は、ローカル・システムのデフォルトのアウトバウンド・エージェントが関数によって選択されます。

関数	WF_STANDARD.GETAGENTS
結果タイプ	ブール
前提条件アクティビティ	メッセージの受信
関数によって取得される項目属性	サブスクリプション GUID
関数によって設定される項目属性	宛先エージェント、送信元エージェント

イベント・メッセージの送信（ノード3）

このイベント・アクティビティは、イベント・メッセージを、ローカル・システムのアウトバウンド・エージェントから、指定されたインバウンド・エージェントに送信します。

イベント・アクション	送信
前提条件アクティビティ	エージェント詳細の取得
アクティビティによって取得される項目属性	イベント・メッセージ、送信元エージェント、宛先エージェント

テキストの比較（ノード 4）

この「標準」関数アクティビティは、2つのテキストの値を比較します。このノードでは、プロセスは「受信確認が必要ですか？」項目属性をチェックして、ノード3で送信されるイベント・メッセージに受信者からの受信確認が必要かどうかを判断します。

プロセスを開始したサブスクリプションにパラメータ名と値のペア「ACKREQ=Y」が含まれていた場合、ワークフロー・エンジンはプロセスの開始時に「受信確認が必要ですか？」項目属性をYに設定します。この場合、プロセスはノード4からノード5に進みます。それ以外の場合は、ノード4からノード6に直接進みます。

関数	WF_STANDARD.COMPARE
結果タイプ	比較
前提条件アクティビティ	なし
アクティビティによって取得される項目属性	受信確認が必要ですか？

受信確認メッセージの受信（ノード 5）

このイベント・アクティビティは、ワークフロー送信プロトコル受信確認イベントの受信を待機します。つまり、ワークフロー・イベント・プロトコル・プロセスのノード3から送信されたイベント・メッセージに対して、それを受信したシステムが返す受信確認イベントを待機します。

イベント・アクション	受信
イベント・フィルタ	oracle.apps.wf.event.wf.ack
前提条件アクティビティ	テキストの比較

テキストの比較（ノード 6）

この「標準」関数アクティビティは、2つのテキストの値を比較します。このノードで、プロセスは「受信確認を送信しますか？」項目属性をチェックして、着信した元のメッセージの受信確認を送信するかどうかを判断します。

元のメッセージに受信確認が必要な場合、ワークフロー・エンジンはプロセスの開始時に、「受信確認を送信しますか？」項目属性をYに設定します。この場合、プロセスはノード6からノード7に進みます。それ以外の場合、プロセスはノード10で終了します。

関数	WF_STANDARD.COMPARE
結果タイプ	比較
前提条件アクティビティ	なし
アクティビティによって取得される項目属性	受信確認を送信しますか？

宛先エージェントの詳細（ノード7）

この関数アクティビティは、受信確認を必要とする送信元システムのインバウンド・エージェントを選択して、そのエージェントのエージェント詳細を取得します。

関数	WF_STANDARD.GETACKAGENT
結果タイプ	なし
前提条件アクティビティ	テキストの比較
関数によって設定される項目属性	宛先エージェントの確認

受信確認メッセージの送信（ノード8）

このイベント・アクティビティは、「ワークフロー送信プロトコル受信確認」メッセージを、ローカル・システムのアウトバウンド・エージェントからノード7で識別されたインバウンド・エージェントに送信します。

イベント・アクション	送信
前提条件アクティビティ	宛先エージェント詳細
アクティビティによって取得される項目属性	受信確認メッセージ、イベント・キー、宛先エージェントの確認

終了（ノード9および10）

この「標準」関数アクティビティは、プロセスの終了をマークします。

関数	WF_STANDARD.NOOP
結果タイプ	なし
前提条件アクティビティ	なし

ワークフロー送信プロトコル・イベント

ワークフロー送信プロトコル・イベント

このイベントを「イベントの呼出し」Web ページから呼び出すと、「ワークフロー送信プロトコル」項目タイプのワークフロー・イベント・プロトコル・プロセスを使用して、イベントをエージェントに送信できます。このワークフロー・プロセスには、メッセージに受信者からの受信確認が必要かどうか、受信したメッセージの受信確認を送信するかどうかを指定できます。

内部名	oracle.apps.wf.event.wf.send
ジェネレート関数	なし
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

Oracle Workflow には、ワークフロー送信プロトコル・イベントに対して、2つのデフォルト・サブスクリプションが用意されています。一方のサブスクリプションでは、ワークフロー送信プロトコル・イベントがローカルで呼び出されたときに、「ワークフロー送信プロトコル」項目タイプのワークフロー・イベント・プロトコル・プロセスにイベント・メッセージが送信されます。サブスクリプション・パラメータに対して、メッセージに受信確認が必要であることを指定します。このサブスクリプションは、デフォルトで使用可能になっています。カスタマイズ・レベルを「ユーザー」に設定すれば、同じプロパティでこのサブスクリプションのコピーを作成し、アウトバウンドおよびインバウンド・エージェントをそのサブスクリプションに追加し、Workflow イベント・プロトコル・プロセスがイベント・メッセージを送信する宛先を指定することもできます。次の表は、このサブスクリプションに対して定義されているプロパティを示しています。

表 9-24

サブスクリプションのプロパティ	値
システム	<local system>
ソース・タイプ	ローカル
イベント・フィルタ	oracle.apps.wf.event.wf.send
フェーズ	50
ステータス	使用可能
ルール・データ	キー
ルール関数	wf_rule.workflow_protocol
ワークフロー項目タイプ	WFSNDPRT

表 9-24 (続き)

サブスクリプションのプロパティ	値
ワークフロー・プロセス名	WFEVPRTC
パラメータ	ACKREQ=Y
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

もう一方のサブスクリプションでは、「ワークフロー送信プロトコル・イベントが外部ソースから着信したときに、ワークフロー送信プロトコル」項目タイプのワークフロー・イベント・プロトコル・プロセスにイベント・メッセージが送信されます。このサブスクリプションは、デフォルトで使用可能になっています。カスタマイズ・レベルを「ユーザー」に設定すれば、同じプロパティでこのサブスクリプションのコピーを作成し、アウトバウンドおよびインバウンド・エージェントをそのサブスクリプションに追加して、Workflow イベント・プロトコル・プロセスがイベント・メッセージを別のエージェントへ送信するように指定することもできます。次の表は、このサブスクリプションに対して定義されているプロパティを示しています。

表 9-25

サブスクリプションのプロパティ	値
システム	<local system>
ソース・タイプ	外部
イベント・フィルタ	oracle.apps.wf.event.wf.send
フェーズ	50
ステータス	使用可能
ルール・データ	キー
ルール関数	wf_rule.workflow_protocol
ワークフロー項目タイプ	WFSNDPRT
ワークフロー・プロセス名	WFEVPRTC
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

ワークフロー送信プロトコル受信確認イベント

このイベントは、ワークフロー・イベント・プロトコル・プロセスに送信されたイベント・メッセージまたはワークフロー・イベント・プロトコル・プロセスから着信したイベント・メッセージに対して受信確認が必要なときに、発信元のシステムに返送されます。

内部名	oracle.apps.wf.event.wf.ack
ジェネレート関数	なし
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

Oracle Workflow には、ワークフロー送信プロトコル受信確認イベントに対して、1つのデフォルト・サブスクリプションが用意されています。このサブスクリプションでは、ワークフロー送信プロトコル受信確認イベントが外部ソースから着信したときに、「ワークフロー送信プロトコル」項目タイプのワークフロー・イベント・プロトコル・プロセスにイベント・メッセージが送信されます。このサブスクリプションは、デフォルトで使用可能になっています。次の表は、このサブスクリプションに対して定義されているプロパティを示しています。

表 9-26

サブスクリプションのプロパティ	値
システム	<local system>
ソース・タイプ	外部
イベント・フィルタ	oracle.apps.wf.event.wf.ack
フェーズ	50
ステータス	使用可能
ルール・データ	キー
ルール関数	wf_rule.workflow_protocol
ワークフロー項目タイプ	WFSNDPRT
ワークフロー・プロセス名	WFEVPRTC
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

関連項目：

8-52 ページ [「イベント・サブスクリプションの定義」](#)

10

デモンストレーション用ワークフロー・プロセス

この章では、Oracle Workflow に用意されているデモンストレーション用のワークフロー・プロセスについて説明します。

サンプル・ワークフロー・プロセス

Oracle Workflow には、次のサンプル・ワークフロー・プロセスが含まれています。各プロセスは、様々な Oracle Workflow 機能の統合例です。これらのプロセスを使用して、Oracle Workflow のインストールをチェックできます。

- 購買申請プロセス： 結果ベースの分岐、並列分岐、サブプロセス、タイムアウト、ループおよび通知での PL/SQL 文書の統合を示します。10-5 ページの「[購買申請プロセス](#)」を参照してください。
- ドキュメント・レビュー・プロセス： 文書管理の統合（この機能は今後使用する目的で確保されています）とループを示します。10-36 ページの「[ドキュメント・レビュー・プロセス](#)」を参照してください。
- エラー・チェック・プロセス： Oracle Alert の定期アラート機能をワークフロー・プロセスでシミュレーションする方法および標準「待機」アクティビティの使用法を示します。10-42 ページの「[エラー・チェック・プロセス](#)」を参照してください。
- イベント・システム・デモンストレーション・プロセス： 2つのシステム間のビジネス・イベントの送受信と外部 Java 関数アクティビティの使用法の例を示します。10-52 ページの「[イベント・システム・デモンストレーション](#)」を参照してください。

注意： Oracle Workflow のスタンドアロン版を使用している場合は、Workflow Configuration Assistant により、購買申請およびイベント・システム・デモンストレーション・プロセスのみがインストールされます。

Oracle Application に組み込まれている Workflow を使用している場合は、AutoUpgrade により、ドキュメント・レビューおよびエラー・チェック・プロセスのみがインストールされます。この2つのサンプル・ワークフロー・プロセスは、サポートするデータ・モデルの作成を必要としません。

これらのサンプル・ワークフローは、「ワークフロー・デモンストレーション」ホーム・ページまたは「プロセスの開始」Web ページから開始できます。次の URL を使用して、「ワークフロー・デモンストレーション」ホーム・ページにアクセスできます。

```
<webagent>/wf_demo.home
```

注意： 「ワークフロー・デモンストレーション」Web ページには、Oracle Workflow ホーム・ページからもアクセスできます。『Oracle Workflow 管理者ガイド』の「Oracle Workflow ホーム・ページへのアクセス」を参照してください。

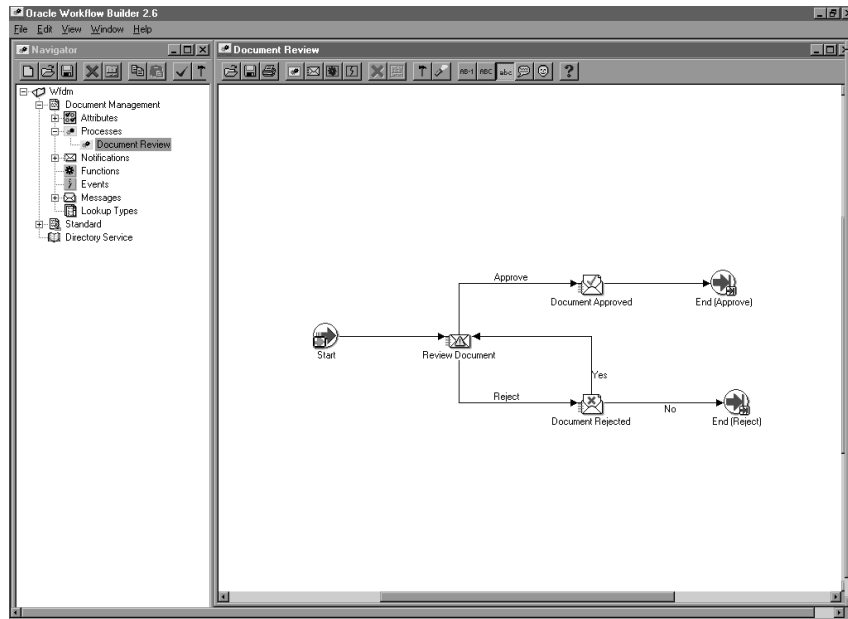
「ワークフロー・デモンストレーション」 ホーム・ページでは、右側のフレームに通知のワークリストが表示され、左側のフレームには、他の Web ページから各サンプル・ワークフローを開始するためのリンクがリストされます。

関連項目：

7-2 ページ「ワークフロー定義のテスト」

サンプル・ワークフローのプロセス・ダイアグラムの表示

Oracle Workflow Builder の「プロセス」ウィンドウに、サンプル・プロセスのプロセス・ダイアグラムを表示できます。



▶ **Oracle Workflow Builder でのサンプル・プロセスの表示**

1. 「ファイル」メニューから「オープン」を選択します。使用している Oracle Workflow データベースに接続して、必要な項目タイプを選択します。

または、PC 上の Oracle Workflow の wf/data/<language> サブディレクトリにある、どのサンプル・ワークフロー定義ファイルに接続することもできます。

2. データ・ソースを展開し、そのデータ・ソース内の項目タイプのブランチを展開します。
3. 「プロセス」のブランチを展開してプロセス・アクティビティをダブルクリックし、そのプロセスのダイアグラムを「プロセス」ウィンドウに表示します。

購買申請プロセス

購買申請プロセスは、品目を購入するための新規購買申請を作成するときに開始されるワークフロー・プロセスの一例です。購買申請プロセスは、承認階層と承認限度情報が格納される2つの表に基づいています。

このデモンストレーションでは、購買申請を発行すると、プロセスによって通知が承認階層の次のマネージャに送信され、購買申請が承認されます。承認担当マネージャの承認限度額が購買申請金額より低ければ、その購買申請を承認できる限度額を持つマネージャが見つかるまで、購買申請は承認階層内で次に上位のマネージャへと順に転送されます。各中間マネージャは、その購買申請を承認して次に上位のマネージャにまわす必要があります。適切な承認限度を持つマネージャが購買申請を承認すると、「承認する」という結果でプロセスが終了します。

次の場合には、プロセスが「否認」という結果で終了することがあります。

- いずれかのマネージャが購買申請を否認した場合
- 購買申請金額が最大の承認限度を超えている場合
- 購買申請者を管理するマネージャがいない場合

Oracle Workflow のスタンドアロン版を使用している場合は、このサンプル・プロセスを設定し、実行できます。Oracle Application に組み込まれている Workflow を使用している場合、このプロセスは、主にデモンストレーションではなく説明のための一例とを考えてください。Oracle Application に組み込まれている Workflow には、このデモンストレーションの設定と実行に必要なファイルが用意されていません。

注意： Oracle Applications または Oracle Self-Service Web Applications と統合された実行可能ワークフロー・プロセスの詳細は、該当する Oracle Applications のユーザーズ・ガイドまたはオンライン・マニュアルを参照してください。C-2 ページの「[Oracle E-Business Suite 埋込みの事前定義済のワークフロー](#)」を参照してください。

注意： Oracle Self-Service Web Applications では、ここで説明するサンプル・プロセスとは異なるバージョンの購買申請プロセスが事前定義されています。この項で説明するサンプル・プロセスは、あくまでもデモンストレーション用であり、実作業用ではありません。

このサンプル・ワークフローは、デモンストレーション・データ・モデルに基づいています。このデータ・モデルには、データの入った2つの表が組み込まれています。一方の表では従業員の承認階層を管理し、他方では各従業員の費用限度額を管理します。この2つの表が、購買申請の承認に使用するデータベース・アプリケーションを構成しています。また、

このデータ・モデルには、このサンプル実装に使用されている Oracle Workflow ユーザーとロールを識別するディレクトリ・サービスも組み込まれています。

架空の購買申請に基づいて購買申請プロセスを実行するには、2つの方法があります。つまり、スクリプトを実行する方法と、Web ベースのインタフェースを使用して購買申請を発行する方法です。どちらの方法でも、購買申請を作成した従業員の氏名、購買申請金額、購買申請番号、購買申請摘要、購買申請プロセス所有者および開始するワークフロー・プロセスの名称を指定する必要があります。

この項では、購買申請プロセスの詳細を説明し、このワークフローの各アクティビティが何を達成するのか理解できるようにします。

購買申請データ・モデルのインストール

購買申請データ・モデルは、Oracle Workflow のスタンドアロン版でのみインストールされます。データ・モデルは、Workflow Configuration Assistant によって自動的にインストールされます。インストールで使用されるファイルは、Oracle Workflow Server のディレクトリ構造の demo および demo//<language> サブディレクトリにコピーされます。

注意： 購買申請プロセスのデモンストレーションを正常に機能させるには、インストール後に、Oracle Workflow の必須設定手順を実行する必要があります。『Oracle Workflow 管理者ガイド』の「設定の概要」を参照してください。

インストール時には、次の処理が行われます。

- スクリプト `wfdemou.sql` がコールされ、後述のシード・データ表にリストされたユーザーごとに、データベース・アカウントが作成されます。このスクリプトは、パブリック権限とシノニムを作成し、これらのアカウントが Oracle Workflow の Web ベースのユーザー・インタフェースにフルアクセスできるようにします。

注意： セキュリティ上の理由から、インストール・プロセスでは、これらのユーザー・アカウントは作成後に自動的にロックされます。アカウントを使用する前に、`wfdemou1.sql` というスクリプトを使用して、これらのアカウントをロック解除する必要があります。このスクリプトは、Oracle ホームの `wf/demo` サブディレクトリにあります。SQL*Plus を使用して SYSTEM データベース・アカウントに接続し、次のコマンドを使用してスクリプトを実行します。

```
sqlplus SYSTEM/<SYSTEM pwd> @wfdemou1
```

SYSTEM アカウントおよびパスワードの詳細は、Oracle データベース管理者に確認してください。

- スクリプト `wfdemoc.sql` がコールされ、シード・データで2つの表が作成されます。これらの表は、ワークフローを使用可能なデモンストレーション・データベース・アプリケーションを構成します。
 - `WF_REQDEMO_EMP_HIERARCHY`: 従業員の承認階層が保存されます。承認チェーンは、これらの従業員のユーザー ID で構成されます。BLEWIS、KWALKER、CDOUGLAS および SPIERSON というように、各従業員は昇順で表示され、最大の権限を持つ従業員が最後に表示されます。
 - `WF_REQDEMO_EMP_AUTHORITY`: 各従業員の支払限度額が保存されます。各従業員の限度額は、BLEWIS:500、KWALKER:1000、CDOUGLAS:2000、SPIERSON:3000 というように、従業員のユーザー ID の後に表示されます。
- また、スクリプト `wfdemoc.sql` では、シード・データも `WF_LOCAL_ROLES` 表と `WF_LOCAL_USER_ROLES` 表に挿入されます。次の表は、スクリプトによってシードされるユーザーおよびロールを示しています。

表 10-1

User	ADMIN ロール	MANAGERS ロール	WORKERS ロール	OTHERS ロール
SYSADMIN	yes			
WFADMIN	yes			
BLEWIS			yes	
KWALKER			yes	
CDOUGLAS			yes	yes
SPIERSON		yes		yes

注意: 各ユーザーは、電子メール・アドレス `WFINVALID` を持ち、各ロールはロール名と同じ電子メール・アドレスを持ちます。Directory Service API の `SetAdHocUserAttr` または `SetAdHocRoleAttr` をコールすると、ユーザーおよびロールの電子メール・アドレスを他の値に変更できません。また、すべてのユーザーおよびロールへの電子メール通知が1つの電子メール受信ボックスに入るようにする場合は、通知メーラーを構成するときに、`Test Address` 構成パラメータにテスト用電子メール・アドレスを指定します。詳細は、Oracle Application に組み込まれている Workflow の場合は Oracle Applications Manager オンライン・ヘルプの Oracle Workflow Manager の項、スタンドアロン版の Oracle Workflow の場合は Oracle Enterprise Manager オンライン・ヘルプの Oracle Workflow Manager の項を参照してください。

注意： BLEWIS 以外のユーザーは、すべて通知環境設定 MAILHTML を持ち、これによって、「通知」Web ページから通知を参照する他に、電子メールを介して通知を個々に受け取ることができます。BLEWIS は通知環境設定 SUMMARY を持ち、「通知」Web ページから通知を参照する以外に、現在オープンしている全通知の要約を、定期的に電子メールで受け取ることができます。通知メーラーを電子メールの通知を配信するように設定する必要があるため、注意してください。

注意： 購買申請データ・モデルのユーザーとロールを含めるには、Oracle Workflow のディレクトリ・サービス・ビューを WF_LOCAL_ROLES 表と WF_LOCAL_USER_ROLES 表にマップする必要があります。『Oracle Workflow 管理者ガイド』の「Oracle Workflow のディレクトリ・サービスの設定」を参照してください。

- スクリプト wfdemos.sql および wfdemob.sql がコールされ、WF_REQDEMO および WF_DEMO と呼ばれるパッケージの PL/SQL の仕様と本文が作成されます。これらのパッケージには、次の内容が含まれます。
 - デモンストレーション・ホーム・ページに関連した PL/SQL ストアド・プロシージャ
 - 購買申請プロセスのワークフローで使用される関数アクティビティでコールされる PL/SQL ストアド・プロシージャ
 - 購買申請プロセスのデモンストレーション用に Web ベースのインタフェース・ページを生成するために、Oracle Workflow の Web Agent によってコールされる PL/SQL プロシージャの WF_REQDEMO.Create_Req
- ワークフロー・リソース・ジェネレータ・プログラムが実行され、wfdemo.msg からデータベースにメッセージがロードされます。メッセージは、購買申請プロセスのデモンストレーション用に、Web ベースのインタフェース・ページによって使用されます。
- 購買申請プロセスのワークフロー定義が、wfdemo.wft からデータベースにロードされます。このプロセスは、Oracle Workflow Builder で表示できます。

購買申請ワークフローの開始

購買申請ワークフローを開始するには、次のいずれかの方法を使用できます。

- スクリプト `gwfrund.sql` を実行します。
- 「ワークフロー・デモンストレーション」 ホーム・ページから、「購買申請デモンストレーション」 Web ページにアクセスします。
- 「プロセスの開始」 Web ページを使用します。7-2 ページの「ワークフロー定義のテスト」を参照してください。

また、ユーザーが、購買申請プロセスのワークフローを自動的に開始する購買申請を作成できるように、カスタムのエンド・ユーザー・アプリケーション・インタフェースを作成することもできます。ただし、ユーザーが購買申請をアプリケーション・データベースに保存するときに、購買申請プロセスを開始する `WF_REQDEMO.StartProcess` と同じような PL/SQL ストアド・プロシージャをアプリケーションがコールするように、アプリケーション・インタフェースをカスタマイズする必要があります。10-25 ページの「[StartProcess 関数の例](#)」を参照してください。

▶ `wfrund.sql` の実行

1. 次のコマンドを入力し、スクリプト `wfrund.sql` を SQL*Plus で実行します。

```
sqlplus <username>/<password>@<alias> @wfrund.sql  
<req_num> <req_desc> <req_amount> <requestor>  
<req_process_owner> <process_int_name> <item_type>
```

`<username>/<password>@<alias>` を、ユーザー名、パスワードおよびデモンストレーション・データ・モデルをインストールしたデータベース・アカウントの別名に置き換えます。

`<req_num>` を、特定の購買申請を識別する購買申請番号に置き換えます。

注意： 購買申請番号がプロセスの項目キーとして使用されるので、シングルバイト・キャラクタのみ使用できます。

`<req_desc>` を、特定の購買申請を識別するエンド・ユーザー定義の摘要に置き換えます。

`<req_amount>` を購買申請金額に、`<requestor>` を購買申請作成者（従業員の承認階層に表示される従業員）の氏名に、`<req_process_owner>` を購買申請プロセスの所有者（従業員の承認階層に表示される従業員）の氏名に、`<process_int_name>` をプロセス・アクティビティの内部名（この場合は `REQUISITION_APPROVAL`）に、`<item_type>` をワークフロー・プロセスが関連付けられている項目タイプの内部名に置き換えます。

2. このスクリプトの完了後に、SQL> プロンプトから `Commit` と入力し、トランザクションを保存して、SQL*Plus を終了します。

- 承認階層に基づいて、購買申請作成者または作成者のマネージャとしてログオンし、プロセスを完了まで進める一連の通知メッセージに従い、応答できます。『Oracle Workflow ユーザーズ・ガイド』の「電子メールによる通知の閲覧」および「Web ブラウザによる通知の表示」を参照してください。

また、ワークフロー・モニターを使用して、ワークフロー・プロセスのステータスを表示することもできます。『Oracle Workflow 管理者ガイド』の「プロセス検索」Web ページの使用」を参照してください。

▶ 「購買承認申請」 Web ページの使用

- Web ブラウザに次の URL を入力して「ワークフロー・デモンストレーション」 Web ページにアクセスします。次に、「購買承認申請」リンクをクリックして、「購買承認申請」 Web ページを表示します。

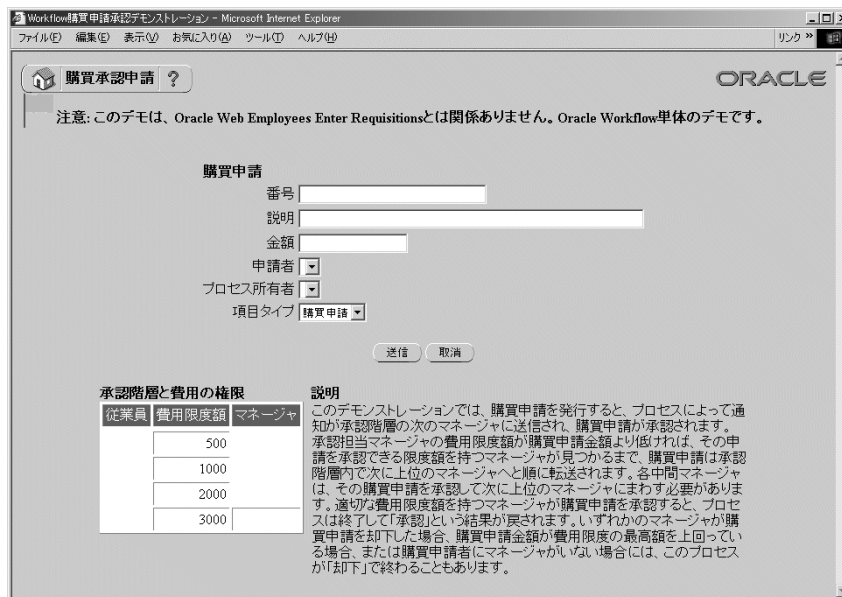
```
<webagent>/wf_demo.home
```

<webagent> は、Web サーバーで Oracle Workflow 用に構成された Web エージェントのベース URL を表します。『Oracle Workflow 管理者ガイド』の「グローバル・ユーザー設定の設定」を参照してください。

あるいは、次の URL を入力して「購買承認申請」 Web ページを直接表示できます。

```
<webagent>/wf_reqdemo.create_req
```

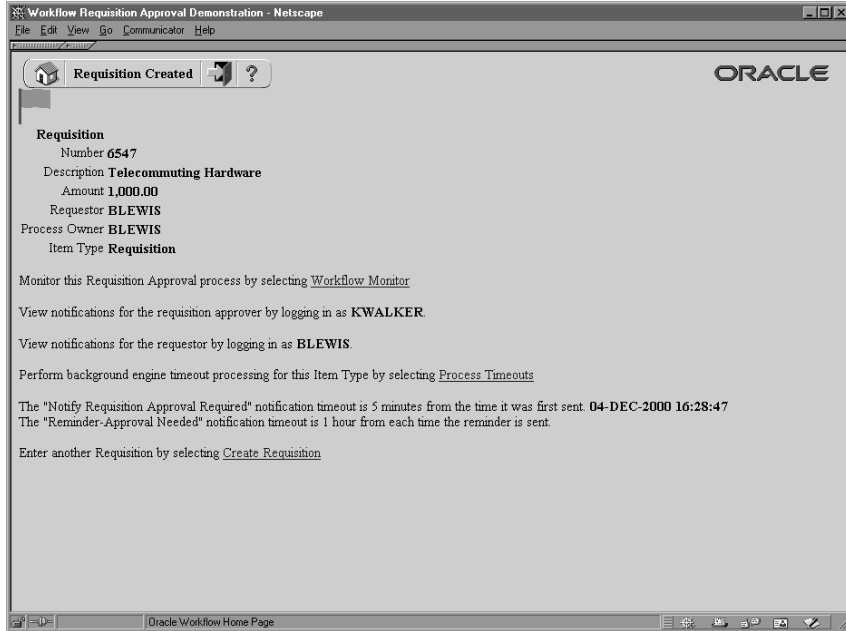
注意： どちらのページにもセキュリティが適用されるため、現行 Web セッションで有効なワークフロー・ユーザーとしてログオンしていない場合は、ページが表示される前に有効なワークフロー・ユーザーとしてのログオンを求めるプロンプトが表示されます。



2. 一意の購買申請番号を入力します。

注意： 購買申請番号がプロセスの項目キーとして使用されるので、シングルバイト・キャラクタのみ使用できます。

3. 特定の購買申請摘要を半角英数字 80 文字以内で指定します。
4. 購買申請金額を入力します。この金額は書式なしの数値として入力する必要があります。
5. ドロップ・ダウン・フィールドを使用して、購買申請者とプロセス所有者を指定します。これらのドロップ・ダウン・リストに表示される名前は、デモンストレーション・データ・モデル内のロール名に限定されています。
6. 「購買申請」の入力フィールドの次に、「承認階層と費用の権限」の表、および購買申請デモンストレーション・プロセスがどのように動作するかの説明があります。「承認階層と費用の権限」の表は、デモンストレーション・データ・モデルの中身の要約です。
7. 「送信」を選択して購買申請プロセスを開始し、「購買申請が作成されました」確認ページに移動します。



8. 確認ページには、どのロールでログインすると、プロセスの通知を参照できるかという情報の他に、ワークフロー・モニターへの HTML リンクがあり、ここで、「ダイアグラム表示」を選択すると、ADMIN モードで送信した購買申請のプロセス・ダイアグラムが表示されます。『Oracle Workflow 管理者ガイド』の「ワークフロー・モニター」を参照してください。
9. 「プロセスのタイムアウト回数」HTML リンクを選択して、バックグラウンド・エンジンでタイムアウト通知を検索し、タイムアウト時に実行される予定の次のアクティビティを実行します。

このリンクの下には、プロセスにタイムアウトが発生する時期を示す 2 つのメッセージが表示されます。
10. 別の購買申請を「購買申請デモンストレーション」Web ページに入力して送信する場合は、「購買申請の作成」HTML リンクを選択します。

「購買申請」項目タイプ

購買申請プロセスには、「購買申請」と呼ばれる項目タイプが関連付けられています。購買申請に関連付けられているワークフロー・プロセスは、現在2つあります。「購買承認申請」と「承認者に通知」です。

購買申請のプロパティ画面を見ると、維持タイプが「一時」に、維持日数が0になっています。これは、この項目タイプの作業項目に関連付けられているランタイム・データは、終了直後に削除の対象となることを意味します。また、WF_REQDEMO.SELECTOR というセレクト関数がコールされることもわかります。このセレクト関数は、指定した項目タイプに複数のプロセスが存在する場合に、実行するプロセス名を戻す PL/SQL ストアド・プロシージャのサンプルです。このセレクト関数の例では、実行するプロセスとして、REQUISITION_APPROVAL または「購買承認申請」を返します。

「購買申請」項目タイプには、複数の属性も関連付けられています。これらの属性は、デモンストレーション・アプリケーションの表にある情報を参照します。属性は、通知アクティビティと同様に、プロセス全体を通して関数アクティビティによって使用され保存されます。次の表は、「購買申請」項目タイプの属性を示しています。

表 10-2

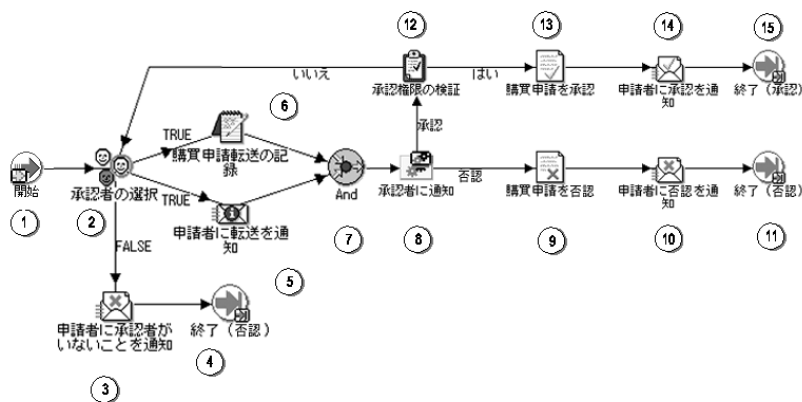
表示名	説明	型	長さ / 書式 / 選択肢タイプ
転送元ユーザー名	購買申請の転送元のユーザー名	ロール	
転送先ユーザー名	購買申請の転送先のユーザー名	ロール	
申請者のユーザー名	購買申請作成者のユーザー名	ロール	
購買申請金額	購買申請金額	数値	9,999,999,999.99
購買申請番号	購買申請の一意の識別子	テキスト	
モニター URL	モニター URL	URL	
購買申請摘要	購買申請の一意のユーザー識別子	テキスト	80
購買申請プロセス所有者	購買申請所有者のユーザー名	ロール	
購買申請の督促文書	PL/SQL により作成される購買申請の督促文書	ドキュメント	
備考	備考	テキスト	
督促用の備考	前の応答者の備考を保存するための備考	テキスト	

購買承認申請プロセスの概要

購買承認申請プロセスのプロパティを表示するには、ナビゲータ・ツリーでプロセスを選択し、「編集」メニューから「プロパティ」を選択します。購買承認申請プロセスには、プロセスの完了時に結果が「承認」になるか「否認」になるか（標準項目タイプに関連付けられた「承認」選択肢タイプの選択肢コード）を示す「承認」という結果タイプがあります。このプロセス・アクティビティも実行可能です。つまり、最上位レベルのプロセスとして開始できます。

プロセス・アクティビティの「詳細」プロパティ画面では、購買申請にエラー・プロセスが割り当てられており、プロセスでエラーが起きた場合にのみ起動されることが示されます。このエラー・プロセスには WFERROR という項目タイプが関連付けられており、DEFAULT_ERROR と呼ばれます。たとえば、承認階層にリストされていない従業員によって作成された購買申請で、購買承認申請プロセスを起動すると、「承認者の選択」アクティビティの実行を試みたときに、ワークフロー・エンジンはエラーを起こします。このエラーにより、デフォルト・エラー・プロセスである WFERROR/DEFAULT_ERROR が実行されます。11-8 ページの「デフォルト・エラー・プロセス」を参照してください。

購買承認申請プロセスの「プロセス」ウィンドウを表示すると、このプロセスが 12 個の一意のアクティビティで構成されていることがわかります。そのうちのいくつかが再利用され、ワークフロー・ダイアグラムに表示される 15 個のアクティビティ・ノードを構成しています。プロセスのアクティビティを詳しく説明するために、次のように各ノードに番号を付けて参照しやすくしてあります。番号自体は、プロセス・ダイアグラムには含まれません。



スクリプト wfrund.sql を実行するか、「購買申請デモンストレーション」Web ページを使用して購買申請を発行すると、購買申請ワークフローが開始されます。どちらの場合も、購買申請者、購買申請番号、購買申請金額、購買申請摘要およびプロセス所有者を指定する必要があります。10-9 ページの「購買申請ワークフローの開始」を参照してください。

ワークフローは、ノード 1 の「開始」アクティビティから開始します。

ノード 2 では、プロセスは購買申請の承認者を選択しようとしています。購買申請の承認者が見つからなければ、申請者に通知され、プロセスは「否認」の最終プロセス結果で終了します。承認者が見つかると、その承認者が誰なのかが申請者に通知され、購買否認が承認者に転送されていることがアプリケーション内で関数により記録されます。この 2 つのアクティビティは、どちらもノード 8 で実際に承認者に通知される前に完了する必要があります。

ノード 8 は、指定された期間中に購買申請を承認するように、承認者に要求するサブプロセスです。その時刻までに承認者が応答しなければ、サブプロセスによって「タイムアウト」アクティビティが実行され、承認者が応答するまで督促通知が送信され続けます。承認者が購買申請を否認すると、購買申請はノード 9 で否認されたものとして更新され、ノード 10 で申請者に通知されます。プロセスはこの時点で、「否認」という結果で終了します。

承認者が購買申請を承認すると、プロセスはノード 12 に移動し、購買申請金額が承認者の支払限度額の範囲内かどうかを検証されます。範囲内であれば、購買申請はノード 13 で承認され、ノード 14 で申請者に通知されます。この場合、プロセスは「承認」という結果で終了します。

購買申請プロセス・アクティビティ

ここでは、各アクティビティの説明をアクティビティの表示名ごとに示します。アクティビティのコンポーネントは、関数アクティビティによってコールされる PL/SQL ストアド・プロシージャ以外すべて、グラフィカルな Oracle Workflow Builder で作成できます。関数アクティビティは、Oracle Advanced Queuing を統合することで、データベース外の関数を実行できます。また、Oracle Database で作成し、格納した PL/SQL ストアド・プロシージャも実行できます。購買申請プロセスの関数アクティビティはすべて、PL/SQL ストアド・プロシージャを実行します。購買申請プロセスで使用される PL/SQL ストアド・プロシージャのネーミング規則は、次のとおりです。

`WF_REQDEMO.<PROCEDURE>`

WF_REQDEMO は、購買申請プロセスに使用されるすべてのプロシージャがグループ化されたパッケージの名前です。<PROCEDURE> は、プロシージャ名です。

いくつかのアクティビティについてはより詳細に説明し、その設計根拠についても説明します。10-28 ページの「[関数アクティビティの例](#)」および 10-33 ページの「[通知アクティビティの例](#)」を参照してください。

開始（ノード1）

これは、単にプロセスの開始をマークする「標準」関数アクティビティです。

関数 WF_STANDARD.NOOP

結果タイプ なし

前提条件アクティビティ なし

承認者の選択（ノード2）

この関数アクティビティでは、架空の従業員の承認階層表がチェックされ、購買申請の次の承認者が決定されます。また、このアクティビティでは、前の承認者の名前か、その購買申請がまだ一度も承認されていない場合は作成者の名前が保存されます。承認者が見つかる、このプロセスでは TRUE を表す値「T」が戻され、それ以外の場合は FALSE を表す値「F」が戻されます。

関数 WF_REQDEMO.SelectApprover

結果タイプ ブール

前提条件アクティビティ なし

申請者に承認者がいないことを通知（ノード3）

このアクティビティでは、購買申請に該当する承認者が見つからなかったことが、購買申請作成者に通知されます。メッセージには、購買申請番号、購買申請摘要、購買申請金額および最終承認者がいればその氏名を表示する「送信」属性が含まれます。メッセージには、通知ヘッダーに表示される、通知を送信したロールの「送信」属性も含まれます。

このアクティビティは、プロセス・ノード3で発生します。ノードのプロパティ画面を表示すると、「申請者のユーザー名」という項目タイプ属性に名前が保存されている実行者に、このアクティビティが割り当てられていることがわかります。

メッセージ 購買承認申請者が見つからない

結果タイプ なし

前提条件アクティビティ 承認者の選択

申請者に転送を通知（ノード5）

このアクティビティでは、購買申請が承認のために転送されたことが、購買申請作成者に通知されます。メッセージには、購買申請番号、購買申請摘要、購買申請金額、購買申請の転送先となる承認者の氏名、直前の承認者がいればその氏名、および購買申請に追加された最新の注釈などを表示する「送信」属性が含まれます。メッセージには、通知ヘッダーに表示される、通知を送信したロールの「送信」属性も含まれます。

ノードのプロパティ画面を表示すると、「申請者のユーザー名」という項目タイプ属性に名前が保存されている実行者に、このアクティビティが割り当てられていることがわかります。

メッセージ	購買申請の転送
結果タイプ	なし
前提条件アクティビティ	承認者の選択

購買申請転送の記録（ノード6）

現在、このアクティビティでは何も処理されません。このアクティビティは、プロセス内で記録関数を統合できる場所を表わしています。このワークフローを統合する購買 / 購買申請アプリケーションがある場合は、このアクティビティをカスタマイズし、PL/SQL ストアド・プロシージャを実行して購買 / 購買申請アプリケーション表を更新し、購買申請が次の承認者に転送されたことを示すようにできます。

関数	WF_REQDEMO.Forward_Req
結果タイプ	なし
前提条件アクティビティ	承認者の選択

And（ノード7）

この「標準」関数アクティビティでは、フロー内の並行する2つ以上の分岐のすべてのアクティビティが完了した場合にのみ、分岐がマージされます。

関数	WF_STANDARD.ANDJOIN
結果タイプ	なし
前提条件アクティビティ	このアクティビティに進むアクティビティが、少なくとも2つは必要です。

承認者に通知（ノード 8）

このアクティビティは、購買申請を承認または否認する処理を実行する必要があることを、承認者に通知するサブプロセスです。このサブプロセスを表示するには、ナビゲータ・ツリーの「プロセス」のブランチの下にある「承認者に通知」をダブルクリックします。このサブプロセスはその承認者に通知を送信し、承認者が指定された時間内に応答しない場合は、その承認者に処理を実行させるための別の督促通知を送信します。10-21 ページの「[承認者に通知](#)」サブプロセスの概要を参照してください。

結果タイプ 承認

前提条件アクティビティ 承認者の選択

購買申請を否認（ノード 9）

現在、このアクティビティでは何も処理されません。このアクティビティは、プロセス内で関数を統合できる場所を表わしています。このワークフローを統合する購買 / 購買申請アプリケーションがある場合は、このアクティビティをカスタマイズし、PL/SQL ストアド・プロシージャを実行して購買 / 購買申請アプリケーション表を更新し、購買申請が否認されたことを示すようにできます。

関数 WF_REQDEMO.Reject_Req

結果タイプ なし

前提条件アクティビティ 承認者の選択、承認者に通知

申請者に否認を通知（ノード 10）

このアクティビティでは、購買申請が否認されたことが、購買申請作成者に通知されます。メッセージには、購買申請番号、購買申請摘要、購買申請金額、購買申請を否認したマネージャの氏名、およびそのマネージャからの注釈などを表示する「送信」属性が含まれます。メッセージには、通知ヘッダーに表示される、通知を送信したロールの「送信」属性も含まれます。

このアクティビティ・ノードのプロパティ画面を表示すると、「申請者のユーザー名」という項目タイプ属性に名前が保存されている実行者に、このアクティビティが割り当てられていることがわかります。

メッセージ 購買申請が否認された

結果タイプ なし

前提条件アクティビティ 承認者に通知

承認権限の検証（ノード 12）

この関数アクティビティでは、現行の承認者にその購買申請の承認権限があるかどうかを検証されます。このプロセスでは、購買申請金額が承認者の承認限度額と比較され、Yes の場合は値 Y が、No の場合は値 N が戻されます。ビジネス・ルールで承認者による承認額が制限されない場合は、このアクティビティを削除してプロセスをカスタマイズできます。

関数 WF_REQDEMO.VerifyAuthority

結果タイプ はい / いいえ

前提条件アクティビティ 承認者の選択、承認者に通知

購買申請を承認（ノード 13）

現在、このアクティビティでは何も処理されません。このアクティビティは、プロセス内で関数を統合できる場所を表わしています。このワークフローを統合する購買 / 購買申請アプリケーションがある場合は、このアクティビティをカスタマイズし、PL/SQL ストアド・プロセスを実行して購買 / 購買申請アプリケーション表を更新し、購買申請が承認されたことを示すようにできます。

関数 WF_REQDEMO.Approve_Req

結果タイプ なし

前提条件アクティビティ 承認者の選択、承認者に通知、承認権限の検証

申請者に承認を通知（ノード 14）

このアクティビティでは、購買申請が承認されたことが、購買申請作成者に通知されます。メッセージには、購買申請番号、購買申請摘要、購買申請金額、承認者名およびその承認者からの注釈を表示する「送信」属性が含まれます。メッセージには、通知ヘッダーに表示される、通知を送信したロールの「送信」属性も含まれます。

このアクティビティ・ノードのプロパティ画面を表示すると、「申請者のユーザー名」という項目タイプ属性に名前が保存されている実行者に、このアクティビティが割り当てられていることがわかります。

メッセージ 購買承認申請済

結果タイプ なし

前提条件アクティビティ 承認者の選択、承認者に通知、承認権限の検証

終了（ノード 4、11 および 15）

この関数アクティビティでは、プロセスの終了がマークされます。このアクティビティ自体には結果タイプはありませんが、プロセス内のこのアクティビティの各ノードには、プロセスの結果を割り当てる必要があります。プロセスの結果は、アクティビティ・ノードのプロパティ画面で割り当てます。購買申請プロセス・アクティビティには「承認」結果タイプがあるため、各「終了」アクティビティ・ノードには、「承認」選択肢タイプの選択肢コードと一致するプロセスの結果が必要です。

