

Oracle Applications

システム管理者ガイド

リリース 11*i*

2000 年 8 月

部品番号 : J02502-01

Oracle Applications システム管理者ガイド, リリース 11i

部品番号 : J02502-01

原本名 : Oracle Applications System Administrator's Guide

原本部品番号 : A75396-01

Copyright © 1996, 2000, Oracle Corporation. All rights reserved.

Printed in Japan.

制限付権利の説明

プログラムの使用、複製または開示は、オラクル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に関する法律により保護されています。

当ソフトウェア（プログラム）のリパース・エンジニアリング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ことがあります。オラクル社は本ドキュメントの無謬性を保証しません。

* オラクル社とは、Oracle Corporation（米国オラクル）または日本オラクル株式会社（日本オラクル）を指します。

危険な用途への使用について

オラクル社製品は、原子力、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を用途として開発されておりません。オラクル社製品を上述のようなアプリケーションに使用することについての安全確保は、顧客各位の責任と費用により行ってください。万一かかる用途での使用によりクレームや損害が発生いたしましても、日本オラクル株式会社と開発元である Oracle Corporation（米国オラクル）およびその関連会社は一切責任を負いかねます。当プログラムを米国国防総省の米国政府機関に提供する際には、『Restricted Rights』と共に提供してください。この場合次の Legend が適用されます。

Restricted Rights Legend

Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication and disclosure of the Programs shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are "restricted computer software" and use, duplication and disclosure of the Programs shall be subject to the restrictions in FAR 52.227-14, Rights in Data -- General, including Alternate III (June 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

このドキュメントに記載されているその他の会社名および製品名は、あくまでその製品および会社を識別する目的にのみ使用されており、それぞれの所有者の商標または登録商標です。

目次

1 Oracle Applications セキュリティの概要

職責の定義	1-2
職責に関するその他の注意	1-3
要求セキュリティ・グループの定義	1-4
「職責」 ウィンドウ	1-6
職責ブロック	1-7
有効日	1-7
使用可能：自	1-7
データ・グループ	1-7
要求グループ	1-8
メニュー除外ブロック	1-8
Self-Service Applications セキュリティ	1-9
「ユーザー」 ウィンドウ	1-9
ユーザー・ブロック	1-10
職責ブロック	1-12
セキュリティ属性	1-12
職責別ユーザー・レポート	1-13
レポート・パラメータ	1-13
レポート見出し	1-13
列見出し	1-13
有効職責レポート	1-14
レポート・パラメータ	1-14
レポート見出し	1-14
列見出し	1-14
有効ユーザー・レポート	1-15
レポート・パラメータ	1-15

レポート見出し	1-15
列見出し	1-15
「職責別レポートおよびレポート・セット」レポート	1-16
レポート・パラメータ	1-16
レポート見出し	1-17
機能セキュリティ	1-17
機能セキュリティの概要	1-17
用語	1-17
フォームおよびサブ機能	1-19
機能、メニューおよび「ナビゲータ」ウィンドウ	1-20
機能セキュリティの動作方法	1-20
機能セキュリティの実装	1-22
新規メニュー体系の定義	1-24
メニューの定義についての注意	1-24
アップグレード時のカスタム・メニューの保存	1-25
Oracle HRMS、Oracle Sales and Marketing 用特殊機能	1-26
機能セキュリティの要約	1-27
「フォーム機能」ウィンドウ	1-28
フォーム機能ブロック	1-28
摘要	1-28
フォーム	1-29
Web HTML および Web ホスト	1-29
「メニュー」ウィンドウ	1-30
メニュー・ブロック	1-31
ツリーの表示	1-31
メニュー項目ブロック	1-32
メニュー・ビューア	1-33
機能性	1-34
機能セキュリティ・レポート	1-34
メニュー・レポート	1-34
レポート・パラメータ	1-35
レポート見出し	1-35
列見出し	1-35

2 ユーザー監査およびデータ監査の概要

ユーザー・アクティビティの監査	2-1
データベース行の変更の監査	2-1

ユーザー・アクティビティの監査	2-2
主な機能	2-3
サインオン監査の設定	2-3
「アプリケーション・モニター」の使用	2-5
「このレコードについて」ウィンドウ	2-6
失敗ログインの通知	2-7
サインオン監査レポート	2-7
「ユーザーのモニター」ウィンドウ	2-8
前提条件	2-8
ユーザーのモニター・ブロック	2-8
サインオン監査コンカレント要求レポート	2-9
レポート・パラメータ	2-9
レポート見出し	2-10
列見出し	2-10
サインオン監査フォーム・レポート	2-11
レポート・パラメータ	2-11
レポート見出し	2-12
列見出し	2-12
サインオン監査職責レポート	2-13
レポート・パラメータ	2-14
レポート見出し	2-14
列見出し	2-15
サインオン監査不成功ログイン・レポート	2-15
レポート・パラメータ	2-16
レポート見出し	2-16
列見出し	2-17
サインオン監査ユーザー・レポート	2-17
レポート・パラメータ	2-18
レポート見出し	2-18
列見出し	2-18
監査証跡データについてのレポート	2-20
監査証跡	2-20
リリース 11i の監査証跡の設定	2-20
監査証跡表等の更新レポート	2-22
リリース 11 の監査証跡表、トリガーおよびビュー	2-22
監査表の変更	2-28
監査情報についてのレポート	2-29
監査証跡の使用禁止と、監査データのアーカイブ	2-30

「導入の監査」ウィンドウ	2-31
導入の監査ブロック	2-32
「監査グループ」ウィンドウ	2-32
監査グループ・ブロック	2-33
監査表ブロック	2-34
「監査表」ウィンドウ	2-35
監査表の定義ブロック	2-35
監査列ブロック	2-36

3 コンカレント・プログラムとコンカレント要求の管理

コンカレント・プログラムおよびコンカレント要求の概要	3-1
コンカレント要求の多言語サポート	3-2
プログラムの要求セット別編成	3-3
要求セットの定義	3-3
要求セット・ステージ	3-4
要求セットおよび所有者	3-9
システム管理者の要求セット権限	3-10
要求セットの非互換性	3-11
要求セット内のパラメータの共有	3-12
要求セット・レポート	3-15
レポート・パラメータ	3-15
レポート見出し	3-15
プログラムの要求グループ別編成	3-15
要求セキュリティ・グループ	3-16
要求グループでのコードの使用	3-17
コードを使用した「要求の発行」ウィンドウのカスタマイズ	3-17
「要求の発行」ウィンドウのカスタマイズ	3-18
レポート・グループ職責レポート	3-20
レポート・パラメータ	3-21
プログラム非互換性ルール of 定義	3-21
非互換プログラムおよび単独実行プログラム	3-21
コンカレント衝突ドメイン	3-22
非互換性ルールの適用	3-24
データ・グループの定義	3-25
データ・グループの概要	3-25
データ・グループの使用	3-26
データ・グループの修正	3-28

カスタム・コンカレント・プログラム	3-29
ログ・ファイル名および出力ファイル名	3-29
Oracle ツール・コンカレント・プログラム	3-30
Pro*C コンカレント・プログラム	3-31
ホスト言語のコンカレント・プログラム	3-34
コンカレント要求の発行 (CONCSUB)	3-35
プログラム定義のコピーと変更	3-41
コンカレント・プログラムのコピーとプログラム名の変更	3-41
プログラムの優先順位の変更	3-42
互換性がないプログラムのリストの修正	3-43
コンカレント・プログラムのパラメータ	3-43
要求パラメータの動作の制御	3-44
プログラムのパラメータの変更例	3-50
コンカレント・プログラム詳細レポート	3-52
レポート・パラメータ	3-52
レポート見出し	3-52
コンカレント・プログラム・レポート	3-53
レポート・パラメータ	3-53
レポート見出し	3-53
「要求グループ」ウィンドウ	3-53
要求グループ・ブロック	3-54
要求ブロック	3-54
「コンカレント・プログラム実行ファイル」ウィンドウ	3-55
コンカレント・プログラム実行ファイル・ブロック	3-55
コンカレント・プログラム・ウィンドウ	3-57
コンカレント・プログラム・ブロック	3-57
ウィンドウへのコピー	3-62
「セッション管理」ウィンドウ	3-62
「データ・グループ」ウィンドウ	3-67
データ・グループ・ブロック	3-68
アプリケーション -Oracle ID 対ブロック	3-68

4 コンカレント処理の管理

コンカレント・プロセスの概要	4-1
コンカレント要求、コンカレント・プログラムおよびコンカレント・プロセス	4-1
Oracle Enterprise Manager との統合	4-3
Oracle Applications Manager	4-3
Oracle Management Pack for Oracle Applications	4-4