関数 WF_STANDARD.NOOP

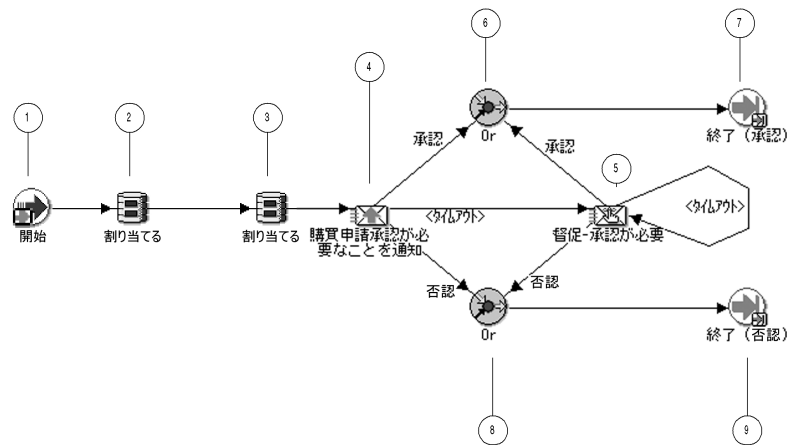
結果タイプ なし

前提条件アクティビティ 開始

「承認者に通知」サブプロセスの概要

「承認者に通知」サブプロセスのプロパティを表示するには、そのプロセス・アクティビティをナビゲータ・ツリーで選択し、「編集」メニューから「プロパティ」を選択します。「承認者に通知」サブプロセスには「承認」結果タイプがあります。この結果タイプは、サブプロセスの完了時に、結果が（「承認」選択肢タイプの選択肢コードに基づいて）「承認」または「否認」になることを示します。これは実行可能なプロセスではありません。つまり、最上位プロセスとしては実行できませんが、上位の別プロセスからコールされたときのみサブプロセスとして実行されます。

「承認者に通知」サブプロセスの「プロセス」ウィンドウを表示すると、このサブプロセスが6個の一意のアクティビティで構成されていることがわかります。そのうちのいくつかは再利用され、ワークフロー・ダイアグラムに表示される9個のアクティビティ・ノードを構成しています。プロセスのアクティビティを詳しく説明するために、次のように各ノードに番号を付けて参照しやすくしてあります。番号自体は、プロセス・ダイアグラムには含まれません。



このサブプロセスは、ノード1の「開始」アクティビティから開始します。ノード2では、「備考」項目属性に前の承認者が格納したコメントがあれば、「督促 - 承認が必要」通知で使用するためにそのコメントが「督促用の備考」項目属性に割り当てられます。「備考」項目属性はその後消去されるので、現在の承認者がコメントを格納することができるようになります。

ノード4で、プロセスは指定された期間内に購買申請を承認するように、現在の承認者に通知します。承認者が購買申請を承認すると、サブプロセスはノード7で終了し、最上位レベルの購買申請プロセスに結果「承認」を戻します。承認者が購買申請を否認すると、サブプロセスはノード9で終了し、最上位の購買申請プロセスに結果「否認」を戻します。

承認者が期日までに応答しなければ、サブプロセスはノード5へのタイムアウト・トランジションに進み、承認者に購買申請の承認を求める督促通知を送信します。また、ノード5にもタイムアウト値が割り当てられており、承認者がその時刻までに督促通知に応答しなければ、サブプロセスは次のタイムアウト・トランジションに進み、ループをたどってノード5に戻り、承認者に別の督促を送信します。このループは、承認者が購買申請を承認または否認するまで繰り返され、その時点でサブプロセスはノード7または9で終了します。

「承認者に通知」サブプロセスのアクティビティ

ここでは、「承認者に通知」サブプロセスの各アクティビティについて、アクティビティの表示名別に説明します。

開始（ノード1）

これは、単にサブプロセスの開始をマークする「標準」関数アクティビティです。

関数	WF_STANDARD.NOOP
結果タイプ	なし
前提条件アクティビティ	なし

割当（ノード2）

これは、値を項目属性に割り当てる「標準」関数アクティビティです。この「割当」アクティビティ・ノードで、「備考」項目属性の値が「督促用の備考」項目属性に割り当てられます。

関数	WF_STANDARD.ASSIGN
結果タイプ	なし
前提条件アクティビティ	なし

割当（ノード3）

これは、値を項目属性に割り当てる「標準」関数アクティビティです。この「割当」アクティビティ・ノードで、前の値を消去するために「備考」項目属性に空白が割り当てられます。

関数	WF_STANDARD.ASSIGN
結果タイプ	なし
前提条件アクティビティ	なし

購買申請承認が必要なことを通知（ノード4）

このアクティビティでは、購買申請を承認または否認する必要があることが承認者に通知されます。このアクティビティは5分以内に完了しないとタイムアウトになります。

メッセージには、通知送信時に購買申請番号、購買申請摘要、購買申請金額、前の承認者名および購買申請作成者名を表示する「送信」属性が含まれます。メッセージ内のメッセージ属性表にある属性を表示し、メッセージに通知履歴表を含めるために、特殊な WF_NOTIFICATION() メッセージ関数がコールされます。メッセージには、通知ヘッダーに表示される、通知を送信したロールの「送信」属性も含まれます。

メッセージには、特殊な「RESULT」属性と「応答」属性が含まれます。「RESULT」属性には「処理」という表示名があり、「承認」と呼ばれる選択肢タイプから、「APPROVE」または「REJECT」の値で応答を戻すように、承認者に要求します。承認者の選択した値が、ワークフロー・エンジンが次に進むアクティビティを決める結果となります。

「応答」属性は「備考」と呼ばれ、通知の応答にオプションのコメントを入れるよう承認者に求めます。

このアクティビティ・ノードのプロパティ画面を表示すると、「転送先ユーザー名」という項目タイプ属性に名前が保存されている実行者に、このアクティビティが割り当てられていることがわかります。

メッセージ	購買申請に承認が必要
結果タイプ	承認
前提条件アクティビティ	承認者の選択

督促 - 承認が必要（ノード5）

このアクティビティは、「購買申請承認が必要なことを通知」アクティビティが完了前にタイムアウトになった場合にのみ発生します。このアクティビティでは、購買申請を承認または否認する必要があることを示す督促通知が承認者に送られます。

メッセージには、通知送信時に購買申請番号、購買申請摘要、購買申請金額、前の承認者名および購買申請作成者名を表示する「送信」属性が含まれます。メッセージに通知履歴表を含めるために、特殊な WF_NOTIFICATION() メッセージ関数がコールされます。メッセージには、通知ヘッダーに表示される、通知を送信したロールの「送信」属性も含まれます。

メッセージには、特殊な「RESULT」属性と「応答」属性が含まれます。「RESULT」属性には「処理」という表示名があり、「承認」と呼ばれる選択肢タイプから、「APPROVE」または「REJECT」の値で応答を戻すように、承認者に要求します。承認者の選択した値が、ワークフロー・エンジンが次に進むアクティビティを決める結果となります。

「応答」属性は「備考」と呼ばれ、通知の応答にオプションのコメントを入れるよう承認者に求めます。

このアクティビティ・ノードのプロパティ画面を表示すると、「転送先ユーザー名」という項目タイプ属性に名前が保存されている実行者に、このアクティビティが割り当てられていることがわかります。

ることがわかります。このアクティビティ・ノードのプロパティ画面を表示すると、「転送先ユーザー名」という項目タイプ属性に名前が保存されている実行者に、このアクティビティが割り当てられていることがわかります。

メッセージ	購買承認申請の督促
結果タイプ	承認
前提条件アクティビティ	承認者の選択、購買申請承認が必要なことを通知

Or（ノード6および8）

この「標準」関数アクティビティでは、フロー内の複数の並列分岐の1つでアクティビティが完了すると、それらの分岐がマージされます。

関数	WF_STANDARD.ORJOIN
結果タイプ	なし
前提条件アクティビティ	なし

終了（ノード7および9）

この関数アクティビティでは、サブプロセスの終了がマークされます。このアクティビティ自体には結果タイプはありませんが、サブプロセス内のこのアクティビティの各ノードには、プロセスの結果を割り当てる必要があります。プロセスの結果は、アクティビティ・ノードのプロパティ画面で割り当てます。「承認者に通知」プロセス・アクティビティには「承認」結果タイプがあるため、各「終了」アクティビティ・ノードには、「承認」選択肢タイプの選択肢コードと一致するプロセスの結果が必要です。

関数	WF_STANDARD.NOOP
結果タイプ	なし
前提条件アクティビティ	開始

StartProcess 関数の例

wfrund.sql と「購買承認申請」Web ページでは、どちらも PL/SQL ストアド・プロシージャ WF_REQDEMO.StartProcess がコールされ、購買申請プロセスが実行されます。

StartProcess を詳しく説明するために、このプロシージャをいくつかのセクションに分け、参照しやすいように各セクションに「1⇒」というような番号を付けてあります。この番号と矢印自体はプロシージャの一部ではありません。

```

1⇒ procedure StartProcess (RequisitionNumber in varchar2,
                           RequisitionDesc in varchar2,
                           RequisitionAmount in number,
                           RequestorUsername in varchar2,
                           ProcessOwner in varchar2,
                           Workflowprocess in varchar2 default null,
                           item_type in varchar2 default null) is
2⇒   ItemType varchar2(30) := nvl(item_type, 'WFDEMO');
   ItemKey varchar2(30) := RequisitionNumber;
   ItemUserKey varchar2(80) := RequisitionDesc;
3⇒   begin
       wf_engine.CreateProcess (itemtype => ItemType,
                               itemkey => ItemKey,
                               process => WorkflowProcess );
4⇒   wf_engine.SetItemUserKey (itemtype => itemtype,
                               itemkey => itemkey,
                               userkey => ItemUserKey);
5⇒   wf_engine.SetItemAttrText (itemtype => itemtype,
                               itemkey => itemkey,
                               aname => 'REQUISITION_NUMBER',
                               avalue => RequisitionNumber);
6⇒   wf_engine.SetItemAttrText (itemtype => itemtype,
                               itemkey => itemkey,
                               aname => 'REQUISITION_DESCRIPTION',
                               avalue => ItemUserKey);
7⇒   wf_engine.SetItemAttrNumber (itemtype => itemtype,
                                  itemkey => itemkey,
                                  aname => 'REQUISITION_AMOUNT',
                                  avalue => RequisitionAmount);
8⇒   wf_engine.SetItemAttrText (itemtype => itemtype,
                                  itemkey => itemkey,
                                  aname => 'REQUESTOR_USERNAME',
                                  avalue => RequestorUsername);
9⇒   wf_engine.SetItemAttrText (itemtype => itemtype,
                                  itemkey => itemkey,
                                  aname => 'FORWARD_TO_USERNAME',
                                  avalue => RequestorUsername);
10⇒  wf_engine.SetItemAttrText (itemtype => itemtype,
                                itemkey => itemkey,

```

```

                                aname => 'REQUISITION_PROCESS_OWNER',
                                avalue => ProcessOwner);
11=> wf_engine.SetItemAttrText (itemtype => itemtype,
                                itemkey => itemkey,
                                aname => 'MONITOR_URL',
                                avalue => wf_monitor.GetDiagramUrl
                                    (WF_CORE.Translate('WF_WEB_AGENT'),
                                     itemtype,itemkey,'NO'));
12=> wf_engine.SetItemAttrText (itemtype => itemtype,
                                itemkey => itemkey,
                                aname => 'REM_DOCUMENT',
                                avalue => 'PLSQL:wf_reqdemo.
                                    reminder_req_document/'
                                    ||Itemtype||':'||Itemkey);
13=> wf_engine.SetItemOwner (itemtype => itemtype,
                                itemkey => itemkey,
                                owner => ProcessOwner);
14=> wf_engine.StartProcess (itemtype => itemtype,
                                itemkey => itemkey );
15=> exception
    when others then
        wf_core.context ('WF_REQDEMO', 'StartProcess',RequisitionNumber,
                        RequisitionAmount,RequestorUsername,ProcessOwner,
                        Workflowprocess);

        raise;
16=> end StartProcess;

```

1⇒ このセクションはプロシージャの仕様部で、**StartProcess** に渡す必要のあるパラメータのリストが入っています。wfrund.sql スクリプトに渡した値または「購買申請デモンストレーション」Web ページに入力したフィールド値 (WF_REQDEMO.Create_Req) と同じパラメータ値が使用されます。

2⇒ プロシージャ本体の宣言部は、このセクションから始まります。**StartProcess** は、ワークフロー・エンジンの複数の PL/SQL API コールで構成されています。『Oracle Workflow API リファレンス』の「Workflow Engine API」を参照してください。

これらの API にはいずれも項目タイプと項目キーの入力が必要であるため、Itemtype と Itemkey はローカル引数として定義されています。引数 Itemtype は「WFDEMO」として定義されています。「WFDEMO」は「購買申請」項目タイプの内部名です。引数 Itemkey は、StartProcess プロシージャに渡される RequisitionNumber パラメータの値です。

注意： プロセス・インスタンスの項目キーには、シングルバイト・キャラクターのみ使用できます。マルチバイトの値を含めることはできません。

3⇒ プロシージャ本体の実行可能部分は、このセクションから始まります。このセクションは、CreateProcess Workflow Engine API をコールします。この API では、購買申請プロ

セスの新しいランタイム・インスタンスが作成されます。このインスタンスの内部名は WFDEMO で、新しいインスタンスは指定された項目タイプおよび項目キーで識別されます。『Oracle Workflow API リファレンス』の「CreateProcess」を参照してください。

注意： wfrund.sql スクリプトに `<process_int_name>` の値を渡さなければ、「購買申請」項目タイプのセレクタ関数によって、どのプロセスを実行するかが判別されます。

4⇒ このセクションでは、SetItemUserKey Workflow Engine API がコールされ、購買申請プロセスの新規のランタイム・インスタンスがエンドユーザー・キーでマークされます。エンドユーザー・キーを使用すると、ユーザーは表示されたプロセス・インスタンスを問い合わせで識別できます。『Oracle Workflow API リファレンス』の「SetItemUserKey」を参照してください。

5、6、7、8、9、10、11、12 ⇒ これらのセクションでは、SetItemAttributeText または SetItemAttributeNumber Workflow Engine API がコールされ、このプロセスに対して定義された項目タイプ属性の値が設定されます。属性は、それぞれ REQUISITION_NUMBER、REQUISITION_DESCRIPTION、REQUISITION_AMOUNT、REQUESTOR_USERNAME、FORWARD_TO_USERNAME、REQUISITION_PROCESS_OWNER、MONITOR_URL および REM_DOCUMENT です。『Oracle Workflow API リファレンス』の「SetItemAttribute」を参照してください。

13⇒ このセクションでは、SetItemOwner Workflow Engine API がコールされ、購買申請プロセスの新規のランタイム・インスタンスがプロセス所有者のユーザー名でマークされます。ユーザーは、プロセス所有者を指定してプロセス・インスタンスを問い合わせできます。『Oracle Workflow API リファレンス』の「SetItemOwner」を参照してください。

14⇒ このセクションでは、例外が発生した場合に WF_CORE.CONTEXT() をコールしてエラー・スタックにコンテキスト情報を含め、エラーの原因箇所を見つけやすいようにします。『Oracle Workflow API リファレンス』の「CONTEXT」を参照してください。

15⇒ このセクションでは、Oracle Workflow Engine の StartProcess API がコールされ、指定した項目タイプと項目キーの購買申請プロセスが開始されます。『Oracle Workflow API リファレンス』の「StartProcess」を参照してください。

16⇒ このセクションで、プロセスの終了がマークされます。

関数アクティビティの例

通常、関数アクティビティには、「アクティビティ」プロパティ画面で次の情報を指定する必要があります。

- アクティビティの内部名
- アクティビティの表示名
- アクティビティの結果タイプで、「なし」または事前定義済の選択肢タイプ名
- アクティビティがコールする PL/SQL ストアド・プロシージャの名前

また、関数アクティビティによってコールされる PL/SQL ストアド・プロシージャは、特定の API に準拠している必要があります。6-3 ページの「[関数アクティビティがコールする PL/SQL プロシージャの標準 API](#)」を参照してください。

サーバーの Oracle Workflow ディレクトリ構造の demo サブディレクトリ内で、購買申請プロセスに使用される WF_REQDEMO ストアド・プロシージャ・パッケージを作成するスクリプトを表示できます。

例：承認者の選択

「承認者の選択」関数アクティビティでは、PL/SQL ストアド・プロシージャ WF_REQDEMO.SelectApprover がコールされ、デモンストレーション・データ・モデルの従業員承認者階層に基づいて、次の承認者が判別されます。

結果タイプ

このアクティビティでは、承認者が見つかった場合は応答「T」、見つからなかった場合は応答「F」が戻されます。応答の候補は、標準項目タイプに関連付けられている「ブール」という名前の選択肢タイプで定義されます。

PL/SQL ストアド・プロシージャ

ここでは、この関数アクティビティでコールされる PL/SQL ストアド・プロシージャについて詳しく説明します。このプロシージャの各セクションには、参照しやすいように「1⇒」のような番号が付いています。

```
procedure SelectApprover (itemtype in varchar2,
                          itemkey in varchar2,
                          actid in number,
                          funcmode in varchar2,
                          resultout out varchar2) is
1⇒  l_forward_from_username varchar2(320);
   l_forward_to_username varchar2(320);
2⇒  begin
   if (funcmode = 'RUN') then
       l_forward_to_username := wf_engine.GetItemAttrText (
                               itemtype => itemtype,
                               itemkey => itemkey,
                               aname => 'FORWARD_TO_USERNAME');
3⇒  if (l_forward_to_username is null) then
       l_forward_to_username := wf_engine.GetItemAttrText (
                               itemtype => itemtype,
                               itemkey => itemkey,
                               aname => 'REQUESTOR_USERNAME');
   end if;
4⇒  l_forward_from_username := l_forward_to_username;
5⇒  wf_engine.SetItemAttrText (itemtype => itemtype;
                              itemkey => itemkey,
                              aname => 'FORWARD_FROM_USERNAME';
                              avalue => l_forward_from_username);
6⇒  l_forward_to_username := wf_reqdemo.GetManager(
                              l_forward_from_username);
7⇒  wf_engine.SetItemAttrText (itemtype => itemtype;
                              itemkey => itemkey,
                              aname => 'FORWARD_TO_USERNAME';
                              avalue => l_forward_to_username);
8⇒  if (l_forward_to_username is null) then
       resultout := 'COMPLETE:F';
   else
       resultout := 'COMPLETE:T';
   end if;
9⇒  end if;
10⇒ if (funcmode = 'CANCEL') then
       resultout := 'COMPLETE';
       return;
   end if;
11⇒ if (funcmode = 'TIMEOUT') then
       resultout := 'COMPLETE';
```

```

        return;
    end if;
12⇒ exception
        when others then
            wf_core.context('WF_REQDEMO', 'SelectorApprover', itemtype,
                itemkey, actid, funcmode);
            raise;
13⇒ end SelectApprover;

```

1⇒ このセクションで、ローカル引数 `l_forward_from_username` および `l_forward_to_username` が宣言されます。

2⇒ `funcmode` の値が `RUN` であれば、Workflow Engine API `GetItemAttrText` をコールして決定された `FORWARD_TO_USERNAME` 項目タイプ属性の値に `l_forward_to_username` が割り当てられ、この購買申請が承認のために最後に転送された承認者の名前が取り出されます。『Oracle Workflow API リファレンス』の「`GetItemAttribute`」を参照してください。

3⇒ `l_forward_to_username` の値が `NULL` であれば、購買申請は承認のために転送されたことがないことを意味します。この場合は、Workflow Engine API `GetItemAttrText` をコールして判別された `REQUESTOR_USERNAME` 項目タイプ属性の値が割り当てられます。

4⇒ `l_forward_from_username` に `l_forward_to_username` の値が割り当てられません。

5⇒ このセクションでは、Workflow Engine API `SetItemAttrText` がコールされ、`l_forward_from_username` の値が `FORWARD_FROM_USERNAME` 項目タイプ属性に設定されます。

6⇒ このセクションでは、関数 `GetManager` がコールされ、`WF_REQDEMO_EMP_HIERARCHY` 表から `l_forward_from_username` に格納されていた前の承認者のマネージャが戻され、そのマネージャの名前が `l_forward_to_username` に割り当てられます。

7⇒ このセクションでは、Workflow Engine API `SetItemAttrText` がコールされ、`l_forward_to_username` の値が `FORWARD_TO_USERNAME` 項目タイプ属性に設定されません。

8⇒ `l_forward_to_username` が `NULL` であれば、階層には前の承認者より上位のマネージャがないことを意味し、`resultout` が `COMPLETE:F` に設定されます。それ以外の場合は、`resultout` が `COMPLETE:T` に設定されます。

9⇒ このセクションでは、`funcmode = 'RUN'` かどうかのチェックが終了します。

10⇒ `funcmode` の値が `CANCEL` であれば、`resultout` が `COMPLETE` に設定されます。

11⇒ `funcmode` の値が `TIMEOUT` であれば、`resultout` が `COMPLETE` に設定されます。

12⇒ このセクションでは、エラー・スタックにコンテキスト情報を含めてエラーの原因箇所を見つけやすくするために、例外が発生した場合に WF_CORE.CONTEXT をコールします。『Oracle Workflow API リファレンス』の「CONTEXT」を参照してください。

13⇒ このセクションで、プロシージャの終了がマークされます。

例：承認権限の検証

「承認権限の検証」関数アクティビティでは、PL/SQL ストアド・プロシージャ WF_REQDEMO.VerifyAuthority がコールされ、購買申請金額が承認者の支払限度額内かどうかを検証されます。これは、ストアド・プロシージャで実現したビジネス・ルールに基づいて結果を戻す、自動化された関数アクティビティの例です。

結果タイプ

このアクティビティでは、プロシージャが完了して承認者に購買申請の承認権限があるかどうかを示すときに、結果として「Yes」または「No」が戻されます。これらの結果値は、標準項目タイプに関連付けられている「Yes/No」という名前の選択肢タイプで定義されます。

PL/SQL ストアド・プロシージャ

ここでは、この関数アクティビティでコールされる PL/SQL ストアド・プロシージャについて詳しく説明します。このプロシージャの各セクションには、参照しやすいように「1⇒」のような番号が付いています。また、プロシージャ内で使用されるローカル引数を示す場合は「l_」という表記を使用しています。

```

procedure VerifyAuthority (itemtype in varchar2,
                           itemkey in varchar2,
                           actid in number,
                           funcmode in varchar2,
                           resultout out varchar2) is
1⇒  l_forward_to_username varchar2(320);
    l_requisition_amount number;
    l_spending_limit number;
2⇒  begin
    if (funcmode = 'RUN') then
        l_requisition_amount := wf_engine.GetItemAttrNumber (
                               itemtype => itemtype,
                               itemkey => itemkey,
                               aname => 'REQUISITION_AMOUNT');
3⇒  l_forward_to_username := wf_engine.GetItemAttrText (
                               itemtype => itemtype,
                               itemkey => itemkey,
                               aname => 'FORWARD_TO_USERNAME');
4⇒  if (wf_reqdemo.checkSpendingLimit(l_forward_to_username,
                                       l_requisition_amount)) then
        resultout := 'COMPLETE:Y';

```

```

else
    resultout := 'COMPLETE:N';
end if;
end if;
5⇒ if (funcmode = 'CANCEL') then
    resultout := 'COMPLETE:Y';
    return;
end if;
6⇒ if (funcmode = 'TIMEOUT') then
    resultout := 'COMPLETE:Y';
    return;
end if;
7⇒ exception
    when others then
        wf_core.context('WF_REQDEMO', 'VerifyAuthority', itemtype,
            itemkey, actid, funcmode);
        raise;
8⇒ end VerifyAuthority;

```

1⇒ このセクションで、ローカル引数 `l_forward_to_username`、`l_requisition_amount` および `l_spending_limit` が宣言されます。

2⇒ `funcmode` の値が `RUN` であれば、Workflow Engine API `GetItemAttrNumber` をコールして判別された `REQUISITION_AMOUNT` 項目タイプ属性の値が `l_requisition_amount` に設定されます。『Oracle Workflow API リファレンス』の「`GetItemAttribute`」を参照してください。

3⇒ このセクションでは、Workflow Engine API `GetItemAttrText` をコールして判別された `FORWARD_TO_USERNAME` 項目タイプ属性の値が、`l_forward_to_username` に設定されます。

4⇒ このセクションでは、現行の承認者の関数 `CheckSpendingLimit` がコールされ、購買申請金額が承認者の支払限度額以内かどうか判別されます。購買申請金額が `l_spending_limit` の値以下であれば、その承認者に承認権限があることを意味し、`resultout` に `COMPLETE:Y` が設定されます。それ以外の場合は、`resultout` に `COMPLETE:N` が設定されます。

5⇒ `funcmode` の値が `CANCEL` であれば、`resultout` が `COMPLETE:` に設定されます。

6⇒ `funcmode` の値が `TIMEOUT` であれば、`resultout` が `COMPLETE:` に設定されます。

7⇒ このセクションでは、エラー・スタックにコンテキスト情報を含めてエラーの原因箇所を見つけやすくするために、例外が発生した場合に `WF_CORE.CONTEXT` をコールします。『Oracle Workflow API リファレンス』の「`CONTEXT`」を参照してください。

8⇒ このセクションで、プロシージャの終了がマークされます。

通知アクティビティの例

購買申請プロセスには、ユーザーに情報メッセージを送信する複数の通知アクティビティが含まれています。ただし、「承認者に通知」サブプロセスには、ユーザーからの応答を要求する通知アクティビティも含まれています。

通知アクティビティの場合は、「アクティビティ」プロパティ画面で次の情報を定義する必要があります。

- アクティビティの内部名
- アクティビティの表示名
- アクティビティの結果タイプで、「なし」または事前定義済の選択肢タイプ名
- 通知が送信される事前定義済のメッセージの名前

例：購買申請承認が必要なことを通知

「購買申請承認が必要なことを通知」アクティビティでは、「購買申請に承認が必要」というメッセージが承認マネージャに送信されます。このメッセージは、マネージャに対して購買申請を承認または否認するように要求するもので、メッセージ本文には購買申請の詳細が表示されます。

結果タイプ

マネージャの応答によって、プロセスが次に進むアクティビティが決まります。考えられる応答として「承認」または「否認」が、「承認」選択肢タイプで定義されています。この2つの値は、メッセージの特殊な「結果」属性（表示名、「処理」）で定義されます。「アクティビティ」プロパティ画面の「結果タイプ」フィールドで定義されているように、これらの値は通知アクティビティの結果の候補でもあります。

メッセージ

通知の内容は、「購買申請に承認が必要」というメッセージ内で定義されます。

件名	Requisition &REQUISITION_NUMBER, &REQUISITION_DESCRIPTION for &REQUISITION_AMOUNT requires approval
本文	WF_NOTIFICATION(ATTRS,REQUISITION_NUMBER, REQUISITION_AMOUNT,REQUISITION_DESCRIPTION, FORWARD_FROM_USERNAME,REQUESTOR_USERNAME) WF_NOTIFICATION(HISTORY)

メッセージの件名または本文のうち、アンパサンド「&」で始まるメッセージ属性は、通知の送信時にランタイム値に置き換えられるトークンです。ただし、トークンの置換が正しく行われるためには、件名またはメッセージ本文で参照されるすべてのメッセージ属性が、「送信」ソースでメッセージ属性として定義されている必要があります。

この例では、メッセージ本文は `WF_NOTIFICATION()` という特殊なメッセージ関数への 2 つのコールから構成されています。最初のコールによって、メッセージ内のメッセージ属性表にある購買申請の購買申請番号、購買申請摘要、購買申請金額、前の承認者名および購買申請作成者名が表示されます。2 番目のコールによって、メッセージ内の通知履歴表が表示されます。また、メッセージには、通知ヘッダーに表示される、通知を送信したロールの `#FROM_ROLE` という「送信」属性も含まれます。3-30 ページの「[#FROM_ROLE 属性](#)」および 3-36 ページの「[WF_NOTIFICATION\(\) メッセージ関数](#)」を参照してください。

このメッセージには、「処理」という特殊な結果メッセージ属性と「備考」という「応答」メッセージ属性も含まれます。

結果のメッセージ属性は、メッセージのプロパティ画面の「結果」タブで定義されています。結果の属性によって、承認者は、指定した選択肢タイプによって与えられる値リストから値を選択して、応答するように要求されます。続いて、この応答が「購買申請承認が必要なことを通知」アクティビティの結果となります。この場合、使用できる応答値は、承認選択肢タイプで定義されているように、「承認」または「否認」です。この結果によって、プロセスが次にどのアクティビティに進むかが決まります。

「応答」メッセージ属性「ノート」のタイプは、「テキスト」です。この属性は、通知への応答時にオプションのコメントを入力するように、承認者に求めます。

注意： メッセージの内容を表示するには、ナビゲータ・ツリーでメッセージをダブルクリックするか、メッセージを選択して「編集」メニューから「プロパティ」を選択します。

プロセス・ノードのプロパティ

「承認者に通知」サブプロセス・ダイアグラムに「購買申請承認が必要なことを通知」アクティビティ・ノードのプロパティを表示すると、「プロセス開始」アクティビティでも「終了」アクティビティでもないため、このノードが「標準」に設定されていることがわかります。

また、「実行者」が「転送先ユーザー名」項目タイプ属性に設定されていることもわかります。これは、名前が項目タイプ属性「転送先ユーザー名」に格納されているユーザーに通知が送信されることを示します。「転送先ユーザー名」の値は、購買申請プロセスの「承認者の選択」アクティビティによって事前に決定されています。

ドキュメント・レビュー・プロセス

注意： 文書管理機能は今後使用する目的で確保されています。ドキュメント・レビュー・プロセスに関するこの説明は、単なる参考として記載されています。

ドキュメント・レビュー・プロセスは、通知とドキュメント管理システムを統合し、添付文書をレビューして承認するように、承認者に要求します。Oracle Workflow 管理ロール内のユーザーは、Oracle Workflow の「プロセスの開始」Web ページまたは「ワークフロー・デモンストレーション」Web ページからドキュメント・レビュー・プロセスを開始できます。プロセスを開始するには、「項目キー」、「ユーザー・キー」、「プロセス所有者」、「文書の送信」、「文書の所有者」および「文書のレビューア」の項目属性値を指定します。

注意： プロセス・インスタンスの項目キーには、シングルバイト・キャラクターのみ使用できます。マルチバイトの値を含めることはできません。

レビュー・プロセスのプロセス定義は、Oracle Workflow のスタンドアロン版では、Workflow Configuration Assistant によって自動的にインストールされます。Oracle Application に組み込まれている Workflow では、AutoUpgrade によって自動的にインストールされます。

このデモンストレーションで、ドキュメント・レビュー要求を送信すると、プロセスは、指定したレビュー担当に文書の承認をを求める通知を送ります。オプションで、レビュー担当が応答時に代替文書を提供することもできます。レビュー担当が文書を承認した場合、プロセスは「承認」という結果で終了します。レビュー担当が文書を否認した場合、申請者は文書の承認を求めて再送信できます。申請者が文書の再送信を選択すると、プロセスはループ・バックし、「文書のレビュー」通知を送ります。再送信しなければ、プロセスは「否認」という結果で終了します。

関連項目：

10-2 ページ [「サンプル・ワークフロー・プロセス」](#)

7-2 ページ [「ワークフロー定義のテスト」](#)

「文書管理」項目タイプ

ドキュメント・レビュー・プロセスは、「文書管理」という項目タイプに関連付けられています。この項目タイプは、ドキュメント管理システムの統合に関連するデモンストレーション・ワークフロー・プロセスをすべて識別します。「文書管理」に関連付けられているワークフロー・プロセスは、現在、「レビュー」1つのみです。

ドキュメント管理のプロパティ画面を見ると、維持タイプが「一時」に、維持日数が0になっています。これは、この項目タイプの作業項目に関連付けられているランタイム・データは、終了直後に削除の対象となることを意味します。Web ベース・インタフェースからドキュメント管理プロセスを開始したときに、開始するプロセスが指定されていたため、項目タイプはセクタ関数を持ちません。

「文書管理」項目タイプには、複数の属性も関連付けられています。これらの属性は、ドキュメント管理プロセスの開始時の情報を記録します。属性は、プロセス全体を通して通知アクティビティと関数アクティビティによって使用され保存されます。次の表は、「文書管理」項目タイプの属性を示しています。

表 10-3

表示名	説明	型	長さ / 書式 / 選択肢タイプ
送信ドキュメント	レビューのために送信されるドキュメント	ドキュメント	フレーム・ターゲット - 新規ウィンドウ
文書の所有者	レビューのために送信されるドキュメントの所有者	ロール	
文書のレビューア	ドキュメントのレビュー担当のロール	ロール	
コメント	レビュー担当が入力するコメント	テキスト	
応答ドキュメント	レビュー後に編集を経たドキュメント	ドキュメント	フレーム・ターゲット - 新規ウィンドウ

ドキュメント・レビュー・プロセスの概要

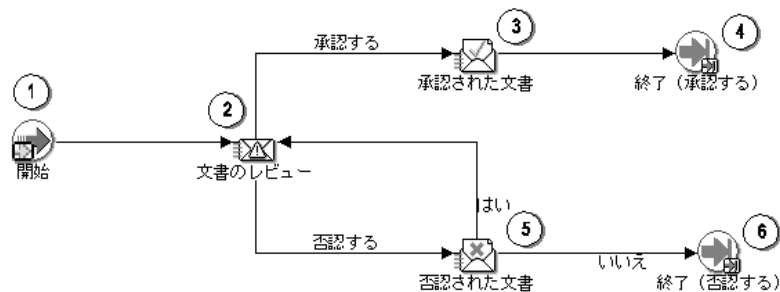
ドキュメント・レビュー・プロセスのプロパティを表示するには、ナビゲータ・ツリーでプロセスを選択し、「編集」メニューから「プロパティ」を選択します。ドキュメント・レビュー・プロセスには、プロセスの完了時に結果が「承認」になるか「否認」になるか（標準項目タイプに関連付けられた「承認」選択肢タイプの選択肢コード）を示す「承認」という結果タイプがあります。このプロセス・アクティビティも実行可能です。つまり、最上位レベルのプロセスとして開始できます。

プロセス・アクティビティの「詳細」プロパティ画面は、「ドキュメント・レビュー」に、WFERROR というエラー項目タイプと DEFAULT_ERROR というエラー・プロセスが関連付けられていることを示します。「ドキュメント・レビュー」にエラーが起きると、項目タイプ WFERROR の DEFAULT_ERROR プロセスが自動的に開始されます。現状では、DEFAULT_ERROR プロセスによって管理者にエラーが通知され、エラーのプロセスを再実行、中止または続行するオプションが提供されます。

ドキュメント・レビュー・プロセスのプロセス・ウィンドウを表示すると、プロセスが6つのアクティビティ・ノードから構成されていることがわかります。プロセスのアクティビティを詳しく説明するために、次のように各ノードに番号を付けて参照しやすくしてあります。番号自体は、プロセス・ダイアグラムには含まれません。

ワークフローは、ノード1の「開始」アクティビティから開始します。

ノード2の「文書のレビュー」では、プロセスは、ドキュメントのレビュー担当に通知を送り、「文書の送信」のレビューと承認を求めます。レビュー担当がドキュメントを承認すると、プロセスはノード3の「承認された文書」に移り、申請者に承認を通知します。この場合、プロセスは「承認する」という結果で終了します。レビュー担当がドキュメントを否認すると、プロセスはノード5の「否認された文書」に移り、申請者に否認を通知します。申請者は、ドキュメントが承認されるように再送信するか、結果を受け入れるかを選択できます。ドキュメントが再送信される場合、プロセスはノード2の「文書のレビュー」に戻ります。申請者が結果を受け入れた場合、プロセスは「否認する」という結果で終了します。



ドキュメント・レビュー・プロセスのアクティビティ

開始（ノード1）

これは、単にプロセスの開始をマークする「標準」関数アクティビティです。

- 関数： WF_STANDARD.NOOP
- 結果タイプ： なし
- 必須： Yes
- 前提条件アクティビティ： なし
- 関数によって取り出されるアクティビティ属性： なし
- 関数によって設定される項目属性： なし
- 関数によって取り出される項目属性： なし

文書のレビュー（ノード2）

これは、ドキュメントのレビュー担当に承認を求めるメッセージを送る通知アクティビティです。オプションで、応答時にドキュメントを編集して返すように求めることもできます。メッセージには、2つの「送信」属性が含まれます。1つは、ドキュメントの所有者を表示するもので、もう1つは、通知に添付情報として表示される文書属性です。添付アイコンを選択すると、ウィンドウが新たに開き、ドキュメントが表示されます。メッセージには、「コメント」と「応答ドキュメント」を要求する2つの「応答」属性も含まれます。

- メッセージ： ドキュメント送信
- 結果タイプ： 承認
- 必須： Yes
- 前提条件アクティビティ： なし
- 通知によって取り出されるアクティビティ属性： なし

承認された文書（ノード3）

これは、ドキュメント・レビューの申請者に、ドキュメントが承認されたことを伝えるメッセージを送る通知アクティビティです。メッセージには、「文書の所有者」、「文書のレビュー担当」、「文書の送信」へのリンク、レビュー担当からの「コメント」およびオプションの「応答ドキュメント」添付を表示する、5つの「送信」属性が含まれます。「応答ドキュメント」添付を選択すると、ウィンドウが新たに開かれ、ドキュメントが表示されます。

- メッセージ： ドキュメントの応答： 承認
- 結果タイプ： 承認
- 必須： Yes
- 前提条件アクティビティ： ドキュメントのレビュー
- 通知によって取り出されるアクティビティ属性： なし

終了（ノード4および6）

これは、単にプロセスの終了をマークする「標準」関数アクティビティです。

- 関数： WF_STANDARD.NOOP
- 結果タイプ： なし
- 必須： Yes
- 前提条件アクティビティ： なし
- 関数によって取り出されるアクティビティ属性： なし
- 関数によって設定される項目属性： なし
- 関数によって取り出される項目属性： なし

否認された文書（ノード 5）

これは、ドキュメント・レビューの申請者に、ドキュメントが否認されたことを伝えるメッセージを送る通知アクティビティです。メッセージには、「文書の所有者」、「文書のレビュー担当」、「文書の送信」へのリンクおよびオプションの「応答ドキュメント」添付を表示する、4つの「送信」属性が含まれます。添付アイコンを選択すると、ウィンドウが新たに開き、ドキュメントが表示されます。メッセージには、レビュー担当からのコメントを表示する1つの「応答」属性も含まれており、ドキュメント・レビュー要求が再送信される場合に、申請者からのコメントを要求します。

- メッセージ： ドキュメントの応答： 否認
- 結果タイプ： Yes/No
- 必須： Yes
- 前提条件アクティビティ： ドキュメントのレビュー
- 通知によって取り出されるアクティビティ属性： なし

エラー・チェック・プロセス

エラー・チェック・プロセスは、Oracle Workflow の項目アクティビティ・ステータス表でエラーのアクティビティを検索します。このプロセスの主な目的は、Oracle Alert の定期アラートと同じ機能を持つワークフロー・プロセスを設計する場合の、Oracle Workflow の使用方法を示すことです。エラー・チェック・プロセスは、指定した間隔でデータベース例外のチェックを行うように設定できます。

エラー・チェック・プロセスのプロセス定義は、Oracle Workflow のスタンドアロン版では、Workflow Configuration Assistant によって自動的にインストールされます。Oracle Application に組み込まれている Workflow では、AutoUpgrade によって自動的にインストールされます。

このデモンストレーションでエラー・チェック・プロセスを開始すると、プロセスはワークフロー項目アクティビティ・ステータス表をチェックし、エラーのあるアクティビティを検索する関数を実行します。エラーが見つかったら、指定されたアラートの宛先にエラーをリストした通知を送ります。1 回のみ実行するか、指定した間隔で実行するかの設定に応じて、エラー・チェック・プロセスは続行または終了します。特定の間隔で実行するように指定した場合、プロセスは必要な時間だけ待機し、ワークフロー項目アクティビティ・ステータス表を再チェックします。待機 / チェックのループが続き、プロセスは指定した終了日に終了します。

エラー・チェック・プロセスは、「プロセスの開始」Web ページまたは「ワークフロー・デモンストレーション」Web ページから開始できます。「項目キー」、「ユーザー・キー」、「プロセス所有者」、「警告の宛先」、プロセスの「開始日」と「終了日」およびエラーをチェックする「頻度」（曜日、日付、時刻、日数または 1 回のみ）を指定する必要があります。

注意： プロセス・インスタンスの項目キーには、シングルバイト・キャラクターのみ使用できます。マルチバイトの値を含めることはできません。

関連項目：

10-2 ページ [「サンプル・ワークフロー・プロセス」](#)

7-2 ページ [「ワークフロー定義のテスト」](#)

「定期アラート」項目タイプ

エラー・チェック・プロセスは、「定期アラート」という項目タイプに関連付けられています。この項目タイプは、Oracle Workflow 全体で、定期アラート機能の実装に関連するプロセスをすべて識別します。定期アラートに関連するワークフロー・プロセスは、現在、2つあります。エラー・チェックとユーザー定義の警告処理です。

定期アラートのプロパティ画面を見ると、維持タイプが「一時」に、維持日数が0になっています。これは、この項目タイプの項目インスタンスに関連付けられているランタイム・データが、終了直後に削除の対象となることを意味します。Web ベース・インタフェースから定期アラート・プロセスを開始した場合、開始するプロセスが指定されているため、セレクト関数の指定はありません。

「定期アラート」項目タイプには、複数の属性も関連付けられています。これらの属性は、定期アラート・プロセスの開始時の情報を記録します。属性は、プロセス全体を通して通知アクティビティと関数アクティビティによって使用され保存されます。次の表は、「定期アラート」項目タイプの属性を示しています。

表 10-4

表示名	説明	型	長さ / 書式 / 選択肢タイプ
警告の宛先	アラート例外が検出された場合に通知するロール	ロール	
開始日	アラート・チェックの開始日 (デフォルトは今日)	日付	DD-MON-YYYY HH24:MI:SS
終了日	アラート・チェックの終了日 (デフォルトは、2010年12月31日 12:12:12 PM)	日付	DD-MON-YYYY HH24:MI:SS
頻度	アラートをチェックする頻度	選択肢	待機モード
頻度： 毎月何日	頻度が「毎月何日」の場合、日付が必須	選択肢	毎月何日
頻度： 毎週何曜日	頻度が「毎週何曜日」の場合、曜日が必須	選択肢	毎週何曜日
頻度： 毎日何時	時間はオプション、ただし「待機」アクティビティで「頻度」の値に使用	日付	HH24:MI
頻度： 日	頻度が「相対時間」の場合、日数が必須	数値	
1回のみ	アラートを1回のみ実行	選択肢	はい / いいえ

エラー・チェック・プロセスの概要

エラー・チェック・プロセスのプロパティを表示するには、ナビゲータ・ツリーでプロセスを選択し、「編集」メニューから「プロパティ」を選択します。エラー・チェック・プロセスは実行可能です。つまり、最上位レベルのプロセスとして開始できます。

プロセス・アクティビティの「詳細」プロパティ画面は、「エラー・チェック」に、WFERROR というエラー項目タイプと DEFAULT_ERROR というエラー・プロセスが関連付けられていることを示します。「エラー・チェック」にエラーが起きると、項目タイプ WFERROR の DEFAULT_ERROR プロセスが自動的に開始されます。現状では、DEFAULT_ERROR プロセスによって管理者にエラーが通知され、エラーのプロセスを再試行、中止または続行するオプションが提供されます。

エラー・チェック・プロセスの「プロセス」ウィンドウを表示すると、このプロセスが6個の一意のアクティビティで構成されていることがわかります。そのうちのいくつかが再利用され、ワークフロー・ダイアグラムに表示される9個のアクティビティ・ノードを構成しています。プロセスのアクティビティを詳しく説明するために、次のように各ノードに番号を付けて参照しやすくしてあります。番号自体は、プロセス・ダイアグラムには含まれません。