Oracle Applications Manager	4-4
Oracle Management Pack for Oracle Applications	4-5
コンカレント要求のライフ・サイクル	4-6
要求のステータスおよび出力を表示する方法	4-7
エンド・ユーザー・レポートおよびログ・ファイルのアクセス権限の設定	4-8
Report Review Agent の起動	4-9
Reports Viewer の定義	4-10
ログ・ファイルのタイプ	4-10
レポート出力ファイルへのオペレーティング・システム・アクセス	4-12
コンカレント要求のステータスの変更	4-12
要求のフェーズおよびステータスの変更	4-13
「保留」または「無効」要求の優先順位の変更	4-13
コンカレント・プロセス用のファイルおよび表の管理	4-14
保守のための提案	4-15
「コンカレント要求やマネージャ・データのページ」プログラム	4-16
レポート・オプション	4-16
コンカレント・プロセスのユーザー・プロファイル設定値	4-19
マネージャおよびマネージャの稼働シフトの定義	4-22
稼働シフトの定義	4-25
稼働シフトによる処理作業負荷のバランス化	4-28
時間ベース・キューの使用方法	4-29
完了したコンカレント要求レポート	4-31
レポート・パラメータ	4-31
レポート見出し	4-32
マネージャ別稼働シフト・レポート	4-32
レポート・パラメータ	4-32
レポート見出し	4-32
稼働シフト・レポート	4-33
レポート・パラメータ	4-33
レポート見出し	4-33
特定なプログラムのみを実行するためのマネージャの特殊化	4-33
特殊化ルールの概要	4-33
特殊化ルールの定義	4-34
例 - 特殊化ルールの使用	4-38
組合せ特殊化ルールの定義	4-43
組合せルールの使用	4-45
特殊化ルールと組合せルールの違い	4-48
プログラムの要求タイプ別グループ化	4-49

コンカレント・マネージャの制御	4-50
マネージャの状態	4-50
「マネージャの管理」フォームからマネージャを制御する	4-52
内部コンカレント・マネージャのオペレーティング・システムからの制御	4-53
パラレル・コンカレント・プロセスの概要	4-58
パラレル・コンカレント・プロセスについて	4-59
パラレル・コンカレント・プロセスの環境	4-59
パラレル・コンカレント・プロセスの実行方法	4-60
パラレル・コンカレント・プロセスの管理	4-62
コンカレント・マネージャの定義	4-62
コンカレント・マネージャの管理	4-63
データベース・インスタンス、マネージャの位置およびファイルの配布	4-64
パラレル・コンカレント・プロセスの例	4-65
パラレル・コンカレント・プロセス - 単一データベース・インスタンス (集中および共有ログ/出力/実行ファイル・システム)	4-66
パラレル・コンカレント・プロセス - Oracle Parallel Server (集中および共有ログ/出力/実行ファイル・システム)	4-67
パラレル・コンカレント・プロセス - 単一データベース・インスタンス (分散ログ/出力/実行ファイル)	4-68
パラレル・コンカレント・プロセス - ノード1がデータベース・サーバーとして動作し、 ノード2～5がコンカレント・マネージャ・サーバーとして動作する場合	4-69
パラレル・コンカレント・プロセスのインストール	4-70
「コンカレント・マネージャの管理」ウィンドウ	4-71
コンカレント・マネージャ・ブロックの管理	4-71
「コンカレント・プロセス」ウィンドウ	4-73
「要求」ウィンドウ	4-76
要求診断ウィンドウ	4-76
「コンカレント・マネージャ」ウィンドウ	4-76
「コンカレント・マネージャ」ブロック	4-76
並列コンカレント・プロセス詳細	4-78
「稼動シフト」ウィンドウ	4-79
「特殊化ルール」ウィンドウ	4-80
「稼動シフト」ウィンドウ	4-81
「組合せ特殊化ルール」ウィンドウ	4-82
組合せ特殊化ルール・ブロック	4-83
特殊化ルール・ブロック	4-83
「コンカレント要求タイプ」ウィンドウ	4-84
コンカレント要求タイプ・ブロック	4-85
「ノード」ウィンドウ	4-85

ノード・ブロック	4-85
「ビューア・オプション」 ウィンドウ	4-86
「ビューア・オプション」 ブロック	4-86

5 プリンタ

プリンタおよび印刷の概要	5-1
管理者用要約	5-1
Oracle Reports および Oracle Applications の印刷	5-4
プリンタ・タイプ、印刷形式およびプリンタ・ドライバ	5-5
キャラクタ・モード印刷またはビットマップ印刷の設定	5-6
プリンタ・タイプの定義およびプリンタの登録	5-8
印刷形式	5-9
プリンタ・ドライバ	5-9
プリンタの設定	5-10
印刷設定の相互関係	5-11
プリンタ設定情報の要求時読み込み	5-12
Oracle Applications の印刷サポートのカスタマイズ	5-13
カスタム・プリンタ・ドライバの作成	5-14
初期化文字列および再設定文字列の使用	5-18
スプール・ファイルの使用	5-19
カスタム SRW ドライバの作成	5-20
UNIX での Postscript 印刷	5-21
例 - enscript を使用した Postscript の印刷	5-22
プリンタおよび印刷形式割当の階層	5-24
プリンタ割当の階層	5-25
印刷形式割当の階層	5-25
システム管理者によるプリンタおよび印刷形式の設定	5-26
エンド・ユーザーによるプリンタおよび印刷形式の設定	5-27
「プリンタ・タイプ」 ウィンドウ	5-28
プリンタ・タイプ・ブロック	5-28
プリンタ・ドライバ・ブロック	5-28
「プリンタ」 ウィンドウ	5-29
プリンタ・ブロック	5-29
「印刷形式」 ウィンドウ	5-30
印刷形式ブロック	5-30
「プリンタ・ドライバ」 ウィンドウ	5-31
プリンタ・ドライバ・ブロック	5-32