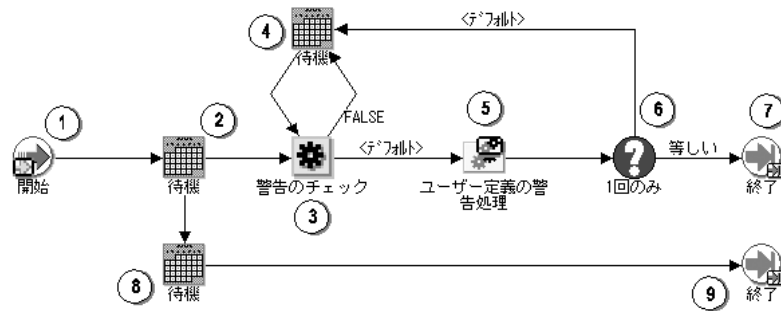
「プロセスの開始」Web ベース・インタフェースからプロセスを起動すると、エラー・チェック・ワークフローが開始します。「項目キー」、「ユーザー・キー」、「プロセス所有者」、「警告の宛先」、プロセスの「開始日」と「終了日」およびエラーをチェックする「頻度」（曜日、日付、時刻、日数または1回のみ）を指定する必要があります。

注意： プロセス・インスタンスの項目キーには、シングルバイト・キャラクターのみ使用できます。マルチバイトの値を含めることはできません。

ワークフローは、ノード1の「開始」アクティビティから開始します。

ノード2でプロセスは休止し、開始日まで待機します。待機時間が経過すると、プロセスはノード3で、ワークフロー項目アクティビティ・ステータス表でエラーを検索する関数アクティビティを実行します。関数アクティビティがエラーを検出できなければ、プロセスはノード4を実行します。ここで、プロセスの開始時に指定された頻度に基づいた期間だけ休止します。頻度をベースにした待機時間が経過すると、プロセスはステータス表でエラーを再検索します。エラーを検出した場合、プロセスはノード5で、アラートの宛先にエラーの通知を送るプロセス・アクティビティを実行します。その後、プロセスはノード6を実行し、エラー・チェック・プロセスを1回のみ実行するかどうかを評価します。プロセスの実行が1回のみの場合、プロセスはノード7で終了します。1回のみでない場合、プロセスはノード4の頻度ベースの「待機」アクティビティに戻ります。

ノード2～5の一連のアクティビティが実行される間に、プロセスは並行してノード8へも移ります。これは、指定した終了日に達するまで、作業項目のスキャンをループさせたままにする別の「待機」アクティビティです。指定の終了日になると、プロセスはノード9で終了します。



エラー・チェック・プロセスのアクティビティ

開始（ノード1）

これは、単にプロセスの開始をマークする「標準」関数アクティビティです。

- 関数： WF_STANDARD.NOOP
- 結果タイプ： なし
- 必須： Yes
- 前提条件アクティビティ： なし
- 関数によって取り出されるアクティビティ属性： なし
- 関数によって設定される項目属性： なし
- 関数によって取り出される項目属性： なし

待機（ノード2）

これは、指定した時間プロセスを休止させる「標準」関数アクティビティです。

プロセスで「待機」アクティビティを使用するには、「待機」アクティビティを終了させるために、待機時間が経過したかどうかを評価するバックグラウンド・エンジンを、1つ以上設定する必要があります。

- 関数： WF_STANDARD.WAIT
- 結果タイプ： なし
- 必須： Yes
- 前提条件アクティビティ： なし
- 関数によって取り出されるアクティビティ属性
 - 待機モード： 定数、絶対日
 - 絶対日： 項目属性、開始日
- 関数によって設定される項目属性： なし
- 関数によって取り出される項目属性： なし

警告のチェック（ノード3）

これは、ワークフロー項目アクティビティ・ステータス表の行をスキャンして、ERROR ステータスのアクティビティを検索する関数アクティビティです。

- 関数： WF_ALERT.CHECKALERT
- 結果タイプ： ブール
- 必須： Yes
- 前提条件アクティビティ： なし
- 関数によって取り出されるアクティビティ属性： なし
- 関数によって作成される項目属性： LAST_CHECKED
- 関数によって取り出される項目属性： LAST_CHECKED
- 関数によって設定される項目属性： LAST_CHECKED

待機（ノード4）

これは、指定した時間プロセスを休止させる「標準」関数アクティビティです。

プロセスで「待機」アクティビティを使用するには、「待機」アクティビティを終了させるために、待機時間が経過したかどうかを評価するバックグラウンド・エンジンを、1つ以上設定する必要があります。

- 関数： WF_STANDARD.WAIT
- 結果タイプ： なし
- 必須： Yes
- 前提条件アクティビティ： なし
- 関数によって取り出されるアクティビティ属性
 - 待機モード： 項目属性、頻度
 - 絶対日： 項目属性、開始日
 - 毎月何日： 項目属性、毎月何日
 - 毎週何曜日： 項目属性、毎週何曜日
 - 相対時間： 項目属性、頻度： 日
 - 毎日何時： 項目属性、頻度： 毎日何時
- 関数によって設定される項目属性： なし
- 関数によって取り出される項目属性： なし

ユーザー定義の警告処理（ノード5）

このアクティビティは、アラート例外が検出されるたびに一連のアクティビティを実行するサブプロセスです。このサブプロセスを参照するには、ナビゲータ・ツリーで「プロセス」のブランチの下にある「ユーザー定義の警告処理」をダブルクリックします。現状では、サブプロセスは、検出されたエラーに関する通知をアラートの宛先に送ります。

- 結果タイプ： なし
- 必須： Yes
- 前提条件アクティビティ： なし
- 関数によって取り出されるアクティビティ属性： なし

1 回のみ (ノード 6)

これは、値を別の値と比較する「標準」関数アクティビティです。

- 関数: WF_STANDARD.COMPARE
- 結果タイプ: 比較
- 必須: Yes
- 前提条件アクティビティ: なし
- 関数によって取り出されるアクティビティ属性
 - テスト値: 項目属性、1 回のみ
 - 参照値: 定数、Y
- 関数によって設定される項目属性: なし
- 関数によって取り出される項目属性: なし

終了 (ノード 7 および 9)

これは、単にプロセスの終了をマークする「標準」関数アクティビティです。

- 関数: WF_STANDARD.NOOP
- 結果タイプ: なし
- 必須: Yes
- 前提条件アクティビティ: なし
- 関数によって取り出されるアクティビティ属性: なし
- 関数によって設定される項目属性: なし
- 関数によって取り出される項目属性: なし

待機（ノード 8）

これは、指定した時間プロセスを休止させる「標準」関数アクティビティです。

プロセスで「待機」アクティビティを使用するには、「待機」アクティビティを終了させるために、待機時間が経過したかどうかを評価するバックグラウンド・エンジンを、1つ以上設定する必要があります。

- 関数： WF_STANDARD.WAIT
- 結果タイプ： なし
- 必須： Yes
- 前提条件アクティビティ： なし
- 関数によって取り出されるアクティビティ属性
 - 待機モード： 定数、絶対日
 - 絶対日： 項目属性、終了日
- 関数によって設定される項目属性： なし
- 関数によって取り出される項目属性： なし

ユーザー定義の警告処理プロセスの概要

ユーザー定義の警告処理プロセスのプロパティを表示するには、ナビゲータ・ツリーでプロセスを選択し、「編集」メニューから「プロパティ」を選択します。ユーザー定義の警告処理プロセスは、実行可能ではありません。つまり、最上位レベルのプロセスとして開始できず、上位レベルのプロセスからコールする必要があります。

プロセス・アクティビティの「詳細」プロパティ画面では、このプロセス・アクティビティに関連するエラー項目タイプおよびエラー・プロセスがないことがわかります。エラーが起きた場合、開始されるエラー・プロセスは、親プロセスであるエラー・チェックに関連付けられているエラー項目タイプとエラー・プロセスによって決定されます。

ユーザー定義の警告処理プロセスの「プロセス」ウィンドウを表示すると、プロセスが3つの異なるアクティビティから構成されていることがわかります。プロセスのアクティビティを詳しく説明するために、次のように各ノードに番号を付けて参照しやすくしてあります。番号自体は、プロセス・ダイアグラムには含まれません。

ユーザー定義の警告処理プロセスは、エラー・チェック・プロセスのサブプロセスとして開始されます。

ワークフローは、ノード1の「開始」アクティビティから開始します。ノード2で、プロセスは検出されたエラーをリストした通知をアラートの宛先へ送ります。この時点でプロセスは終了します。



ユーザー定義の警告処理プロセスのアクティビティ

開始（ノード1）

これは、単にプロセスの開始をマークする「標準」関数アクティビティです。

- 関数： WF_STANDARD.NOOP
- 結果タイプ： なし
- 必須： Yes
- 前提条件アクティビティ： なし
- 関数によって取り出されるアクティビティ属性： なし
- 関数によって設定される項目属性： なし
- 関数によって取り出される項目属性： なし

警告受信者への通知（ノード2）

これは、指定したアラートの宛先にエラー・レポートを送る通知アクティビティです。

メッセージには、エラー・レポートという1つの送信属性が含まれます。これは、PL/SQL プロシージャ WF_ALERT.ErrorReport によって作成される値を持つ PL/SQL 文書属性です。

- メッセージ： 例外の検出された FYI
- 結果タイプ： なし
- 必須： No
- 前提条件アクティビティ： なし

終了（ノード3）

これは、単にプロセスの終了をマークする「標準」関数アクティビティです。

- 関数： WF_STANDARD.NOOP
- 結果タイプ： なし
- 必須： Yes
- 前提条件アクティビティ： なし
- 関数によって取り出されるアクティビティ属性： なし
- 関数によって設定される項目属性： なし
- 関数によって取り出される項目属性： なし

イベント・システム・デモンストレーション

イベント・システム・デモンストレーションは、2つのシステム間でビジネス・ドキュメントを送信するイベントの使用例です。一方のシステムで発注を入力して、デモンストレーション・プロセスを開始します。Oracle Workflow は発注 XML 文書を生成して、この文書をもう一方のシステムに送信します。発注先のシステムは発注を処理して、発注承認、アドバンスト出荷通知および請求書に相当する3つのXML文書を発注元のシステムに返送します。

Oracle Workflow のスタンドアロン版を使用している場合は、このサンプル・プロセスを開始できます。Oracle Application に組み込まれている Workflow を使用している場合、このプロセスは、主にデモンストレーションではなく説明のための一例と考えてください。

Oracle Application に組み込まれている Workflow には、このデモンストレーションの設定と実行に必要なファイルが用意されていません。

注意： Oracle Applications または Oracle Self-Service Web Applications と統合された実行可能ワークフロー・プロセスの詳細は、該当する Oracle Applications のユーザーズ・ガイドまたはオンライン・マニュアルを参照してください。C-2 ページの「[Oracle E-Business Suite 埋込みの事前定義済みのワークフロー](#)」を参照してください。

イベント・システム・デモンストレーションを実行する前に、デモンストレーションで使用するバイヤー・システムとサプライヤ・システムを設定する必要があります。

注意： 2つのシステムは、個別に設定したり、同じシステムをバイヤーおよびサプライヤとして使用することができます。

イベント・システム・デモンストレーション・プロセスは、「ワークフロー・デモンストレーション」Web ページのバイヤー・システムから開始できます。プロセスを開始するときに、発注についてオーダー番号、項目番号、項目の説明、納品日、合計金額およびオーダー・リクエスト・ロールを指定する必要があります。

注意： オーダー番号がプロセスの項目キーで使用されるので、シングルバイト・キャラクタのみ使用できます。

発注を送信すると、注文情報がデータベース表に挿入され、オーダー番号をイベント・キーとして発注イベントが呼び出されます。発注イベントを呼び出すと、このイベントへのサブスクリプションから、「ローカル」ソース・タイプを持つ2つのサブスクリプションがトリガーされます。一方のサブスクリプションでは、関連IDがイベント・メッセージに追加されます。関連IDは、接頭辞POとイベント・キー（オーダー番号）で構成されます。

発注イベントへのもう一方のサブスクリプションには、すべてのイベント・データが必要です。このため、イベント・マネージャはイベントのジェネレート関数を実行して、有効な発注 XML 文書を作成します。もう一方のサブスクリプションでは、「イベント・システムのデモ」項目タイプのバイヤー:最上位の発注プロセスにイベントが送信されます。ワークフロー・エンジンにより、関連 ID をイベント・キーとしてこのプロセスの新しいインスタンスが作成されます。

バイヤー:最上位の発注プロセスで発注を処理しているときに、標準の外部 Java 関数アクティビティによってオーダー・リクエストの名前が発注 XML 文書から取得されます。発注プロセスは、この情報を基にしてオーダー・リクエストに通知を送信します。発注プロセスは、発注イベント・メッセージをサプライヤ・システムに送信し、サプライヤからの応答を待機します。

発注イベントがサプライヤ・システムに着信すると、このイベントへのサブスクリプションのうち、「外部」ソース・タイプを持つ2つのサブスクリプションがトリガーされます。一方のサブスクリプションでは、イベント・メッセージの関連 ID の構成が、接頭辞 SO とイベント・キー（オーダー番号）に変更されます。もう一方のサブスクリプションでは、「イベント・システムのデモ」項目タイプのサプライヤ:最上位のオーダー・プロセスにイベントが送信されます。ワークフロー・エンジンにより、新しい関連 ID をイベント・キーとしてこのプロセスの新しいインスタンスが作成されます。

サプライヤ:最上位のオーダー・プロセスで発注が処理されているときに、標準の外部 Java 関数アクティビティによってオーダー番号と説明が発注 XML 文書から取得されます。次の XML 文書を含むイベント・メッセージが、オーダー・プロセスからバイヤー・システムに送信されます。

- 発注承認
- アドバンスト出荷通知
- 請求書

バイヤー・システムでは、これらのイベントによって「外部」ソース・タイプのサブスクリプションがトリガーされます。各イベントには、2つのサブスクリプションが割り当てられています。一方のサブスクリプションでは、接頭辞 PO とイベント・キー（オーダー番号）で構成される関連 ID が追加されます。もう一方のサブスクリプションでは、バイヤー:最上位の発注プロセスにイベント・メッセージが送信されます。このとき、イベント・メッセージの送信元プロセスは、関連 ID によって照合されます。発注プロセスは、イベント・メッセージを受信すると、オーダー・リクエストに通知します。3つの応答文書がすべて着信すると、プロセスは完了します。

イベント・システム・デモンストレーション・データ・モデルのインストール

イベント・システム・データ・モデルは、Oracle Workflow のスタンドアロン版でのみインストールされます。データ・モデルは、Workflow Configuration Assistant によって自動的にインストールされます。インストールで使用されるファイルは、Oracle Workflow Server のディレクトリ構造の demo および demo//<Language> サブディレクトリにコピーされます。

注意： イベント・システム・デモンストレーションを正常に機能させるには、インストール後に、Oracle Workflow の必須設定手順を実行する必要があります。『Oracle Workflow 管理者ガイド』の「設定の概要」を参照してください。

インストール時には、次の処理が行われます。

- スクリプト wfevdemc.sql がコールされ、WF_EVENTDEMO_ITEMS および WF_EVENTDEMO_PO という 2 つの表が作成されます。表 WF_EVENTDEMO_ITEMS には、発注時に選択可能な項目が格納されます。イベント・システム・デモンストレーション・プロセスでは、発注情報によって表 WF_EVENTDEMO_PO が更新されます。
- スクリプト wfevdems.sql および wfevdemb.sql がコールされ、WF_EVENTDEMO というパッケージの PL/SQL 仕様と本体が作成されます。このパッケージの内容は次のとおりです。
 - PL/SQL スタアド・プロシージャ。イベント・システム・デモンストレーション・ワークフローで使用される関数アクティビティによってコールされます。
 - PL/SQL プロシージャの WF_EVENTDEMO.Create_Order および WF_EVENTDEMO.Track_Order。Oracle Workflow の Web Agent によってコールされ、イベント・システム・デモンストレーション・プロセスのデモンストレーションに対して、Web ベースのインタフェース・ページを生成します。
- イベント・システム・デモンストレーション・ワークフロー定義が、wfevdeme.wft からデータベースにロードされます。このプロセスは、Oracle Workflow Builder で表示できます。
- イベント・システム・デモンストレーション・プロセスのデータ・モデルには、サンプルの購買申請プロセスで使用されるデモンストレーション・ディレクトリ・サービスも含まれます。10-6 ページの「[購買申請データ・モデルのインストール](#)」を参照してください。

注意： セキュリティ上の理由から、インストール・プロセスでは、デモンストレーション・ディレクトリ・サービスのユーザーのデータベース・アカウントは作成後に自動的にロックされます。アカウントを使用する前に、wfdemoul.sql というスクリプトを使用して、それらのアカウントをロック解除する必要があります。10-6 ページの「[購買申請データ・モデルのインストール](#)」を参照してください。

イベント・システム・デモンストレーション・ワークフローの開始

イベント・システム・デモンストレーションには、2つのワークフロー対応システム（バイヤー・システムとサプライヤ・システム）が必要です。2つのシステムは、個別に設定したり、同じシステムをバイヤーおよびサプライヤとして使用することができます。イベント・システム・デモンストレーションを実行する前に、使用するシステムを設定する必要があります。10-55 ページの「[イベント・システム・デモンストレーション・ワークフローの設定](#)」を参照してください。

2つのシステムを設定したら、「イベント: Buyer Workbench」デモンストレーション Web ページを使用して、イベント・システム・デモンストレーション・ワークフローを開始できます。ワークフローを開始したら、サプライヤ・システムの「[オーダーの追跡](#)」デモンストレーション Web ページを使用して、イベント・システム・デモンストレーション・ワークフローの処理を継続できます。10-56 ページの「[Buyer Workbench からのイベント・システム・デモンストレーション・ワークフローの開始](#)」および 10-58 ページの「[サプライヤ・システムでのイベント・システム・デモンストレーション・ワークフローの継続](#)」を参照してください。

▶ イベント・システム・デモンストレーション・ワークフローの設定

1. Oracle Workflow の 2 つのインストレーションを個別に使用する場合は、一方をバイヤー・システム、もう一方をサプライヤ・システムとして指定します。
2. Oracle Workflow の 2 つのインストレーションを個別に使用する場合は、2 つのシステムを相互にサインアップして、イベント・メッセージを交換します。8-63 ページの「[システムのサインアップ](#)」を参照してください。
3. Oracle Workflow の 2 つのインストレーションを個別に使用する場合は、バイヤー・システム上で、「ローカル」ソース・タイプおよびフェーズ 2 を持つ demo.oracle.apps.wf.po.create イベントへの事前定義済サブスクリプションを検索します。サプライヤ・システムの標準 WF_IN エージェントを宛先エージェントとして選択して、このサブスクリプションを編集します。8-52 ページの「[イベント・サブスクリプションの定義](#)」を参照してください。
4. Oracle Workflow の 2 つのインストレーションを個別に使用する場合は、サプライヤ・システム上で、「外部」ソース・タイプおよびフェーズ 2 を持つ demo.oracle.apps.wf.po.create イベントへの事前定義済サブスクリプションを検索します。バイヤー・システムの標準 WF_IN エージェントを宛先エージェントとし

て選択して、このサブスクリプションを編集します。8-52 ページの「[イベント・サブスクリプションの定義](#)」を参照してください。

5. Java 関数アクティビティ・エージェントが、システムで動作していることを確認します。『Oracle Workflow 管理者ガイド』の「[Java 関数アクティビティ・エージェントの設定](#)」を参照してください。
6. オプションで、バックグラウンド・エンジンがシステムで 10 ～ 30 秒おきに実行されるようにスケジュールします。『Oracle Workflow 管理者ガイド』の「[バックグラウンドのワークフロー・エンジンの設定](#)」を参照してください。

「イベント: オーダーの追跡」ページの「プロセス・オーダー」リンクを選択して、デモンストレーション中にバックグラウンド・エンジンを手動で実行することもできます。

▶ Buyer Workbench からのイベント・システム・デモンストレーション・ワークフローの開始

1. バイヤー・システム上の Web ブラウザに次の URL を入力して、「ワークフロー・デモンストレーション」Web ページにアクセスします。

```
<webagent>/wf_demo.home
```

<webagent> は、Web サーバーで Oracle Workflow 用に構成された Web エージェントのベース URL を表します。『Oracle Workflow 管理者ガイド』の「[グローバル・ユーザー設定の設定](#)」を参照してください。

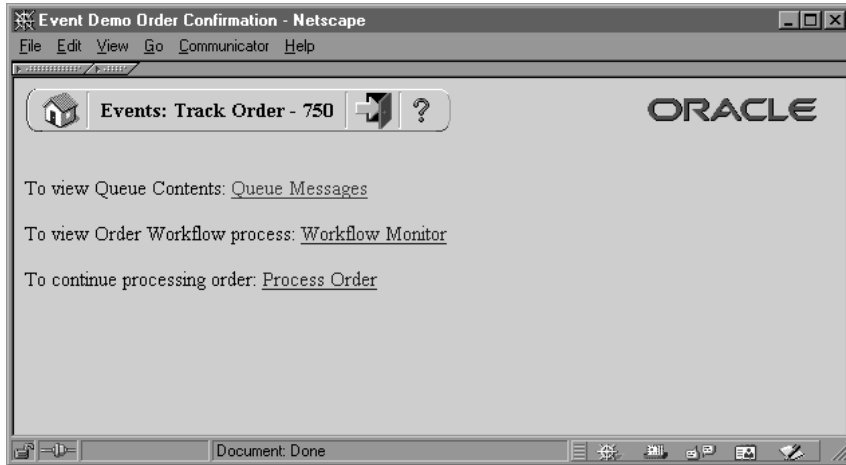
「イベント: Buyer Workbench」リンクを選択して、「Buyer Workbench」Web ページを表示します。

または、次の URL を入力して「Buyer Workbench」Web ページを直接表示することもできます。

```
<webagent>/wf_eventdemo.create_order
```

注意： どちらのページにもセキュリティが適用されるため、現行 Web セッションで有効なワークフロー・ユーザーとしてログオンしていない場合は、ページが表示される前に有効なワークフロー・ユーザーとしてのログオンを求めるプロンプトが表示されます。

2. 発注に対して一意のオーダー番号を入力します。オーダー番号は、呼び出される発注イベントのイベント・キーになります。
3. 項目番号を選択します。
4. 項目の説明を半角英数字 40 文字以内で入力します。
5. 納品日を「DD-MON-YYYY」の形式で入力します。
6. 発注の合計金額を入力します。この金額は書式なしの数値として入力する必要があります。
7. オーダー・リクエストのロール名を選択します。
8. 「送信」ボタンを選択して発注を送信し、「イベント：オーダーの追跡」ページを表示します。「取消」ボタンを選択し、発注を送信しないで「ワークフロー・デモンストレーション」ページに戻ることもできます。



9. 「イベント：オーダーの追跡」ページで、「キュー・メッセージ」リンクを選択し、「イベント・システム・ローカル・キュー」ページを表示して、ビジネス・イベント・システムのキューにあるメッセージを確認します。8-69 ページの「ローカル・キューの確認」を参照してください。
10. 「ワークフロー・モニター」リンクを選択し、ワークフロー・モニターでパイヤー：最上位の発注ワークフロー・プロセスの進行状況を確認します。『Oracle Workflow 管理者ガイド』の「ワークフロー・モニター」を参照してください。
11. 「プロセス・オーダー」リンクを選択し、バックグラウンド・エンジンを実行して、パイヤー・ワークフローが発注の処理を継続できるように遅延アクティビティを処理します。

▶ **サプライヤ・システムでのイベント・システム・デモンストレーション・ワークフローの継続**

1. サプライヤ・システムで、Web ブラウザに次の URL を入力して「ワークフロー・デモンストレーション」Web ページにアクセスします。

```
<webagent>/wf_demo.home
```

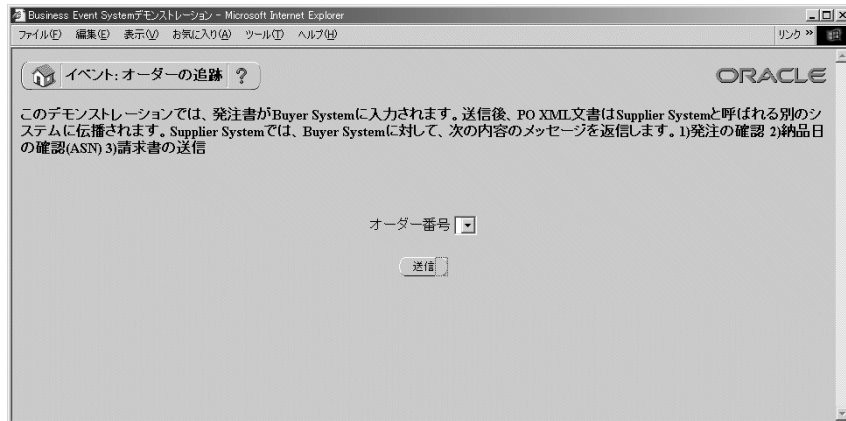
<webagent> は、Web サーバーで Oracle Workflow 用に構成された Web エージェントのベース URL を表します。『Oracle Workflow 管理者ガイド』の「グローバル・ユーザー設定の設定」を参照してください。

「イベント：オーダーの追跡」リンクを選択して、「イベント：オーダーの追跡」Web ページを表示します。

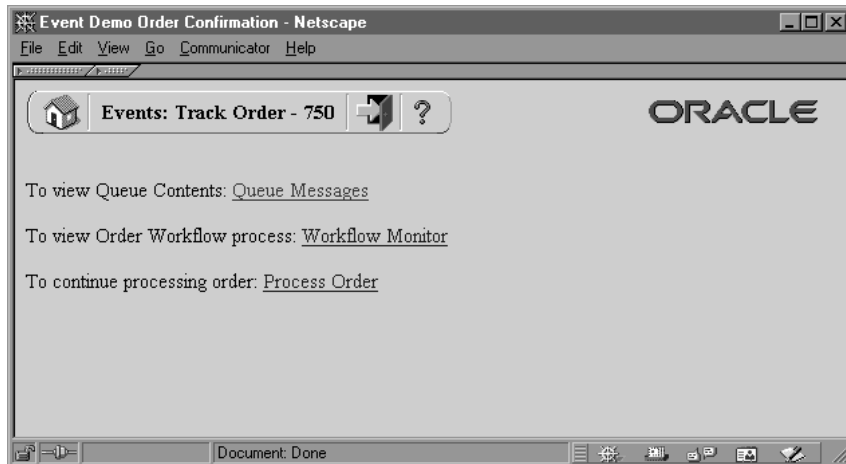
または、次の URL を入力して「Buyer Workbench」Web ページを直接表示することもできます。

```
<webagent>/wf_eventdemo.track_order
```

注意： どちらのページにもセキュリティが適用されるため、現行 Web セッションで有効なワークフロー・ユーザーとしてログオンしていない場合は、ページが表示される前に有効なワークフロー・ユーザーとしてのログオンを求めるプロンプトが表示されます。



2. 発注のオーダー番号を選択して、「送信」ボタンを選択します。



3. 表示される「イベント:オーダーの追跡」ページで、「キュー・メッセージ」リンクを選択して「イベント・システム・ローカル・キュー」ページを表示し、ビジネス・イベント・システムのキューにあるメッセージを確認します。8-69 ページの「ローカル・キューの確認」を参照してください。
4. 「ワークフロー・モニター」リンクを選択し、ワークフロー・モニターでサプライヤ:最上位のオーダー・プロセスの進行状況を確認します。『Oracle Workflow 管理者ガイド』の「ワークフロー・モニター」を参照してください。
5. 「プロセス・オーダー」リンクを選択し、バックグラウンド・エンジンを実行して、発注の処理を続けます。この手順を 2 回実行して、サプライヤ・ワークフローの両方の遅延アクティビティを処理します。

「イベント・システムのデモ」項目タイプ

イベント・システム・デモンストレーションは、「イベント・システムのデモ」という項目タイプに関連付けられています。現在、次の 11 のワークフロー・プロセスがイベント・システム・デモンストレーションに関連付けられています。

- バイヤー:最上位の発注
- バイヤー:サプライヤへの発注送信
- バイヤー:サプライヤの発注承認の受信
- バイヤー:アドバンスト出荷通知
- バイヤー:サプライヤの請求の受信
- サプライヤ:最上位のオーダー
- サプライヤ:オーダー詳細の取得
- サプライヤ:クレジット・チェック
- サプライヤ:在庫チェック
- サプライヤ:アドバンスト出荷通知
- サプライヤ:サプライヤの請求書の送信

「イベント・システムのデモ」項目タイプの詳細を **Workflow Builder** に表示するには、「ファイル」メニューから「オープン」を選択します。次に、データベースに接続して、「イベント・システムのデモ」項目タイプを選択するか、ファイル・システムの `<ORACLE_HOME>/wf/Data/<language>` サブディレクトリにある `wfefdeme.wft` というファイルに接続します。

「イベント・システムのデモ」のプロパティ画面を見ると、維持タイプが「一時」に、維持日数が 0 になっています。これは、この項目タイプの作業項目に関連付けられているランタイム・データは、終了直後に削除の対象となることを意味します。

「イベント・システムのデモ」項目タイプには、複数の属性も関連付けられています。これらの属性は、ワークフローのアプリケーション表にある情報を参照します。属性は、プロセス全体を通して関数アクティビティ、通知アクティビティおよびイベント・アクティビティによって使用され保存されます。次の表は、「イベント・システムのデモ」項目タイプの属性を示しています。

表 10-5

表示名	説明	型	長さ/書式/ 参照タイプ/ デフォルト値
イベント名	元のイベントの内部名	テキスト	
イベント・キー	元のイベントの特定のインスタンスを一意に識別するイベント・キー	テキスト	
イベント・メッセージ	元のイベントのイベント・メッセージ	イベント	
発注ステータス	発注のステータス	テキスト	CREATED
発注承認イベント	発注承認イベントの名前	テキスト	
発注承認イベント・キー	発注承認イベントの特定のインスタンスを一意に識別するイベント・キー	テキスト	
発注 ASN イベント	アドバンスト出荷通知イベントの名前	テキスト	
発注 ASN イベント・キー	アドバンスト出荷通知イベントの特定のインスタンスを一意に識別するイベント・キー	テキスト	
発注 ASN イベント・メッセージ	アドバンスト出荷通知イベント・メッセージ	イベント	
オーダー・リクエスト	オーダーをリクエストしたユーザーの名前	テキスト	BLEWIS
宛先エージェント / システム 1	イベント・メッセージを受信するパイヤー・システム上のインバウンド・エージェント (書式は <agent>@<system>)	テキスト	
送信元エージェント / システム 1	イベント・メッセージを送信するパイヤー・システム上のアウトバウンド・エージェント (書式は <agent>@<system>)	テキスト	

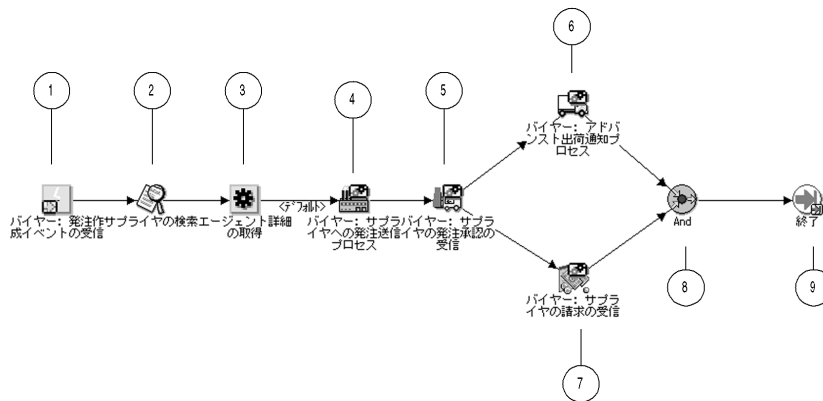
表 10-5 (続き)

表示名	説明	型	長さ / 書式 / 参照タイプ / デフォルト値
宛先エージェント / システム 2	イベント・メッセージを受信するサプライヤ・システム上のインバウンド・エージェント (書式は <agent>@<system>)	テキスト	
送信元エージェント / システム 2	イベント・メッセージを送信するサプライヤ・システム上のアウトバウンド・エージェント (書式は <agent>@<system>)	テキスト	
項目番号	項目番号	テキスト	
項目の説明	項目の説明	テキスト	
サブスクリプション GUID	サブスクリプションのグローバル一意識別子	テキスト	

バイヤー：最上位の発注プロセスの概要

バイヤー：最上位の発注プロセスのプロパティを表示するには、ナビゲータ・ツリーでプロセスを選択し、「編集」メニューから「プロパティ」を選択します。このプロセス・アクティビティは実行可能です。つまり、最上位レベルのプロセスとして実行できます。

バイヤー：最上位の発注プロセスの「プロセス」ウィンドウを表示すると、プロセスが9つの異なるアクティビティから構成されていることがわかります。プロセスのアクティビティを詳しく説明するために、次のように各ノードに番号を付けて参照しやすくしてあります。番号自体は、プロセス・ダイアグラムには含まれません。



バイヤー:最上位の発注ワークフローは、「Buyer Workbench」デモンストレーション Web ページから発注を送信して、発注イベントを呼び出すと開始します。事前定義のサブスク립ションにより、接頭辞 PO とイベント・キー（オーダー番号）で構成される相関 ID が追加され、イベント・メッセージがバイヤー:最上位の発注プロセスに送信されます。10-56 ページの「Buyer Workbench からのイベント・システム・デモンストレーション・ワークフローの開始」を参照してください。

発注プロセスは、ノード 1 の「バイヤー:発注作成イベントの受信」アクティビティから開始します。この発注のサプライヤを検索し、目的のアウトバウンド・エージェントおよびインバウンド・エージェントのエージェント詳細を取得したら、発注をサプライヤに送信します。

ノード 4 は、発注イベント・メッセージからオーダー・リクエストの名前を取得し、発注をサプライヤに送信して、発注がサプライヤに送信されたことをオーダー・リクエストに通知するサブプロセスです。

ノード 5 は、サプライヤからの発注承認イベント・メッセージを待機するサブプロセスです。発注承認が指定した期間内に着信しない場合は、タイムアウト・アクティビティを実行し、発注承認を受信するまで、サプライヤが応答しないことをオーダー・リクエストに繰り返し通知します。承認が着信すると、サブプロセスはオーダー・リクエストに承認を通知します。

発注承認が着信すると、最上位プロセスは、サプライヤからのアドバンスト出荷通知および請求書を待機します。ノード 6 は、アドバンスト出荷通知を受信し、オーダーが出荷されたことをオーダー・リクエストに通知するサブプロセスです。ノード 7 は、請求書を受信し、そのことをオーダー・リクエストに通知するサブプロセスです。これらのサブプロセスが完了すると、プロセスは終了します。

バイヤー：最上位の発注プロセスのアクティビティ

ここでは、プロセスの各アクティビティについて、アクティビティの表示名別に説明します。

バイヤー：発注作成イベントの受信（ノード1）

このイベント・アクティビティは、イベント・マネージャによってバイヤー：最上位の発注プロセスに送信された発注イベント・メッセージを受信して、新しい項目を開始します。

イベント・アクション	受信
イベント・フィルタ	demo.oracle.wf.b2b.po.create
前提条件アクティビティ	なし
アクティビティによって設定される項目属性	イベント名、イベント・キー、イベント・メッセージ

サプライヤの検索（ノード2）

現在、このアクティビティでは何も処理されません。このアクティビティに対して、発注されている項目などの条件に基づいて、発注のサプライヤを検索する関数を統合できます。

関数	WF_EVENTDEMO.FINDSUPPLIER
結果タイプ	なし
前提条件アクティビティ	バイヤー：発注作成イベントの受信

エージェント詳細の取得（ノード3）

この関数アクティビティは、イベント・メッセージをワークフロー・プロセスに送信するサブスクリプションから、エージェント詳細を取得します。このアクティビティは、メッセージを送信するバイヤー・システム上のアウトバウンド・エージェントおよびメッセージを受信するサプライヤ・システム上のインバウンド・エージェントの、エージェント詳細を取得します。

関数	WF_STANDARD.GETAGENTS
結果タイプ	ブール
前提条件アクティビティ	サプライヤの検索
関数によって取得される項目属性	サブスクリプション GUID
関数によって設定される項目属性	送信元エージェント / システム 1、宛先エージェント / システム 2

バイヤー：サプライヤへの発注送信プロセス（ノード4）

このアクティビティは、発注イベント・メッセージからオーダー・リクエストの名前を取得し、発注をサプライヤに送信して、発注がサプライヤに送信されたことをオーダー・リクエストに通知するサブプロセスです。このサブプロセスを参照するには、ナビゲータ・ツリーで「プロセス」のブランチの下にある「バイヤー：サプライヤへの発注書の送信」をダブルクリックします。10-67 ページの「[バイヤー：サプライヤへの発注送信サブプロセスの概要](#)」を参照してください。

結果タイプ なし

前提条件アクティビティ エージェント詳細の取得

バイヤー：サプライヤの発注承認の受信（ノード5）

このアクティビティは、サプライヤから発注承認イベント・メッセージを待機するサブプロセスです。発注承認が指定した期間内に着信しない場合は、タイムアウト・アクティビティを実行し、発注承認を受信するまで、サプライヤが応答しないことをオーダー・リクエストに繰り返し通知します。承認が着信すると、サブプロセスはオーダー・リクエストに承認を通知します。

このサブプロセスを参照するには、ナビゲータ・ツリーで「プロセス」のブランチの下にある「バイヤー：サプライヤの発注承認の受信」をダブルクリックします。10-69 ページの「[バイヤー：サプライヤの発注承認の受信サブプロセスの概要](#)」を参照してください。

結果タイプ なし

前提条件アクティビティ バイヤー：サプライヤへの発注送信

バイヤー：アドバンスト出荷通知プロセス（ノード6）

このアクティビティは、サプライヤからアドバンスト出荷通知を受信し、オーダーが出荷されたことをオーダー・リクエストに通知するサブプロセスです。このサブプロセスを表示するには、ナビゲータ・ツリーの「プロセス」のブランチの下にある「バイヤー：アドバンスト出荷通知」をダブルクリックします。10-73 ページの「[バイヤー：アドバンスト出荷通知サブプロセスの概要](#)」を参照してください。

結果タイプ なし

前提条件アクティビティ バイヤー：サプライヤの発注承認の受信

バイヤー：サプライヤの請求の受信（ノード 7）

このアクティビティは、サプライヤから請求書を受信し、請求書が届いたことをオーダー・リクエストに通知するサブプロセスです。このサブプロセスを参照するには、ナビゲータ・ツリーで「プロセス」のブランチの下にある「バイヤー：サプライヤの請求の受信」をダブルクリックします。10-75 ページの「[バイヤー：サプライヤの請求の受信サブプロセスの概要](#)」を参照してください。

結果タイプ なし

前提条件アクティビティ バイヤー：サプライヤの発注承認の受信

And（ノード 8）

この「標準」関数アクティビティでは、フロー内の並行する 2 つ以上の分岐のすべてのアクティビティが完了した場合にのみ、分岐がマージされます。

関数 WF_STANDARD.ANDJOIN

結果タイプ なし

前提条件アクティビティ バイヤー：アドバンスト出荷通知、バイヤー：サプライヤの請求の受信

終了（ノード9）

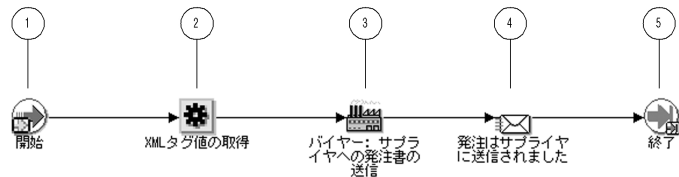
この「標準」関数アクティビティは、プロセスの終了をマークします。

関数	WF_STANDARD.NOOP
結果タイプ	なし
前提条件アクティビティ	And

バイヤー：サプライヤへの発注送信サブプロセスの概要

バイヤー：サプライヤへの発注送信サブプロセスのプロパティを表示するには、そのプロセス・アクティビティをナビゲータ・ツリーで選択し、「編集」メニューから「プロパティ」を選択します。このプロセス・アクティビティは、実行可能ではありません。つまり、最上位レベルのプロセスとして開始できず、上位レベルのプロセスからコールする必要があります。

バイヤー：サプライヤへの発注送信サブプロセスの「プロセス」ウィンドウを表示すると、サブプロセスが5つの異なるアクティビティから構成されていることがわかります。プロセスのアクティビティを詳しく説明するために、次のように各ノードに番号を付けて参照しやすくしてあります。番号自体は、プロセス・ダイアグラムには含まれません。



このサブプロセスは、ノード1の「開始」アクティビティから開始します。ノード2で、発注イベント・メッセージからオーダー・リクエストの名前を取得します。次に、発注をサプライヤに送信し、発注が送信されたことをオーダー・リクエストに通知します。この時点でサブプロセスは終了します。

バイヤー：サプライヤへの発注送信サブプロセスのアクティビティ

ここでは、サブプロセスの各アクティビティについて、アクティビティの表示名別に説明します。

開始（ノード 1）

この「標準」関数アクティビティは、プロセスの開始をマークします。

関数	WF_STANDARD.NOOP
結果タイプ	なし
前提条件アクティビティ	なし

XML タグ値の取得（ノード 2）

この「標準」外部 Java 関数アクティビティは、イベント・メッセージに設定されている特定の XML タグの値を取得します。このノードで、プロセスは発注イベント・メッセージからオーダー・リクエストの名前を取得します。

関数	oracle.apps.fnd.wf.XMLGetTagValue
結果タイプ	なし
前提条件アクティビティ	開始
関数によって取得される項目属性	イベント・メッセージ
関数によって設定される項目属性	オーダー・リクエスト

バイヤー：サプライヤへの発注書の送信（ノード 3）

このイベント・アクティビティは、バイヤー・システム上のアウトバウンド・エージェントおよびサプライヤ・システム上のインバウンド・エージェントを使用して、発注イベント・メッセージをサプライヤに送信します。これらのエージェントは、バイヤー：最上位の発注プロセスの「エージェント詳細の取得」アクティビティによって識別されます。

イベント・アクション	送信
前提条件アクティビティ	XML タグ値の取得
アクティビティによって取得される項目属性	イベント・メッセージ、送信元エージェント / システム 1、宛先エージェント / システム 2

発注はサプライヤに送信されました（ノード4）

このアクティビティは、発注がサプライヤに送信されたことをオーダー・リクエストに通知します。メッセージには、通知の送信時に発注番号を表示する「送信」属性が含まれます。

このアクティビティ・ノードのプロパティ画面を表示すると、「オーダー・リクエスト」という項目タイプ属性に名前が保存されている実行者に、このアクティビティが割り当てられていることがわかります。

メッセージ	発注はサプライヤに送信されました
結果タイプ	なし
前提条件アクティビティ	バイヤー：サプライヤへの発注送信

終了（ノード5）

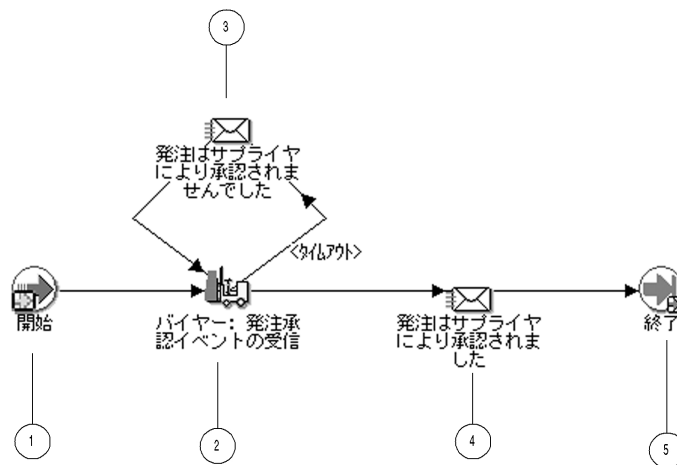
この「標準」関数アクティビティは、プロセスの終了をマークします。

関数	WF_STANDARD.NOOP
結果タイプ	なし
前提条件アクティビティ	発注はサプライヤに送信されました

バイヤー：サプライヤの発注承認の受信サブプロセスの概要

バイヤー：サプライヤの発注承認の受信サブプロセスのプロパティを表示するには、そのプロセス・アクティビティをナビゲータ・ツリーで選択し、「編集」メニューから「プロパティ」を選択します。このプロセス・アクティビティは、実行可能ではありません。つまり、最上位レベルのプロセスとして開始できず、上位レベルのプロセスからコールする必要があります。

バイヤー：サプライヤの発注承認の受信サブプロセスの「プロセス」ウィンドウを表示すると、サブプロセスが5つの異なるアクティビティから構成されていることがわかります。プロセスのアクティビティを詳しく説明するために、次のように各ノードに番号を付けて参照しやすくしてあります。番号自体は、プロセス・ダイアグラムには含まれません。



このサブプロセスは、ノード1の「開始」アクティビティから開始します。ノード2で、サプライヤから発注承認イベント・メッセージを待機します。発注承認が指定した期間内に着信しない場合は、タイムアウト・アクティビティを実行し、発注承認を受信するまで、サプライヤが応答しないことをオーダー・リクエストに繰り返し通知します。承認が着信すると、サブプロセスはオーダー・リクエストに承認を通知します。この時点でサブプロセスは終了します。

バイヤー：サプライヤの発注承認の受信サブプロセスのアクティビティ

ここでは、サブプロセスの各アクティビティについて、アクティビティの表示名別に説明します。

開始（ノード 1）

この「標準」関数アクティビティは、プロセスの開始をマークします。

関数	WF_STANDARD.NOOP
結果タイプ	なし
前提条件アクティビティ	なし

バイヤー：発注承認イベントの受信（ノード 2）

このイベント・アクティビティは、サプライヤから発注承認イベント・メッセージを待機します。このアクティビティは 1 日以内に完了しないとタイムアウトになります。

イベント・アクション	受信
イベント・フィルタ	demo.oracle.wf.b2b.po.ack
前提条件アクティビティ	開始
アクティビティによって 設定される項目属性	イベント名、イベント・キー、イベント・メッセージ

発注はサプライヤにより承認されませんでした（ノード 3）

このアクティビティは、「バイヤー：発注承認イベントの受信」アクティビティがタイムアウトになり、完了しなかった場合にのみ発生します。このアクティビティは、サプライヤがまだ発注を承認していないことをオーダー・リクエスタに通知します。メッセージには、通知の送信時に発注番号を表示する「送信」属性が含まれます。

このアクティビティ・ノードのプロパティ画面を表示すると、「オーダー・リクエスタ」という項目タイプ属性に名前が保存されている実行者に、このアクティビティが割り当てられていることがわかります。

メッセージ	発注未承認
結果タイプ	なし
前提条件アクティビティ	バイヤー：発注承認イベントの受信

発注はサプライヤにより承認されました（ノード4）

このアクティビティは、発注がサプライヤによって承認されたことをオーダー・リクエストに通知します。メッセージには、通知の送信時に発注番号を表示する「送信」属性が含まれます。

このアクティビティ・ノードのプロパティ画面を表示すると、「オーダー・リクエスト」という項目タイプ属性に名前が保存されている実行者に、このアクティビティが割り当てられていることがわかります。

メッセージ	発注承認済
結果タイプ	なし
前提条件アクティビティ	バイヤー：発注承認イベントの受信

終了（ノード5）

この「標準」関数アクティビティは、プロセスの終了をマークします。

関数 WF_STANDARD.NOOP

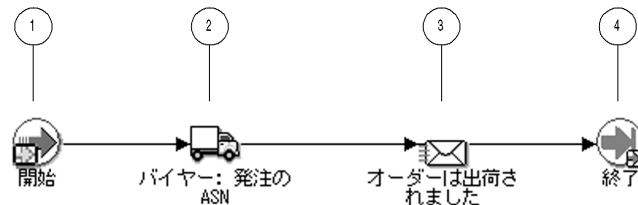
結果タイプ なし

前提条件アクティビティ 発注はサプライヤにより承認されました

バイヤー：アドバンスト出荷通知サブプロセスの概要

バイヤー：アドバンスト出荷通知サブプロセスのプロパティを表示するには、そのプロセス・アクティビティをナビゲータ・ツリーで選択し、「編集」メニューから「プロパティ」を選択します。このプロセス・アクティビティは、実行可能ではありません。つまり、最上位レベルのプロセスとして開始できず、上位レベルのプロセスからコールする必要があります。

バイヤー：アドバンスト出荷通知サブプロセスの「プロセス」ウィンドウを表示すると、サブプロセスが4つの異なるアクティビティから構成されていることがわかります。プロセスのアクティビティを詳しく説明するために、次のように各ノードに番号を付けて参照しやすくしてあります。番号自体は、プロセス・ダイアグラムには含まれません。



このサブプロセスは、ノード1の「開始」アクティビティから開始します。ノード2で、サプライヤからアドバンスト出荷通知を待機します。アドバンスト出荷通知が着信すると、オーダーが出荷されたことをオーダー・リクエストに通知します。この時点でサブプロセスは終了します。

バイヤー：アドバンスト出荷通知サブプロセスのアクティビティ

ここでは、サブプロセスの各アクティビティについて、アクティビティの表示名別に説明します。

開始（ノード1）

この「標準」関数アクティビティは、プロセスの開始をマークします。

関数	WF_STANDARD.NOOP
結果タイプ	なし
前提条件アクティビティ	なし

バイヤー：発注の ASN（ノード2）

このイベント・アクティビティは、サプライヤからアドバンスト出荷通知イベント・メッセージを待機します。

イベント・アクション	受信
イベント・フィルタ	demo.oracle.wf.b2b.po.asn
前提条件アクティビティ	開始
アクティビティによって 設定される項目属性	イベント名、イベント・キー、イベント・メッセージ

オーダーは出荷されました（ノード3）

このアクティビティは、アドバンスト出荷通知が着信したことをオーダー・リクエストに通知します。つまり、オーダーがサプライヤから出荷されたことを通知します。メッセージには、通知の送信時に発注番号を表示する「送信」属性が含まれます。

このアクティビティ・ノードのプロパティ画面を表示すると、「オーダー・リクエスト」という項目タイプ属性に名前が保存されている実行者に、このアクティビティが割り当てられていることがわかります。

メッセージ	発注のアドバンスト出荷通知
結果タイプ	なし
前提条件アクティビティ	バイヤー：発注の ASN

終了（ノード4）

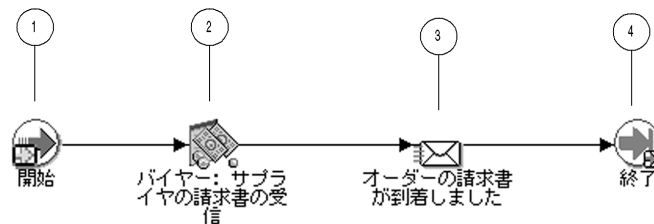
この「標準」関数アクティビティは、プロセスの終了をマークします。

関数	WF_STANDARD.NOOP
結果タイプ	なし
前提条件アクティビティ	オーダーは出荷されました

バイヤー：サプライヤの請求の受信サブプロセスの概要

バイヤー：サプライヤの請求の受信サブプロセスのプロパティを表示するには、そのプロセス・アクティビティをナビゲータ・ツリーで選択し、「編集」メニューから「プロパティ」を選択します。このプロセス・アクティビティは、実行可能ではありません。つまり、最上位レベルのプロセスとして開始できず、上位レベルのプロセスからコールする必要があります。

バイヤー：サプライヤの請求の受信サブプロセスの「プロセス」ウィンドウを表示すると、サブプロセスが4つの異なるアクティビティから構成されていることがわかります。プロセスのアクティビティを詳しく説明するために、次のように各ノードに番号を付けて参照しやすくしてあります。番号自体は、プロセス・ダイアグラムには含まれません。



このサブプロセスは、ノード1の「開始」アクティビティから開始します。ノード2で、サプライヤから請求書を待機します。請求書が着信したときに、請求が届いたことをオーダー・リクエストに通知します。この時点でサブプロセスは終了します。

バイヤー：サプライヤの請求の受信サブプロセスのアクティビティ

ここでは、サブプロセスの各アクティビティについて、アクティビティの表示名別に説明します。

開始（ノード 1）

この「標準」関数アクティビティは、プロセスの開始をマークします。

関数	WF_STANDARD.NOOP
結果タイプ	なし
前提条件アクティビティ	なし

バイヤー：サプライヤの請求書の受信（ノード 2）

このイベント・アクティビティは、サプライヤから請求書イベント・メッセージを待機します。

イベント・アクション	受信
イベント・フィルタ	demo.oracle.wf.b2b.po.invoice
前提条件アクティビティ	開始
アクティビティによって設定される項目属性	イベント名、イベント・キー、イベント・メッセージ

オーダーの請求書が到着しました（ノード 3）

このアクティビティは、サプライヤから発注の請求書が着信したことをオーダー・リクエストに通知します。メッセージには、通知の送信時に発注番号を表示する「送信」属性が含まれます。

このアクティビティ・ノードのプロパティ画面を表示すると、「オーダー・リクエスト」という項目タイプ属性に名前が保存されている実行者に、このアクティビティが割り当てられていることがわかります。

メッセージ	サプライヤ発注請求書
結果タイプ	なし
前提条件アクティビティ	バイヤー：サプライヤの請求書の受信

終了（ノード4）

この「標準」関数アクティビティは、プロセスの終了をマークします。

関数 WF_STANDARD.NOOP

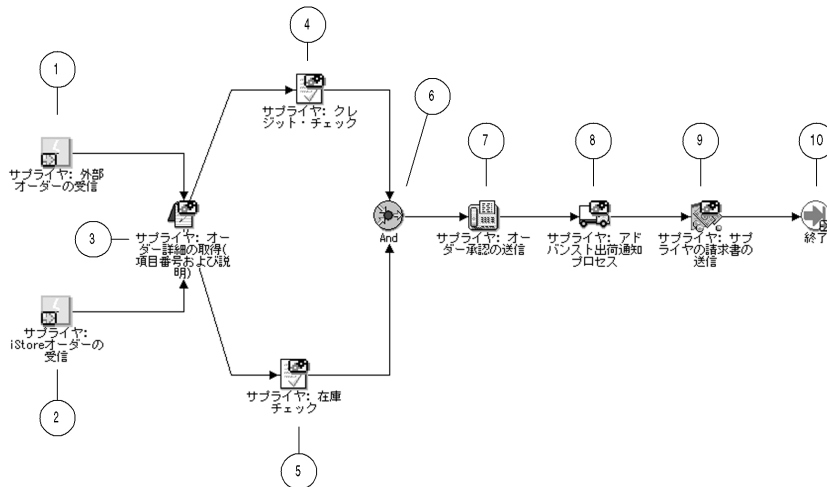
結果タイプ なし

前提条件アクティビティ オーダーの請求書が到着しました

サプライヤ：最上位のオーダー・プロセスの概要

サプライヤ：最上位のオーダー・プロセスのプロパティを表示するには、ナビゲータ・ツリーでプロセスを選択し、「編集」メニューから「プロパティ」を選択します。このプロセス・アクティビティは実行可能です。つまり、最上位レベルのプロセスとして実行できます。

サプライヤ：最上位のオーダー・プロセスの「プロセス」ウィンドウを表示すると、プロセスが10の異なるアクティビティから構成されていることがわかります。プロセスのアクティビティを詳しく説明するために、次のように各ノードに番号を付けて参照しやすくしてあります。番号自体は、プロセス・ダイアグラムには含まれません。



サプライヤ：最上位のオーダー・ワークフローは、サプライヤ・システムがバイヤー・システムから発注イベントを受信したときに開始します。事前定義のサブスクリプションにより、関連 ID の構成が接頭辞 PO とイベント・キー（オーダー番号）に変更され、イベントがサプライヤ：最上位のオーダー・プロセスに送信されます。

このプロセスは、ノード1またはノード2に発注が着信したときに開始します。着信先は、発注のソースによって異なります。

ノード3は、項目番号および項目の説明とともに、目的のアウトバウンドおよびインバウンド・エージェントのエージェント詳細を取得して、応答をバイヤーに送信するサブプロセスです。

ノード4は、オーダーの合計コストが \$2000 を超える場合にクレジット・チェックを実行するサブプロセスです。ノード5は、在庫チェックを実行して、オーダー項目の在庫を確認するサブプロセスです。クレジット・チェックと在庫チェックが完了すると、発注の承認をバイヤーに送信します。

ノード8は、オーダーの出荷詳細をチェックして、アドバンスト出荷通知をバイヤーに送信するサブプロセスです。ノード9は、請求書を作成し、請求書をバイヤーに送信するサブプロセスです。請求書の送信後に、プロセスは終了します。

サプライヤ:最上位のオーダー・プロセスのアクティビティ

ここでは、プロセスの各アクティビティについて、アクティビティの表示名別に説明します。

サプライヤ:外部オーダーの受信 (ノード1)

このイベント・アクティビティは、発注イベント・メッセージを受信します。このメッセージは、イベント・マネージャがバイヤーからイベントを受信したときに、サプライヤ:最上位のオーダー・プロセスに送信されます。

イベント・アクション 受信

イベント・フィルタ demo.oracle.wf.b2b.po.create

前提条件アクティビティ なし

**アクティビティによって
設定される項目属性** イベント名、イベント・キー、イベント・メッセージ

サプライヤ:iStore オーダーの受信 (ノード2)

このイベント・アクティビティは、サプライヤの iStore アプリケーションによってオーダーが作成されたときに送信される、発注イベント・メッセージを受信します。現在、このイベントは定義されていません。このアクティビティに対して、別のソースから受信した発注を統合できます。

イベント・アクション 受信

イベント・フィルタ demo.oracle.wf.istore.po.create

前提条件アクティビティ なし

**アクティビティによって
設定される項目属性** イベント名、イベント・キー、イベント・メッセージ

サプライヤ:オーダー詳細の取得 (ノード 3)

このアクティビティは、項目番号およびオーダー項目の説明とともに、目的のアウトバウンドおよびインバウンド・エージェントのエージェント詳細を取得して、応答をバイヤーに送信するサブプロセスです。

このサブプロセスを参照するには、ナビゲータ・ツリーで「プロセス」のブランチの下にある「サプライヤ:オーダー詳細の取得」をダブルクリックします。10-81 ページの「[サプライヤ:オーダー詳細の取得サブプロセスの概要](#)」を参照してください。

結果タイプ なし

前提条件アクティビティ サプライヤ:外部オーダーの受信またはサプライヤ:iStore オーダーの受信

サプライヤ:クレジット・チェック (ノード 4)

このアクティビティは、オーダーの合計コストが \$2000 を超える場合にクレジット・チェックを実行するサブプロセスです。

このサブプロセスを表示するには、ナビゲータ・ツリーの「プロセス」のブランチの下にある「サプライヤ:クレジット・チェック」をダブルクリックします。10-85 ページの「[サプライヤ:クレジット・チェック・サブプロセスの概要](#)」を参照してください。

結果タイプ なし

前提条件アクティビティ サプライヤ:オーダー詳細の取得

サプライヤ:在庫チェック (ノード 5)

このアクティビティは、在庫チェックを実行して、オーダー項目の在庫を確認するサブプロセスです。

このサブプロセスを表示するには、ナビゲータ・ツリーの「プロセス」のブランチの下にある「サプライヤ:在庫チェック」をダブルクリックします。10-87 ページの「[サプライヤ:在庫チェック・サブプロセスの概要](#)」を参照してください。

結果タイプ なし

前提条件アクティビティ サプライヤ:オーダー詳細の取得

And (ノード 6)

この「標準」関数アクティビティでは、フロー内の並行する 2 つ以上の分岐のすべてのアクティビティが完了した場合にのみ、分岐がマージされます。

関数 WF_STANDARD.ANDJOIN

結果タイプ なし

前提条件アクティビティ サプライヤ: クレジット・チェック、サプライヤ: 在庫チェック

サプライヤ: オーダー承認の送信 (ノード 7)

このイベント・アクティビティは、サプライヤ・システム上のアウトバウンド・エージェントおよびバイヤー・システム上のインバウンド・エージェントを使用して、発注承認イベント・メッセージをバイヤーに送信します。これらのエージェントは、サプライヤ: オーダー詳細の取得サブプロセスの「エージェント詳細の取得」アクティビティによって識別されません。

イベント・アクション 送信

前提条件アクティビティ And

アクティビティによって取得される項目属性 イベント・メッセージ、送信元エージェント / システム 2、宛先エージェント / システム 1

サプライヤ: アドバンスト出荷通知プロセス (ノード 8)

このアクティビティは、オーダーの出荷詳細をチェックして、アドバンスト出荷通知をバイヤーに送信するサブプロセスです。

このサブプロセスを表示するには、ナビゲータ・ツリーの「プロセス」のブランチの下にある「サプライヤ: アドバンスト出荷通知」をダブルクリックします。10-89 ページの「[サプライヤ: アドバンスト出荷通知サブプロセスの概要](#)」を参照してください。

結果タイプ なし

前提条件アクティビティ サプライヤ: オーダー承認の送信

サプライヤ: サプライヤの請求書の送信 (ノード 9)

このアクティビティは、発注の請求書を作成し、請求書をバイヤーに送信するサブプロセスです。

このサブプロセスを参照するには、ナビゲータ・ツリーで「プロセス」のブランチの下にある「サプライヤ: サプライヤの請求書の送信」をダブルクリックします。10-92 ページの「[サプライヤ: サプライヤの請求書の送信サブプロセスの概要](#)」を参照してください。

結果タイプ なし

前提条件アクティビティ サプライヤ: アドバンスト出荷通知

終了 (ノード 10)

この「標準」関数アクティビティは、プロセスの終了をマークします。

関数 WF_STANDARD.NOOP

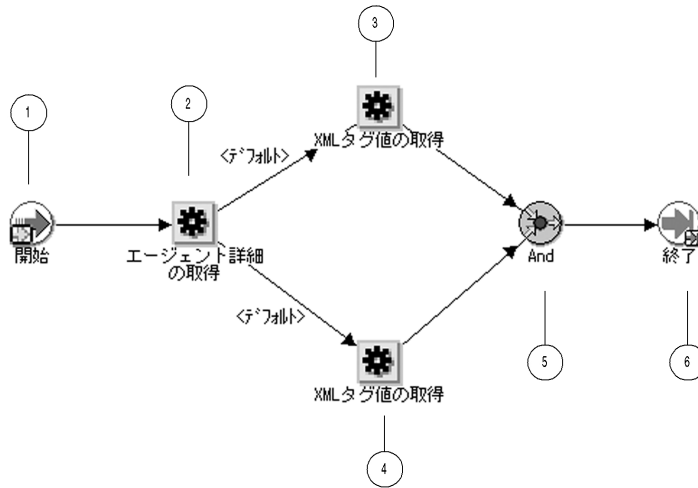
結果タイプ なし

前提条件アクティビティ サプライヤ: サプライヤの請求書の送信

サプライヤ: オーダー詳細の取得サブプロセスの概要

サプライヤ: オーダー詳細の取得サブプロセスのプロパティを表示するには、そのプロセス・アクティビティをナビゲータ・ツリーで選択し、「編集」メニューから「プロパティ」を選択します。このプロセス・アクティビティは、実行可能ではありません。つまり、最上位レベルのプロセスとして開始できず、上位レベルのプロセスからコールする必要があります。

サプライヤ: オーダー詳細の取得サブプロセスの「プロセス」ウィンドウを表示すると、サブプロセスが 5 つの異なるアクティビティで構成されていることがわかります。そのうちの 1 つが再利用され、ワークフロー・ダイアグラムに表示される 6 つのアクティビティ・ノードを構成しています。プロセスのアクティビティを詳しく説明するために、次のように各ノードに番号を付けて参照しやすくしてあります。番号自体は、プロセス・ダイアグラムには含まれません。



このサブプロセスは、ノード1の「開始」アクティビティから開始します。ノード2で、目的のアウトバウンド・エージェントおよびインバウンド・エージェントのエージェント詳細を取得して、応答をバイヤーに送信します。次に、オーダー項目の項目番号と項目の説明を発注から取得します。項目番号と項目の説明の取得が完了すると、サブプロセスは終了します。

サプライヤ：オーダー詳細の取得サブプロセスのアクティビティ

ここでは、サブプロセスの各アクティビティについて、アクティビティの表示名別に説明します。

開始（ノード1）

この「標準」関数アクティビティは、プロセスの開始をマークします。

関数	WF_STANDARD.NOOP
結果タイプ	なし
前提条件アクティビティ	なし

エージェント詳細の取得（ノード2）

この関数アクティビティは、イベント・メッセージをワークフロー・プロセスに送信するサブスクリプションから、エージェント詳細を取得します。このアクティビティは、応答メッセージを送信するサプライヤ・システム上のアウトバウンド・エージェントおよび応答メッセージを受信するバイヤー・システム上のインバウンド・エージェントの、詳細を取得します。

関数	WF_STANDARD.GETAGENTS
結果タイプ	ブール
前提条件アクティビティ	開始
関数によって取得される項目属性	サブスクリプション GUID
関数によって設定される項目属性	送信元エージェント / システム 2、宛先エージェント / システム 1

XML タグ値の取得（ノード3）

この「標準」外部 Java 関数アクティビティは、イベント・メッセージに設定されている特定の XML タグの値を取得します。このノードで、プロセスは発注イベント・メッセージから項目番号を取得します。

関数	oracle.apps.fnd.wf.XMLGetTagValue
結果タイプ	なし
前提条件アクティビティ	エージェント詳細の取得
関数によって取得される項目属性	イベント・メッセージ
関数によって設定される項目属性	項目番号

XML タグ値の取得（ノード 4）

この「標準」外部 Java 関数アクティビティは、イベント・メッセージに設定されている特定の XML タグの値を取得します。このノードで、発注イベント・メッセージから項目の説明を取得します。

関数 oracle.apps.fnd.wf.XMLGetTagValue

結果タイプ なし

前提条件アクティビティ エージェント詳細の取得

**関数によって取得される
項目属性** イベント・メッセージ

**関数によって設定される
項目属性** 項目の説明

And（ノード 5）

この「標準」関数アクティビティでは、フロー内の並行する 2 つ以上の分岐のすべてのアクティビティが完了した場合にのみ、分岐がマージされます。

関数 WF_STANDARD.ANDJOIN

結果タイプ なし

前提条件アクティビティ XML タグ値の取得（ノード 3）、XML タグ値の取得（ノード 4）

終了（ノード 6）

この「標準」関数アクティビティは、プロセスの終了をマークします。

関数 WF_STANDARD.NOOP

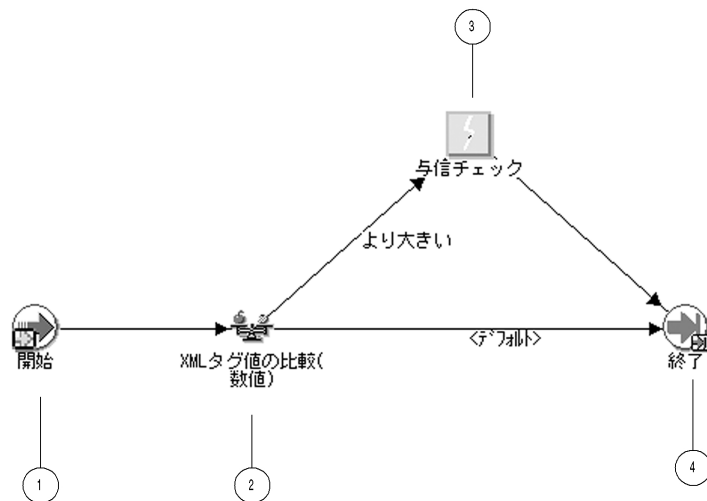
結果タイプ なし

前提条件アクティビティ And

サプライヤ:クレジット・チェック・サブプロセスの概要

サプライヤ:クレジット・チェック・サブプロセスのプロパティを表示するには、そのプロセス・アクティビティをナビゲータ・ツリーで選択し、「編集」メニューから「プロパティ」を選択します。このプロセス・アクティビティは、実行可能ではありません。つまり、最上位レベルのプロセスとして開始できず、上位レベルのプロセスからコールする必要があります。

サプライヤ:クレジット・チェック・サブプロセスの「プロセス」ウィンドウを表示すると、サブプロセスが4つの異なるアクティビティから構成されていることがわかります。プロセスのアクティビティを詳しく説明するために、次のように各ノードに番号を付けて参照しやすくしてあります。番号自体は、プロセス・ダイアグラムには含まれません。



このサブプロセスは、ノード1の「開始」アクティビティから開始します。ノード2で、オーダーの合計コストが\$2000を超えるかどうかをチェックします。合計コストが\$2000以下の場合、プロセスはこの時点で終了します。合計コストが\$2000を超える場合、バイヤーに対してクレジット・チェックを実行した後でプロセスは終了します。

サプライヤ:クレジット・チェック・サブプロセスのアクティビティ

ここでは、サブプロセスの各アクティビティについて、アクティビティの表示名別に説明します。

開始 (ノード 1)

この「標準」関数アクティビティは、プロセスの開始をマークします。

関数	WF_STANDARD.NOOP
結果タイプ	なし
前提条件アクティビティ	なし

XML タグ値の比較 (数値) (ノード 2)

この「標準」外部 Java 関数アクティビティは、イベント・メッセージに設定されている特定の XML タグの値とテスト値を比較します。このノードで、発注イベント・メッセージの合計コストとテスト値 \$2000 を比較します。

関数	oracle.apps.fnd.wf.XMLCompareTag
結果タイプ	比較
前提条件アクティビティ	開始
関数によって取得される項目属性	イベント・メッセージ

クレジット・チェック (ノード 3)

このアクティビティは、発注の合計コストが \$2000 を超える場合にのみ発生します。このイベント・アクティビティは、クレジット・チェック・イベントを呼び出します。現在、このイベントは定義されていません。このアクティビティに対して、クレジット・チェックを続合できます。

イベント・アクション	呼出し
前提条件アクティビティ	XML タグ値の比較 (数値)
アクティビティによって取得される項目属性	イベント・キー

終了（ノード4）

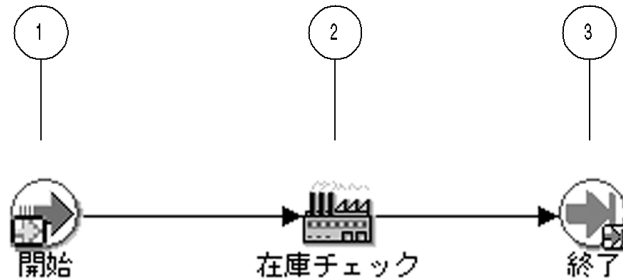
この「標準」関数アクティビティは、プロセスの終了をマークします。

関数	WF_STANDARD.NOOP
結果タイプ	なし
前提条件アクティビティ	XML タグ値の比較（数値）

サプライヤ：在庫チェック・サブプロセスの概要

サプライヤ：在庫チェック・サブプロセスのプロパティを表示するには、そのプロセス・アクティビティをナビゲータ・ツリーで選択し、「編集」メニューから「プロパティ」を選択します。このプロセス・アクティビティは、実行可能ではありません。つまり、最上位レベルのプロセスとして開始できず、上位レベルのプロセスからコールする必要があります。

サプライヤ：在庫チェック・サブプロセスの「プロセス」ウィンドウを表示すると、サブプロセスが3つの異なるアクティビティから構成されていることがわかります。プロセスのアクティビティを詳しく説明するために、次のように各ノードに番号を付けて参照しやすくしてあります。番号自体は、プロセス・ダイアグラムには含まれません。



このサブプロセスは、ノード1の「開始」アクティビティから開始します。ノード2で、在庫チェックを実行して、オーダー項目の在庫を確認します。この時点でサブプロセスは終了します。

サプライヤ:在庫チェック・サブプロセスのアクティビティ

ここでは、サブプロセスの各アクティビティについて、アクティビティの表示名別に説明します。

開始（ノード1）

この「標準」関数アクティビティは、プロセスの開始をマークします。

関数 WF_STANDARD.NOOP

結果タイプ なし

前提条件アクティビティ なし

在庫チェック（ノード2）

現在、このアクティビティでは何も処理されません。このアクティビティに対して、在庫チェックを実行する関数を統合できます。

関数 WF_EVENTDEMO.STOCKCHECK

結果タイプ なし

前提条件アクティビティ 開始

終了（ノード3）

この「標準」関数アクティビティは、プロセスの終了をマークします。

関数 WF_STANDARD.NOOP

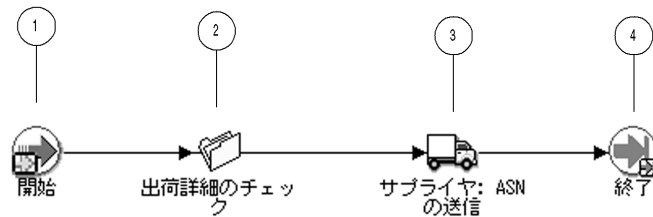
結果タイプ なし

前提条件アクティビティ 在庫チェック

サプライヤ: アドバンスト出荷通知サブプロセスの概要

サプライヤ: アドバンスト出荷通知サブプロセスのプロパティを表示するには、そのプロセス・アクティビティをナビゲータ・ツリーで選択し、「編集」メニューから「プロパティ」を選択します。このプロセス・アクティビティは、実行可能ではありません。つまり、最上位レベルのプロセスとして開始できず、上位レベルのプロセスからコールする必要があります。

サプライヤ: アドバンスト出荷通知サブプロセスの「プロセス」ウィンドウを表示すると、サブプロセスが4つの異なるアクティビティから構成されていることがわかります。プロセスのアクティビティを詳しく説明するために、次のように各ノードに番号を付けて参照しやすくしてあります。番号自体は、プロセス・ダイアグラムには含まれません。



このサブプロセスは、ノード1の「開始」アクティビティから開始します。ノード2で、発注の出荷詳細をチェックします。次に、アドバンスト出荷通知をバイヤーに送信します。この時点でサブプロセスは終了します。

サプライヤ: アドバンスト出荷通知サブプロセスのアクティビティ

ここでは、サブプロセスの各アクティビティについて、アクティビティの表示名別に説明します。

開始 (ノード 1)

この「標準」関数アクティビティは、プロセスの開始をマークします。

関数	WF_STANDARD.NOOP
結果タイプ	なし
前提条件アクティビティ	なし

出荷詳細のチェック（ノード2）

現在、このアクティビティでは何も処理されません。このアクティビティに対して、発注の出荷詳細をチェックする関数を統合できます。

関数	WF_EVENTDEMO.CHECKSHIPPING
結果タイプ	なし
前提条件アクティビティ	開始

サプライヤ:ASNの送信（ノード3）

このイベント・アクティビティは、サプライヤ・システム上のアウトバウンド・エージェントおよびバイヤー・システム上のインバウンド・エージェントを使用して、アドバンスト出荷通知・メッセージをバイヤーに送信します。これらのエージェントは、サプライヤ:オーダー詳細の取得サブプロセスの「エージェント詳細の取得」アクティビティによって識別されます。

イベント・アクション	送信
前提条件アクティビティ	出荷詳細のチェック
アクティビティによって取得される項目属性	イベント・メッセージ、送信元エージェント / システム2、宛先エージェント / システム1

終了（ノード4）

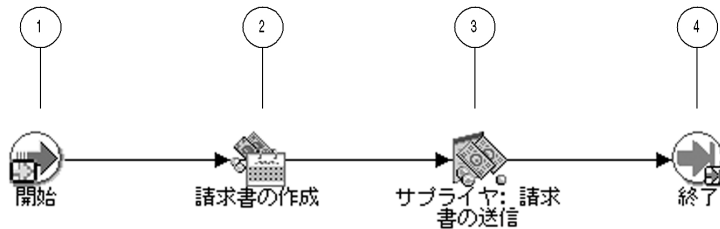
この「標準」関数アクティビティは、プロセスの終了をマークします。

関数	WF_STANDARD.NOOP
結果タイプ	なし
前提条件アクティビティ	サプライヤ:ASNの送信

サプライヤ: サプライヤの請求書の送信サブプロセスの概要

サプライヤ: サプライヤの請求書の送信サブプロセスのプロパティを表示するには、そのプロセス・アクティビティをナビゲータ・ツリーで選択し、「編集」メニューから「プロパティ」を選択します。このプロセス・アクティビティは、実行可能ではありません。つまり、最上位レベルのプロセスとして開始できず、上位レベルのプロセスからコールする必要があります。

サプライヤ: サプライヤの請求書の送信サブプロセスの「プロセス」ウィンドウを表示すると、サブプロセスが4つの異なるアクティビティから構成されていることがわかります。プロセスのアクティビティを詳しく説明するために、次のように各ノードに番号を付けて参照しやすくしてあります。番号自体は、プロセス・ダイアグラムには含まれません。



このサブプロセスは、ノード1の「開始」アクティビティから開始します。ノード2で、発注のオーダー項目について請求書を作成します。次に、請求書をバイヤーに送信します。この時点でサブプロセスは終了します。

サプライヤ: サプライヤの請求書の送信サブプロセスのアクティビティ

ここでは、サブプロセスの各アクティビティについて、アクティビティの表示名別に説明します。

開始 (ノード 1)

この「標準」関数アクティビティは、プロセスの開始をマークします。

関数	WF_STANDARD.NOOP
結果タイプ	なし
前提条件アクティビティ	なし

請求書の作成 (ノード 2)

現在、このアクティビティでは何も処理されません。このアクティビティに対して、発注のオーダー項目について請求書を作成する関数を統合できます。

関数	WF_EVENTDEMO.CREATEINVOICE
結果タイプ	なし
前提条件アクティビティ	開始

サプライヤ: 請求書の送信 (ノード 3)

このイベント・アクティビティは、サプライヤ・システム上のアウトバウンド・エージェントおよびバイヤー・システム上のインバウンド・エージェントを使用して、請求書イベント・メッセージをバイヤーに送信します。これらのエージェントは、サプライヤ: オーダー詳細の取得サブプロセスの「エージェント詳細の取得」アクティビティによって識別されます。

イベント・アクション 送信

前提条件アクティビティ 請求書の作成

アクティビティによって取得される項目属性 イベント・メッセージ、イベント・キー、送信元エージェント / システム 2、宛先エージェント / システム 1

終了 (ノード 4)

この「標準」関数アクティビティは、プロセスの終了をマークします。

関数 WF_STANDARD.NOOP

結果タイプ なし

前提条件アクティビティ サプライヤ: 請求書の送信

発注イベント

このイベントは、「Buyer Workbench」デモンストレーション・ページから発注を送信したときに、バイヤー・システム上で呼び出されます。

内部名 demo.oracle.wf.b2b.po.create

ジェネレート関数 wf_eventdemo.generatexml

所有者名 Oracle Workflow

所有者タグ FND

カスタマイズ・レベル 制限

Oracle Workflow には、発注イベントに対して、4つのデフォルト・サブスクリプションが用意されています。最初のサブスクリプションでは、発注イベントがローカルで呼び出されたときに、関連 ID がイベント・メッセージに追加されます。関連 ID は、接頭辞 PO とイベント・キー (オーダー番号) で構成されます。このサブスクリプションは、デフォルトで使用可能になっています。次の表は、このサブスクリプションに対して定義されているプロパティを示しています。

表 10-6

サブスクリプションのプロパティ	値
システム	<local system>
ソース・タイプ	ローカル
イベント・フィルタ	demo.oracle.wf.b2b.po.create
フェーズ	1
ステータス	使用可能
ルール・データ	キー
ルール関数	wf_eventdemo.derivecorrelationid
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

2 番目のサブスクリプションでは、発注イベントがローカルで呼び出されたときに、「イベント・システムのデモ」項目タイプのバイヤー:最上位の発注プロセスにイベント・メッセージが送信されます。このサブスクリプションは、デフォルトで使用可能になっています。次の表は、このサブスクリプションに対して定義されているプロパティを示しています。

表 10-7

サブスクリプションのプロパティ	値
システム	<local system>
ソース・タイプ	ローカル
イベント・フィルタ	demo.oracle.wf.b2b.po.create
フェーズ	2
ステータス	使用可能
ルール・データ	メッセージ
ルール関数	wf_rule.workflow_protocol
ワークフロー項目タイプ	WFEVDEME
ワークフロー・プロセス名	RCVPOCRT
送信エージェント	WF_OUT@<local system>
宛先エージェント	WF_IN@<local system>

表 10-7 (続き)

サブスクリプションのプロパティ	値
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

3 番目のサブスクリプションでは、発注イベントが外部ソースから着信したときに、イベント・メッセージ内の相関 ID が変更されます。相関 ID は、接頭辞 so とイベント・キー（オーダー番号）で構成されます。このサブスクリプションは、デフォルトで使用可能になっています。次の表は、このサブスクリプションに対して定義されているプロパティを示しています。

表 10-8

サブスクリプションのプロパティ	値
システム	<local system>
ソース・タイプ	外部
イベント・フィルタ	demo.oracle.wf.b2b.po.create
フェーズ	1
ステータス	使用可能
ルール・データ	キー
ルール関数	wf_eventdemo.derivecorrelationid
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

4 番目のサブスクリプションでは、発注イベントが外部ソースから呼び出されたときに、「イベント・システムのデモ」項目タイプのサブライヤ:最上位のオーダー・プロセスにイベント・メッセージが送信されます。このサブスクリプションは、デフォルトで使用可能になっています。次の表は、このサブスクリプションに対して定義されているプロパティを示しています。

表 10-9

サブスクリプションのプロパティ	値
システム	<local system>

表 10-9 (続き)

サブスクリプションのプロパティ	値
ソース・タイプ	外部
イベント・フィルタ	demo.oracle.wf.b2b.po.create
フェーズ	2
ステータス	使用可能
ルール・データ	キー
ルール関数	wf_rule.workflow_protocol
ワークフロー項目タイプ	WFEVDEME
ワークフロー・プロセス名	SUPPORCV
送信エージェント	WF_OUT@<local system>
宛先エージェント	WF_IN@<local system>
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

発注承認イベント

このイベントは、発注イベントが受信および処理された後で、サプライヤ・システムからバイヤー・システムに発注承認として送信されます。

内部名	demo.oracle.wf.b2b.po.ack
ジェネレート関数	なし
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

Oracle Workflow には、発注承認イベントに対して、2つのデフォルト・サブスクリプションが用意されています。最初のサブスクリプションでは、発注承認イベントが外部ソースから着信したときに、**相関 ID** がイベント・メッセージに追加されます。相関 ID は、接頭辞 **PO** とイベント・キー (オーダー番号) で構成されます。このサブスクリプションは、デフォルトで使用可能になっています。次の表は、このサブスクリプションに対して定義されているプロパティを示しています。

表 10-10

サブスクリプションのプロパティ	値
システム	<local system>
ソース・タイプ	外部
イベント・フィルタ	demo.oracle.wf.b2b.po.ack
フェーズ	1
ステータス	使用可能
ルール・データ	キー
ルール関数	wf_eventdemo.derivecorrelationid
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

2 番目のサブスクリプションでは、発注承認イベントが外部ソースから呼び出されたときに、「イベント・システムのデモ」項目タイプのバイヤー：最上位の発注プロセスにイベント・メッセージが送信されます。このサブスクリプションは、デフォルトで使用可能になっています。次の表は、このサブスクリプションに対して定義されているプロパティを示しています。

表 10-11

サブスクリプションのプロパティ	値
システム	<local system>
ソース・タイプ	外部
イベント・フィルタ	demo.oracle.wf.b2b.po.ack
フェーズ	2
ステータス	使用可能
ルール・データ	キー
ワークフロー項目タイプ	WFEVDEME
ワークフロー・プロセス名	RCVPOCRT
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

アドバンスト出荷通知イベント

このイベントは、発注イベントが受信および処理された後で、サプライヤ・システムからバイヤー・システムにアドバンスト出荷通知として送信されます。

内部名	demo.oracle.wf.b2b.po.asn
ジェネレート関数	なし
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

Oracle Workflow には、アドバンスト出荷通知イベントに対して、2つのデフォルト・サブスクリプションが用意されています。最初のサブスクリプションでは、アドバンスト出荷通知イベントが外部ソースから着信したときに、**相関 ID** がイベント・メッセージに追加されます。相関 ID は、接頭辞 **PO** とイベント・キー（オーダー番号）で構成されます。このサブスクリプションは、デフォルトで使用可能になっています。次の表は、このサブスクリプションに対して定義されているプロパティを示しています。

表 10-12

サブスクリプションのプロパティ	値
システム	<local system>
ソース・タイプ	外部
イベント・フィルタ	demo.oracle.wf.b2b.po.asn
フェーズ	1
ステータス	使用可能
ルール・データ	キー
ルール関数	wf_eventdemo.derivecorrelationid
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

2番目のサブスクリプションでは、アドバンスト出荷通知イベントが外部ソースから呼び出されたときに、「イベント・システムのデモ」項目タイプのバイヤー：最上位の発注プロセスにイベント・メッセージが送信されます。このサブスクリプションは、デフォルトで使用可能になっています。次の表は、このサブスクリプションに対して定義されているプロパティを示しています。

表 10-13

サブスクリプションのプロパティ	値
システム	<local system>
ソース・タイプ	外部
イベント・フィルタ	demo.oracle.wf.b2b.po.asn
フェーズ	2
ステータス	使用可能
ルール・データ	キー
ワークフロー項目タイプ	WFEVDEME
ワークフロー・プロセス名	RCVPOCRT
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

請求書イベント

このイベントは、発注イベントが受信および処理された後で、サプライヤ・システムからバイヤー・システムに請求書として送信されます。

内部名	demo.oracle.wf.b2b.po.invoice
ジェネレート関数	なし
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

Oracle Workflow には、請求書イベントに対して、2つのデフォルト・サブスクリプションが用意されています。最初のサブスクリプションでは、請求書イベントが外部ソースから着信したときに、関連 ID がイベント・メッセージに追加されます。関連 ID は、接頭辞 PO と イベント・キー（オーダー番号）で構成されます。このサブスクリプションは、デフォルトで使用可能になっています。次の表は、このサブスクリプションに対して定義されているプロパティを示しています。

表 10-14

サブスクリプションのプロパティ	値
システム	<local system>

表 10-14 (続き)

サブスクリプションのプロパティ	値
ソース・タイプ	外部
イベント・フィルタ	demo.oracle.wf.b2b.po.invoice
フェーズ	1
ステータス	使用可能
ルール・データ	キー
ルール関数	wf_eventdemo.derivecorrelationid
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

2 番目のサブスクリプションでは、請求書イベントが外部ソースから着信したときに、「イベント・システムのデモ」項目タイプのバイヤー:最上位の発注プロセスにイベント・メッセージが送信されます。このサブスクリプションは、デフォルトで使用可能になっています。次の表は、このサブスクリプションに対して定義されているプロパティを示しています。

表 10-15

サブスクリプションのプロパティ	値
システム	<local system>
ソース・タイプ	外部
イベント・フィルタ	demo.oracle.wf.b2b.po.invoice
フェーズ	2
ステータス	使用可能
ルール・データ	キー
ワークフロー項目タイプ	WFEVDEME
ワークフロー・プロセス名	RCVPOCRT
所有者名	Oracle Workflow
所有者タグ	FND
カスタマイズ・レベル	制限

11

エラー処理

この章では、Oracle Workflow のワークフロー・プロセスおよびイベント・サブスクリプション処理でのエラー処理方法について説明します。

エラー処理

Oracle Workflow には、ワークフロー・プロセスとイベント・サブスクリプション処理の両方に対するデフォルトのエラー処理が用意されています。デフォルトの処理は、「システム:エラー」と呼ばれる特殊な項目タイプに用意されているエラー・プロセスによって定義されています。また、イベントの場合は、「エラー」ソース・タイプの **Unexpected** イベントに対する特殊なサブスクリプションによって定義されています。さらに、カスタム・エラー・プロセスとサブスクリプションを作成することで、独自のカスタム・エラー・プロセスを定義することもできます。

ワークフロー・プロセスのエラー処理

通常、コール元ではエラーに応答する方法がわからないため、ワークフロー実行中に発生したエラーをコール元に直接戻すことはできません。実際には、コール元は担当のオペレータがいないバックグラウンド・エンジンである場合があります。そのかわり **Oracle Workflow** では、**Oracle Workflow Builder** でワークフロー・プロセスを作成するときにエラー処理プロセスを指定して、エラー発生時の処理を定義できるようになっています。

ワークフローの設計時にアクティビティの「詳細」プロパティ・ページで、プロセス、関数、またはイベント・アクティビティにエラー処理プロセスを割り当てることができます。エラー処理プロセスが定義されている項目タイプとエラー処理プロセス自体について、内部名を指定する必要があります。3-70 ページの「[オプションのアクティビティ詳細の定義](#)」を参照してください。

The image shows a dialog box titled "Navigator Control Properties" with a close button (X) in the top right corner. The dialog has four tabs: "Activity", "Details", "Roles", and "Access". The "Details" tab is selected. Inside the dialog, there are several input fields and a dropdown menu:

- Error Item Type:** A text box containing "WFERROR".
- Error Process:** A text box containing "DEFAULT_ERROR".
- Effective:** An empty text box.
- On Revisit:** A dropdown menu with "Reset" selected.
- Version:** A text box containing "0".

At the bottom of the dialog, there are four buttons: "OK", "Cancel", "Apply", and "Help".