6 Oracle Applications ヘルプ

Oracle Applications ヘルプのカスタマイズ	6-1
ヘルプ・ファイルのダウンロードおよびアップロード	6-1
編集するためのヘルプ・ファイルのダウンロード	6-2
新規または変更されたヘルプ・ファイルのアップロード	6-2
ヘルプ・ファイルのリンク	6-3
特別リンク構文	6-4
ウィジェットの全容	6-5
クロス・アプリケーション・リンク	6-5
関連トピック・リンク	6-6
コンテキスト依存ヘルプ	6-6
検索インデックスの更新	6-7
ヘルプ・ナビゲーション・ツリーのカスタマイズ	6-7
グローバル環境でのヘルプのカスタマイズ	6-8
異言語間でのリンク	6-8
ローカライズされたヘルプ・ファイルのリンク	6-9

7 アプリケーション DBA の職務

アプリケーション DBA の職務の概要	7-1
Oracle ユーザー名	7-1
Oracle ユーザー名の登録	7-2
Oracle ユーザー・ウィンドウ	7-3
前提条件	7-5
Oracle ユーザー・ブロック	7-5
「コンカレント衝突ドメイン」ウィンドウ	7-7
アプリケーション・ウィンドウ	7-8
前提条件	7-8
アプリケーション・ブロック	7-8
ネットワーク・テスト・ウィンドウ	7-9
フォルダの管理	7-10
「言語」ウィンドウ	7-13
言語ブロック	7-13
「地域」ウィンドウ	7-13
地域ブロック	7-13
Oracle Applications の生産資源コンシューマ・グループ	7-14
生産資源コンシューマ・グループの割当て	7-14

生産資源コンシューマ・グループ割当ての階層	7-15
-----------------------------	------

8 Oracle Applications におけるコスト・ベース最適化

Oracle Applications とコスト・ベース最適化	8-1
CBO のコンカレント・プログラム	8-1
表統計の収集	8-2
バックアップ表統計	8-2
表統計の復元	8-2
スキーマ統計の収集	8-3
列統計の収集	8-3
全列統計の収集	8-3
全インデックス列の統計を分析	8-3
FND_STATS パッケージ	8-3
CREATE_STAT_TABLE プロシージャ	8-4
BACKUP_TABLE_STATS	8-4
BACKUP_SCHEMA_STATS プロシージャ	8-5
RESTORE_SCHEMA_STATS プロシージャ	8-6
RESTORE_TABLE_STATS プロシージャ	8-6
RESTORE_COLUMN_STATS プロシージャ	8-7
GATHER_SCHEMA_STATS プロシージャ	8-8
GATHER_SCHEMA_STATISTICS プロシージャ	8-9
GATHER_INDEX_STATS プロシージャ	8-10
GATHER_TABLE_STATS プロシージャ	8-11
GATHER_COLUMN_STATS プロシージャ	8-12
GATHER_ALL_COLUMN_STATS プロシージャ	8-14
ANALYZE_ALL_COLUMNS プロシージャ	8-14
LOAD_XCLUD_STATS プロシージャ	8-15
FND_CTL	8-16
FND_SESS_CTL のパラメータ	8-16
例	8-16

9 ユーザー・プロファイル

ユーザー・プロファイル設定の概要	9-1
主な機能	9-1
ユーザー・プロファイル・オプションの設定	9-2
プロファイル・オプションをパラメータまたはセグメントのデフォルト値として使用する場合	9-3
ユーザー・プロファイル・オプションの例	9-4

ユーザー・プロファイル・オプション値レポート	9-5
レポート・パラメータ	9-5
レポート見出し	9-5
システム・プロファイル値ウィンドウ	9-6
プロファイル値ブロック	9-6
「システム・プロファイル値の検索」ブロック	9-7

10 プロセス・ナビゲーションの管理

プロセス・ナビゲータ処理の作成	10-1
プロセス・ナビゲータ処理の作成	10-2

11 文書連番

文書連番とは	11-1
文書連番の定義	11-3
文書カテゴリの定義	11-5
文書連番の割当	11-6
文書採番と文書入力	11-7
文書連番ウィンドウ	11-8
文書連番ブロック	11-8
「文書カテゴリ」ウィンドウ	11-10
文書カテゴリ・ブロック	11-10
「連番割当」ウィンドウ	11-11
連番割当ブロック	11-11
割当リージョン	11-12
文書フレックスフィールド	11-12