Oracle Workflow には、「システム:エラー」と呼ばれる特別な項目タイプが用意されています。この項目タイプには、任意のプロセスで汎用的なエラー処理に使用できるデフォルト・エラー・プロセスと「再試行のみ」プロセスという、エラー・プロセスが含まれています。ただし、「システム:エラー」項目タイプのエラー・プロセスは変更できません。これらのエラー・プロセスで使用できない機能を組み込むには、各自の項目タイプにカスタム・エラー処理プロセスを作成する必要があります。エラー・プロセスは、エラー・コードに基づく分岐、通知の送信、失敗したアクティビティのリセット、再試行またはスキップなどを自動化するルールを使用した、エラー処理が可能です。

注意: 特定のビジネス・ルールの非互換性によるエラーを処理する場合は、エラー・プロセスを利用するのではなく、エラーの状況をワークフロー・プロセス定義にモデル化する必要があります。たとえば、ビジネス・プロセスの前提条件が満たされないために関数アクティビティにエラーが発生する可能性がある場合は、そのような状況が発生したときに該当するルールに通知を送信して状況を修正するプロセスをモデル化して、ワークフロー・プロセスが先に進めるようにします。この状況をワークフロー・プロセスにモデル化せず、エラー・プロセスをアクティブにするエラーを利用すると、ワークフロー・プロセス全体のステータスが「エラー」になり、ワークフロー管理者がそのエラーを処理するまで停止します。

実行時に、ワークフロー・エンジンでは各関数アクティビティの前にセーブポイントを設定し、関数アクティビティによって生成されるエラーが検出されるようにしています。アクティビティで、処理されない例外が生成された場合、ワークフロー・エンジンはセーブポイントまでロールバックし、アクティビティのステータスを **ERROR** に設定します。

注意: このため、関数アクティビティの PL/SQL プロシージャでは、ユーザーがコミットすることはありません。コミットは、コール側のアプリケーションが行うため、ワークフロー・エンジンではコミットを発行しません。

その後、ワークフロー・エンジンは、エラーが発生したアクティビティを実行してエラー・プロセスに進み、対応するエラー・プロセスが見つかるまで各親プロセス・アクティビティをチェックします。ワークフロー・エンジンがエラー・プロセスを検出すると、エラーを処理するためにそのプロセスを起動します。実行中のプロセスにエラー・プロセスが割り当てられていない場合は、ワークフロー・エンジンがセーブポイントまでロールバックし、アクティビティのステータスを **ERROR** に設定しますが、エラー・プロセスは起動されません。

注意: 起動されるエラー・プロセスに対して、エラー項目タイプとエラー・プロセスの両方を指定する必要があります。

項目タイプのエラー通知のカスタマイズ

デフォルト・エラー・プロセスと「再試行のみ」プロセスの両方によって、管理者にエラーの発生を知らせる通知が送信されます。独自のプロセス・アクティビティにこれらのプロセスの1つをエラー・プロセスとして割り当てる場合は、独自の項目タイプに定義した項目タイプ属性を使用して、エラー通知の動作を制御することができます。このようにすることで、事前定義済みのエラー・プロセスを変更しないで、エラー処理をカスタマイズすることができます。

デフォルトでは、デフォルト・エラー・プロセスと「再試行のみ」プロセスによって、エラー通知が SYSADMIN ロールに送信されます。項目タイプ内に WF_ADMINISTRATOR という名前の項目タイプ属性を作成し、通知を送信するロールにこの属性の値を設定することで、エラー通知の受信者として別のロールを指定することができます。項目タイプに WF_ADMINISTRATOR 属性が含まれていると、デフォルト・エラー・プロセスおよび「再試行のみ」プロセスは、デフォルトを無視して、この属性に指定されたロールへエラー通知を送信します。

たとえば、購買承認申請ワークフローがあり、このワークフローで起きた問題の解決は、システム管理者ではなく、購買管理者が行うとします。購買承認申請ワークフローを所有する項目タイプに、項目属性 WF_ADMINISTRATOR を定義し、WF_ADMINISTRATOR 属性を PO_ADMIN などの購買管理者ロールに設定できます。

さらに、独自の項目タイプ内の項目タイプ属性を使用して、デフォルト・エラー・プロセスまたは「再試行のみ」プロセス内のエラー通知のアクティビティがタイムアウトになるかどうかを制御することができます。ERROR_TIMEOUT という名前の数値タイプの項目タイプ属性を作成し、この属性の値を希望するタイムアウト間隔（分）に設定します。ワークフロー・エンジンは、この属性のタイムアウト値を、アクティビティの開始日からの相対オフセットとして解釈します。ERROR_TIMEOUT 属性に、NULL 値または値 0（ゼロ）が入っている場合や、まったく定義されていない場合は、エラー通知アクティビティにタイムアウトはありません。

関連項目：

11-7 ページ 「システム:エラー」項目タイプと項目属性」

11-8 ページ 「デフォルト・エラー・プロセス」

11-12 ページ 「再試行のみ」プロセス」

イベント・サブスクリプション処理のエラー処理

イベント・マネージャでは、イベント・サブスクリプションのルール関数によって返されるステータス・コードを使用して、イベントのサブスクリプション処理のステータスを監視します。ルール関数から「WARNING」または「ERROR」のステータス・コードが返され、サブスクリプションの処理中に警告状態またはエラーが発生したことを示している場合、イベント・マネージャはイベント・メッセージを WF_ERROR エージェントに関連付けられている標準 WF_ERROR キューに格納します。「WARNING」ステータスの場合は、イベントのサブスクリプション処理が続行されます。「ERROR」ステータスの場合は、イベントのサブスクリプション処理が中止され、イベントに対して実行されたすべてのサブスクリプションがロールバックされます。

注意： エラーの場合、イベントを再試行することによってシステム管理者がエラー処理することができるように、適切であれば、すべてのサブスクリプション処理がロールバックされます。この方法によって、Oracle Workflow では、サブスクリプション処理が再び開始されたときにどのサブスクリプションも二重に処理されないことが保証されます。

注意： ルール関数で例外が発生した場合は、イベント・マネージャによってイベントへのすべてのサブスクリプションがロールバックされ、コール側アプリケーションにエラーが呼び出されます。この場合、イベント・メッセージは WF_ERROR キューに格納されません。

WF_ERROR エージェントは、Oracle Workflow をインストールするとローカル・システムに自動的に定義される、エラー処理のための標準エージェントです。WF_ERROR エージェントを監視するには、リスナーをスケジュールする必要があります。このリスナーによって WF_ERROR キューのイベント・メッセージがデキューされると、そのメッセージに対して「エラー」ソース・タイプが割り当てられます。イベント・マネージャは、「エラー」ソース・タイプを持つそのイベントまたは Any イベントへのサブスクリプションを、ローカル・システム単位に検索および実行します。サブスクリプションが見つからない場合は、「エラー」ソース・タイプの Unexpected イベントへのサブスクリプションがローカル・システム単位に実行されます。

Oracle Workflow には、「エラー」ソース・タイプの Unexpected イベントに対して、1つの事前定義済サブスクリプションが用意されています。カスタム・エラー・サブスクリプションをイベントに定義していない場合は、このサブスクリプションによってデフォルトのエラー処理が実行されます。イベント・メッセージは、「システム:エラー」項目タイプの「デフォルト・イベント・エラー」プロセスに送信されます。

注意： Unexpected イベントまたはそのイベントへの事前定義済のエラー・サブスクリプションの定義は、変更したり使用不可にしないでください。これらを変更したり使用不可にすると、イベント・マネージャはイベントおよびサブスクリプション処理に対してデフォルトのエラー処理を実行できなくなります。

「デフォルト・イベント・エラー」プロセスでは、システム管理者に通知が送信されます。警告状態の場合、応答する必要はありません。エラーの場合、システム管理者はイベント・サブスクリプション処理を強制終了または再試行することができます。

特定のイベントに対してカスタム・エラー処理を設定するには、「エラー」ソース・タイプのイベントへのサブスクリプションを定義し、サブスクリプション・アクションとして実行するカスタム処理を指定します。この場合、このエラー・イベントは Unexpected イベントでなくなるため、イベント・マネージャはデフォルトのエラー処理を実行しません。デフォルトのエラー処理のかわりに、設定したカスタム・エラー処理が実行されます。

Unexpected イベント

Oracle Workflow でも、予期しないイベントの処理にデフォルト・イベント・エラー・プロセスが使用されます。外部ソースからイベントが受信されたときにローカル・システムにそのイベントに対するサブスクリプションがない場合は、イベント・マネージャによって、「外部」ソース・タイプの Unexpected イベントに対するサブスクリプションが自動的に検索されます。Oracle Workflow には、イベント・メッセージをデフォルト・イベント・エラー・プロセスに送信する、Unexpected イベント用に事前定義済の「外部」サブスクリプションが用意されています。

デフォルト・イベント・エラー・プロセスからシステム管理者に Unexpected イベントが通知されます。通知を受けたシステム管理者は、イベント・サブスクリプション処理を中止または再試行することができます。たとえば、イベントを処理するサブスクリプションを作成してから、イベント・メッセージを再度エンキューすると、新しいサブスクリプションをトリガーすることができます。

注意： Oracle Workflow には、ローカルで呼び出されたイベントに対するサブスクリプションがないときにイベント・メッセージをデフォルト・イベント・エラー・プロセスに送信する、Unexpected イベント用の事前定義済「ローカル」サブスクリプションが用意されています。ただしこのサブスクリプションは、サブスクリプションを必要としない多くのローカル・イベントが呼び出される可能性があるため、デフォルトで無効になっています。このサブスクリプションを使用可能にする場合は、ローカル・システム上で呼び出されるすべてのイベントを考慮してからトリガーしてください。

関連項目：

9-18 ページ「Unexpected イベント」

11-15 ページ「デフォルト・イベント・エラー・プロセス」

「システム:エラー」項目タイプと項目属性

「システム:エラー」項目タイプの詳細を参照するには、「ファイル」メニューから「オープン」を選択してからデータベースに接続し、「システム:エラー」項目タイプを選択するか、`<drive>:¥<ORACLE_HOME>¥wf¥Data¥<Language>` サブディレクトリにあるファイル `wferror.wft` に接続します。

「システム:エラー」項目タイプには、次の項目属性があります。

- エラー・アクティビティ ID
- エラー・アクティビティ・ラベル
- エラー割当済ユーザー
- エラー項目タイプ
- エラー項目キー
- エラー・ユーザー・キー
- エラー・メッセージ
- エラー名
- エラー通知 ID
- エラー結果コード
- エラー・スタック
- エラー・モニター URL
- タイムアウト値
- イベント名
- イベントの詳細
- イベント・メッセージ
- イベント・キー
- イベント・データの URL
- イベント・サブスクリプション
- エラー・タイプ

これらの項目属性は、「デフォルト・エラー・プロセス」、「再試行のみ」および「デフォルト・イベント・エラー・プロセス」というエラー・プロセスを構成する関数、通知およびイベント・アクティビティによって参照されます。

注意： 独自の項目タイプにカスタムのエラー処理プロセスを作成すると、Oracle Workflow では、エラー処理プロセスをコールするときに、前述の項目属性が自動的に設定されます。これらの項目属性は、プロセスになれば Oracle Workflow によって作成されます。ただし、メッセージなど独自のエラー処理プロセスでこれらの項目属性を参照する場合は、最初に Oracle Workflow Builder を使用して、これらの項目属性をプロセスの項目タイプの項目属性として作成する必要があります。

デフォルト・エラー・プロセス

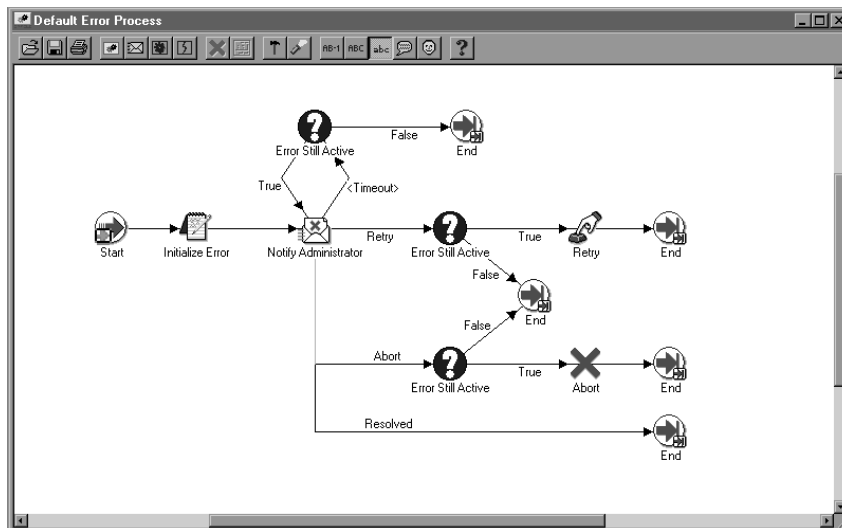
DEFAULT_ERROR は、「デフォルト・エラー・プロセス」の内部名です。このエラー処理プロセスの目的は、次のとおりです。

- プロセスにエラーが発生した場合に、管理者に通知を送ります。
- 管理者にエラーに関する情報を提供します。
- 管理者によるプロセスの中止、エラーになったアクティビティの再試行、またはエラーの原因となった問題の解決を可能にします。

エラーがアクティブでなくなると、プロセスは自動的に終了します。

「デフォルト・エラー・プロセス」は編集できませんが、このプロセスには柔軟性があり、その動作をカスタマイズできます。「デフォルト・エラー・プロセス」をコールする項目タイプ内に WF_ADMINISTRATOR および ERROR_TIMEOUT という 2 つの項目タイプ属性を定義して、実行されるエラー処理を制御することができます。

- WF_ADMINISTRATOR: Oracle Workflow がエラー通知を送信するロールを指定します。デフォルトは SYSADMIN ロールです。
- ERROR_TIMEOUT: エラー通知がタイムアウトするかどうかを指定します。



「エラー・プロセスの初期化」関数アクティビティ

「エラー・プロセスの初期化」アクティビティは、WF_STANDARD.INITIALIZEERRORS という名前の PL/SQL プロシージャをコールします。このプロシージャは、エラーになったプロセスの項目タイプに、内部名 WF_ADMINISTRATOR で項目タイプ属性の定義があるかどうかを判別します。定義があれば、後続の通知アクティビティの実行者である「管理者へ通知」を、WF_ADMINISTRATOR に格納されているロールに設定します。定義がなければ、後続の通知アクティビティは、デフォルト実行者である「システム管理者」に設定されたままになります。

「エラー・プロセスの初期化」アクティビティでは、エラーになったプロセスの項目タイプの項目属性 WF_ADMINISTRATOR がチェックされるため、エラー・プロセスを変更せずに、特定のプロセスでエラーが起きた場合の通知の送信先を指定できます。

「管理者へ通知」通知アクティビティ

「管理者へ通知」アクティビティは、「デフォルトの再試行エラー」メッセージを、実行者（システム管理者または項目タイプの WF_ADMINISTRATOR 項目属性に格納されているロール）に送ります。このメッセージは、指定のプロセスでエラーが起きたことと、応答が必要であることを示します。応答オプションとその結果の処理は、次のとおりです。

- プロセスの異常終了：「未解決のエラー」アクティビティを実行し、エラーがまだ存在するかをチェックします。存在する場合は、「プロセスを中止」関数アクティビティをコールして、「デフォルト・エラー・プロセス」は終了します。

- プロセスの再試行：「未解決のエラー」アクティビティを実行し、エラーがまだ存在するかを検証します。存在する場合は、「再試行」関数アクティビティをコールして、「デフォルト・エラー・プロセス」を終了します。
- プロセスの解決： なんらかの外部的手段またはワークフロー・モニターへの埋込み URL リンクを使用して、エラーになったプロセスが直接処理されたため、「デフォルト・エラー・プロセス」を終了します。

注意： 通知メッセージに埋め込まれたモニター URL により、エラーのプロセスがすべての管理者権限付きでワークフロー・モニターに表示されます。プロセスの一部の再試行、スキップまたはロールバックなどの処理を行って、エラーを解決できます。

「デフォルトの再試行エラー」メッセージの件名と本文は次のとおりです。

Subject: Error in Workflow &ERROR_ITEM_TYPE/&ERROR_ITEM_KEY &ERROR_MESSAGE

Body: An Error occurred in the following Workflow.

```
Item Type = &ERROR_ITEM_TYPE
Item Key = &ERROR_ITEM_KEY
User Key = &ERROR_USER_KEY

Error Name = &ERROR_NAME
Error Message = &ERROR_MESSAGE
Error Stack = &ERROR_STACK

Activity Id = &ERROR_ACTIVITY_ID
Activity Label = &ERROR_ACTIVITY_LABEL
Result Code = &ERROR_RESULT_CODE
Notification Id = &ERROR_NOTIFICATION_ID
Assigned User = &ERROR_ASSIGNED_USER

&MONITOR
```

「管理者へ通知」通知アクティビティには、動的タイムアウト値が割り当てられています。エラーの起きたプロセスの項目タイプをチェックして、内部名が `ERROR_TIMEOUT` である項目タイプ属性を検索します。`ERROR_TIMEOUT` は、数値型の属性です。ワークフロー・エンジンは、この属性値をアクティビティの開始日からの相対オフセットとして解釈し、「管理者へ通知」のタイムアウト値を分単位で決定します。`ERROR_TIMEOUT` に、NULL 値または値 0（ゼロ）が入っている場合や、まったく定義されていない場合は、「管理者へ通知」にタイムアウトはありません。

「未解決のエラー」関数アクティビティ

「管理者へ通知」アクティビティがタイムアウトになった場合や、結果として異常終了または再試行を戻した場合、ワークフロー・エンジンは「未解決のエラー」関数アクティビティを開始します。

「未解決のエラー」アクティビティは、WF_STANDARD.CHECKERRORACTIVE と呼ばれる PL/SQL プロシージャをコールします。「未解決のエラー」アクティビティの目的は、エラー処理を続ける前に、エラーの起きたプロセスが、まだエラー状態にあるかどうかを判別することです。エラー状態にある場合、「未解決のエラー」は TRUE を戻し、ワークフロー・エンジンは、他の通知の送信、エラーの起きたプロセスの中止または再試行のうち、いずれか適切なトランジションを取ります。エラーの起きたプロセスがエラー状態になれば、このアクティビティは FALSE を戻し、プロセス・ダイアグラムで示されたとおり、エラー処理プロセスは終了します。

「再試行」関数アクティビティ

「再試行」関数アクティビティは、WF_STANDARD.RESETERROR と呼ばれる PL/SQL プロシージャを実行し、エラーになったアクティビティを消去して再実行します。このプロシージャは、WF_ENGINE.HandleError API をコールして、アクティビティを再実行します。

「プロセスを中止」関数アクティビティ

「プロセスを中止」関数アクティビティは、PL/SQL プロシージャ WF_STANDARD.ABORTPROCESS を実行し、続いて WF_ENGINE.AbortProcess API をコールして、エラーの起きたプロセスを中止します。

関連項目：

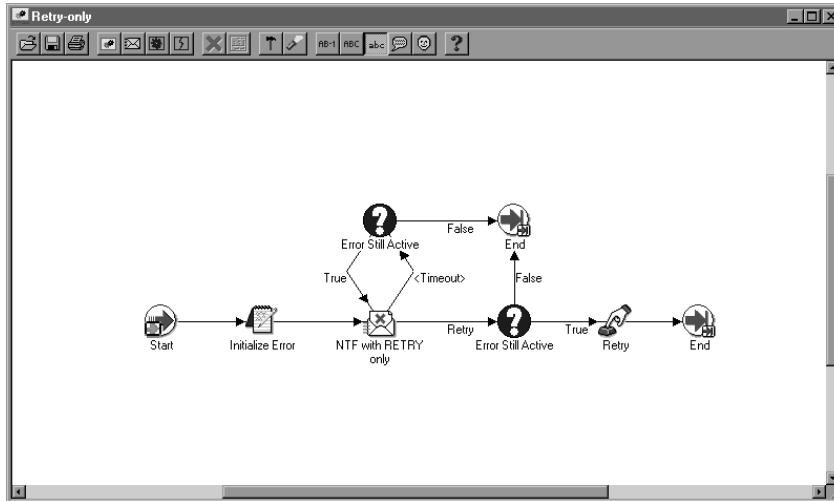
『Oracle Workflow API リファレンス』の「Workflow CORE API」

「再試行のみ」プロセス

「再試行のみ」エラー・プロセスの内部名は、RETRY_ONLYです。このエラー処理プロセスの目的は、プロセスにエラーが発生した場合に管理者に警告することと、エラーになったプロセスを再試行するよう管理者に要求することです。エラーがアクティブでなくなると、プロセスは自動的に終了します。

「再試行のみ」プロセスは編集できませんが、このプロセスには柔軟性があり、その動作をカスタマイズできます。「再試行のみ」プロセスをコールする項目タイプ内に WF_ADMINISTRATOR および ERROR_TIMEOUT という 2つの項目タイプ属性を定義して、実行されるエラー処理を制御することができます。

- WF_ADMINISTRATOR: Oracle Workflow がエラー通知を送信するロールを指定します。デフォルトは SYSADMIN ロールです。
- ERROR_TIMEOUT: エラー通知がタイムアウトするかどうかを指定します。



「エラー・プロセスの初期化」関数アクティビティ

「エラー・プロセスの初期化」アクティビティは、WF_STANDARD.INITIALIZEERRORS という名前の PL/SQL プロシージャをコールします。このプロシージャは、エラーになったプロセスの項目タイプに、内部名 WF_ADMINISTRATOR で項目タイプ属性の定義があるかどうかを判別します。定義があれば、後続の通知アクティビティである「エラー通知（再試行のみプロセス用）」の実行者を、WF_ADMINISTRATOR に格納されているロールに設定します。定義がなければ、後続の通知アクティビティは、デフォルト実行者である「システム管理者」に設定されたままになります。

「エラー・プロセスの初期化」アクティビティでは、エラーになったプロセスの項目タイプの項目属性 WF_ADMINISTRATOR がチェックされるため、エラー・プロセスを変更せずに、特定のプロセスでエラーが起きた場合の通知の送信先を指定できます。

「エラー通知（再試行のみプロセス用）」通知アクティビティ

「エラー通知（再試行のみプロセス用）」アクティビティは、「再試行（オプション）」メッセージを、実行者（システム管理者または項目タイプの WF_ADMINISTRATOR 項目属性に格納されているロール）に送ります。このメッセージは、指定のプロセスでエラーが起きたことを示し、管理者にエラーとなったアクティビティを再試行するように要求します。その後、エラー・プロセスは、「再試行」関数アクティビティに移り、「再試行のみ」エラー・プロセスを終了します。

注意： 通知メッセージに埋め込まれた URL リンクにより、エラーとなったプロセスがすべての管理者権限付きでワークフロー・モニターに表示されます。ワークフロー・モニターでプロセスの一部の再試行、スキップまたはロールバックなどの処理を行って、エラーを解決できます。

「再試行（オプション）」メッセージの件名と本文は次のとおりです。

Subject: Error in Workflow &ERROR_ITEM_TYPE/&ERROR_ITEM_KEY &ERROR_MESSAGE

Body: An Error occurred in the following Workflow.

```
Item Type = &ERROR_ITEM_TYPE
Item Key = &ERROR_ITEM_KEY
User Key = &ERROR_USER_KEY

Error Name = &ERROR_NAME
Error Message = &ERROR_MESSAGE
Error Stack = &ERROR_STACK

Activity Id = &ERROR_ACTIVITY_ID
Activity Label = &ERROR_ACTIVITY_LABEL
Result Code = &ERROR_RESULT_CODE
Notification Id = &ERROR_NOTIFICATION_ID
Assigned User = &ERROR_ASSIGNED_USER
```

&MONITOR

「エラー通知（再試行のみプロセス用）」通知アクティビティには、動的タイムアウト値が割り当てられています。エラーの起きたプロセスの項目タイプをチェックして、内部名が `ERROR_TIMEOUT` である項目属性を検索します。`ERROR_TIMEOUT` は、数値型の属性です。ワークフロー・エンジンは、この属性のタイムアウト値を、アクティビティの開始日からの分単位の相対オフセットとして解釈します。`ERROR_TIMEOUT` に、`NULL` 値または値 `0`（ゼロ）が入っている場合や、まったく定義されていない場合、「エラー通知（再試行のみプロセス用）」にタイムアウトはありません。

「未解決のエラー」関数アクティビティ

「エラー通知（再試行のみプロセス用）」アクティビティがタイムアウトになると、ワークフロー・エンジンは「未解決のエラー」関数アクティビティを開始します。

「未解決のエラー」アクティビティは、`WF_STANDARD.CHECKERRORACTIVE` と呼ばれる PL/SQL プロシージャをコールします。「未解決のエラー」アクティビティの目的は、エラー処理を続ける前に、エラーの起きたプロセスが、まだエラー状態にあるかどうかを判別することです。エラー状態にある場合、「未解決のエラー」は `TRUE` を戻し、ワークフロー・エンジンは「エラー通知（再試行のみプロセス用）」通知アクティビティに戻って、管理者に別の通知を送信します。エラーの起きたプロセスがエラー状態になれば、このアクティビティは `FALSE` を戻し、プロセス・ダイアグラムで示されたとおり、エラー処理プロセスは終了します。

「再試行」関数アクティビティ

「再試行」関数アクティビティは、`WF_STANDARD.RESETERROR` と呼ばれる PL/SQL プロシージャを実行し、エラーになったアクティビティを消去して再実行します。このプロシージャは、`WF_ENGINE.HandleError` API をコールして、アクティビティを再実行します。

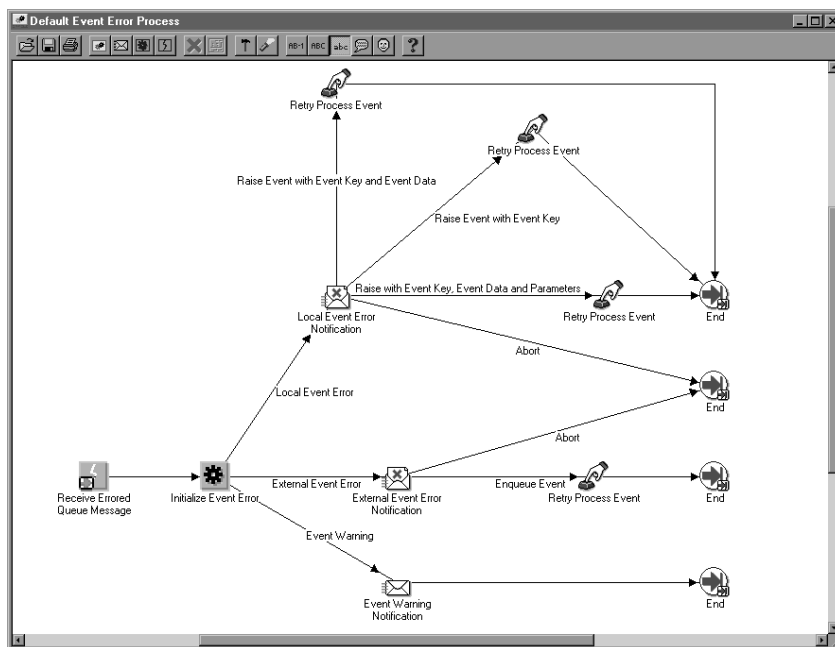
関連項目：

『Oracle Workflow API リファレンス』の「Workflow CORE API」

デフォルト・イベント・エラー・プロセス

DEFAULT_EVENT_ERROR は、ビジネス・イベント・システムのデフォルト・イベント・エラー・プロセスの内部名です。このエラー処理プロセスの目的は、次のとおりです。

- イベント・サブスクリプションを処理しているときにエラーまたは警告状態が発生すると、管理者に通知を送信します。
- 管理者にエラーに関する情報を提供します。
- 管理者がイベント・サブスクリプション処理を中止または再試行できるようにします。



エラー・キュー・メッセージの受信イベント・アクティビティ

エラー・キュー・メッセージの受信イベント・アクティビティでは、サブスクリプションを処理しているときに、エラーまたは警告状態が発生したイベント・メッセージを受信します。たとえば、サブスクリプションのルール関数から **ERROR** または **WARNING** ステータス・コードが返された場合、あるいは予期しないイベントが着信した場合は、イベント・マネージャによってデフォルトのサブスクリプションが実行され、イベント・メッセージがデフォルト・イベント・エラー・プロセスに送信されます。9-2 ページの「[事前定義済ワークフロー・イベント](#)」を参照してください。

エラー・キュー・メッセージの受信イベント・アクティビティでは、イベント名、イベント・キーおよびすべてのイベント・メッセージを項目タイプ属性に格納します。

「イベント・エラーの初期化」関数アクティビティ

「エラー・プロセスの初期化」アクティビティは、`WF_STANDARD.INITIALIZEEVENTERROR` という PL/SQL プロシージャをコールします。このプロシージャによってエラー・タイプが決まります。

- イベント警告： 警告状態が発生しましたが、サブスクリプション処理は続行しました。このエラー・タイプでは、ワークフロー・エンジンによってイベント警告通知が送信されます。
- 外部イベント・エラー： エラーが発生して、外部ソースから着信したイベントのサブスクリプション処理が中止されました。このエラー・タイプでは、ワークフロー・エンジンによって「外部イベント・エラー通知」が送信されます。
- ローカル・イベント・エラー： エラーが発生して、ローカル・システムで発生したイベントのサブスクリプション処理が中止されました。このエラー・タイプでは、ワークフロー・エンジンによって「ローカル・イベント・エラー通知」が送信されます。

「イベント警告通知」アクティビティ

ワークフロー・エンジンでは、エラーが発生したイベントのエラー・タイプが「イベント警告」であるときに、「イベント警告通知」アクティビティを開始します。このアクティビティは、「デフォルト・イベント警告」メッセージをシステム管理者に送信して、サブスクリプションを処理しているときに警告状態が発生したことを通知します。このメッセージは FYI メッセージであるため、応答する必要はありません。

「デフォルト・イベント警告」メッセージの件名と本文は次のとおりです。

Subject: Event WARNING : &EVENT_NAME / &EVENT_KEY

Body: A Warning occurred in the following Event

Subscription: &EVENT_SUBSCRIPTION

Event Error Name: &ERROR_NAME

Event Error Message: &ERROR_MESSAGE

Event Error Stack: &ERROR_STACK

Event Data: &EVENT_DATA_URL

Other Event Details: &EVENT_DETAILS

「外部イベント・エラー通知」アクティビティ

ワークフロー・エンジンでは、エラーが発生したイベントのエラー・タイプが「外部イベント・エラー」であるときに「外部イベント・エラー通知」アクティビティを開始します。このアクティビティは、「デフォルトの外部イベント・エラー」メッセージをシステム管理者に送信します。このメッセージは、外部ソースから着信したイベントのサブスクリプションを処理しているときにエラーが発生したことで、応答が必要であることを通知します。応答アクションとその結果の処理は、次のとおりです。

- プロセスを中止：サブスクリプション処理を中止し、デフォルト・イベント・エラー・プロセスを終了します。たとえば、イベント・メッセージに含まれているイベント・データが破損した場合、システム管理者はそのイベント・メッセージに対するサブスクリプション処理を中止できます。
- エンキュー・イベント：「プロセス・イベントの再試行」アクティビティを実行して、イベント・メッセージを送信したキューに戻し、デフォルト・イベント・エラー・プロセスを終了します。このイベント・メッセージは、-1の優先度でエンキューされるため、次にリスナーが動作するときに最初にデキューされます。

システム管理者は、エラーを訂正してからイベントを再度エンキューすることができます。たとえば、予期しないイベントを処理するサブスクリプションを作成してから、イベント・メッセージを再度エンキューすると、新しいサブスクリプションが処理されます。

「デフォルトの外部イベント・エラー」メッセージの件名と本文は次のとおりです。

Subject: External Event &ERROR_TYPE : &EVENT_NAME / &EVENT_KEY

Body: An Error occurred in the following Event

Subscription: &EVENT_SUBSCRIPTION

Event Error Name: &ERROR_NAME

Event Error Message: &ERROR_MESSAGE

Event Error Stack: &ERROR_STACK

Event Data: &EVENT_DATA_URL

Other Event Details: &EVENT_DETAILS

「ローカル・イベント・エラー通知」アクティビティ

ワークフロー・エンジンでは、エラーが発生したイベントのエラー・タイプが「ローカル・イベント・エラー」であるときに、「ローカル・イベント・エラー通知」アクティビティを開始します。このアクティビティは、「デフォルトのローカル・イベント・エラー」メッセージをシステム管理者に送信します。このメッセージは、ローカル・システムで発生したイベントのサブスクリプションを処理しているときにエラーが発生したことで、応答が必要であることを通知します。応答オプションとその結果の処理は、次のとおりです。

- プロセスを中止： サブスクリプション処理を中止し、デフォルト・イベント・エラー・プロセスを終了します。
- イベント・キーによるイベントの呼出し： 「プロセス・イベントの再試行」アクティビティを実行して、イベント名とイベント・キーだけでイベントを再度呼び出し、デフォルト・イベント・エラー・プロセスを終了します。
- イベント・キーとイベント・データによるイベントの呼出し： 「プロセス・イベントの再試行」アクティビティを実行して、イベント名、イベント・キーおよびイベント・データだけでイベントを再度呼び出し、デフォルト・イベント・エラー・プロセスを終了します。
- パラメータ、イベント・キーおよびイベント・データによるイベントの呼出し： 「プロセス・イベントの再試行」アクティビティを実行して、イベント名、イベント・キー、イベント・データおよびパラメータでイベントを再度呼び出し、デフォルト・イベント・エラー・プロセスを終了します。

システム管理者は、イベントを再度呼び出すときに、イベント・マネージャに提供する情報のレベルを選択できます。たとえば、最初に渡したイベント・データにエラーがある場合は、イベント名とイベント・キーだけでイベントを再度呼び出します。このとき、イベントのジェネレート関数を使用して、イベント・データを再生成することができます。

システム管理者は、エラーを訂正してからイベントを再度呼び出すこともできます。

「デフォルトのローカル・イベント・エラー」メッセージの件名と本文は次のとおりです。

Subject: Local Event &ERROR_TYPE : &EVENT_NAME / &EVENT_KEY

Body: An Error occurred in the following Event

Subscription: &EVENT_SUBSCRIPTION

Event Error Name: &ERROR_NAME

Event Error Message: &ERROR_MESSAGE

Event Error Stack: &ERROR_STACK

Event Data: &EVENT_DATA_URL

Other Event Details: &EVENT_DETAILS

「プロセス・イベントの再試行」関数アクティビティ

「プロセス・イベントの再試行」アクティビティは、PL/SQL プロシージャ `WF_STANDARD.RETRYRAISE` を実行します。このプロシージャでは、システム管理者が選択した通知応答に応じて、エラーが発生したイベントのエンキューまたは呼出しを再度行います。「プロセス・イベントの再試行」アクティビティを開始できる応答とそれに応じた処理は、次のとおりです。

- エンキュー・イベント： エラーが発生した外部イベント・メッセージを、送信元のキューに戻します。このイベント・メッセージは、-1 の優先度でエンキューされるため、次にリスナーが動作するときに最初にデキューされます。
- イベント・キーによるイベントの呼出し： イベント名とイベント・キーだけで、エラーが発生したローカル・イベントを再度呼び出します。
- イベント・キーとイベント・データによるイベントの呼出し： イベント名、イベント・キーおよびイベント・データだけで、エラーが発生したローカル・イベントを再度呼び出します。
- パラメータ、イベント・キーおよびイベント・データによるイベントの呼出し： イベント名、イベント・キー、イベント・データおよびパラメータで、エラーが発生したローカル・イベントを再度呼び出します。

関連項目：

『Oracle Workflow API リファレンス』の「Workflow CORE API」

8-2 ページ「[ビジネス・イベントの管理](#)」

A

Oracle Workflow 開発者のナビゲーション・パス

この付録では、Oracle Applications に対してシードされた Oracle Workflow 職責での、Oracle Workflow 開発者 Web ページへのナビゲーション・パスについて説明します。

Oracle Workflow 開発者のナビゲーション・パス

次の表に、Oracle Applications に対してシードされた Oracle Workflow 職責での、Oracle Workflow 開発者 Web ページへのナビゲーション・パスを示します。システム管理者が職責、メニューおよびナビゲーション・パスをカスタマイズしている可能性もあります。

表 A-1

Web ページ	標準のナビゲーション・パス
7-6 ページ「開発者スタジオ」	ワークフロー管理者 Web アプリケーション: 「開発者スタジオ」またはワークフロー管理者 Web (新規): 「開発者スタジオ」またはワークフロー管理者イベント・マネージャ: 「開発者スタジオ」
8-70 ページ「イベント・マネージャ」	ワークフロー管理者 Web アプリケーション: 「ビジネス・イベント」またはワークフロー管理者 Web (新規): 「ビジネス・イベント」またはワークフロー管理者イベント・マネージャ: 「ビジネス・イベント」
8-66 ページ「イベント・システム・ローカル・キュー」	ワークフロー管理者 Web アプリケーション: 「イベント・キュー・サマリー」またはワークフロー管理者イベント・マネージャ: 「イベント・キュー・サマリー」

注意: ワークフロー管理者 Web (新規) の職責は、Oracle Workflow で追加された Oracle Applications Framework ベースの機能が参照できるように用意されています。ワークフロー管理者 Web アプリケーションおよびワークフロー管理者イベント・マネージャの従来の職責でも同じ機能を利用できます。

B

Oracle Workflow Builder のメニューとツールバー

この付録では、Oracle Workflow Builder のメニューとツールバーについて説明します。

Oracle Workflow Builder のメニュー

Oracle Workflow Builder のメイン・メニュー・バーには、次のメニューが含まれています。

- ファイル
- 編集
- 表示
- ウィンドウ
- ヘルプ

「ファイル」メニュー

「ファイル」メニューを使用すると、次のような処理を実行できます。

新規作成	項目タイプを定義するための新規作業領域を作成します。
クイック・スタート・ウィザード	ワークフロー・プロセス定義の設計を開始できるフレームワークを作成します。2-19 ページの「 クイック・スタート・ウィザードの概要 」を参照してください。
オープン	データベースやファイルへの接続を求めるプロンプトが表示され、データ・ストアが開きます。2-13 ページの「 項目タイプのオープンと保存 」を参照してください。
ストアを閉じる	選択したデータ・ストアを閉じます。このメニュー・オプションを選択できるのは、ナビゲータがアクティブ・ウィンドウになっている場合に限られます。
上書き保存	現在接続されているデータベースまたはファイルに変更内容を保存します。2-13 ページの「 項目タイプのオープンと保存 」を参照してください。
別名保存	変更内容を、指定したファイルまたはデータベースに任意の有効日付で保存します。

- ショートカット作成** Oracle Workflow Builder の現行セッションのデスクトップにショートカット・アイコンを作成します。ショートカット名の入力を求めるプロンプトが表示されます。ショートカットを選択すると、Oracle Workflow Builder が実行され、ショートカットの作成時に選択したデータ・ストアに自動的に接続して項目タイプをロードし、その時点でロードされて開いていたプロセス・ウィンドウが開きます。データ・ストアがデータベースの場合は、ショートカットを実行すると、Oracle Workflow Builder の起動前に、データベースのパスワード入力を求めるプロンプトが表示されます。この機能を使用できるのは、Windows NT 4.0 以上で Oracle Workflow Builder を実行する場合のみです。Microsoft Windows NT の旧バージョンでは、ショートカットの概念がサポートされていません。4-25 ページの「ワークフロー・プロセスのショートカット・アイコンの作成」を参照してください。
- 検証** 現行データ・ストア内のすべてのプロセス定義を検証します。「更新」を使用して、プロセスの最新の検証レポートを表示します。4-23 ページの「プロセス定義の検証」を参照してください。
- ダイアグラムの印刷** アクティブなプロセス・ウィンドウに表示されているプロセス・ダイアグラムを印刷します。4-22 ページの「プロセスの印刷」を参照してください。
- 項目タイプの表示 / 非表示** 「項目タイプの表示」ウィンドウが表示され、現行データ・ストアのどの項目タイプをナビゲータ・ツリーで表示または非表示にするかを指定します。
- データベースからルールをロード** Oracle Workflow ディレクトリ・サービスのルールが、現行データベース・ストアから Oracle Workflow Builder にロードされ、ナビゲータ・ツリーの「ディレクトリ・サービス」ブランチや、ルールを参照するプロパティ画面のドロップダウン・フィールドから表示できるようになります。このメニュー・オプションを選択できるのは、現行データ・ストアがデータベースの場合に限られます。4-26 ページの「ルール」を参照してください。
- 終了** Oracle Workflow Builder を終了します。

「編集」メニュー

「編集」メニューは、「ナビゲータ」ウィンドウを選択した場合とプロセス・ウィンドウを選択した場合で表示が異なります。次のメニュー・オプションは、「ナビゲータ」ウィンドウを選択した場合にのみ表示され、「ナビゲータ」ウィンドウにのみ適用されます。

新規作成	プロパティ画面を表示し、項目タイプ、関数アクティビティ、プロセス・アクティビティ、通知アクティビティ、イベント・アクティビティ、メッセージ、選択肢タイプ、選択肢コードまたは属性を新規作成します。
コピー	選択したオブジェクトをナビゲータ・ツリーにコピーします。
貼り付け	オブジェクトを、クリップボードからナビゲータ・ツリーで選択したブランチに貼り付けます。
削除	選択したオブジェクトをナビゲータ・ツリーから削除します。
検索	「検索」ウィンドウが表示され、検索基準を入力してナビゲータ・ツリーでオブジェクトを検索できます。2-7 ページの「 ナビゲータ・ツリーでのオブジェクトの検索 」を参照してください。
再検索	前に「検索」ウィンドウで定義したのと同じ基準を使用して、ナビゲータ・ツリーでオブジェクトを検索します。
プロパティ	選択したオブジェクトのプロパティ画面を表示します。
プロセスの詳細	選択したプロセス・アクティビティのプロセス・ウィンドウを開きます。
属性の移動	選択した属性をリスト内で上下に移動し、ナビゲータ・ツリーの現在のブランチに表示されている属性の順序を変更します。

次のメニュー・オプションが表示されるのは、プロセス・ウィンドウを選択している場合に限られ、選択したプロセス・ウィンドウにのみ適用されます。

選択の削除	選択したオブジェクトをプロセス・ウィンドウから削除します。
プロパティ	選択したアクティビティ・ノードのプロパティ画面を表示します。
ダイアグラムのコピー	アクティブなプロセス・ウィンドウに表示されているプロセス・ダイアグラムを、クリップボードにコピーします。4-22 ページの「 プロセス・ダイアグラムをクリップボードにコピー 」を参照してください。

「表示」メニュー

「表示」メニューを使用すると、Oracle Workflow Builder の表示を変更できます。

- フォント** 「フォント」プロパティ画面を表示します。このプロパティ画面を使用して、「ナビゲータ」ウィンドウやプロセス・ウィンドウに表示されるテキストのフォントの設定を変更します。変更は、それ以降のすべての Oracle Workflow Builder セッションに適用されます。4-24 ページの「[Oracle Workflow Builder のフォントの変更](#)」を参照してください。
- ログ->表示** 「ログ」ウィンドウの表示と非表示を切り替えます。「メッセージ・ログ」ウィンドウには、ワークフロー・ビルダーからのエラーと無関係なメッセージが表示されます。
- ログ->詳細** Oracle Workflow Builder のデバッグ・モードのオンとオフを切り替えます。「詳細」をオンにすると、デバッグ・モードがオンになり、より詳細なメッセージが「ログ」ウィンドウに書き込まれます。このモードをオンにすると Oracle Workflow Builder のパフォーマンスが大幅に低下するため、オラクル社カスタマ・サポート・センターの担当者から特に指示がないかぎり、「詳細」をオンにしないでください。
- ログ->ファイルに保存** 「メッセージ・ログ」ウィンドウの今後の内容がすべてファイルに書き込まれます。「表示」メニューから「Log Show」を選択して、ログ・ファイルの位置と名前を指定します。
- ログ->手前に表示** 「メッセージ・ログ」ウィンドウがアクティブ・ウィンドウとして一番手前に表示されます。
- グリッド指定** すべてのプロセス・ウィンドウのグリッド表示のオンとオフを切り替えます。
- 「デザイナーでラベルを表示」サブメニュー** オプションのサブメニューを使用すると、アクティビティ・ノード・ラベルに表示される情報を制御できます。「インスタンス・ラベル」、「内部名」、「表示名」、「実行者」または「コメント」を選択します。
- デザイナーでラベルを表示->インスタンス・ラベル** プロセス・ダイアグラム内のアクティビティ・ノードのラベルとして、ノード・ラベルを使用します。この設定は、明示的に変更するまで、すべてのプロセス・ダイアグラムとすべての Oracle Workflow Builder セッションに適用されます。
- デザイナーでラベルを表示->内部名** プロセス・ダイアグラム内のアクティビティ・ノードのラベルとして、アクティビティの内部名を使用します。この設定は、明示的に変更するまで、すべてのプロセス・ダイアグラムとすべての Oracle Workflow Builder セッションに適用されます。

デザイナーでラベルを表示 → 表示名 プロセス・ダイアグラム内のアクティビティ・ノードのラベルとして、アクティビティの表示名を使用します。この設定は、明示的に変更するまで、すべてのプロセス・ダイアグラムとすべての Oracle Workflow Builder セッションに適用されます。

デザイナーでラベルを表示 → 実行者 プロセス・ダイアグラム内のアクティビティ・ノードのラベルとして、アクティビティの実行者を使用します。実行者が存在しない関連アクティビティとプロセス・アクティビティには、ラベルは表示されません。この設定は、明示的に変更するまで、すべてのプロセス・ダイアグラムとすべての Oracle Workflow Builder セッションに適用されます。

ラベルの表示 → コメント プロセス・ダイアグラム内のアクティビティ・ノードのラベルとして、そのアクティビティのコメントを使用します。コメントが付いていないアクティビティには、ラベルは表示されません。この設定は、明示的に変更するまで、すべてのプロセス・ダイアグラムとすべての Oracle Workflow Builder セッションに適用されます。

開発者モード 表示を標準のプレゼンテーション・モードと開発者モードに切り替えます。開発者モードでは、アイコンはすべて特定のオブジェクト・タイプ / サブタイプのデフォルト・アイコンに戻り、サブプロセスにはトップ・レベルのプロセス・アイコンと異なるアイコンが表示され、ナビゲータ・ツリーでは、オブジェクトが内部名で表示およびソートされます。属性は内部名で表示されますが、ソートはされないので注意してください。

アクティブ・ウィンドウが「ナビゲータ」ウィンドウの場合は、次のメニュー・オプションも表示されます。

ウィンドウを分割 「ナビゲータ」ウィンドウを縦方向または横方向に分割します。

アクティブ・ウィンドウがプロセス・ウィンドウの場合は、次のメニュー・オプションも表示されます。

概要 プロセスの「概要」ウィンドウを表示します。4-21 ページの「[プロセス概要の表示](#)」を参照してください。

ナビゲータでプロセスを表示 このメニュー・オプションを選択すると、プロセス・ダイアグラム・ウィンドウに表示されている現行プロセスに対応するプロセス・アクティビティが、「ナビゲータ」ウィンドウに表示されます。

オーバーレイ・イメージの表示 アイコンにオーバーレイ・イメージがある場合、その表示または非表示を切り替えます。たとえば、プロセスの「開始」アクティビティと「終了」アクティビティには、それぞれ緑と赤の矢印のオーバーレイ・イメージがあります。

「ウィンドウ」メニュー

「ウィンドウ」メニューには、開いているすべてのアプリケーション・ウィンドウの名前が表示されます。ウィンドウ名を選択して、ウィンドウをアクティブにします。また、次のメニュー項目も選択できます。

- 重ねて表示** 開いているウィンドウがカスケード表示（重ねて表示）されます。
- 並べて表示** 開いているウィンドウがタイル表示（並べて表示）されます。

「ヘルプ」メニュー

「ヘルプ」メニューを使用すると、Oracle Workflow Builder の使用方法に関するヘルプを表示できます。

- 目次** Oracle Workflow Builder の使用方法のヘルプを表示します。
- Oracle Workflow Builder
のバージョン情報** Oracle Workflow Builder の現行バージョンとアクセス・レベルを表示します。「アクセス・レベル」フィールドでアクセス・レベルを編集し、「OK」を選択して変更を適用することもできます。

Oracle Workflow Builder のツールバー

Oracle Workflow Builder では、「ナビゲータ」ウィンドウとプロセス・ウィンドウにツールバーが表示されます。

ナビゲータのツールバー

ナビゲータのツールバーには、次のボタンが含まれ、ナビゲータ・ツリーから選択したオブジェクトにのみ適用されます。



新規ストア： ナビゲータ・ツリーに新規のデータ・ストアのブランチが作成されます。



オープン： 「オープン」ウィンドウを表示し、ファイルまたはデータベースから保存されている項目タイプを開きます。



上書き保存： 選択したデータ・ストアの変更内容が、現在接続中のデータベースまたはファイルに保存されます。選択したデータ・ストアがデータベースやファイルに接続されていない場合は、「オープン」ウィンドウを表示してデータベースやファイルに接続します。



削除： 選択したオブジェクトを削除します。



プロパティ： 選択したオブジェクトのプロパティ画面を表示します。



コピー： 選択したオブジェクトをコピーします。



貼り付け： コピーしたオブジェクトを現行オブジェクト・ブランチに貼り付けます。



検証： プロセス定義を検証します。



開発者モード： 「開発者」と「プレゼンテーション」の表示を切り替えます。



検索： 「検索」ウィンドウを表示し、検索基準を指定してナビゲータ・ツリーでオブジェクトを検索します。



クイック・スタート・ウィザード： クイック・スタート・ウィザードを実行し、ワークフロー・プロセス定義の作成を開始します。



ヘルプ: Oracle Workflow Builder の使用方法のヘルプを表示します。



新規オブジェクト: 選択したオブジェクト・ブランチ (項目タイプ、「プロセス」、「通知」、「関数」、「メッセージ」または「選択肢タイプ」) に応じて、そのオブジェクト・タイプのプロパティ画面が表示され、オブジェクトを新規作成します。

「プロセス」ウィンドウのツールバー

プロセス・ウィンドウのツールバーには、次のボタンが含まれ、現行のプロセス・ウィンドウから選択したオブジェクトにのみ適用されます。



オープン: 「オープン」ウィンドウを表示し、ファイルまたはデータベースから保存されている項目タイプを開きます。



上書き保存: 選択したデータ・ストアの変更内容が、現在接続中のデータベースまたはファイルに保存されます。選択したデータ・ストアがデータベースやファイルに接続されていない場合は、「オープン」ウィンドウを表示してデータベースやファイルに接続します。



ダイアグラムの印刷: 現行のプロセス・ダイアグラムを印刷します。



新規プロセス: プロセス・アクティビティ・ノードのプロパティ画面を表示し、新規プロセス・アクティビティを作成します。



新規通知: 通知アクティビティ・ノードのプロパティ画面を表示し、新規通知アクティビティを作成します。



新規関数: 関数アクティビティ・ノードのプロパティ画面を表示し、新規関数アクティビティを作成します。



新規イベント: イベント・アクティビティ・ノードのプロパティ画面を表示し、新規イベント・アクティビティを作成します。



選択の削除: 選択したオブジェクトを削除します。



プロパティ: 選択したオブジェクトのプロパティ画面を表示します。



開発者モード: 「開発者」と「プレゼンテーション」の表示を切り替えます。



検索: プロセスの「概要」ウィンドウを表示します。



インスタンス・ラベルの表示: プロセス・ウィンドウのノード・アクティビティ・ラベルとして、インスタンス・ラベルを表示します。



内部名の表示: プロセス・ウィンドウのノード・アクティビティ・ラベルとして、ノードの内部名を表示します。



表示名の表示: プロセス・ウィンドウのノード・アクティビティ・ラベルとして、ノードの表示名を表示します。



コメントの表示: プロセス・ウィンドウのノード・アクティビティ・ラベルとして、ノードのコメントを表示します。



実行者の表示: プロセス・ウィンドウのノード・アクティビティ・ラベルとして、ノードの実行者を表示します。



ヘルプ: Oracle Workflow Builder の使用方法のヘルプを表示します。

他の Oracle 製品への Oracle Workflow の実装

この付録では、Oracle E-Business Suite および Oracle テクノロジ・スタックに組み込まれているワークフローとビジネス・イベント・システムについて説明します。また、各ワークフロー、イベントおよびサブスクリプションのカスタマイズに関するオラクル社のサポート・ポリシーについて説明します。

Oracle E-Business Suite 埋込みの事前定義済みのワークフロー

次にあげる事前定義済みのワークフロー・プロセスは、Oracle Workflow を使用してカスタマイズできます。各ワークフローの詳細は、各製品のユーザーズ・ガイドまたは構成ガイドを参照してください。

注意： Oracle Applications の製品には、Account Generator の機能を使用して、動的に会計フレックスフィールドの組合せを作成するものもあります。Account Generator には、あらかじめ定義されている汎用的なワークフロー関数が用意されており、それぞれの Oracle Application 製品では、事前に定義した Account Generator のプロセス内でこれを使用します。ここでは、各製品に関する Account Generator のプロセスについては記載していません。詳細は各製品のユーザーズ・ガイドに記載されています。Account Generator 機能の一般的な内容については、『Oracle Applications フレックス・フィールド・ガイド』を参照してください。

関連項目：

C-39 ページ [「Oracle Workflow のサポート・ポリシー」](#)

適用されるテクノロジー

Application Implementation Wizard

Application Implementation Wizard には、一連のワークフロー・プロセスが表示されるため、指示に従って Oracle Applications を設定し、実装できます。また、このウィザードを使用すると、各タスクを実行し、インストールに合わせて相互に依存する Oracle Applications を構成できます。実装作業を容易に進行できるように、Application Implementation Wizard では類似する設定タスクが論理グループにまとめられています。

ウィザードに表示される手順の順序は、実装するアプリケーション・モジュールによって決まります。このため、複数のアプリケーション・モジュールを実装する場合に、重複する設定手順を実行する必要がなくなります。

ワークフロー・プロセスの詳細と使用方法は、『Application Implementation Wizard User's Guide』を参照してください。

Oracle Application Object Library

Oracle Application Object Library には、コンカレント・マネージャの処理を Oracle Applications のワークフロー・プロセスに統合するときに使用できるように、一連の「標準」関数アクティビティが用意されています。これらの「標準」関数アクティビティは、コンカレント・マネージャ関数の項目タイプに関連付けられています。5-23 ページの「[コンカレント・マネージャの標準アクティビティ](#)」を参照してください。

Oracle Common Application Components

HTML カレンダー

JTF カレンダー・ワークフロー： 新規グループと公開カレンダーの承認についてはカレンダー管理者へ、既存のグループ・カレンダー・サブスクリプションの承認についてはグループ・カレンダー管理者へ、カレンダー要求を追跡およびルーティングします。このプロセスでは、アポイントメントの招待者と出席者に案内状も送信します。

JTF タスク・リマインダ： スケジュールされたアポイントメントの督促を選択します。たとえば、ユーザーがアポイントメントを作成するときに「リマインドする」ドロップダウン・リストから「15 分前」を選択すると、スケジュールされた会議の始まる 15 分前にワークフロー通知を受け取ります。

エスカレーション・マネージャ

エスカレーション通知の応答ワークフロー： エスカレーション文書が作成されたとき、またはステータス、レベル、所有者、目標日付などの一定の文書属性が更新されたときに、エスカレーション文書内で識別されるエスカレーションの所有者、その所有者の管理者、または顧客や従業員にそのことを知らせるために、通知を送信します。

ビジネス・ルール・モニター

ビジネス・ルール・モニターのメイン・プロセス： このプロセスは、ビジネス・ルール・モニターのループするワークフローで動き、すべてのアクティブなビジネス・ルールをチェックして、ルール違反がないかどうかを識別します。

ビジネス・ルール・モニターのタスク・プロセス： このプロセスは、ビジネス・ルール・モニターのメイン・プロセスが「タスク」オブジェクトに関連するビジネス・ルール違反を識別したときにのみ使用されます。ビジネス・ルール・モニターのタスク・プロセスは、ルールで識別されたワークフロー情報に基づいて、以後のアクティビティで動きます。たとえば、「タスクのエスカレート（通知のみ）」ワークフローがビジネス・ルールで選択されている場合は、ワークフローの属性ウィンドウで識別されたユーザーにワークフロー通知が送信されます。

ビジネス・ルール・モニターのサービス要求プロセス： このプロセスは、ビジネス・ルール・モニターのメイン・プロセスがサービス要求オブジェクトに関連するビジネス・ルール違反を識別したときにのみ使用されるという点を除けば、ビジネス・ルール・モニターのタスク・プロセスと同じです。

ビジネス・ルール・モニターの欠陥プロセス： このプロセスは、ビジネス・ルール・モニターのメイン・プロセスが欠如オブジェクトに関連するビジネス・ルール違反を識別したときにのみ使用されるという点を除けば、ビジネス・ルール・モニターのタスク・プロセスと同じです。

タスク・マネージャ

ワークフロー-タスク・マネージャ： タスク・マネージャは、ワークフローからタスク・マネージャ・プロセスを使用して、タスクが作成または変更されたときに担当者へ知らせる通知を送信します。

タスク・マネージャによって、グループまたはチームへの通知の送信がサポートされます。タスクがグループまたはチームに割り当てられると、ワークフロー・プロセスは、グループ用の一覧にある電子メール・アドレスへ1通の通知のみを送信するのではなく、グループの各メンバーに個別にワークフロー通知を送信します。

通知は、タスク・マネージャの HTML 版とフォーム版の両方に自動的に送信されるか、または手動でフォーム版のみに送信されるようにすることができます。

- 手動ワークフロー通知（フォームのみ）： 所有者およびその代理人の情報を使用してタスクが作成された後に「タスク」ウィンドウの「ワークフロー起動」ボタンをクリックして、タスクの作成または更新についての通知を手動で所有者と代理人へ送信します。
- 自動ワークフロー通知： 次の条件の1つが満たされる場合に、ワークフロー通知が自動的に送信されます。
 - タスクを保存する前に、フォーム・ベースの「タスク」ウィンドウでタスクの「自動通知」チェック・ボックスが選択されている。
 - HTML ベースの「タスクの作成」ウィンドウで「通知」チェック・ボックスが選択されている。
 - 「タスク・タイプ」設定ウィンドウで「通知」チェック・ボックスが選択されている。

自動ワークフロー通知が有効にされていると、次の場合にタスクのワークフロー通知が自動的に送信されます。

- タスクが作成または削除される場合は、所有者とその代理人全員が通知を受け取ります。
- タスクが新しい所有者に再割当てされる場合は、新旧の所有者が通知を受け取ります。
- 追加、再割当てまたは削除のいずれかによってタスクの代理人が変更される場合は、所有者および新旧の代理人が通知を受け取ります。
- タスクのステータス、優先度、タイプ、および計画済、スケジュール済、実際の開始日と終了日が更新される場合は、所有者と代理人全員が通知を受け取ります。

- タスク所有者が自分のステータスを更新する場合は、所有者と代理人全員が通知を受け取ります。

タスク・マネージャのデフォルトのワークフロー・プロセスを使用しない場合は、Oracle Workflow を使用して新しいワークフロー・プロセスを定義し、カスタム・ワークフローをタスク・タイプに割り当てることができます。

ビジネス・インテリジェンス

Oracle Business Intelligence System

BIS 例外管理： 汎用的なワークフロー・プロセスで、BIS カスタマが Performance Management Framework の一部として使用するためのテンプレートです。実際のパフォーマンスが、要求するパフォーマンス要件を満たさない場合は、このプロセスによって、埋込みレポート URL を使用した基本的な訂正処置の通知が送信されます。OBIS 訂正処理の項目タイプの他のすべてのプロセスは、BIS 例外管理に類似しています。

Process Manufacturing Intelligence

プロセス製造在庫回転率プロセス： 在庫品の回転の実績値が、在庫回転率レポートで定義されている目標値に満たない場合に、指定された責任者に対して通知が送信されます。在庫回転率レポートは、Process Manufacturing BIS で使用されます。

契約

Oracle Core Contracts

承認プロセス： ユーザーが契約の承認プロセスを開始すると、このワークフローは、承認するように契約を指定された個人にルーティングします。契約が承認された後に、署名した通知が指定された個人に送信されます。

変更承認プロセス： 契約を変更する前に、変更要求が承認される必要があります。このワークフローでは、契約の変更要求を指定された承認者へルーティングします。変更要求が承認された後に、契約が更新できるようになります。

Oracle Service Contracts

サービス要求の作成ワークフロー： 関連する条件が満たされたときにこのワークフローを使用してサービス要求が自動的に作成されるように、サービス契約の明細に対応するイベントを設定することができます。サービス要求の詳細を含む通知も作成され、契約ランチパッドの受信ボックスで参照することができます。

電子更改ワークフロー： このワークフローは、保証契約延長の認可を電子更改プロセスで受けるときに起動されます。最初にテンプレート・セットが作成され、QA で契約がチェックされ、関連する表紙付きの見積書（契約の下書き）が生成され、電子メールで送信され、更新ステータスが更新されます。セルフサービスの更新を使用してユーザーがログインする場合は、契約の見積りを受け入れ、購買発注書を使用して契約金額を支払うことを決定すると、ワークフローによって契約管理者へ通知が送信されます。管理者の応答に応じて、契約は適切なステータスに設定されます。

導入ベース取引通知ワークフロー： システム・プロファイルの「OKS: 導入ベース統合メッセージの使用可能化」が「Yes」に設定されている場合は、一定のトランザクション処理中にシステム・プロファイルの「OKS: 導入ベース統合メッセージを送るユーザー名」で識別されるユーザーへ、このワークフローによって通知が送信されます。これには、新しい項目インスタンスが作成されるとき、終了するとき、新しい所有者に転送されるとき、置換または戻されるとき、または大量の項目インスタンスが分割されるときが含まれます。

会計関係のアプリケーション

Oracle Cash Management

CE 取引明細書伝送通知プロセス： このワークフローでは、「銀行伝送詳細」ウィンドウで定義されたユーザーへ、転送ステータスの電子メール通知が送信されます。このワークフロー・プロセスは、銀行取引明細書の取得プログラムを実行して自分の取引銀行からローカル・ディレクトリへ銀行取引明細書を転送すると、開始されます。

CE 例外レポート伝送通知プロセス： このワークフローでは、「銀行伝送詳細」ウィンドウで定義されたユーザーへ、転送ステータスの電子メール通知が送信されます。このワークフロー・プロセスは、Retrieve Payment Exceptions プログラムを実行して自分の取引銀行からローカル・ディレクトリへ支払例外レポートを転送すると、開始されます。

XML 銀行取引明細書の処理： このワークフローは、インポート・プロセスが起動された後に始まります。インポート・プロセスが正常に起動されたかどうか、また正常に終了したかどうかを知らせるために、指定された資金管理者へ通知が送信されます。

Oracle Collections

破産ステータス： ある取引先が支払不能者としてフラグを立てられると、次の複数のプロセスが開始されます。

- 破産に対する処理をし、詳細データを収集し、破産による負債を追跡できるかどうかを判断するために、専門家または管理者に HTML ユーザー・インタフェースで承認が送信されます。破産に対応する戦略が最終的に割り当てられます。
- すべての滞納に対して、滞納ステータスが「破産」に設定されます。
- 該当する事例または顧客のその他すべての項目（つまり滞納）が「破産」に設定されず。
- 該当組織のすべての担当者に対し、TCA 内に「担当なし」フラグが設定されます。または、一般消費者に対する回収の場合は、保証人または連帯保証人に対して「担当なし」フラグが設定されます。
- 賃貸契約管理の回収のみの場合は、請求書発行を停止するために必要な変更を検討して実行するのに適切な関係者に、通知が送信されます。
- 賃貸契約管理の回収のみの場合は、適切な関係者に割当のデフォルト破産通知が送信されます。

破産ステータス・ワークフローは、回収者が Oracle Collections の「延滞」タブで単一の延滞に対し新しく「破産」ステータスを作成すると (Forms のユーザー・インタフェース)、開始されます。

延滞の現行ステータス通知： 延滞状況のステータスが変化したときに、そのステータスについての通知が代理人または代理人の管理者に送信されます。たとえば、延滞状況が終了したり（つまり、通常の実行が行われているとみなされる）、延滞に対して「破産」または「貸倒償却」などの新しいステータスが設定されたりすると、ステータス内の変更を記述するために通知が送信されます。

回収延滞与信保留： 回収者は、「延滞」タブにある対応するチェック・ボックスをクリックして、顧客に対する信用販売を続けることを要求することができます。次に、適切な管理者に通知が送信され、その要求が検討されて承認または否認されます。承認を求める適切な管理者へのルーティングは、Oracle Resource Manager で定義されたレポート階層に沿って行われます。要求が承認されると「延滞」タブの「承認」チェック・ボックスがオンにされるので、回収者は、要求が承認されたことを確認することができます。ただし、与信保持の実際の設定は、手動で行うことになります。

回収延滞サービス保留： 回収者は、「延滞」タブにある対応するチェック・ボックスをクリックして、顧客に対するサービスを続けることを要求することができます。次に、適切な管理者に通知が送信され、その要求が検討されて承認または否認されます。承認を求める適切な管理者へのルーティングは、Oracle Resource Manager で定義されたレポート階層に沿って行われます。要求が承認されると「延滞」タブの「承認」チェック・ボックスがオンにされるので、回収者は、要求が承認されたことを確認することができます。

サード・パーティに差押を通知： 資産を取り戻すために第三者を専門家が指定すると、その当事者へ、資産、差押えおよびその他の情報について詳しく知らせる電子メール通知が送信されます。専門家は、HTML インタフェースを通じて第三者を指名します。

延滞ステータス承認： 延滞に対して、差押え、原価償却または訴訟など、回収者は様々な戦略を提言することができます。(破産については、別途処理されます。)ただし、これらのステータスに対応する戦略を開始する前に管理者によって検討され、承認されることが必要です。延滞ステータス承認ワークフローは、回収者が「延滞」タブの「新規」ボタンをクリックして新しいステータスを選択し、プロセスを開始するための詳細な情報を入力し、新しいステータスを保存すると、開始されます。システムによって定義されたリソースの階層に従って、要求がルーティングされます。

延滞資産ワークフロー： 専門家は、延滞事例に関連する資産に対して資産評価が行われるように要求することができます。このワークフローによって、要求を検討し、情報を取得して専門家に返信するように、適切な管理者へ通知が送信されます。この要求は、「延滞」タブから選択した資産選択ウィンドウおよび減価償却ウィンドウで開始されます。

戦略履行メーラー： 戦略の一環として、督促通知、延滞分の請求書のコピーなどの電子メールによる文書を延滞先に送信するために、1-1 Fulfillment Server に対して要求を行うことができます。このワークフローで、このような要求が管理されます。

戦略カスタム・ワークフロー： シードされたワークフローとは異なるカスタムの作業項目項目を顧客が戦略の一環として実装しようとする場合は、カスタムの戦略ワークフローをカスタマイズすることができます。カスタムの戦略作業項目ワークフローは、ワークフロー内で識別される必要のあるパラメータと、作業項目が完了した後に識別される必要があるパラメータの1例です。

回収戦略ワークフロー： 戦略とは一連の回収作業項目であり、回収計画を作成するためにグループ化されます。各作業項目には、完了プロセスの一部としてワークフローがあります。回収戦略ワークフローとは、各戦略の回収作業項目に結び付けられたすべてのワークフローを完了に導くメイン・ワークフロー・プロセスです。

Oracle General Ledger

仕訳承認プロセス： 仕訳を転記する前に仕訳バッチの承認を要求できます。承認階層を作成し、ユーザーごとに承認限度を定義します。仕訳承認プロセスは、仕訳バッチを転記したときに開始されます。このプロセスは、承認階層に基づいて、承認を要求するユーザーに仕訳を自動的にルーティングします。

自動配賦プロセス： 戻し自動配賦を生成すると、このワークフロー・プロセスによって自動配賦プロセスが開始され、自動配賦に定義されている一括配賦または定型仕訳バッチが検証および生成されます。また、生成された各バッチに仕訳承認が必要かどうかを確認され、必要に応じて承認を要求するユーザーにバッチが送信され、承認結果が該当するユーザーに通知されます。自動配賦プロセスでエラーが発生した場合は、指定されたユーザーが、自動配賦プロセスをロールバックし、転記された仕訳を取り消すことができます。

グローバル会社間システム： グローバル会社間システム（以前の CENTRA）は、リリース 11i の拡張機能で、リリース 11 にバックポートされています。グローバル会社間システムは、複数の会社間の取引を交換するための環境を提供します。送信側の会社が会社間取引を開始して受信側の会社に承認を要求するか、送信側の会社が会社間取引を取り消すか破棄すると、ワークフロー・プロセスによって受信側の会社に通知されます。送信側の会社によって開始された会社間取引を受信側の会社が承認または否認すると、送信側の会社に通知されます。また、しきい金額を設定して、通知量を制限することもできます。このワークフロー・プロセスは、送信側の会社が会社間取引の送信、取消または破棄を行うか、受信側の会社が会社間取引を否認または承認したときに開始されます。

グローバル連結システムのクロス・インスタンス・データ転送： グローバル連結システムを自動化して、遠隔地の子会社の元帳データベースのインスタンスを中央の統合データベースのインスタンスに統合することができます。オプションで、ワークフロー通知を使用して、インスタンス間の統合ステータスをユーザーに通知します。各通知で、ソース・データベース名、マッピング・ルール、帳簿、グループ ID およびコンカレント要求 ID を含む統合転送の詳細をユーザーに提供します。中央の統合データベースで **Journal Import** または **AutoPost** が自動的に実行されるようにユーザーが選択すると、ワークフロー通知で、コンカレント要求の成功または失敗のステータスが知らされます。**Journal Import** または **AutoPost** が失敗すると、ワークフロー通知によって、中央統合データベースの要求ログを詳しく検討するように促されます。

Oracle Grants Accounting

Grants Accounting ワークフロー・プロセス： **Grants Accounting** ワークフロー・プロセスでは、賦払いが選択されるかレポートが必要なことを主要なメンバーに通知します。**Budget** サブプロセスでは、予算が送信され承認を必要としていることを、予算承認者または裁定マネージャに通知します。

このワークフロー・プロセスは、次の時点で開始されます。

- 賦払いが選択されたとき
- レポートが必要なとき
- 予算が送信されたとき
- 予算が承認 / ベースライン化されたとき

Oracle Grants Proposal

Proposal Approval プロセス： **Proposal Approval** プロセスは、提案を送信して承認を要求したときに開始されます。

通知が承認者に送信されると、このワークフロー・プロセスは、その承認者からの回答を取得してから、提案承認階層マップ内の次の承認者に進みます。

すべての承認者が提案を承認すると、その提案は承認されます。いずれかの承認者が提案を否認すると、その提案は否認されます。提案を送信して承認を要求するユーザーには、承認プロセスのステージ単位に承認ステータスを通知します。

Notify Approval サブプロセス： **Proposal Approval** プロセスの **Notify Approval** サブプロセスは、承認階層マップ内の次の承認者が選択されているときに開始されます。

Notify Approval サブプロセスでは、承認を必要とする提案が保留中であることを承認者に通知します。承認者は、提案を承認または否認します。

承認者が所定の時間内に提案を承認または否認しなかった場合、その承認者は定期的に督促を受け取ります。組織は、タイムアウトを設定して、督促を送信する期間を定義します。デフォルトでは、タイムアウトは設定されていません。

Notify Proposal Members プロセス： **Notify Proposal Members** プロセスでは、提案の担当者に通知を送信します。

Oracle Internet Expenses

AP 経費精算プロセス： Oracle Internet Expenses では、AP 経費精算ワークフロー・プロセスを使用し、Internet Expenses に入力された経費精算書に対してマネージャの承認および経理担当者の検査を行います。AP 経費精算プロセスは、ユーザーが経費精算書を送信した時点で起動され、経費精算書が否認された時点または経費精算書がマネージャによって承認され経理担当者によって検査された時点で終了します。承認および検査された場合は、AP 経費精算プロセスによって、経費精算書が買掛 / 未払金請求書インポートプログラムで使用できるようになります。AP 経費精算プロセスは、マネージャの承認および経理担当者の検査の際のキー・イベントで、従業員に通知を行います。

AP クレジット・カード・プロセス： AP クレジット・カード・ワークフロー・プロセスでは、従業員に対して作成された支払、および T&E (travel and entertainment) クレジット・カードの請求額としてクレジット・カード発行者に対して作成された支払を、従業員およびマネージャに通知します。また、未処理のクレジット・カード取引のうち、経費精算書に送信されていないもの、または経費精算書に送信されているが経費精算書が AP Expense Report プロセスを完了していないものを、従業員およびマネージャに通知を送信します。支払通知は、支払が Oracle Payables で作成されたときに生成されます。未処理の請求通知は、Credit Card Outstanding Charges Report プログラムを Oracle Payables から実行したときに送信されます。

退職従業員ワークフロー・プロセス： 申請者、従業員、管理者または承認者が退職すると、経費精算書が失われたり、ワークフロー承認プロセスの途中で経費精算書にアクセスできなくなったりする可能性があります。さらに、会社支払または両支払とマークされた会社のクレジット・カード支出が含まれた経費精算書は請求する相手がいない上、従業員の退職に先立って支払義務のあるクレジット・カード支出を分類することもあります。その結果、クレジット・カード会社への支払処理が遅延する可能性があります。退職者ワークフロー・プロセスによって、このような経費精算書の処理が継続できるようになり、関連する支払がタイムリーに行われるようになります。

Procurement Cards

次の調達カード・ワークフローでは、セルフサービス従業員が調達カード取引の検証および承認を行うことができます。

AP 調達カード従業員確認ワークフロー・プロセス： AP 調達カード従業員確認ワークフロー・プロセスでは、カード所有者との調達取引を通知および確認します。このワークフロー・プロセスは、Oracle Payables から従業員カード取引検証の配分プログラムを送信したときに開始されます。このプロセスでは、従業員の調達カードに請求された取引をその従業員に通知し、オプションで従業員による検証を要求します。

AP 調達カード・マネージャ承認取引プロセス： AP 調達カード・マネージャ承認取引ワークフロー・プロセスでは、検証済のカード所有者との調達カード取引を通知および確認します。このワークフロー・プロセスは、Oracle Payables からマネージャ・カード取引承認の配分プログラムを送信したときに開始されます。マネージャ・ワークフロー・プロセスは、AP 調達カード従業員確認ワークフローが取引を完了していないと使用できません。このプロセスでは、従業員による調達カード取引をマネージャに通知し、オプションでマネージャによる承認を要求します。

Oracle Labor Distribution

Effort Report Notification プロセス： **Labor Distribution** のワークフロー機能は、成果レポートを組織全体に自動的にルーティングし、注意が必要な成果レポートまたは完了しているプロセスに関して、ユーザーに電子通知を配信します。

Effort Report Notification ワークフロー・プロセスには、次のサブプロセスが含まれます。

- 承認
- 通知

Effort Report Notification ワークフロー・プロセスは、成果レポートが作成されたときに、**Labor Distribution** で開始されます。

通知は、成果レポートの承認者に送信します。成果レポートが承認されると、成果レポートは管理者に送信し、証明を要求します。成果レポートの作成者は、成果レポートのステータスを監視できます。

Distribution Adjustment Approval Notification プロセス： **Labor Distribution** のワークフロー機能は、**Distribution Adjustment Approval Notification** を組織全体に自動的にルーティングし、注意が必要な配布調整または完了しているプロセスに関して、ユーザーに電子通知を配信します。このプロセスは、配布バッチが送信されたときに開始されます。

Oracle Payables

AP オープン・インタフェース・インポート・プロセス： このワークフローで、オープン・インタフェース表のデータの検証が自動化されます。たとえば、このプロセスを修正し、オープン・インタフェース表内ですべての会計コード組合せを検証できます。無効なコード組合せがある場合は、その通知を指定したユーザーに送信して訂正させることができます。必要であれば、無効なコード組合せを明示したデフォルト値で上書きするように、このプロセスを設定できます。**Oracle Workflow** を使用すると、特定の業務要件にあわせてワークフロー・ルールを追加することも可能です。請求書は、このプロセスを通過すると、**Oracle Payables Application** の表にインポートする準備ができたこととなります。オープン・インタフェース・インポート・プロセスを開始するには、「要求の発行」ウィンドウから「買掛 / 未払金オープン・インタフェース・ワークフロー」を発行します。

請求書承認ワークフロー： このワークフローでは、指定された各承認者へ承認のために請求書がルーティングされます。このワークフローでは、請求書に承認が必要かどうかを判断するために、**Oracle Approvals Management (OAM)** 内で定義された承認ルールが使用されます。請求書に承認が必要な場合は、このワークフローで、承認者リストの各承認者へ順番に請求書の承認をオンラインで求めていきます。たとえば、100,000 ドルを超える請求書の場合、最高財務責任者の承認、次に最高経営責任者の承認が必要であるというように、ルールを定義することができます。

支払メッセージの処理ワークフロー： このワークフローによって **XML** 支払メッセージが作成され、取引銀行へ送信されます。次に、支払を作成したユーザーへ通知が送信されます。また、**XML** 支払を生成する支払バッチの確認も管理されます。

支払指示エラーの受信ワークフロー： XML 支払機能を使用していて、支払指示エラーの表示メッセージを受信するように Oracle XML Gateway を設定している場合は、取引銀行からこのメッセージが送信されると、このワークフローによって、支払指示エラーが含まれた通知が適切なユーザーへ送信されます。

支払通知メッセージの受信ワークフロー： XML 支払機能を使用していて、支払通知表示 XML メッセージを受信するように Oracle XML Gateway を設定している場合は、取引銀行からこのメッセージが送信されると、このワークフローによって、支払の助言が含まれた通知が適切なユーザーへ送信されます。

経費精算書および調達カードを含む Oracle Internet Expenses 関連のワークフローの詳細は、C-11 ページの「[Oracle Internet Expenses](#)」を参照してください。

Oracle Public Sector Budgeting

Distribute Worksheet Workflow プロセス： Distribute Worksheet Workflow プロセスでは、ワークシートを配布し、その事実をユーザーに通知します。このプロセスは、ワークシートが配布するときに開始します。

Submit Worksheet Workflow プロセス： Submit Worksheet Workflow プロセスでは、ワークシートを送信します。ユーザー定義のパラメータに基づいて、制約の検証、ワークシートの処理、コピーおよびマージを実行します。ワークシートは、予算ステージを経過してから、必要な承認を得るために承認プロセスにルーティングされます。また、このプロセスは、ワークシートの凍結および凍結解除も行います。通知は、プロセスを開始するユーザーおよび承認者に送信します。

このプロセスは、次の時点で開始されます。

- ワークシートの制約を検証するとき
- ワークシートを凍結するとき
- ワークシートの凍結を解除するとき
- ワークシートを次のステージに移動するとき
- ワークシートをコピーするとき
- ワークシートをマージするとき
- ワークシートを送信するとき

Distribute Budget Revision Workflow プロセス： Distribute Budget Revision Workflow では、予算改訂を配布し、その事実をユーザーに通知します。このプロセスは、予算改訂を配布するときに開始されます。

Submit Budget Revisions Workflow： Submit Budget Revisions Workflow プロセスでは、予算改訂を送信します。ユーザー定義のパラメータに基づいて、制約の検証およびその他の予算改訂処理を実行します。Submit Budget Revisions プロセスでは、予算改訂を承認プロセスにルーティングし、予算改訂のステータスおよび基準値を更新します。また、予算引当お

よび総勘定元帳への改訂の転記も行います。予算改訂の凍結および凍結解除も実行します。通知は、プロセスを開始するユーザーおよび承認者に送信します。

このプロセスは、次の時点で開始されます。

- 予算改訂の制約を検証するとき
- 予算改訂を凍結するとき
- 予算改訂の凍結を解除するとき
- 予算改訂を送信するとき

Oracle Public Sector Financials (International)

ITR 承認： 国内取引の承認プロセスは、国内費用が承認のために送信されると開始されます。このプロセスでは、費用発生元の適切な承認者へ、承認階層に基づいてサービスの明細が自動的にルーティングされます。費用発生元に承認された場合は、そのサービス明細が費用受取り側の適切な承認者へルーティングされます。

EXP 承認： 交換プロトコル承認プロセスは、転送ユニットが転送されると開始されます。このプロセスでは、承認プロファイルに基づいた適切な承認者へ、転送ユニット認証のための要求が送信されます。転送ユニット内に含まれるダイアログ・ユニットは、承認者の応答に基づいて処理されます。

契約取引約定ワークフロー・プロセス： このワークフローは、契約約定が承認のために送信されると開始されます。このプロセスでは、承認階層の設定に基づいて、適切な承認者へ文書が自動的にルーティングされます。承認者は、文書を承認または否認します。予算上の裁量が可能な場合は、承認者が文書を承認するときに、資金がチェックされ、確保されます。文書を送信して承認を要求するユーザーには、承認プロセスのステージ単位に承認ステータスを通知します。

調査承認プロセス： このプロセスは、一件書類のメンテナンス・トランザクションが承認者へ送信されると開始されます。このプロセスでは、一件書類に添付された承認階層に基づいて、一件書類のメンテナンス資金振替の承認を要求する通知が適切な承認者へ送信されます。承認者の応答に基づいて、債務仕訳帳が取り消され、実際の仕訳帳が作成されて、資金の振替が完了します。

Oracle Receivables

クレジット・メモ申請承認プロセス： このワークフローでは、組織の内部管理階層、または Oracle Receivables で定義されている承認制限を使用して、承認するクレジット・メモの申請をルーティングします。この申請が承認されると、Oracle Receivables にクレジット・メモが自動的に作成されます。申請が否認された場合は、申請者に対して承認されなかった理由を通知します。

クレジット・メモ申請ワークフローは、iReceivables から開始します。iReceivables は、Web ベースのセルフサービス・アプリケーションです。iReceivables に登録されたユーザーは、標準 Web ブラウザを使用して、Receivables アカウント情報にアクセスできます。iReceivables ユーザーが「係争請求」を選択すると、Receivables によって、指定した額が係争中になり、クレジット・メモ申請プロセスが開始され、申請が承認プロセスにルーティングされます。

文書転送メッセージ・ワークフロー： このワークフローでは、XML 請求文書が作成され、取引先へ送信されます。このワークフローは、次の 2 つの項目タイプから構成されます。

- 「AR 文書の転送」項目タイプ： Oracle Receivables 内で、ユーザーは文書転送計画および文書転送コンカレント・プログラムを実行して XML 文書を送信することができます。文書転送計画プログラムによって、転送用にトランザクションがスケジュールされます。文書転送では、文書を作成および転送するために Oracle XML Gateway をコールするワークフロー・プロセスによって、サブスクライブされるビジネス・イベントが呼び出されます。
- 「AR 通知」項目タイプ： 文書転送コンカレント・プログラムで技術的な、または転送のエラーが発生すると、転送ステータスは「障害発生」に変化し、例外処理が行われるように、システム管理者または Receivables ユーザーにワークフロー通知が送信されます。

さらに、Receivables ユーザーは、取引先の支払部門から送信された確認メッセージを、Oracle XML Gateway 経由で受信することができます。これらのメッセージで、文書のインポート・ステータスを確認します。Receivables プログラムによってメッセージが受信され、必要に応じて、例外処理のため、Receivables ユーザーに自動的にワークフロー通知が送信されます。

AR 与信管理申請プロセス・ワークフロー： このワークフローでは、回収や勘定分析を管理したり与信データを予想したりするだけでなく、与信決定を行ったり実装したりします。

このワークフローは、オーダーの保留などの与信関連のイベントまたはアプリケーションの実行のどちらかによって与信申請が生成されると、開始されます。ワークフローでは、最初に与信検討のプロセスを自動的に完了させようとします。ワークフローの機能になんらかの理由で障害が発生した場合は、適切な信用分析家へ処置を求めて信用分析がルーティングされます。

他の担当者によって与信勧告が検討され、承認される必要がある組織では、承認階層に沿って勧告がルーティングされます。

- 適切な担当者によって勧告が承認された場合は、ワークフローは自動的に実装されません。

- 承認階層に沿ってルーティングされる過程で勧告が否認された場合は、適切な担当者へ通知が送信され、ケース・フォルダが信用判断で更新されます。

Oracle US Federal Financials

予算実行トランザクション承認プロセス： 予算トランザクションが承認プロセスでルーティングされます。承認を求めて文書をルーティングするために、**Budget Execution** 内の「予算ユーザーの定義」ウィンドウ内で定義された承認コントロールと階層が使用されます。予算実行トランザクション承認プロセスは、次のウィンドウ内で「承認 ...」ボタンをクリックすると開始されます。

- 充当の入力
- 資金配分の入力
- 取引の再プログラミング
- 予算取引要約

トランザクションが承認のために送信されると、資金チェック・プロセスが開始され、十分な資金が利用できるかどうかを検査されます。資金チェックに通らないと、トランザクションは承認されません。トランザクションは、総勘定元帳へ転送される前に承認されている必要があります。

HRMS アプリケーション

Oracle Federal Human Resources

GHR 人事処理プロセス： データ・エントリの人事申請処理 RPA フォームをルーティングできるようにし、署名および最終的な承認前の検討を行って、データベースを更新します。ユーザーは、エージェントの実行に基づいて、個人、グループボックスまたはルーティング・グループのルーティング・リストに RPA を送信できます。RPA がルーティングされると、処理の履歴が記録されます。履歴を参照すると、実行された処理、実行者および実行日がわかります。

GHR 職階摘要プロセス： データ・エントリの職階摘要フォームをルーティングできるようにし、署名、検討および分類を行います。ユーザーは、エージェントの実行に基づいて、個人、グループボックスまたはルーティング・グループのルーティング・リストに職階摘要フォームを送信できます。職階摘要フォームがルーティングされると、処理の履歴が記録されます。履歴を参照すると、実行された処理、実行者および実行日がわかります。

GHR 等級変更プロセス： 職階変更 (WGI) の処理を、手動の作業を介さずに自動化できるようにします。デフォルトの WGI プロセスは、WGI が承認されたことを **Personnel Office** に自動的に通知するため、ユーザーは応答する必要がありません。WGI プロセスは、導入の際にエージェントの実行に基づいて様々な方法で設定できます。

Oracle Human Resources

タスク・フロー項目タイプ： Oracle Human Resources には、あらかじめ定義されているワークフロー（HR Task Flow）の項目タイプが用意されており、これを使用してタスク・フローを設定できます。HR Task Flow の項目タイプには、HR アプリケーションの各ウィンドウの関数アクティビティが含まれており、このアクティビティによって1つのタスク・フローに統合できます。これらの定義済の関数アクティビティを使用して、それぞれのタスク・フローに対するワークフロー・プロセスをモデル化できます。また、関数アクティビティにはアクティビティ属性が含まれており、この属性の設定によって、対象のアプリケーション・ウィンドウで、ボタン・ラベルの作成やボタンの位置定義を行うことができます。

HR Task Flow の項目タイプでは、タスク・フローを設定および管理するためのフォームを使用することもできます。Oracle Workflow と統合すると、グラフィカルな Oracle Workflow Builder を使用し、一連のウィンドウを簡単に設計および作成できます。

Oracle Internet Time

PA タイムカード承認： このプロセスは、従業員が Oracle Internet Time でタイムカードを送信したときに開始されます。このワークフローは、すべてのタイムカードを自動的に承認するか、事前に決定された承認プロセスを介してタイムカードをルーティングするように構成できます。この承認プロセスでは、マネージャおよび従業員に通知を送信し、タイムカードが会社の方針に準拠していることを確認し、マネージャ承認レベルをチェックします。送信されたタイムカードのステータスは、すべての承認プロセスにわたって監視できます。

Oracle Self-Service Human Resources

候補者内定承認プロセス： 「ライン・マネージャ・ダイレクト・アクセス」の「候補者内定」オプションを使用して提示された内定が、承認階層の該当するマネージャに送信されます。承認階層の最後の承認者が内定を承認すると、内定通知を印刷、署名し、候補者に送信して候補者の応答を待つようにという通知が、ワークフローによって人事部門に送られます。候補者が内定に応答すると、元のマネージャに通知され、ワークフローが完了します。このプロセスの実行中は、内定のステータスが元のマネージャに絶えず通知されます。

キャリア・マネージメント・レビュー・プロセス： レビュー担当者に「評価と査定」の通知が送信されます。

360 度評価プロセス： 特定の人に、グループの一員として評価を行う必要があるという通知を送信します。

訂正に対する承認プロセス： 個人情報の住所の変更には承認が必要でも電話番号の変更には承認が必要ない、というようにワークフロー・プロセスを構成することができます。そのかわり、職責の条件によって承認の要件を変更し、従業員が変更したレコードは承認が必要でも、管理者が変更したレコードには承認が不要、というようにすることができます。

従業員情報のリリース： 従業員が従業員情報のリリース関数を有効にして、アクセス権限のない他の従業員や、時には管理者と、従業員同士の情報を共有できるようにするとこのワークフローが開始されます。同様に、管理者はこの関数を使用して、ある従業員についての情報を別の管理者と共有することができます。

ユーザー名の作成プロセス：すでに HR データベースにレコードがあるユーザーが自身のセルフサービスのユーザー名とパスワードを作成するためのページが表示されます。

報酬ディストリビューション：報奨を配分する管理者の監督者へ自動的に通知が送信されます。管理者の監督者は、配分を承認または否認します。

その他のプロセス： Oracle Self-Service HR には、従業員およびマネージャが、承認された個人情報の詳細を Web 上で参照、更新および表示するプロセスが含まれています。具体的には、次の情報が対象となります。

- 基本詳細（名前、婚姻区分など）
- 住所
- 電話番号
- 連絡先
- 履歴書
- 資格
- 個人コンピテンス・プロフィール
- 学校と大学就学
- 勤務選択
- 従業員登録
- ユーザー名の作成プロセス
- 退職
- 給与
- 新規採用
- 組織管理者
- 休暇管理