- A Oracle Application Object Library のプロファイル・オプション**
- B 登録済みアラートの使用**
- C ローダー**
- D OracleApplications システム管理者の文字モード・フォームおよび対応する GUI ウィンドウ**
- E Oracle Applications System Administrator の設定**
- F TCF サーバーの管理**

Oracle Applications セキュリティの概要

システム管理者は、Oracle Applications のユーザーを定義し、各ユーザーに 1 つ以上の職責を割り当てます。

アプリケーション・ユーザーの定義

アプリケーション・ユーザーを定義すると、新しいユーザーは Oracle Applications にサインオンできるようになります。各アプリケーション・ユーザーにはユーザー名とパスワードが割り当てられます。初期パスワードはシステム管理者が定義しますが、アプリケーション・ユーザーは、最初にサインオンするときに必ず新しい（秘密の）パスワードを入力する必要があります。

アプリケーション・ユーザーを定義するときには、ユーザーに対して 1 つ以上の職責を割り当てます。割り当てられた職責が 1 つのみの場合、そのアプリケーション・ユーザーはサインオン直後にアプリケーションに入ります。

2 つ以上の職責が割り当てられると、そのアプリケーション・ユーザーには、サインオンの後、使用可能な職責をリストしたウィンドウが表示されます。

アプリケーションの権限を定義する職責

職責とは、Oracle Applications での「権限レベル」のことで、組織内における各自の役割の遂行に必要と認められる範囲内で Oracle Applications の機能とデータへのアクセスを許可します。各職責でアクセスできる対象は次のとおりです。

- Oracle General Ledger や Oracle Planning などの特定のアプリケーション。
- 「アメリカ業務」や「ドイツ売上」などの会計帳簿、または「ニューヨーク製造」や「ニューヨーク流通」などの組織。
- ユーザーがナビゲートできる各ウィンドウの制限リスト。たとえば、ある職責では、Oracle Planning の一定のユーザーは予測品目を入力できますが、基準需要計画品目は入力できません。

- 実行できる機能の制限リスト。たとえば、2つの職責が同じ1つのウィンドウにアクセスできるとき、一方の職責で使用するウィンドウには他方のウィンドウにはない追加機能ボタンが表示されます。
- 特定のアプリケーション内のレポート。システム管理者はレポート・グループを1つ以上の職責に割り当てて、ユーザーが選択した職責に応じて、出力可能なレポートを決定できます。

各ユーザーには少なくとも1つ以上の職責が与えられます。また複数のユーザーが同じ職責を共有することもできます。システム管理者は、Oracle Applications で提供されている標準職責をユーザーに割り当てたり、または新しい特別な職責を作成できます。

職責の定義

職責を定義するときには、次に記述するコンポーネントの一部またはすべてを、その職責に割り当てます。

データ・グループ (必須)

データ・グループは、Oracle Applications 製品および ORACLE ID 間のマッピングを定義します。データ・グループにより、職責のフォーム、コンカレント・プログラムおよびそれに関連するレポートのアカウントを処理する Oracle データベースが決定されます。参照：[「データ・グループの定義」](#) (3-25)。

要求セキュリティ・グループ (オプション)

要求セキュリティ・グループは、特定の職責のもとでアプリケーション・ユーザーが実行できる要求や要求セットを含むコンカレント・プログラムを定義します。参照：[「要求セキュリティ・グループの定義」](#) (1-4)。参照：[「プログラムの要求グループ別編成」](#) (3-15)。

メニュー (必須)

メニューとは、「ナビゲータ」ウィンドウに表示される各アプリケーション機能（フォーム）を階層型に配置したものです。メニューは、「ナビゲータ」ウィンドウには表示されないが、ある職責で使用可能なアプリケーション機能の範囲を定義している非フォーム機能（サブ機能）を指すこともあります。それぞれの職責は、各メニューと関連付けられています。参照：[「機能セキュリティの概要」](#) (1-17)。

機能とメニュー除外 (オプション)

職責では、その職責で使用可能なアプリケーション機能を制限するために、オプションで機能やメニューの除外規定をその職責に対応付けることができます。参照：[「機能セキュリティの概要」](#) (1-17)。

職責に関するその他の注意

事前定義済みの職責

Oracle Applications の全製品には、事前定義済みの職責がインストールされています。Oracle Applications 製品にインストールされている事前定義済みの職責の名前については、関連のリファレンス・ガイドを参照してください。

その他にも、職責（データ・グループ、要求セキュリティ・グループ、メニューおよび機能）を定義するのに役立つ主要コンポーネントのインスタンスが、Oracle Applications 用に事前定義されています。