Oracle Self-Service HR には、従業員がクラスに登録するプロセス、および応募するプロセスも含まれています。

Oracle Time and Labor

OTL シード承認ワークフロー： Oracle Time and Labor には、タイムカードを承認が必要な適切な担当者へルーティングする単一のワークフローが用意されています。このワークフローをテンプレートとして使用し、定義する承認スタイルに必要な数だけワークフローを作成します。

Oracle Training Administration

OTA Workflow プロセス： Self-Service Oracle Training Administration のワークフローおよび注文管理を介した Training Administration のワークフローで構成されます。

■ Self-Service Oracle Training Administration

- 既存のイベントに空きがあるかどうかを確認し、待機リストへの登録または配置を受講者に通知します。
- イベントの直前に登録の取消しが発生した場合は、コストの転送が発生し、顧客に登録費用を請求します。
- Oracle Training Administration が相互請求を自動作成するように設定されているときは、「転送元」または「転送先」の値が見つからない場合に、イベント所有者に通知します。その場合は、経理上のヘッダーと明細を手動で作成する必要があることを管理者へ知らせます。
- 登録申請の作成者に登録ステータスの変更を通知します。
- 登録者および管理者に登録の取消しを通知します。
- 承認の否認が原因の登録の取消しを登録者へ通知します。
- イベントが予定の日時に近い時期に取り消される場合は、イベント所有者に、登録者を待機リストに手動で移すように促します。
- 自動登録を妨げるデータベースのエラーをトレーニング管理者へ通知します。
- 特定のイベントへの出席レコードが記録、更新または削除されたことを、登録者へ通知します。
- 登録承認後のデータベース・エラーをシステム管理者へ通知します。

■ 注文管理を介した Training Administration

- 受注明細に対する請求を、受講者がコースを完了した後で生成します。
- イベントまたは登録の取消しを受講者に通知します。
- イベントの所有者に登録または注文明細の取消しを通知し、待機リストから受講者を登録します。
- 取り消されたイベントの所有者に記録済のリソースを手動で削除するように促します。

- イベントの最大受講者数が増加したときは、イベントの所有者に通知します。
- 登録しているイベントを顧客が変更したときは、イベントの所有者に通知します。

賃貸契約

Oracle Lease Management

OKL CS 機器交換： このワークフローは、必要な承認を受けた後に装置の詳細部分を更新する機器置交換要求で使用されます。このワークフローには、「要求」タブからアクセスします。

OKL CS 移動引受要求： このワークフローは、必要な承認を受けた後に契約管理者へ通知するために、移動引受要求で使用されます。このワークフローには、「要求」タブからアクセスします。

OKL CS 請求訂正要求： このワークフローは、申請された訂正を契約管理者へ通知するために、請求書修正要求で使用されます。このワークフローには、「取引」タブからアクセスします。

OKL CS 請求払戻要求： このワークフローは、申請された返品を契約管理者へ通知するために、請求払戻要求で使用されます。このワークフローには、「アカウント」タブからアクセスします。

OKL CS 利息タイプの換算： このワークフローは、利息タイプの変更を契約管理者へ通知するために、利息の換算要求で使用されます。このワークフローには、「体系」タブからアクセスします。

OKL CS 契約リース更改： このワークフローは、承認済の契約更新を契約管理者へ通知するために、更改見積で使用されます。このワークフローには、「要求」タブからアクセスします。

OKL - LA: リース契約承認： このワークフローは、契約の有効化に必要な承認を得るために使用されます。このワークフローには、「記帳」ウィンドウからアクセスします。

OKL - LA: 契約資金調達承認： このワークフローは、資金調達に必要な承認を得るために使用されます。このワークフローには、「資金調達要求」ウィンドウからアクセスします。

OKL ストリーム生成 - アウトバウンド： このワークフローは、ストリーム生成のためにアウトバウンドの XML Pricing Engine のトランザクションを開始するために使用されます。このワークフローには、ストリーム生成プロセスからアクセスします。

OKL ストリーム生成 - インバウンド： このワークフローは、ストリーム生成のためにインバウンドの XML Pricing Engine のトランザクションを管理するために使用されます。このワークフローには、ストリーム生成プロセスからアクセスします。

OKL - 第三者保険情報の収集： このワークフローは、顧客が保険証明を提示できず、賃貸者が保険を販売できないときに、第三者の情報を収集するために使用されます。このワークフローには、「コンカレント・マネージャ」からアクセスします。

契約資産管理承認： このワークフローでは、承認を求めて契約ポートフォリオがルーティングされます。このワークフローには、「契約資産管理」からアクセスします。

契約資産管理実行通知： このワークフローでは、契約ポートフォリオの実行を再売買者へ通知します。このワークフローには、「契約資産管理」からアクセスします。

再構築の承認： このワークフローでは、承認を求めて再構築の申請がルーティングされます。このワークフローには、「見積再構築」からアクセスします。

損益の承認： このワークフローでは、承認を求めて契約満了時の損益がルーティングされます。このワークフローには、「終了見積」からアクセスします。

部分見積りの承認： このワークフローでは、承認を求めて部分的な見積りがルーティングされます。このワークフローには、「終了見積」からアクセスします。

出荷指示： このワークフローは、資産の輸送明細を送信するために使用されます。このワークフローには、「見積連絡」からアクセスします。

内部輸送部門に通知： このワークフローは、資産の輸送明細を社内の輸送部門へ知らせるために使用されます。このワークフローには、「見積連絡」からアクセスします。

見積の送信： このワークフローは、見積りを送信するために使用されます。このワークフローには、「見積連絡」からアクセスします。

買戻しの受入： このワークフローでは、買戻しの見積りを受け入れることをユーザーに知らせます。このワークフローには、「見積受入」からアクセスします。

終了見積受入： このワークフローでは、契約満了時の見積りを受け入れることをユーザーに知らせます。このワークフローには、「見積受入」からアクセスします。

見積再構築の受入： このワークフローでは、再構築の見積りを受け入れることをユーザーに知らせます。このワークフローには、「見積受入」からアクセスします。

再販担当者に通知： このワークフローでは、新しい資産配分を再売買チームへ知らせます。このワークフローには、「資産返却」からアクセスします。

回収部門に通知： このワークフローでは、取戻しの場合の返却の更新を回収代理人へ知らせます。このワークフローには、「資産返却」からアクセスします。

再所有エージェントに通知： このワークフローでは、新しい差押え申請を外部の差押え代理人へ知らせます。このワークフローには、「資産返却」からアクセスします。

プロバイダに通知： このワークフローでは、資産返却をサービスおよびメンテナンス業者へ知らせます。このワークフローには、「資産返却」からアクセスします。

資産修理： このワークフローでは、資産修理の要請が承認を求めてルーティングされます。このワークフローには、「資産状態」からアクセスします。

所有権返還要求： このワークフローでは、契約の失効時に権利証書を返却するように、第三者の権利保有者へ申請します。このワークフローには、「再販」からアクセスします。

再販注文サイクル： このワークフローでは、注文管理内で再売買される資産の注文サイクルを定義します。このワークフローには、「再販」からアクセスします。

メンテナンス・アプリケーション

Oracle Enterprise Asset Management

EAM 作業要求承認プロセス： このプロセスが **Enterprise Asset Management** パラメータ（「自動承認」チェック・ボックス）で使用可能にされた場合は、作業申請を出された資産を現在所有している部門に割り当てられた職責を持つユーザーが、作成された作業申請に対処できるように電子メール通知を受け取ります。これらの通知は、ログインに使用する **EAM 保守ユーザー** という職責の範囲内で表示することができます。通知を受け取ったユーザーはすべて、作業申請にアクセスして、そのステータスを変更したり、作業申請のログに情報を追加したりすることができます。一人のユーザーが作業申請を承認すると、同じ承認グループに属する他のユーザーのワークリストからその通知が削除されます。また、承認や情報の追加のために、通知を別のユーザーに再度割り当てることもできます。

このプロセスが使用可能にされると、「作業指示の待機（承認済）」ステータスの作業申請が作成されます。使用可能でない場合は、「オープン」ステータスの作業申請が作成されます。

製造業関連のアプリケーション

Oracle Engineering

設計変更プロセス： 設計変更が適切な承認担当に送信されます。

Oracle Inventory

INV: 移動オーダー承認： 移動指示の承認を求めて通知と督促が承認者へ送信され、提案された移動の通知が申請者と在庫表に関連する通知リストに載っている従業員へ送信されません。

資材不足メッセージ： 資材不足の品目の受領トランザクションが処理されるごとに、プランナへ通知が送信されます。このワークフロー通知は、トランザクション・マネージャによってマークされたすべての資材トランザクションをチェックするコンカレント・プログラムから開始されます。

Oracle Master Scheduling/MRP

計画例外メッセージ： 資材およびリソースの計画を生成するときに訂正処置をすばやくできるようにするために、自動的にプロセス例外を使用可能にします。例外メッセージは、特定の職責にある主要な担当者へルーティングされます。社内の担当者と同様に、サプライヤおよび顧客も担当者にすることができます。

Oracle Process Manufacturing (OPM)

OPM Inventory

品目の活用化ワークフロー：新しく作成された品目のステータスを承認済の在庫ステータスに変更する前に通知が送信されます。このワークフローでは、品目の作成を承認する必要があるユーザー個人へ通知が送信されます。ルーティングの結果、承認されればその品目は有効になり、承認されなければ無効のままです。品目の承認が保留されている場合、品目のプロパティを変更することはできません。

ロット失効 / 再テスト・ワークフロー：定義された、ロットの期限切れおよび再テストの数日前の日付になると、ユーザーに通知が送信されます。このワークフローがないと、ロットの期限切れや再テストの情報は、ユーザーが問い合わせないかぎり、表示されません。ロット期限切れ / 再テスト・ワークフローによって、ロットの期限切れおよび再テストの日付より前の定義済の日数の日付になると承認者へ通知されることで、情報が表示されるようになります。

OPM Quality Management

サンプル作成通知：仕様および実質的な仕様検証規則に関連するサンプル計画によって申請されたとおりの物理的なサンプルを作成する通知が送信されます。このワークフローは、在庫数量の向上、トランザクションの受信、バッチ・ステップ・リリース、またはロットの期限切れまたは再テストにトリガーされるようにすることができます。通知には、サンプル計画情報のリストと「サンプル」ウィンドウへのリンクが含まれます。このウィンドウには、サンプル・イベントについての情報を入力します。

テスト通知：特定のサンプルに対してテストを実行し、結果を入力するように指示する通知が送信されます。このワークフローは、すでに仕様に関連付けられているか、または一定のテストを再テストするために結果処理から選択したサンプルを作成することによって開始されます。プロセス内で指定されたサンプルの各試験者は、予想されるテスト結果についての通知を受け取ります。

サンプル処分通知：いったんテストが完了したサンプルの最終的な処置について指示されます。関連する仕様が必要な場合には、すべてのテストが評価され、終了した結果が記録されたときに、サンプルの処置を、承認、特例付き承認または否認に変更することができます。

複合結果通知：同じサンプル・イベント内のサンプルのセット全体の結果を複合するかどうかを質問されます。このワークフローは、サンプル・イベント内のすべてのサンプル（サンプル計画に必要なサンプルの数に基づく）のテストが終了し、各サンプルに最終的なサンプル処置が割り当てられたときに開始されます。通知には、「複合結果」ウィンドウへのリンクが含まれます。このウィンドウにサンプル・イベントについての情報が表示されます。

サンプル・グループ処分通知：サンプル・イベントには、サンプル計画で必要なサンプル数に基づいた最終的な処置が要求されていることが通知されます。このワークフローは、サンプル・イベント内のすべてのサンプルの最終的な処置が、受諾、特例付き受諾、否認または取消しになったときに起動されます。

Oracle Project Manufacturing

間接 / 資本プロジェクト定義プロセス： このプロセスは、プロジェクト製造プロジェクト定義プロセスのナビゲータ・フローの一部として機能します。このプロセスでは、**Oracle Project Manufacturing** で使用する間接または資本タイプのプロジェクトを設定するときに必要な一連の手順の説明が表示されます。

契約プロジェクト定義プロセス： このプロセスは、プロジェクト製造プロジェクト定義プロセスのナビゲータ・フローの一部として機能します。このプロセスでは、**Oracle Project Manufacturing** で使用する契約タイプのプロジェクトを設定するときに必要な一連の手順に関する説明が表示されます。

Oracle Quality Online

従業員通知： 指摘された欠陥や改善案についての特定のイベントに関心を示した従業員に通知します。

重大な欠陥： 最近重大な欠陥を割り当てられた所有者に通知します。

Oracle Work in Process

中間品出荷： 外注組立部品を、作業現場から出荷場所、さらにサプライヤまで追跡できるようにします。中間物出荷ワークフローは、組立部品を **PO Move** リソース・チャージ・タイプの外注作業のキュー処理手順へ移すとき、または外注作業が最初の処理の場合に、アクティブにされます。

発注希望入手日の変更： 外注作業を再スケジュールするときに、発注書の入用期日を更新するワークフロー・プロセスをアクティブにして、変更のための承認が必要な日付をバイヤーおよびサプライヤに通知します。承認されない場合は、手動で日付を変更することができます。「工程」ウィンドウからアクセスする「発注希望入手日の更新」ウィンドウで、このワークフローをアクティブにすることができます。

オーダーの履行

Oracle Configure To Order (CTO)

CTO 変更オーダー： 出荷組織のプランナに ATO オーダーに加えられた変更の詳細を通知します。このプロセスは、オーダー数量、予定出荷日、要求日または予定到着日が変更され、オーダーに対して個別のジョブ予約またはフロー・スケジュールが存在するときに開始されます。また、構成の変更や受注明細の取消しが発生したときは、予約が存在しない場合でも開始されます。

Oracle Order Management

Order and Line Runnable プロセスおよび Functional サブプロセス： Oracle Order Management には、Seeded Order and Line Runnable プロセスおよび Functional サブプロセスが用意されています。注文ヘッダー・ワークフロー・データは、項目タイプ「OM 注文ヘッダー (OECH)」に定義されています。シード済ヘッダー実行可能プロセスは、標準の承認のある受注と返品を処理するために用意されています。Booking and Close Order 機能サブプロセスもシードされています。Order Line ワークフロー・データは、項目タイプ「OM 受注明細 (OEOL)」に定義されています。Oracle Order Management には、明細レベルの実行可能プロセスが用意されており、標準項目、構成、サービス項目、直送およびその他の項目を処理することができます。受注明細のスケジュール、出荷、履行、請求書インタフェースおよびクローズに関する機能サブプロセスもシードされています。また、シード済機能サブプロセスおよびカスタム・アクティビティを使用して、特定のビジネス要件に合わせてカスタム・プロセスを構成することもできます。シード済プロセスおよびカスタム実行可能プロセスは、特定の受注および明細取引タイプに割り当てることにより、オーダー処理に使用できます。

Change Order Notification: 特定の申請フォームからオーダー変更通知を送信します。職責を選択し、内容を指定すると、受信者およびメッセージの内容が動的に設定されます。項目タイプ「OM 発注変更 (OECHORD)」を使用します。

COGS: 項目タイプ「Generate Cost of Goods Sold Account (OECOGS)」を使用して、売上原価勘定を生成します。

Oracle Warehouse Management

MTL 取引事由ワークフロー： 取引根拠のアクションによって、例外が発生したときに、代替のカスタマイズ可能な関数フローがユーザーに提供されます。このワークフローには、現在 2 つのシードされたワークフロー・プロセスがあります。

- 循環棚卸： 数量の不一致がある場所のサイクル・カウントを要求し、その後通知を送信します。
- WMS N ステップ棚入： システムから提案された場所とは異なる中間的な場所へユーザーが品目を格納できるようにします。システムによって、中間的な場所から提案された場所へ品目を移動するための別のタスクが作成されます。

調達

Oracle Purchasing

調達ワークフロー： 調達ワークフローは、サプライ・チェーンのメンバー全員のための、完全に自動化された柔軟で拡張性のあるトランザクション処理システムです。このワークフローは、より戦略的な部品外注および調達アクティビティに対する主要な実現手段の 1 つです。このワークフローは、文書承認、自動文書作成、変更オーダー、勘定科目生成、通知の送信、価格 / 販売カタログ通知および入金確認 (Self-Service Purchasing のみで使用) のワークフロー・プロセスによって構成されます。

文書承認プロセス： Oracle Purchasing の承認に関するすべてのアクティビティを実行します。具体的には、文書の発行、承認、転送、承認通知、否認などがあります。このプロセスには、購買発注を承認するための発注承認ワークフロー・プロセスと、購買申請を承認するための発注依頼の承認ワークフロー・プロセスが含まれています。

自動文書作成プロセス： 購買申請明細に必要なソース情報が含まれている場合は、承認済みの購買申請明細を使用して、包括契約に対する標準発注または発注リリースが自動的に作成されます。このプロセスは、発注文書作成ワークフローとも呼ばれます。

変更オーダー・プロセス： 変更によって、手動による再承認が必要であるか、自動的に再承認されるかを制御します。手動または自動で再承認されたすべての文書では、文書の改訂を表す番号が増加します。このプロセスは、発注承認ワークフローの一部として機能します。

通知の送信プロセス： 不完全な文書、否認された文書または再承認が必要な文書をロックし、この文書ステータスに関する通知を該当メンバーに通知します。このプロセスは、購買文書ワークフローの発注送信通知としても機能します。

価格 / 販売カタログ通知プロセス： 購買文書オープン・インタフェースを介して送信された価格 / 販売カタログ情報に、設定した金額範囲を超える金額の増分が検出された場合に、購買担当に対して通知を送信します。このプロセスは、発注カタログの価格許容範囲超過通知ワークフローとしても機能します。

Oracle Self-Service Purchasing

入金確認プロセス：申請者に対して、それぞれの申請が受理されたことを表す受領通知を送信します。このプロセスは、発注受入確認ワークフローとしても機能します。

「購買承認申請」プロセス：Web 購買から作成された購買申請が承認のために該当するマネージャに送信され、購買申請のステータスが更新されます。

Oracle Web Suppliers

Supplier Self-Service 登録承認プロセス：Oracle Web Suppliers では、ゲストとしてログインしたユーザーを、会社の仕入先担当として登録できます。このプロセスでは、登録の検証と承認のために、該当するアカウント承認者に通知がルーティングされます。承認者が登録を承認すると、「Supplier Web ユーザー」アカウントがアクティブ化されます。アカウント承認者が登録を否認すると、そのアカウントは非アクティブ化されます。

プロジェクト

Oracle Projects

プロジェクト承認とステータス変更プロセス：プロジェクトがルーティングされ、該当するユーザーにプロジェクト・ステータスの変更が通知されます。たとえば、プロジェクトを承認処理のために送信したり、プロジェクトの完了時に該当する担当者に通知できます。該当するステータスの変更のみでなく、プロジェクトのルーティング先となる担当者の決定に使用するワークフローを選択します。

予算承認プロセス：プロジェクト予算が承認処理とベースライン化のためにルーティングされます。予算タイプのみでなく、予算のルーティング先となる担当者の決定に使用するワークフローを選択します。

ステップ・ダウン配賦プロセス：ステップ・ダウン処理自動配賦セットの実行を自動化します。つまり、配賦実行の作成、配賦トランザクションの生成、配賦トランザクションの解放（プロセスの続行に必要な承認の要求）、コストの配布、およびプロジェクト要約金額の更新を行います。

プロジェクト予算勘定科目ワークフロー：プロジェクトの予算が総勘定元帳の予算に統合されるときに、各プロジェクト予算の明細に勘定科目が割り当てられる必要があります。勘定科目は、プロジェクト予算金額を General Ledger に連動させるために使用されます。プロジェクト予算勘定科目ワークフローによって、勘定科目の生成と割当てプロセスを自動化できるようになります。

販売、マーケティングおよび電子取引

Oracle Incentive Compensation

契約承認プロセス： Oracle Incentive Compensation では、Sales Force Planning モジュール内で契約承認ワークフローが使用され、経営陣による承認の合意を処理し、その合意書を販売スタッフに配布し、営業員がその合意書を受け取るようします。ユーザーが承認に対する合意を送信すると、契約承認プロセスが開始されます。すべての承認が行われて営業担当者が合意書を受け取ると、それが報奨計画としてアクティブにされます。契約承認プロセスでは、この合意書を配布することを忘れないように、販売スタッフが合意書を受け取るように、および販売部門の管理者へ合意書がまだ受け取られていないことを警告するために、メッセージが送信されます。

Oracle iStore

Oracle iStore には、事前定義済の通知および通知メッセージがあるシードされたワークフローが用意されています。iStore アラート・ワークフローにはレポートとは関連のない通知が含まれており、レポート関連の通知は iStore Reports Alerts に属しています。電子メールによる通知メッセージは、様々なイベントに基づいてユーザーへ送信されます。

オーダー関連の通知

オーダー確認 - 通常： 標準的なオーダーを確認します。この通知は、ユーザーが Web 専門ストアに注文すると開始されます。

オーダー確認 - FAX によるオーダーの次の手順： オーダーを送信する残りの手順を説明します。この通知は、ユーザーが注文し、支払方法としてクレジット・カード番号または購入注文書をファックスで送信することを選択すると、開始されます。

オーダー未記帳通知： オーダーがまだ記帳されていないことをシステム管理者へ知らせます。この通知は、ユーザーのオーダーが記帳されていないと開始されます。

オーダーの取消： 取り消されたオーダーを確認します。この通知は、ユーザーが Web 専門ストアを使用してオーダーを取り消すと、開始されます。

共有ショッピング・カート関連の通知

カートの共有確認通知： 共有カート・イベントを確認し、共有カートへのアクセス方法を共有カートの所有者および受信者へ説明します。この通知は、ユーザーが別のユーザーとカートを共有すると、開始されます。

アクセス・レベル変更： アクセス・レベル変更についてユーザーへ通知します。この通知は、共有カートの所有者が受信者のアクセス・レベルを変更すると開始されます。

受取者別カート・アクセスの削除： 共有カートのステータスについてユーザーへ通知します。この通知は、共有カート受信者が共有カートの使用を終了すると、開始されます。

共有の停止：共有カートが所有者によって共有を解除されたことをユーザーへ通知します。この通知は、共有カート受信者が共有カートの共有を停止するときに開始されます。

販売支援関連の通知

販売アシスタンス要求 - 対営業担当：販売代理業者へ販売支援の要請を記述します。この通知は、チェックアウト・フロー中にユーザーが販売支援を要請すると、開始されます。

営業アシスタンス要求：販売支援を要請したユーザーへ確認を送信します。この通知は、チェックアウト・フロー中にユーザーが販売支援を要請すると、開始されます。

ユーザーと登録関連の通知

ユーザー登録：新しく登録されたユーザーを歓迎します。この通知は、ユーザーが専門ストアに登録すると、または B2B ユーザーが B2B プライマリ・ユーザーによって登録されると、開始されます。

ログイン情報の喪失：ユーザーへそのユーザーのユーザー名とパスワードを送信します。この通知は、登録ユーザーがログイン情報を要求したときに開始されます。

契約関連の通知イベント

契約ネゴシエーション要求 - 否認：契約期間の変更申請の否認を知らせます。この通知は、契約管理者が契約期間の変更についてのユーザーの要求を否認すると、開始されます。

契約ネゴシエーション要求 - 対営業担当：契約期間の変更についての申請を記述します。この通知は、ユーザーが契約期間の変更を申請すると、開始されます。

契約ネゴシエーション要求 - 承認：契約期間変更の申請が承認されたことを知らせます。この通知は、契約管理者が契約期間の変更についてのユーザーの申請を承認すると、開始されます。

契約ネゴシエーション要求 - 取消：契約折衝の取消しを知らせます。この通知は、ユーザーが契約期間の変更を申請したときに作成された契約を契約管理者が取り消すと、開始されます。

契約ネゴシエーション要求 - 対ユーザー：契約期間の変更についての申請を確認します。この通知は、ユーザーが契約期間の変更を要求すると、開始されます。

レポート関連の通知イベント

レポート - iStore 履歴要約：ストア管理 UI で指定されたストア・オーダー要約データ出力ビンの情報が含まれています。この通知は、iStore Reports コンカレント・プログラムによってレポート・データが更新されると、開始されます。

レポート - iStore トップ・オーダー：ストア管理 UI で指定されたトップ・オーダー・ビンの日次最上位のオーダーが含まれています。この通知は、iStore Reports コンカレント・プログラムによってレポート・データが更新されると、開始されます。

Oracle Marketing Online

OMO トリガー： ユーザーに設定された一定の条件を監視し、条件が満たされる場合は、ユーザーがトリガーに関連付けたスケジュールを実行します。

サービス

Oracle Service

サービス要求プロセス： サービス要求が組織内の個人にルーティングされ、処理されます。このプロセスをカスタマイズして、サービス担当を選択して通知するのみでなく、組織のサービス・ルールとガイドラインに基づいてサービス要求を自動的に転送し、エスカレートさせることができます。

サービス要求処理と手配プロセス： サービス要求処理が解決のために組織内の個人にルーティングされるだけでなく、現場で作業手配が必要な該当のサービス担当に指示が通知されます。このプロセスをカスタマイズして、作業手配要求を管理、転送またはエスカレートさせることができます。

フィールド・サービス作業手配プロセス： サービス要求データをインタフェース表に対して挿入または更新し、フィールド・サービス・エンジニアに対して、ディスパッチ情報の通知を送信します。このプロセスは、Oracle Mobile Field Service で使用されます。

サプライ・チェーン計画

Oracle Advanced Supply Chain Planning

事前計画例外メッセージ・プロセス： 仕入先、得意先の担当者または社内の担当者に対して、事前計画の例外を通知し、通知を受信した担当者が、対象の計画例外に対して適切な処置を開始できるようにします。

割付 ATP プロセス： 注文予約要求を満たすために異なる供給ソース間で補充が発生した場合または注文スケジュール・プロセスが失敗した場合に、プランナに通知を送信します。

Oracle Demand Planning

MSD 需要計画サイクル・プロセス： MSD 需要計画サイクルでは、需要計画サイクルのすべてのバックグラウンド処理が管理されます。需要計画サイクルは複数のステージで構成されています。それぞれのステージで、実行されるタスクを管理するワークフロー・プロセスが開始されます。各ステージは、「需要計画管理者」画面から開始します。ワークフローが各ステージを処理すると、管理者とユーザーに対して関連するプロセスのステータスを通知します。

需要計画の処理サイクルは、次の5つのステージで構成されており、各ステージはプロセスに対応しています。

- 計画サーバーからのデータのダウンロード： 需要計画サーバーのデータを転送および変換し、そのデータを分析するために Express Server Demand Planning エンジンに渡します。
- 移入指標： 統計的予測を実行し、定義済の各測定単位を入力します。
- プランナに配布： 需要プランナのコミュニティでデータを利用できるようにして、各プランナにデータが準備できていることを通知します。
- 需要プランナからのデータの収集および統合： 需要プランナから送信されたデータを収集し、需要計画エンジンでデータを統合します。このプロセスは、収集期間の終了に指定した日付まで、繰り返し実行することができます。
- 計画サーバーへの統合したデータのアップロード： 変換したデータを計画サーバーに転送します。

需要計画処理サイクル中の任意の時点で、需要計画管理者は「すぐに使用可能な発行の収集」という特殊なプロセスを実行することができます。このプロセスで、サイクルの終了前に、待機中の確認用データ送信が収集されます。

需要計画管理者は、このマスター・プロセスを起動して管理するためのコンカレント要求を実行することによって、5つのステージすべてを一括管理する ODP Master と呼ばれるマスター・ワークフローを実行することもできます。このコンカレント要求は、需要計画エンジン・マスター・ワークフロー・プロセスと呼ばれます。

Oracle E-Business Suite への Oracle Workflow ビジネス・イベント・システムの実装

次にあげる製品では、Oracle Workflow ビジネス・イベント・システムを利用してビジネス・プロセスが統合されます。各機能の詳細は、各製品のユーザーズ・ガイドまたは構成ガイドを参照してください。

関連項目：

C-39 ページ「[Oracle Workflow のサポート・ポリシー](#)」

適用されるテクノロジー

Oracle XML Gateway

Oracle XML Gateway は、Oracle Workflow ビジネス・イベント・システムを利用して、メッセージの作成または取込みを自動的に行うアプリケーション・ビジネス・イベントの発行および承認申請を行います。XML Gateway 実行エンジンと直接対話しながらアウトバウンド・メッセージの生成やインバウンド・メッセージの取込みを行う場合は、シード済ワークフロー関数をワークフロー・プロセスで使用できます。実行エンジンが生成したアウトバウンド・メッセージは、ダウンストリーム・ワークフロー・アクティビティで処理できます。実行エンジンは、ワークフロー・プロセスから渡されたインバウンド・メッセージを取り込みます。

XML Gateway では、XML Gateway 標準項目タイプと XML Gateway エラー・プロセス項目タイプが配信されます。

XML Gateway 標準項目タイプ： XML Gateway 標準項目タイプには、文書配信の呼出しイベントが含まれます。このイベントは、既存のワークフロー・プロセスからビジネス・イベントを呼び出すときに使用します。これにより、既存のワークフロー・プロセスと Oracle XML Gateway をシームレスに統合して、アウトバウンド XML メッセージを作成できます。XML Gateway 標準項目タイプに含まれる関数は、XML 文書の取込み、XML 文書の生成、取引先 XML 文書の生成、文書の送信、XML の変換およびトランザクション・デリバリーの必要性の判別です。

企業間またはアプリケーション間の統合を行うときは、既成の XML メッセージにシードされたイベントおよびイベント・サブスクリプションが Oracle E-Business Suite から配信されるように構成します。

XML Gateway エラー・プロセス項目タイプ： XML Gateway エラー・プロセス項目タイプに含まれるエラー処理プロセスは、Oracle Workflow ビジネス・イベント・システムまたは Oracle XML Gateway によって検出されたエラーを管理します。エラー・プロセスには、デフォルト・エラー・プロセス、ECX エンジン通知プロセス、ECX メイン・エラー・プロセス、ECX メイン・インバウンド・エラー・プロセス、ECX メイン・アウトバウンド・エラー・プロセス、インバウンド・メッセージ用エラー・プロセスおよびアウトバウンド・メッセージ用エラー・プロセスがあります。

XML Gateway エラー・プロセス項目タイプでは、エラー・イベントの受信イベント・アクティビティと送信通知イベントの受信イベント・アクティビティがサポートされます。エラー・イベントの受信は、XML Gateway 実行エンジンがエラーを検出したときに XML Gateway によって使用されます。送信通知イベントの受信は、インバウンド・プロセスに関連するエラーを実行エンジンから通知する必要があるときに、XML Gateway によって使用されます。

Oracle Workflow のエラー処理では、XML Gateway のシステム管理者または取引パートナーに対してエラー通知が有効になっており、ワークフローの再試行および再処理機能を使用できます。XML Gateway エラー・プロセス項目タイプによって提供される関数は、ECX 再処理インバウンド、ECX 再送信アウトバウンド・メッセージ、ECX In Error 詳細の取得、ECX Out Error 詳細の取得、システム管理者ロールの取得および取引先ロールの取得です。

詳細は、『Oracle XML Gateway ユーザーズ・ガイド』を参照してください。

会計関係のアプリケーション

Oracle Payables

電子メール送金通知プログラム： 支払バッチを確定するかクイック支払を作成すると、ビジネス・イベント・システムによってこのプログラムが開始され、送金通知電子メール・アドレスが定義されている各サプライヤに電子メールが自動的に送信されます。

XML 支払の送信： このビジネス・イベントは、OAG XML 支払プログラムを使用する支払または支払バッチをユーザーが作成するときに呼び出されます。このイベントで、XML 支払メッセージを FSP（取引銀行）へ送信する支払メッセージの処理ワークフローが開始されます。

XML 支払確認： このビジネス・イベントは、FSP（取引銀行）が BOD XML メッセージを ERP へ送信すると、呼び出されます。このイベントにより、支払メッセージの処理ワークフローが続行され、完了します。

XML 支払エラー： このビジネス・イベントは、FSP（取引銀行）が支払エラー XML メッセージを ERP へ送信すると、呼び出されます。このイベントにより、支払エラー・メッセージ処理ワークフローが開始されます。

XML 支払助言： このビジネス・イベントは、FSP（取引銀行）が支払助言 XML メッセージを ERP へ送信すると、呼び出されます。このイベントにより、支払助言メッセージ処理ワークフローが開始されます。

Oracle Receivables

Oracle Receivables では、AR 与信管理申請プロセス・ワークフローが与信の検討を終える
と、`oracle.apps.ar.cmgt.CreditRequestRecommendation.implement` ビジネス・イベントが呼
び出されます。カスタムの与信決定を実装する場合は、ビジネス・イベント・システムを使用
して、このイベントに対するサブスクリプションを作成します。

製造業関連のアプリケーション

Oracle Process Manufacturing (OPM)

OPM 製品開発の電子署名イベント

処理の承認： 処理ステータスの変更が要求されたときには常に、製品開発の「ステータス
の変更」ウィンドウで署名の取込みが開始されます。処理のステータスが変更されると、プ
ロセス内の複数のポイントで電子署名を取り込むことができます。イベントをこのウィンド
ウ内で実行している間に、必要な署名がオンラインで取り込まれます。さらに署名が必要な
すべてのユーザーは、処理ステータスが要求されたステータスに変更される前に電子署名を
入力する必要があることを知らせるワークフロー通知を受信します。

工順の承認： ルーティングのステータスが変更されると、プロセス内の複数のポイントで
電子署名を取り込むことができます。ステータスの変更が要求されると、イベントをウィン
ドウ内で実行している間に、必要な署名をオンラインで取り込むことができます。追加の必
要な署名に対する責任を負うユーザーはすべて、ルーティング・ステータスが要求されたス
テータスに変更される前に電子署名が入力される必要があることを知らせるワークフロー通
知を受信します。

フォーミュラの承認： 調整法のステータスが変更されると、プロセス内の複数のポイント
で電子署名を取り込むことができます。ステータスの変更が要求されると、イベントをウイ
ンドウ内で実行している間に、必要な署名をオンラインで取り込むことができます。追加の
必要な署名に対する責任を負うユーザーはすべて、調整法のステータスが要求されたス
テータスに変更される前に電子署名が入力される必要があることを知らせるワークフロー通
知を受信します。

レシピの承認： 手法のステータスが変更されると、プロセス内の複数のポイントで電子署
名を取り込むことができます。ステータスの変更が要求されると、イベントをウィンドウ内
で実行している間に、必要な署名をオンラインで取り込むことができます。追加の必要な
署名に対する責任を負うユーザーはすべて、手法のステータスが要求されたステータスに更
えられる前に電子署名が入力される必要があることを知らせるワークフロー通知を受信し
ます。

妥当性ルールの承認： 検証ルールのステータスが変更されると、プロセス内の複数のポイ
ントで電子署名を取り込むことができます。ステータスの変更が要求されると、イベントを
ウィンドウ内で実行している間に、必要な署名をオンラインで取り込むことができます。追
加の必要な署名に対する責任を負うユーザーはすべて、検証ルールのステータスが要求さ

たステータスに変更される前に電子署名が入力される必要があることを知らせるワークフロー通知を受信します。

OPM プロセス実行の電子署名イベント

バッチのリリース： これは、バッチによって生産が開始され、自動リリースの材料が消費されたことを知らせるイベントです。バッチには実際の開始日があり、自動リリース材料には完了したトランザクションによってサポートされる、割当てが成功した場合の実際の数量があります。

バッチのクローズ： これは、バッチが終了し、以後バッチ・データには変更が行われなことを知らせるイベントです。バッチのクローズによって実際のクローズ日が割り当てられ、クローズしたバッチのステータスが変更されます。いったんバッチがクローズすると、以後そのバッチに調整、挿入または削除を行うことはできません。

OPM 在庫イベント

品目イベント： 既存の品目に変更されるときに、必要な電子署名がオンラインで取り込まれて、その品目を使用する必要があるトランザクションが中断せずに機能できるようになります。

ロット・ステータス・イベント： OPM ロット・マスターの承認とロット・マスターの更新がオンラインで実行されます。「ロット・マスター」ウィンドウを使用してロットまたはサブロットを作成または更新しているときに、データを保存すると電子署名のイベントが呼び出されます。その結果、「電子署名」ウィンドウが表示されます。新規または更新されたデータは、「電子署名」ウィンドウが開いている間に必要な署名のすべてがオンラインで取得されないかぎり、データベースへ確定されません。

数量イベント： 在庫処理トランザクションは、電子署名と電子レコードを取り込むために、調整即時、作成即時、ステータス即時および移動即時の在庫数量トランザクションのいずれに対しても構成可能です。

OPM 品質イベント

品質仕様の承認： 品質仕様の承認は、仕様の承認と仕様妥当性ルールの承認の、2つの独立したイベントから構成されます。両方のイベントとも、電子署名と電子レコードの取込みが必要です。仕様は、ステータスによって、またはワークフローを通じて制御されます。したがって、仕様の作成と更新の両方に対する署名は遅延されます。

結果： 品質結果にも、電子署名と電子レコードのサポートが必要です。すべての結果がセッション中に入力された後、結果の保存時に、このウィンドウによって電子署名イベントが呼び出され、適切なサインオフがオンラインで実行されます。この時点で必要な電子署名のすべてが取り込めない場合、結果をデータベースへ確定されません。一覧にされたすべてのテストの結果は、電子署名イベントが呼び出される順番に入力する必要はありません。1つ以上のテスト結果が入力され保存されると、電子署名が要求されます。以後のセッションでさらに結果が入力されると、その時点で電子署名が要求されます。

サンプル処分： このイベントは、1つのサンプルに対して記録されたすべての結果の最終的な確認と、承認または否認を表します。サンプル処分が、承認、特例付き受入れまたは否認になった後に、この処置を反映するために、ロット・ステータスが変更されます。分析証明レポートと仕様照合機能の両方は、承認されたサンプル処置によって決まります。サンプルが承認されない場合は、これらの機能でそのサンプルは認識されません。

ロット・ステータス変更： このイベントは、仕様の範囲内または範囲外という結果に基づく自動的なロット・ステータスの変更を表しています。

グレード変更： このイベントは、仕様の範囲内または範囲外という結果に基づく自動的なグレード変更を表しています。この変更は、検討中のウェアハウス全体に影響します。

Oracle テクノロジ・スタックへの Oracle Workflow の実装

次にあげる製品では、Oracle Workflow を利用してビジネス・プロセスが定義および統合されます。各機能の詳細は、各製品のユーザーズ・ガイドまたは構成ガイドを参照してください。

関連項目：

C-39 ページ [「Oracle Workflow のサポート・ポリシー」](#)

Oracle Warehouse Builder

Oracle Warehouse Builder には Workflow 配布ウィザードが組み込まれています。このウィザードを使用すれば、マッピングの抽出、変換およびロードを項目タイプ内の関数として Oracle Workflow に配置できます。これらの関数は、Oracle Workflow Builder を使用してワークフロー・プロセスとして定義することができます。ワークフロー・プロセスを設計するときに、ジョブが適切な順序で実行されるように、ジョブの依存関係をマッピング間に指定できます。このワークフロー・プロセスは、Oracle Workflow から実行したり、Oracle Enterprise Manager を使用してスケジュールすることができます。

ジョブをワークフロー・プロセスとして定義することにより、一連のジョブのスケジュール全体を自動化することができます。プロセスの実行時は、プロセスに定義されている順序でジョブが実行されるように Oracle Workflow が管理します。例外が発生した場合、Oracle Workflow はプロセスを終了します。この方法を使用すれば、ウェアハウスの負荷管理およびジョブの更新に必要な手動操作を最小限に抑えることができます。

詳細は、『Oracle Warehouse Builder ユーザーズ・ガイド』の「ロードおよび更新の管理」を参照してください。

Oracle Application Server InterConnect

Oracle Workflow ビジネス・イベント・システムは、Oracle Application Server InterConnect と Oracle Workflow を統合して、ビジネス・プロセスに特化した完全な統合ソリューションを提供します。Oracle Application Server InterConnect および Oracle Workflow を使用すれば、組織のビジネス・プロセスを実現するために、複数のアプリケーション間の業務を連携させることができます。

簡単なビジネス・プロセス定義は、Oracle Application Server InterConnect の主要な機能に定義されているメッセージ交換にすでに組み込まれています。複雑なビジネス・プロセスの場合でも、設計時および実行時に強力な Oracle Workflow ビジネス・プロセス定義と実行サポートを利用することにより、ビジネス・プロセスをわかりやすく管理することができます。たとえば、Oracle Workflow を使用して、例外が発生したときのエラー管理、承認などの決済処理、メッセージの交換（メッセージの送受信など）、処理状態を保存したルーティング、および複数のアプリケーション間の通信に関する複合サービスをモデル化することができます。

Oracle Application Server InterConnect iStudio 設計ツールを使用すれば、一般的なイベントおよびプロセスに対応する Oracle Workflow のビジネス・イベント定義およびサブスク

リブション定義を自動的に生成できます。これらの定義は、iStudio から Oracle Workflow ホーム・ページを表示して確認できます。

また、iStudio ツールを使用して、Oracle Workflow の項目タイプ定義にワークフロー・プロセスを組み込むこともできます。たとえば、Oracle Workflow イベント・アクティビティを含む初期ワークフロー・プロセスを項目タイプに組み込み、iStudio に定義されている「公開」、「サブスクリプト」、「起動」および「実装」アクティビティをそれらのイベント・アクティビティに関連付けることができます。次に、iStudio から Oracle Workflow Builder を起動して、イベント・アクティビティの順序を指定すれば、ワークフロー定義の作成が完了します。このとき、通知や関数などの他のアクティビティを追加することもできます。

たとえば、iStudio を使用して 2 つのイベント・アクティビティを含むワークフロー・プロセスを生成します。このとき、一方のイベント・アクティビティでは CreatePO イベントを受信し、もう一方のイベント・アクティビティでは AcceptPO イベントを送信するとします。次に、Oracle Workflow Builder を使用して、これらのアクティビティの実行を制御するビジネス・プロセスを定義します。たとえば、電子メールを送信する通知アクティビティを追加して、CreatePO イベントの受信後と AcceptPO イベントの送信前に承認を要求します。

Oracle Application Server InterConnect と Oracle Workflow は、Oracle Workflow ビジネス・イベント・システムを介して実行時に通信し、Oracle Advanced Queuing のメッセージ・インフラストラクチャを利用して複数のアプリケーションに定義されているビジネス・プロセスを実行します。

詳細は、『Oracle Application Server InterConnect ユーザーズ・ガイド』の Oracle Workflow の項を参照してください。

Oracle Application Server Wireless

Oracle Workflow と Oracle Application Server Wireless は、ワイヤレス通知を送信できるように統合されます。Oracle Application Server Wireless は、WF_NOTIFICATION_OUT キューに対するサブスクリプトを提供することで、Oracle Workflow と統合します。このサブスクリプトは、キューからの通知メッセージを JMS Text メッセージとしてデキューし、携帯情報端末に送信します。ユーザーが携帯情報端末から応答を送信する場合、Oracle Application Server Wireless はユーザーが携帯情報端末から応答を送信する場合、Oracle Application Server Wireless は適切な通知応答関数をコールして、応答を記録し通知を完了します。詳細は、『Oracle Application Server Wireless 管理者ガイド』および『Oracle Application Server Wireless 開発者ガイド』を参照してください。

Oracle Workflow のサポート・ポリシー

Oracle Workflow は Oracle Applications に組み込まれており、ビジネス・プロセスを自動化し、効率よく作業するために、そのモジュールが使用されます。Oracle Workflow Builder を使用すると、アプリケーション・コードを修正しなくても、既存のビジネス・プロセスを変更することができます。また、Oracle Workflow では、実際のニーズにあわせてビジネス・ルールが変更された場合にも、ワークフロー・プロセスを拡張できるようになっています。また、イベント・マネージャを使用すると、アプリケーション・コードを修正しないでイベント定義およびサブスクリプション定義を変更することができます。

Oracle Workflow を使用して定義済のワークフロー・プロセス、イベントまたはサブスクリプションをカスタマイズする前に、次の「カスタマイズのガイドライン」をよく読み、事前に標準設計、安全な設計および実際の開発について理解しておいてください。これらのガイドラインに従うと、カスタマイズで問題が発生した場合に、オラクル社カスタマ・サポート・センターに対して、支援に必要な重要な情報を提示できます。