職責および要求セキュリティ・グループ

要求グループが職責に割り当てられると、それは要求セキュリティ・グループになります。

ユーザーは、「要求の発行」フォームなどの標準発行フォームからは、自分の職責の要求セキュリティ・グループ内にあるレポート、コンカレント・プログラムおよび要求セットしか実行できません。

- 職責に対してメニューに「要求の発行」フォームが含まれていない場合は、要求セキュリティ・グループを職責に割り当てる必要はありません。
- 要求セキュリティ・グループが職責に割り当てられていない場合、その職責のもとで作業するユーザーは、標準発行フォームからレポート、要求セットまたはその他のコンカレント・プログラムを実行することはできません。

職責および機能セキュリティ

Oracle Applications の GUI ベースのアーキテクチャでは、関連するいくつかの業務機能が 1 つのフォームに集約されます。アプリケーションの機能の一部は、Oracle Applications の個々の機能として識別したり、保護できます（つまり職責に含めたり、職責から除外できます）。

参照：「[機能セキュリティの概要](#)」（1-17）

関連項目

- 「[Oracle Applications セキュリティの概要](#)」（1-1）
- 「[フォーム機能](#)」ウィンドウ（1-28）
- 「[メニュー](#)」ウィンドウ（1-30）
- 「[ユーザー](#)」ウィンドウ（1-9）
- 「[職責](#)」ウィンドウ（1-6）

- 「機能セキュリティの実装」 (1-22)
- 「機能セキュリティの動作方法」 (1-20)
- 「新規メニュー体系の定義」 (1-24)

要求セキュリティ・グループの定義

この章で簡単に説明されている他にも、要求グループおよび要求セキュリティ・グループは、広範囲に及ぶトピック（必ずしもアプリケーション・セキュリティに限定されていない）の一部として、第7章「コンカレント・プログラムおよびレポートの管理」に詳しく述べられています。

参照：

「プログラムの要求グループ別編成」 (3-15)

「要求グループ」ウィンドウ (3-53)

要求セキュリティの使用

システム管理者は要求セキュリティを使用して、ユーザーが「要求の発行」フォームなどの標準発行フォームからユーザーが実行できるレポート、要求セットおよびコンカレント・プログラムを指定します。

要求セキュリティを設定するには、「要求グループ」フォームを使用して要求グループを定義します。システム管理者は、「職責」フォームを使用して、要求グループを職責に割り当てます。この時点で、要求グループは要求セキュリティ・グループになります。参照：「[要求セキュリティ・グループ](#)」 (3-16)。

システム管理者は要求グループを定義して、単一要求、要求セット、またはすべての要求および要求セットを1つのアプリケーションに入れることができます。

すべての要求および要求セットを1つのアプリケーションに入れるように選択すると、ユーザーは、これから新しい要求や要求セット（所有者なし）に自動的にアクセスできます。

要求セキュリティ・グループには、別のアプリケーションからの要求や要求セットを入れることができます。別のアプリケーションからの要求を所有する要求セキュリティ・グループを定義するには、「データ・グループ」についての説明を参照してください。参照：「[データ・グループの定義](#)」 (3-25)。

注意： 要求セキュリティ・グループまたは要求グループは、セキュリティ・グループと同じものではありません。

個々の要求および要求セット

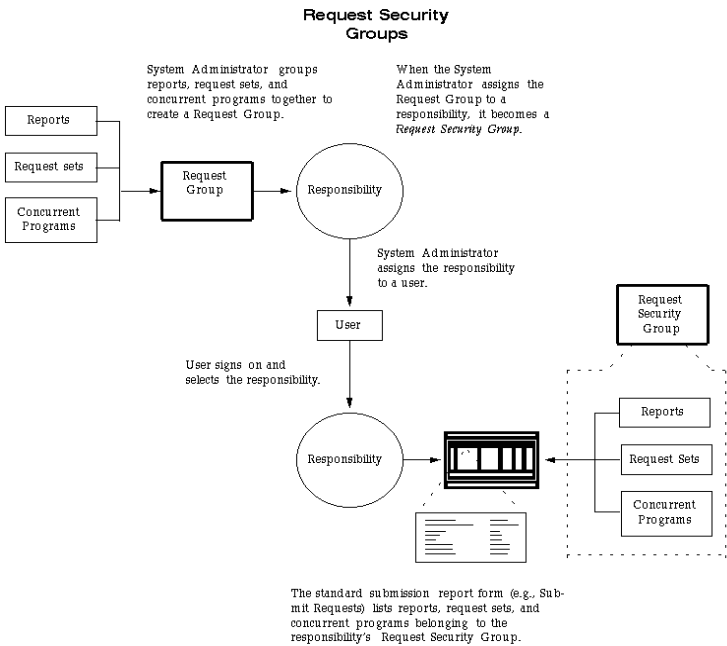
個々の要求セキュリティ・グループには含まれていないが、要求セキュリティ・グループに含まれている要求セットに属しているレポートやコンカレント・プログラムには、次の権限があります。

- ユーザーは、「要求の発行」フォームを使用して、そのユーザーの職責の要求セキュリティ・グループに含まれていない単一要求や要求セットを実行することはできません。
- しかしユーザーは、要求セットがそのユーザーの要求セキュリティ・グループに含まれている場合に、その要求セキュリティ・グループに含まれていない要求が入っている要求セットを実行できます。

システム管理者がセットに入っている要求ではなく、要求セットを要求セキュリティ・グループに割り当てた場合、ユーザーの操作は次のように制約されます。

- 要求セットの定義にある要求情報を編集できません。
- セットに入っている特定の要求の実行を停止できません。
- ユーザーが要求セットの所有者として割り当てられている場合に限り、要求セットから要求を削除したり要求セットに別の要求を追加することにより、要求セットを編集できます。

「要求セキュリティ・グループ」に関する図に、要求セキュリティ・グループと、アプリケーション・ユーザーおよび職責間の関連性が示されています。



関連項目

- [「Oracle Applications セキュリティの概要」](#) (1-1)
- [「フォーム機能」ウィンドウ](#) (1-28)
- [「メニュー」ウィンドウ](#) (1-30)
- [「ユーザー」ウィンドウ](#) (1-9)
- [「職責」ウィンドウ](#) (1-6)
- [「システム管理者の要求セット権限」](#) (3-10)
- [「職責の定義」](#) (1-2)
- [「要求セットおよび所有者」](#) (3-9)

「職責」ウィンドウ

職責を定義するには、このウィンドウを使用します。各アプリケーション・ユーザーには最低 1 つの職責が割り当てられます。職責は、そのユーザーが Oracle Applications または Oracle Self-Service Web Applications にアクセスするかどうか、どのアプリケーション機能をユーザーが使用できるか、どのレポートとコンカレント・プログラムをそのユーザーが実行できるか、およびそれらのレポートとコンカレント・プログラムがどのデータにアクセスできるかを決定します。

注意： 職責は削除できません。職責を使用できないようにするには、「有効日」の「至」フィールドを過去の日付に設定します。変更の効果を確認するには、Oracle Applications を再起動する必要があります。

参照：[「機能セキュリティの概要」](#) (1-17)

前提条件

- 「データ・グループ」ウィンドウで、自分の職責のコンカレント・プログラムがアプリケーションごとに参照する Oracle ユーザー名のリストを作成します。
- 「要求グループ」ウィンドウでは、この職責とともに使用できる要求グループを定義します。
- 「メニュー」ウィンドウでは、この職責に割り当てるときに選択する事前定義済みのメニューを参照します。

職責ブロック

アプリケーション名および職責名を使用して職責を一意に識別します。

職責名

複数の職責を持っている場合は、サインオンするとこの名前が含まれたポップアップ・ウィンドウがオープンします。

アプリケーション

メニューで、他のアプリケーションへのアクセスを定義しておく、ここで使用されるアプリケーション名により、ここでの職責のユーザーが他のアプリケーションのフォームおよび機能にアクセスできなくなることはありません。

職責キー

これは職責を示す一意の名称で、ローダー・プログラムが使用します。ローダーとは、メッセージ、ユーザー・プロファイル、ユーザー・プロファイル値などの情報を Oracle Applications 表にロードするために使用するコンカレント・プログラムのことです。システム内で自分の職責キーが一意であるようにするため、それぞれの職責キーの名前を、この職責に関連したアプリケーション短縮名で開始するようにしてください。

有効日

自 / 至

職責をアクティブにする開始日と非アクティブにする終了日を入力します。開始日のデフォルト値は当日で、終了日を入力しないと職責は永久に有効となります。

職責の情報は監査証跡に使用されるため、職責を削除することはできません。職責は、終了日を当日に設定することにより、いつでも非アクティブ化することができます。非アクティブ化した職責を再度アクティブにするには、終了日を当日以降の日付に変更するか、終了日を消去します。

使用可能：自

職責は、ただ1つのアプリケーション・システムへ関連付けることができます。Oracle Self-Service Web Applications と Oracle Applications のどちらかを選択してください。

データ・グループ

名称 / アプリケーション

データ・グループでは、アプリケーションと Oracle ユーザー名の対が定義されます。

この職責を選択したときに、Oracle ユーザー名フォームが接続されるアプリケーションを選択します。Oracle ユーザー名は、職責がアクセスできるデータベース表と表権限を決めます。

す。取引マネージャは、取引マネージャと同じデータ・グループを割り当てられた職責からの要求のみを処理します。

メニュー

名前を入力するメニューは、すでに Oracle Applications で定義されているものでなければなりません。参照：「[メニュー](#) ウィンドウ」(1-30)。