カスタマイズのガイドライン

1. Oracle Workflow のマニュアルおよびその他の製品固有のユーザーズ・ガイドに記載されているとおりに、すべての設定手順が完了していることを確認します。
2. シードされる未修正のワークフロー、イベントまたはサブスクリプションを、テスト・データベースでテストし、各自の環境に特有の設定およびデータを使用して、正常に稼働することを確認します。
3. ワークフロー、イベントまたはサブスクリプションの固有な問題については、製品固有のユーザーズ・ガイドを参照してください。これらのドキュメント・ソースでは、特に「不可」について記載されています。オラクル社カスタマ・サポート・センターでは、「UNSUPPORTED」と明示されているオブジェクトに関する修正については、サポートしていません。
4. 手順ごとにカスタマイズを行い、各手順が終了した時点で、カスタマイズしたワークフローまたはサブスクリプションをテストします。
5. PL/SQL プロシージャを作成する場合は、Oracle Workflow のマニュアルに記載されている標準の PL/SQL API テンプレートを確認します。エラーの発生時には必ず例外を処理するようにして、エラーが発生したプロシージャに戻れるようにします。
6. カスタマイズしたワークフロー、イベントまたはサブスクリプションが、テスト・データベースで正常に実行できることを確認せずに、本稼働用のデータベースに実装しないでください。テスト・データベースは、本稼働用の設定の特定部分のみが複製されているためです。

カスタマイズに関する問題点の解決

ワークフロー、イベントまたはサブスクリプションのカスタマイズ中に問題が発生した場合は、次のようにします。

- サポート分析者に、修正したワークフローの定義ファイル、イベント定義またはサブスクリプション定義を提示し、可能な場合は、問題が発生した手順を識別します。
- サポート分析者に、修正していないワークフローまたはサブスクリプションを実行した結果を報告します。

サポート対象外のカスタマイズ

次のタイプのカスタマイズは、サポート対象外となります。

1. 保護レベルが 100 未満のワークフロー・オブジェクトは修正できません。
2. 元の保護レベルが 100 未満の場合、そのワークフロー・オブジェクトの保護レベルは変更できません。
3. 100 未満のレベルで保護されているワークフロー・オブジェクトを修正するために、アクセス・レベルを、未定義 (100 未満) のレベルに変更することはできません。
4. 対象ワークフローの製品固有のユーザーズ・ガイドまたはドキュメントの改訂ノートに、明確に「UNSUPPORTED」と記載されているカスタマイズを行うことはできません。具体的には、修正不可と記載されているプロセス、属性、関数アクティビティ、イベント・アクティビティ、通知、選択肢タイプ、メッセージ、イベントまたはサブスクリプションは修正できません。
5. 接頭辞 WF_ または FND_ を付加して、ワークフローの表を手動で修正することはできません (Oracle Workflow のマニュアルに記載されている場合、またはオラクル社サポート・サービスが必要な場合は除きます)。
6. WF_USERS、WF_ROLES および WF_USER_ROLES ビューの事前定義済の実装だけでなく、Oracle Applications 内の元のシステムに用意されているバルク同期ビューも含めて、事前定義済のすべての Oracle Workflow ディレクトリ・サービスのビュー定義に対するカスタマイズ。
7. Oracle Workflow PL/SQL セキュリティ・パッケージである WFA_SEC の事前定義済のすべてのバージョンに対するカスタマイズ。

サポート対象のカスタマイズ

次のタイプのカスタマイズは、サポート対象となります。

1. ワークフロー、イベントまたはサブスクリプションの製品固有のユーザーズ・ガイド、またはドキュメントの改訂ノートで、必須とされているカスタマイズ。
2. 製品固有のユーザーズ・ガイドまたはドキュメントの改訂ノートに記載されているカスタマイズ例。ここで発生する問題点は、記載されているとおりにカスタマイズ例が実行される範囲までは、完全にサポート対象となります。記載とは異なる環境によって発生する問題は、カスタム開発の問題となり、さらに評価する必要があります。次の「3」を参照してください。
3. カスタマイズがサポート対象外か、必須か、またはサポート対象のカスタマイズ例かについて明確ではない場合は、サポートの範囲は次のようになります。まず、ユーザーが前述のカスタマイズのガイドラインに従って問題を切り分けます。評価の結果、切り分けられた問題点が Oracle 製品によるものであるとオラクル社カスタマ・サポート・センターがみなした場合は、オラクル社より解決策が提示されます。Oracle 製品による問題ではない、カスタム開発に対する問題解決は、ユーザーの責任において行ってください。

用語集

アクセス・レベル (Access Level)

0～1000の数値。各ワークフロー・ユーザーは固有のアクセス・レベルで操作します。アクセス・レベルにより、そのユーザーが特定のワークフロー・データを変更できるかどうか定義されます。変更できるのは、自分のアクセス・レベル以上のレベルで保護されているデータのみです。

アクティビティ (Activity)

ビジネス・プロセス中に実行される1単位の作業。

アクティビティ属性 (Activity Attributes)

関数アクティビティの動作を制御するために、その関数アクティビティの外部で定義されているパラメータ。アクティビティ属性を定義するには、「アクティビティ」ウィンドウでアクティビティの「属性」プロパティ画面を表示します。アクティビティ属性に値を割り当てるには、プロセス・ウィンドウで、そのアクティビティ・ノードの「属性値」プロパティ画面を表示します。

イベント (Event)

システム内の他のオブジェクトまたは外部エージェントに関連付けられた、インターネットまたはイントラネット内の状態変化です。

イベント・アクティビティ (Event Activity)

ワークフロー・プロセスに組み込めるように、アクティビティとしてモデル化されたビジネス・イベント。

イベント・キー (Event Key)

イベントのインスタンスを一意に識別する文字列。イベント名、イベント・キーおよびイベント・データは、イベント内で発生したすべてのアクティビティと対話します。

イベント・サブスクリプション (Event Subscription)

特定のイベントと特定のシステムの関連を指定し、イベントのトリガーが発生したときに実行する処理を指定すること。サブスクリプション処理には、カスタム・コードのコール、ワークフロー・プロセスへのイベント・メッセージの送信、またはエージェントへのイベント・メッセージの送信を指定することができます。

イベント・データ (Event Data)

イベントを説明する一連の追加詳細。イベント・データは XML 文書として作成できます。イベント名、イベント・キーおよびイベント・データは、イベント内で発生したすべてのアクティビティと対話します。

イベント・メッセージ (Event Message)

データ型 WF_EVENT_T によって定義された、ビジネス・イベントを伝達するための標準ワークフロー構造。イベント・メッセージは、イベント・データ以外に、イベント名、イベント・キー、アドレッシング属性およびエラー情報を含む、いくつかのヘッダー・プロパティで構成されます。

エージェント (Agent)

システム内の通信の名前付きポイント。

エージェント・リスナー (Agent Listener)

インバウンド・エージェントでイベント・メッセージを処理するサービス・コンポーネントの一種。

外部 Java 関数 (External Java Functions)

Oracle Database の外部で、Java 関数アクティビティ・エージェントによって実行される Java プログラム。

外部関数 (External Functions)

Oracle データベース・サーバーの外部で実行されるプログラム。

関数 (Function)

ビジネス・ルールの定義、アプリケーション内で自動化されているタスクの実行、またはアプリケーション情報の取出しができる PL/SQL ストアド・プロシージャ。ストアド・プロシージャは、標準引数を受け入れて完了結果を戻します。

関数アクティビティ (Function Activity)

PL/SQL ストアド・プロシージャで定義され、自動化されている 1 単位の作業。

結果コード (Result Code)

結果タイプにより定義される、結果値の内部名。

結果タイプ (Result Type)

アクティビティの結果値の候補を含む選択肢タイプの名前。

結果値 (Result Value)

完了したアクティビティから戻される値。

項目 (Item)

ワークフロー・プロセスにより管理される特定のプロセス、ドキュメントまたはトランザクション。たとえば、「購買承認申請」プロセスのワークフローで管理される項目は、Oracle Internet Commerce の「Web 購買依頼」ページで作成される特定の購買申請です。

項目属性 (Item Attribute)

「項目タイプ属性」を参照。

項目タイプ (Item Type)

同じ項目属性セットを共有する特定カテゴリの全項目のグループ。たとえば、「発注依頼」は、Oracle Internet Commerce の「Web 購買依頼」ページで作成された購買申請をすべてグループ化するための項目タイプです。項目タイプは、プロセスを上位レベルでグループ化する手段としても使用されます。

項目タイプ属性 (Item Type Attribute)

特定の項目タイプに関連付けられている機能。「項目属性」と同義。項目タイプ属性は、その項目を保存するアプリケーションで値を検索して設定できる変数として定義されます。項目タイプ属性とその値は、プロセスのすべてのアクティビティに使用可能です。

コスト (Cost)

アクティビティの完了までに必要な処理量をワークフロー・エンジンに通知するために、関数アクティビティまたは通知アクティビティに割当て可能な相対値。大きいコストを割り当てるほど、アクティビティは複雑になり、完了までの所要時間が長くなります。ワークフロー・エンジンは、コストのしきい値で動作するように設定できます。ワークフロー・エンジンのコストのしきい値を超えるアクティビティは、「DEFERRED」に設定され、処理されません。バックグラウンド・エンジンは、延期されたアクティビティをチェックして処理するように設定できます。

サービス・コンポーネント (Service Component)

汎用サービス・コンポーネント・フレームワークで管理できるように、このフレームワーク標準に従って定義された Java プログラムのインスタンス。

サービス・コンポーネント・コンテナ (Service Component Container)

所有する個々のサービス・コンポーネントの実行を管理するサービスまたはサーブレットのインスタンス。コンテナは、そのコンポーネントのステータスを監視し、コンテナ自体とそのコンポーネントの制御イベントを処理します。

サブスクリプション (Subscription)

「イベント・サブスクリプション」を参照。

システム (System)

ホスト・マシンやデータベース・インスタンスなどの論理的に孤立したソフトウェア環境。

実行者 (Performer)

手動によるアクティビティ（通知）を実行するように割り当てられているユーザーまたはロール。プロセスに含まれている通知アクティビティは、実行者に割り当てる必要があります。

選択肢コード (Lookup Code)

選択肢タイプに定義されている値の内部名。

選択肢タイプ (Lookup Type)

事前定義済の値リスト。選択肢タイプのそれぞれの値には、内部名と表示名が付いています。

属性 (Attribute)

「アクティビティ属性」、「項目タイプ属性」または「メッセージ属性」を参照。

タイムアウト (Timeout)

ワークフロー・エンジンがエラー・プロセスまたは代替アクティビティ（定義されている場合）に進む前に、通知アクティビティを実行する必要がある期間。

通知 (Notification)

ユーザーに配信されるメッセージのインスタンス。

「通知」 Web ページ (Notification Web Page)

任意の Web ブラウザで表示し、ワークフロー通知の問合せと応答に使用できる Web ページ。

通知アクティビティ (Notification Activity)

ユーザーによる操作を必要とする 1 単位の作業。通知アクティビティは、作業の完了に必要な情報を含むメッセージをユーザーに送信します。

通知メーラー (Notification Mailer)

ユーザーに対してメール・アプリケーションを介して電子メール通知を送信し、電子メールによる応答を処理するサービス・コンポーネントの一種。

ディレクトリ・サービス (Directory Services)

Oracle Workflow のユーザーおよびロールと、サイトのディレクトリ・リポジトリとのマッピング。

トランジション (Transition)

あるアクティビティの完了とプロセス内の別のアクティビティのアクティブ化を定義する関連。プロセス・ダイアグラムでは、2つのアクティビティを結ぶ矢印がトランジションを表します。

ノード (Node)

「プロセス」ウィンドウに表示されるプロセス・ダイアグラム内のアクティビティのインスタンス。

バックグラウンド・エンジン (Background Engines)

延期されたアクティビティやタイムアウトになったアクティビティを処理する補助的なワークフロー・エンジン。

汎用サービス・コンポーネント・フレームワーク (Generic Service Component Framework)

バックグラウンド Java サービスの管理の簡略化と自動化に役立つ機能。

ビジネス・イベント (Business Event)

「イベント」を参照。

プロセス (Process)

ビジネス目標を達成するために実行する必要があるアクティビティのセット。

プロセス・アクティビティ (Process Activity)

他のプロセスで参照できるように、アクティビティとしてモデル化されているプロセス。

プロセス定義 (Process Definition)

Oracle Workflow Builder に定義されているワークフロー・プロセス。

保護レベル (Protection Level)

データ変更が禁止されているユーザーを表す 0～1000 の数値。ワークフロー・データを定義するときに、誰でも変更できることを示すカスタマイズ可能 (1000) に設定するか、そのデータを定義中のユーザーのアクセス・レベルと同じ保護レベルを割り当てることができます。後者の場合、データを変更できるのは、そのデータの保護レベル以下のアクセス・レベルで操作するユーザーのみです。

メッセージ (Message)

通知アクティビティにより送信される情報。メッセージは、通知アクティビティに関連付ける前に定義する必要があります。メッセージには、件名、優先度、本文が含まれ、1つ以上のメッセージ属性も含まれている場合があります。

メッセージ属性 (Message Attribute)

メッセージが通知で送信されるときに、情報を提供したり応答プロンプトを表示するために、特定のメッセージに対して定義する変数。事前定義済の項目タイプ属性をメッセージ属性として使用できます。「送信」ソースとして定義されたメッセージ属性は、メッセージの送信時にランタイム値で置き換えられます。「応答」ソースとして定義されたメッセージ属性では、メッセージの送信時に応答を求めるプロンプトが表示されます。

ロール (Role)

共通の職責または職階別にグループ化された1人以上のユーザー。

ワークフロー・エンジン (Workflow Engine)

ワークフロー・プロセス定義を実装する Oracle Workflow コンポーネント。ワークフロー・エンジンにより、項目のすべてのアクティビティのステータスが管理され、関数が自動的に実行されて通知が送信され、完了したアクティビティの履歴が保存され、エラー条件が検出されて、エラー・プロセスが開始されます。ワークフロー・エンジンはサーバーの PL/SQL に実装され、エンジン API のコール時にアクティブ化されます。

ワークフロー定義ローダー (Workflow Definitions Loader)

フラット・ファイルとデータベース間でワークフロー定義をアップロードおよびダウンロードするためのコンカレント・プログラム。

索引

記号

#ATTACHED_URLS 属性, 3-34
#FROM_ROLE 属性, 3-30
#HDR 属性, 3-32
#HIDE_MOREINFO 属性, 3-29
#HIDE_REASSIGN 属性, 3-27
#WF_REASSIGN_LOV 属性, 3-26
#WF_SIG_ID 属性, 3-32
#WF_SIG_POLICY 属性, 3-31
#WFM_CANCELED 属性, 3-34
#WFM_CLOSED 属性, 3-35
#WFM_FROM 属性, 3-33
#WFM_HTMLAGENT 属性, 3-33
#WFM_OPEN_INVALID 属性, 3-35
#WFM_OPEN_MAIL_DIRECT 属性, 3-34
#WFM_OPEN_MAIL_FYI 属性, 3-34
#WFM_OPEN_MAIL_PSIG_ONLY 属性, 3-35
#WFM_OPEN_MAIL 属性, 3-34
#WFM_OPEN_MORE_INFO 属性, 3-35
#WFM_OPEN_SIGN 属性, 3-35
#WFM_REPLYTO 属性, 3-33
&#NID, 3-12, 3-13, 3-16, 3-42

A

ACTID, 6-4, 6-13
Advanced Queuing, 8-2
「And」 アクティビティ, 5-3
Any イベント, 9-15
AQ\$_JMS_TEXT_MESSAGE, 8-28, 8-108
ATTACHED_URLS 属性, 3-34

B

「Buyer Workbench」
Web ページ, 10-56

D

DEFAULT_ERROR, 11-8
DEFAULT_EVENT_ERROR, 11-15

F

FROM_ROLE 属性, 3-30
FUNCMODE, 6-5, 6-6

G

GSC ビジネス・イベント・システム制御グループ,
9-36

H

HDR 属性, 3-32
HIDE_MOREINFO 属性, 3-29
HIDE_REASSIGN 属性, 3-27

I

ITEMKEY, 6-4, 6-13
ITEMTYPE, 6-4, 6-13

J

Java API
関数アクティビティ, 6-8
JMS テキスト・メッセージ, 8-28, 8-108

N

「NOOP」 アクティビティ, 5-7

O

Oracle Advanced Queuing, 8-2

Oracle Applications Manager, 1-5

Oracle Workflow Builder, 1-3

概要, 2-2

コマンドラインからの起動, 2-18

保存モード, 2-16, 3-18

ローダー機能, 2-16

Oracle Workflow Manager, 1-5

「Or」 アクティビティ, 5-3

P

Ping の確認イベント, 9-13

PL/SQL, 1-4

文書, 6-15

PL/SQL API

「PL/SQL CLOB」 文書, 6-15

「PL/SQL」 文書, 6-15

イベント・サブスクリプションのルール関数, 6-29

イベント・データ・ジェネレート関数, 6-24

関数アクティビティ, 6-3

キュー・ハンドラ, 6-26

セレクト関数またはコールバック関数, 6-12

PL/SQL BLOB

文書, 6-15, 6-22

PL/SQL CLOB

文書, 6-15, 6-17

PL/SQL スタアド・プロシージャ

作成, 10-15

スクリプト, 10-15

ネーミング規則, 10-15

PL/SQL 文書, 3-5

R

RESULT, 6-5, 6-14

RETRY_ONLY, 11-12

S

SetItemParent API, 5-12

StartProcess 関数

サンプル購買申請プロセス, 10-25

U

Unexpected イベント, 9-18, 11-6

URL

イベント・システム・デモンストレーションの Web ページ, 10-56, 10-58

「購買申請デモンストレーション」 Web ページ, 10-10

"URL 属性

フレーム・ターゲット, 3-47

URL-タイプ属性, 3-3

「URL」メッセージ属性

添付と埋込み, 3-48

User Entry Has Changed イベント, 9-20

W

WF_CONTROL エージェント, 8-29, 8-109

WF_DEFERRED エージェント, 8-29, 8-109

WF_ERROR_QH, 8-28, 8-108

WF_ERROR エージェント, 8-29, 8-109

WF_EVENT_OJMSTEXT_QH, 8-28, 8-108

WF_EVENT_OMB_QH, 8-28, 8-108

WF_EVENT_QH, 8-28, 8-108

WF_IN エージェント, 8-29, 8-109

WF_JMS_IN エージェント, 8-29, 8-109

WF_JMS_OUT エージェント, 8-29, 8-109

WF_NOTIFICATION() メッセージ関数, 3-36

WF_NOTIFICATION_IN エージェント, 8-29, 8-109

WF_NOTIFICATION_OUT エージェント, 8-29, 8-109

WF_OUT エージェント, 8-29, 8-109

WF_REASSIGN_LOV 属性, 3-26

WF_REQDEMO.SelectApprover, 10-28

WF_REQDEMO.StartProcess, 10-9

WF_REQDEMO.VerifyAuthority, 10-19, 10-31

WF_SIG_ID 属性, 3-32

WF_SIG_POLICY 属性, 3-31

WFM_CANCELED 属性, 3-34

WFM_CLOSED 属性, 3-35

WFM_FROM 属性, 3-33

WFM_HTMLAGENT 属性, 3-33

WFM_OPEN_INVALID 属性, 3-35

WFM_OPEN_MAIL_DIRECT 属性, 3-34

WFM_OPEN_MAIL_FYI 属性, 3-34

WFM_OPEN_MAIL_PSIG_ONLY 属性, 3-35
WFM_OPEN_MAIL 属性, 3-34
WFM_OPEN_MORE_INFO 属性, 3-35
WFM_OPEN_SIGN 属性, 3-35
WFM_REPLYTO 属性, 3-33
WFRUND.SQL, 10-9
Workflow Builder のメニュー, B-2
Workflow Designer, 1-3

X

「XML タグ値の取得」アクティビティ, 5-19
「XML タグ値の比較」アクティビティ, 5-20
「XML タグ値の比較 (数値)」アクティビティ, 5-20
「XML タグ値の比較 (テキスト)」アクティビティ,
5-20
「XML タグ値の比較 (日付)」アクティビティ, 5-20
「XML 変換」アクティビティ, 5-22

あ

アイコン
表示, 3-60, 3-63, 3-66, 3-69
アクション
サブスクリプション, 8-44, 8-90
「アクセス」プロパティ画面, 3-18
アクセス保護
カスタマイズの保持, 3-19
アクセス・レベル
インジケータ, 3-18
アクティビティ, 2-10, 3-53
アイコン, 3-60, 3-63, 3-66, 3-69, 3-74
イベント, 3-53, 3-56
エラー・プロセス, 3-71
オプションの詳細, 3-71
関数, 3-53, 3-55
結果タイプ, 3-60, 3-63, 3-69
購買申請プロセス, 10-15
コスト, 3-58
異なるデータ・ストアからのアクセス, 4-8
コピー, 3-72
コンカレント・マネージャ, 5-23
「再試行のみ」プロセス, 11-12
作成, 3-59, 3-61, 3-65, 3-68
サプライヤ: アドバンスト出荷通知プロセス, 10-90
サプライヤ: オーダー詳細の取得プロセス, 10-82
サプライヤ: クレジット・チェックプロセス, 10-86

サプライヤ: 在庫チェックプロセス, 10-88
サプライヤ: 最上位のオーダープロセス, 10-78
サプライヤ: サプライヤの請求書の送信プロセス,
10-93
システム
エラー, 3-53
「詳細 Ping」プロセス, 8-143
「承認者に通知」サブプロセス, 10-22
タイムアウト, 4-11
遅延, 3-58
通知, 3-53, 3-54
デフォルト・イベント・エラー・プロセス, 11-15
デフォルト・エラー・プロセス, 11-8
バージョン番号, 3-72
バイヤー: アドバンスト出荷通知プロセス, 10-74
バイヤー: 最上位の発注プロセス, 10-64
バイヤー: サプライヤの請求の受信プロセス, 10-76
バイヤー: サプライヤの発注承認の受信プロセス,
10-71
バイヤー: サプライヤへの発注送信プロセス, 10-68
プロセス, 3-53, 3-58
分岐の結合, 4-4
「マスター Ping」プロセス, 8-141
有効日, 3-71
ループ内, 3-71
ワークフロー・イベント・プロトコル・プロセス,
9-42
アクティビティ属性, 3-64, 3-67
値の設定, 4-13
アクティビティ・ノード
購買申請プロセス, 10-15
サプライヤ: アドバンスト出荷通知プロセス, 10-90
サプライヤ: オーダー詳細の取得プロセス, 10-82
サプライヤ: クレジット・チェックプロセス, 10-86
サプライヤ: 在庫チェックプロセス, 10-88
サプライヤ: 最上位のオーダープロセス, 10-78
サプライヤ: サプライヤの請求書の送信プロセス,
10-93
「詳細 Ping」プロセス, 8-143
「承認者に通知」サブプロセス, 10-22
バイヤー: アドバンスト出荷通知プロセス, 10-74
バイヤー: 最上位の発注プロセス, 10-64
バイヤー: サプライヤの請求の受信プロセス, 10-76
バイヤー: サプライヤの発注承認の受信プロセス,
10-71
バイヤー: サプライヤへの発注送信プロセス, 10-68
「マスター Ping」プロセス, 8-141

- ワークフロー・イベント・プロトコル・プロセス, 9-42
- アクティビティの結合, 4-4
- 値リスト, 8-85
 - Web インタフェース, 8-25
- アドバンスト出荷通知イベント, 10-99
- アド・ホックのユーザーおよびロール, 4-26

い

- 維持, 3-4
- イベント, 8-5
 - エージェントへの送信, 8-46, 8-93
 - 概要, 8-73
 - 検索, 8-16
 - 更新, 8-17, 8-78
 - 削除, 8-17
 - 作成, 8-78
 - 事前定義済, 9-2
 - 定義, 8-7
 - テスト, 8-84
 - 表示, 8-75
 - 呼出し, 8-5, 8-60, 8-72
 - ワークフロー・プロセスへの送信, 8-45, 8-91
- 「イベント Buyer Workbench」
 - Web ページ, 10-56
- 「イベント」Web ページ, 8-7, 8-10, 8-17, 8-52
- イベント・アクティビティ
 - 作成, 3-65
- 「イベント」アクティビティ, 3-56
- イベント・アクティビティ属性, 3-67
- イベント・アクティビティの詳細, 4-14
- 「イベント オーダーの追跡」
 - Web ページ, 10-58
- イベント・グループ, 8-5
 - 概要, 8-75
 - 更新, 8-80
 - 作成, 8-80
 - 定義, 8-10
- イベント・グループ更新イベント, 9-4
- イベント・グループ削除イベント, 9-4
- イベント・グループ作成イベント, 9-4
- イベント更新イベント, 9-3
- イベント削除イベント, 9-3
- イベント作成イベント, 9-3

- イベント・サブスクリプション, 8-41, 8-86
 - ルール関数, 6-29
- 「イベント・サブスクリプション」Web ページ, 8-59
- イベント・システム・デモンストレーション
 - 開始, 10-55
 - 概要, 10-52
 - 設定, 10-55
 - データ・モデル, 10-54
- イベント・システム・デモンストレーション・プロセス
 - インストール, 10-54
- イベント・システムの同期イベント, 9-9
- 「イベント・システム・ローカル・キュー」Web ページ, 8-69
- イベント・タイプ属性, 3-4
- イベント・データ, 6-24, 8-6, 8-43, 8-73, 8-88
- イベント・ノード, 4-14
- 「イベントの検索」Web ページ, 8-16, 8-57
- イベントの呼出し, 8-5, 8-60, 8-72
- 「イベントの呼出し」Web ページ, 8-60
- 「イベント・プロパティの取得」アクティビティ, 5-16
- 「イベント・プロパティの設定」アクティビティ, 5-17
- 「イベント・プロパティの比較」アクティビティ, 5-18
- イベント・マネージャ, 8-4
 - Oracle Applications, 8-72

う

- 「ウインドウ」メニュー, B-7

え

- エージェント, 8-25
 - ping, 8-137
 - 外部システム, 8-116
 - 概要, 8-105
 - キュー, 8-28, 8-108
 - キュー・ハンドラ, 8-28, 8-108
 - 検索, 8-37
 - 更新, 8-39, 8-119
 - 削除, 8-39
 - 作成, 8-119
 - 定義, 8-34
 - 表示, 8-117
 - プロトコル, 8-26, 8-106
 - 方向, 8-26, 8-106
- 「エージェント」Web ページ, 8-34, 8-39

エージェント・グループ

概要, 8-109

更新, 8-122

作成, 8-122

エージェント・グループ・メンバー更新イベント, 9-7

エージェント・グループ・メンバー削除イベント, 9-7

エージェント・グループ・メンバー作成イベント, 9-7

エージェント更新イベント, 9-6

エージェント削除イベント, 9-6

エージェント作成イベント, 9-6

エージェントの Ping, 8-137

エージェントの Ping イベント, 9-12

「エージェントの検索」 Web ページ, 8-37

エラー処理

イベント・サブスクリプション, 11-5

ワークフロー・プロセス, 11-2

エラー・チェック・プロセス, 10-42

アクティビティ, 10-45

概要, 10-44

エラー・プロセス, 3-71, 11-7

エンキュー

キュー・ハンドラ, 6-27

か

「開始」アクティビティ, 4-5, 5-9

開始イベント, 9-35

開発者スタジオ, 7-6

外部システムの登録, 8-135

外部文書の統合, 3-5

カスタマイズの保持

アクティビティ, 3-19

カスタマイズ・レベル

アクティビティ, 3-7, 3-11, 3-21, 3-42, 3-61,
3-64, 3-67, 3-70, 3-74

イベント, 8-6, 8-74

サブスクリプション, 8-48, 8-89

監視

作業項目, 1-5

関数, 2-11

関数アクティビティ, 3-55

作成, 3-61

標準 Java API, 6-8

標準 PL/SQL API, 6-3

関数アクティビティ属性, 3-8, 3-64

関数アクティビティの例

「承認権限の検証」, 10-31

「承認者の選択」, 10-28

き

キュー

エージェントへの割当て, 8-28, 8-108

確認, 8-68

キュー・ハンドラ, 6-26, 8-28, 8-108

く

グローバル変数, 3-2

け

「継続フロー」アクティビティ, 5-13

結果タイプ

アクティビティ, 3-60, 3-63, 3-69

投票アクティビティ, 3-74

こ

更新イベント, 9-35

購買申請

データ・モデル, 10-6

購買申請承認が必要なことを通知, 10-33

「購買申請デモンストレーション」

Web ページ, 10-9

購買申請プロセス, 10-5

インストール, 10-6

開始, 10-9

概要, 10-14

関数アクティビティの例, 10-28

項目属性

外部文書の統合, 3-5

項目タイプ, 2-10, 3-2

維持タイプ, 3-4

「イベント・システムのデモ」, 10-60

「購買申請」, 10-13

コールバック関数, 3-5

コピー, 3-16

「コンカレント・マネージャ機能」, 5-23

コンテキストの再設定, 6-12

作成, 3-6

「システム:エラー」, 11-7

セレクト関数, 3-5, 6-12

「標準」, 5-2

保存, 2-13

ロード, 2-13

「ワークフロー・エージェントの Ping/ 確認」,
8-139
「ワークフロー送信プロトコル」, 9-39
項目タイプ属性, 3-2, 3-8
「イベント・システムのデモ」, 10-60
「購買申請」, 10-13
「ワークフロー送信プロトコル」, 9-39
項目タイプのロード, 2-13
コールバック関数, 6-12
項目タイプ, 3-5
コマンド, 6-13
コストのしきい値, 3-58
「コンカレント・プログラムの実行」 アクティビティ,
5-23
「コンカレント・プログラムの発行」 アクティビティ,
5-24
「コンカレント・プログラム待ち」 アクティビティ,
5-25
「コンカレント・マネージャ機能」 項目タイプ, 5-23
コンカレント・マネージャのアクティビティ, 5-23

さ

再開イベント, 9-36
再試行エラー, 11-12
作業項目, 2-10
サブスクリプション, 8-41
概要, 8-86
検索, 8-57
更新, 8-59, 8-99
削除, 8-59
作成, 8-99
事前定義済, 9-2
遅延, 8-49, 8-94
定義, 8-52
表示, 8-97
サブスクリプション更新イベント, 9-8
サブスクリプション削除イベント, 9-8
サブスクリプション作成イベント, 9-8
サブスクリプション処理, 8-90
サブプロセス
タイムアウト, 4-11
サプライヤ:アドバンスト出荷通知プロセス
概要, 10-89
サプライヤ:オーダー詳細の取得プロセス
概要, 10-81

サプライヤ:クレジット・チェックプロセス
概要, 10-85
サプライヤ:在庫チェックプロセス
概要, 10-87
サプライヤ:最上位のオーダープロセス
概要, 10-77
サプライヤ:サプライヤの請求書の送信プロセス
概要, 10-92
サンプル・プロセス
イベント・システム・デモンストレーション,
10-52
購買申請, 10-5
サンプル・ワークフロー・プロセス, 10-2

し

シード・イベント・グループ, 9-10
ジェネレート関数, 8-6, 8-74
システム, 8-19
Workflow 対応システムの登録, 8-135
外部, 8-126
概要, 8-125
検索, 8-22
更新, 8-23, 8-132
サインアップ, 8-62, 8-63, 8-126
削除, 8-23
作成, 8-132
定義, 8-20
同期, 8-65, 8-127
表示, 8-129
マスター / コピー, 8-67, 8-129
ローカル, 8-20, 8-125
ローカル・システム識別子, 8-133
「システム」 Web ページ, 8-20, 8-23
「システム:エラー」 項目タイプ, 11-7
システム更新イベント, 9-5
システム削除イベント, 9-5
システム作成イベント, 9-5
システム識別子, 8-63
「システム識別子」 Web ページ, 8-63
「システムの検索」 Web ページ, 8-22
「システムのサインアップ」 Web ページ, 8-63
システムのサインアップ・イベント, 9-14
システムの統合, 8-2
事前定義済イベント, 9-2
「実行時間の比較」 アクティビティ, 5-4

自動レプリケーション
イベント・マネージャのオブジェクト, 8-67, 8-129
「終了」アクティビティ, 4-5, 5-9
「詳細 Ping」プロセス
概要, 8-142
「承認権限の検証」関数アクティビティ, 10-31
「承認者に通知」
通知アクティビティの例, 10-33
「承認者に通知」サブプロセス
概要, 10-21
「承認者の選択」関数アクティビティ, 10-28
ショートカット, 4-25
署名, 3-31
署名 ID, 3-32

す

数値 - タイプ属性, 3-3
「数値の比較」アクティビティ, 5-3
「スレッドの遅延」アクティビティ, 5-6

せ

請求書イベント, 10-100
セーブポイント, 6-3, 6-8
セレクト関数, 3-5, 6-12
選択肢コード
コピー, 3-24
選択肢タイプ, 2-10, 3-20
コピー, 3-23
作成, 3-21
選択肢 - タイプ属性, 3-3

そ

送信者への返送イベント, 9-28
ソース・タイプ, 8-42, 8-87
属性
コピー, 3-17
タイプ, 3-3, 3-9, 3-45
トークンの置換, 3-51
「標準」, 3-53, 5-2
属性タイプ
URL, 3-10, 3-12, 3-45
イベント, 3-10, 3-47
数値, 3-10, 3-45
選択肢, 3-10, 3-45

属性, 3-10
テキスト, 3-10, 3-45
日付, 3-10, 3-45
フォーム, 3-10, 3-13, 3-46
文書, 3-10, 3-13, 3-47
ルール, 3-10, 3-47
属性 - タイプ属性, 3-4
その他の項目タイプ, 2-5

た

ダイアグラムの矢印, 4-2
「待機」アクティビティ, 5-4
タイムアウト, 4-11
動的, 4-12
タイムアウト・トランジション, 4-2, 4-3
単一コンシューマ・キュー, 8-47, 8-93

ち

遅延アクティビティ, 3-58
遅延処理
イベント・サブスクリプション, 8-49, 8-94
着信応答受信イベント, 9-27
中断イベント, 9-35

つ

通知
「From Role」の設定, 3-30
再度割り当てるユーザーのリストの指定, 3-26
「再割当」ボタンの非表示, 3-27
電子メール属性, 3-33
パスワードに基づいた署名の要求, 3-31
ヘッダー属性, 3-32
メッセージ・テンプレートの指定, 3-34
「要求情報」ボタンの非表示, 3-29
ロード・バランス, 5-9
通知 ID トークン, 3-12, 3-13, 3-16, 3-42
「通知」Web ページ, 1-5
「通知」アクティビティ, 5-10
通知アクティビティ, 3-54
カスタム関数との結合, 3-60
「購買申請承認が必要なことを通知」, 10-33
作成, 3-59
通知エラー・イベント, 9-28
通知応答イベント, 9-30

- 通知応答のイベント, 9-30
- 通知関数, 3-60
- 通知完了イベント, 9-30
- 通知完了のイベント, 9-30
- 通知後関数, 3-54
- 通知再割当てイベント, 9-24
- 通知再割当てのイベント, 9-24
- 通知受信グループ, 9-29
- 通知受信者使用不可イベント, 9-28
- 通知送信イベント, 9-23
- 通知送信エラー・イベント, 9-27
- 通知送信グループ, 9-24
- 通知送信のイベント, 9-23
- 通知取消しイベント, 9-24
- 通知取消しのイベント, 9-24
- 通知の再割当て
 - 「再割当て」ボタンの非表示, 3-27
- 通知の再度割り当てる
 - 再度割り当てるユーザーのリストの指定, 3-26
- 通知メーラー
 - イベント, 9-31
- 通知履歴, 3-36
- ツールバー
 - Oracle Workflow Builder, B-8

て

- 「定期アラート」
 - 項目タイプ, 10-43
- 停止イベント, 9-36
- ディスパッチ・モード, 8-51, 8-96
- ディテール・プロセス, 5-12
- ディレクトリ・サービス
 - Builder からの表示, 4-28
 - ナビゲータ・ツリー, 2-4
- テキスト・タイプ属性, 3-3
- 「テキストの比較」アクティビティ, 5-3
- デキュー
 - キュー・ハンドラ, 6-28
- テスト・ハーネス, 7-2
- テスト・プロセスの実行, 7-8
- デフォルト・イベント・エラー・プロセス, 11-15
- デフォルト・エラー・プロセス, 11-8
- デフォルト・トランジション, 4-2
- デモンストレーション
 - ディレクトリ・サービス, 10-7
- 電子署名, 3-31

- 電子メール通知, 1-4

と

- 動的タイムアウト, 4-12
- 動的優先度, 4-12
- 投票アクティビティ, 3-73
 - 結果タイプ, 3-74
- トークンの置換
 - 属性, 3-51
 - 文書タイプのメッセージ属性, 3-13
- ドキュメント・レビュー・プロセス, 10-36
 - アクティビティ, 10-39
 - 概要, 10-38
- トランジション, 4-2
 - 作成, 4-20
 - タイムアウト, 4-2
 - デフォルト, 4-2
 - 任意, 4-2
 - 編集, 4-20

な

- 「内容の添付」チェックボックス, 3-48
- ナビゲーション・パス, A-2
- ナビゲータ・ツリー
 - オブジェクトの検索, 2-7
- ナビゲータのツールバー, B-8

に

- 任意トランジション, 4-2

ね

- ネーミング規則
 - PL/SQL ストアド・プロシージャ, 10-15

の

- ノード
 - 開始と終了, 4-9
 - プロセスへの追加, 4-7
- ノードのアクティビティ
 - 動的優先度, 4-12

は

バージョン管理, 2-8
バージョン番号
 アクティビティ, 3-72
「はい / いいえ投票」アクティビティ, 5-11
バイヤー: アドバンスト出荷通知プロセス
 概要, 10-73
バイヤー: 最上位の発注プロセス
 概要, 10-62
バイヤー: サプライヤの請求の受信プロセス
 概要, 10-75
バイヤー: サプライヤの発注承認の受信プロセス
 概要, 10-69
バイヤー: サプライヤへの発注送信プロセス
 概要, 10-67
パスワードに基づいた署名, 3-31
発注イベント, 10-94
発注承認イベント, 10-97
汎用サービス・コンポーネント・フレームワーク
 イベント, 9-35

ひ

「比較」アクティビティ, 5-3
ビジネス・イベント, 8-5, 8-73
ビジネス・イベント・システム, 1-3
 Ping/ 確認の例, 8-137
 事前定義済イベント, 9-2
 ビジネス・イベントの管理, 8-2
ビジネス・イベント・システム・アプリケーション制
 御グループ, 9-33
ビジネス・イベント・システム制御 Ping イベント,
 9-32
ビジネス・イベント・システム制御グループ, 9-32
日付 - タイプ属性, 3-3
「日付の比較」アクティビティ, 5-3
「表示」メニュー, B-5
標準 API
 PL/SQL BLOB 文書, 6-15, 6-22
 PL/SQL CLOB 文書, 6-15, 6-17
 PL/SQL 文書, 6-15
 イベント・サブスクリプションのルール関数, 6-29
 イベント・データ・ジェネレート関数, 6-24
 関数アクティビティ, 6-3, 6-8
 キュー・ハンドラ, 6-26
 セレクト関数 / コールバック関数, 6-12

標準アクティビティ, 5-2
標準エラー・プロセス, 11-7
標準項目タイプ, 5-2

ふ

「ファイル」メニュー, B-2
フェーズ番号, 8-43, 8-50, 8-88, 8-95
フォーム - タイプ属性, 3-3
フォント
 設定, 4-24
 変更, 4-24
複数コンシューマ・キュー, 8-47, 8-93
フレーム・ターゲット
 URL 属性, 3-47
「フロー待ち」アクティビティ, 5-13
プロセス
 アクティビティ・トランジション, 4-2
 印刷, 4-22
 開始, 4-5
 概要, 4-22
 クリップボードへのコピー, 4-22
 検証, 4-23
 作成, 2-8
 定義
 変更, 2-12
 編集, 2-10, 2-12
 ループ, 6-6
プロセス・アクティビティ, 3-58
 作成, 3-68
「プロセス」ウィンドウ, 4-2
 編集, 4-2
「プロセス」ウィンドウのツールバー, B-9
プロセス・ダイアグラム
 作成, 4-2, 4-7
 ノードの追加, 4-7
「プロセスの開始」アクティビティ, 5-6
「ブロック」アクティビティ, 5-6
プロトコル, 8-26, 8-106
文書, 3-5
「文書管理」
 項目タイプ, 10-37
文書管理の統合, 3-3, 3-5
文書 - タイプ属性, 3-3
文書の統合, 3-3, 3-10, 3-47, 6-15
「文書」メッセージ属性
 添付と埋込み, 3-48

へ

ヘッダー属性, 3-32
「ヘルプ」メニュー, B-7
「編集」メニュー, B-4

ま

「マスター Ping」プロセス
概要, 8-140
マスター / コピー・システム, 8-67, 8-129
「マスター / ディテール連携」アクティビティ, 5-12
使用方法に関する注意, 5-14
マスター / ディテールを連携するアクティビティ,
5-12
マスター・プロセス, 5-12

み

未来日付のイベント, 8-49, 8-94, 8-95

め

メッセージ, 2-10
件名, 3-40, 10-34
コピー, 3-52
作成, 3-38
デフォルト優先度の上書き, 4-12
表示, 10-34
本文, 3-40, 10-34
「メッセージ」ウィンドウ, 3-24
メッセージ関数
WF_NOTIFICATION(), 3-36
メッセージ属性, 3-24, 3-25, 3-44, 10-34
#ATTACHED_URLS, 3-34
#FROM_ROLE, 3-30
#HDR, 3-32
#HIDE_MOREINFO, 3-29
#HIDE_REASSIGN, 3-27
#WF_REASSIGN_LOV, 3-26
#WF_SIG_ID, 3-32
#WF_SIG_POLICY, 3-31
#WFM_CANCELED, 3-34
#WFM_CLOSED, 3-35
#WFM_FROM, 3-33
#WFM_HTMLAGENT, 3-33
#WFM_OPEN_INVALID, 3-35

#WFM_OPEN_MAIL, 3-34
#WFM_OPEN_MAIL_DIRECT, 3-34
#WFM_OPEN_MAIL_FYI, 3-34
#WFM_OPEN_MAIL_PSIG_ONLY, 3-35
#WFM_OPEN_MORE_INFO, 3-35
#WFM_OPEN_SIGN, 3-35
#WFM_REPLYTO, 3-33
「応答」, 3-25, 3-45, 3-50
「送信」, 3-25, 3-45
ソース, 3-25, 3-45
通知メーラー属性, 3-33
メッセージ・テンプレート属性, 3-34
メッセージ・テンプレート, 3-34
メニュー
Oracle Workflow Builder, B-2

も

「モニター URL」アクティビティ, 5-15

や

矢印, 4-2

ゆ

有効
日付, 2-8
有効日, 2-15, 2-17, 3-71
有効日付, 2-17
ユーザー
アド・ホック, 4-26
ユーザー定義の警告処理プロセス
アクティビティ, 10-51
概要, 10-50

よ

要約通知送信イベント, 9-26
要約通知の開始, 9-31

ら

ライセンス保有, 8-74, 8-89

る

ループ, 3-71, 6-6
「ループ・カウンタ」アクティビティ, 5-7
ループ・リセット, 4-4
ルール関数, 6-29
 イベント・サブスクリプション, 8-44, 8-90

れ

例外キュー, 8-71

ろ

ローカル・システム, 8-20, 8-125
ローカル・システム識別子, 8-133
ロード・バランス, 5-9
ロール, 4-26
 Builder からの表示, 4-28
 Workflow Builder へのロード, 4-26
 アド・ホック, 4-26
「ロール」
 タブ, 4-26
 プロパティ画面, 4-28
「ロール解決」アクティビティ, 5-9
ロール・タイプ属性, 3-4
「ロールの拡張」チェック・ボックス, 3-54, 3-60,
 3-74

わ

ワークフロー・イベント・プロトコル・プロセス
 概要, 9-40
ワークフロー・エージェントの Ping/ 確認, 8-137
「ワークフロー・エージェントの Ping/ 確認」
 項目タイプ, 8-139
 項目タイプ属性, 8-139
ワークフロー・エンジン, 1-3
 エラー処理, 11-2
 コストのしきい値, 3-58
ワークフロー送信プロトコル
 サンプル・ワークフロー・プロセス, 9-38
「ワークフロー送信プロトコル」
 項目タイプ, 9-39
ワークフロー送信プロトコル・イベント, 9-45
ワークフロー送信プロトコル受信確認イベント, 9-47
ワークフロー・ダイアグラム

 表示, 10-3
ワークフロー定義
 ソース・コントロール, 2-13
 テスト, 7-2, 7-6
 ロード, 1-4
ワークフロー定義のテスト, 7-8
ワークフロー定義ローダー, 1-4
「ワークフロー・デモンストレーション」ホーム・ペー
 ジ, 10-2
「ワークフローの実行」ページ, 7-8
ワークフロー・プロセス
 サンプル, 10-2
ワークフロー・プロセスの開始, 10-9, 10-55
「割当」アクティビティ, 5-15