Web ホスト名

Web Server がデータベースと異なるマシン上にある場合は、ここにホスト名（URL）を指定する必要があります。指定しなかった場合、Web ホスト名はデフォルトで現行のデータベース・ホスト・サーバーになります。

Web エージェント名

この職責が使用するデータベースの PL/SQL エージェント名を入力します。エージェント名を指定しなかった場合、職責はデフォルトでログオンの時点で有効なエージェント名になります。

要求グループ

名称 / アプリケーション

この職責に要求セキュリティ・グループを割り当てないと、この職責を持ったユーザーは「要求の発行」ウィンドウから要求、要求セット、またはコンカレント・プログラムを実行できません（ただし、ユーザー自身が所有している要求セットは例外です）。ユーザーは、要求グループ・コードを使用してメニュー・パラメータによりカスタマイズした「要求の発行」ウィンドウから要求にアクセスできます。

参照：

「[Oracle Applications セキュリティの概要](#)」(1-1)

「[コードを使用した「要求の発行」ウィンドウのカスタマイズ](#)」(3-17)

「[要求グループ](#)」ウィンドウ」(3-53)

メニュー除外ブロック

アプリケーションの職責へのアクセスを制限するために機能およびメニューの除外ルールを定義します。

タイプ

この職責に対して適用される除外ルールのタイプとして「機能」または「メニュー」を選択します。

- 職責から機能1つを除外すると、その職責のメニュー体系全体からその機能のすべてのオカレンスが除外されます。
- メニューを除外すると、そのメニュー項目のすべて、つまりそこで選択できる機能および機能のメニューのすべてが除外されます。

名称

この職責から除外する機能またはメニューの名前を選択します。ここで指定する機能またはメニューは、あらかじめ Oracle Applications で定義されている必要があります。

Self-Service Applications セキュリティ

Oracle Self-Service Web Applications は、ユーザーがアクセスできる情報を定義するためにデータベース表の列、行および値を使用します。表列は、職責にセキュリティ属性または除外属性として割り当てることができる属性を表しています。それらの属性は、Web Application Dictionary の中で定義します。

関連項目

その他の関連項目

Self-Service のインプリメンテーション・ガイド

除外項目

「値リスト」を使用して有効な名前を選択します。職責には、任意の数の除外属性を割り当てることができます。

セキュリティ属性

「値リスト」を使用して有効な名前を選択します。職責には、任意の数のセキュリティ属性を割り当てることができます。

「ユーザー」ウィンドウ

このウィンドウを使用して、アプリケーション・ユーザーを定義します。アプリケーション・ユーザーは、Oracle Applications または Oracle Self-Service Applications、あるいはその両方の使用権限を与えられたユーザーであり、アプリケーション・ユーザー名によって一意に識別されます。新しく定義されたアプリケーション・ユーザーは、Oracle Applications にサインオンして Oracle Applications のウィンドウでデータにアクセスできます。参照：[「Oracle Applications セキュリティの概要」](#) (1-1)。

ユーザー・ブロック

ユーザー名

アプリケーション・ユーザーは、Oracle Applications にサインオンするために、このユーザー名を入力します。

- ユーザー名は、1 語である必要があります。
- ユーザー名に使用できるのは英数字（A ～ Z および 0 ～ 9）のみです。

ユーザー名には、使用するオペレーティング・システムがファイル名としてサポートする文字セットしか使用できないので注意してください。

提案：ユーザー名には、従業員のイニシャルの後に姓を付けるなど、意味のある名前を付けることをお勧めします。また、グループ・アカウントであれば、グループ・アカウントの目的や性質を表すアプリケーション・ユーザー名を付けることができます。

パスワード

アプリケーション・ユーザーの初期パスワードを入力します。アプリケーション・ユーザーは、Oracle Applications にサインオンするときに自分のユーザー名とともにこのパスワードを入力します。

- パスワードは最短 5 文字以上、最長 100 文字まで使用できます。
- パスワードに使用できるのは英数字（A ～ Z および 0 ～ 9）のみです。それ以外の文字、記号は無効となります。

入力したパスワードは、このウィンドウには示されません。パスワードを入力するときには、タイプミスがないかどうかを確認するために 2 回繰り返して入力します。

入力したアプリケーション・ユーザー名がすでに使用されていて、しかも 2 回のパスワード入力が異なるときには、もとのパスワードは変更されず、自動的に次のフィールドにナビゲートします。

新規のアプリケーション・ユーザーを定義していて、しかも 2 回のパスワード入力が異なるときには、パスワードを再入力するように指示が出ます。新規ユーザーは、2 回のパスワード入力が一致するまで次のフィールドにナビゲートできません。

アプリケーション・ユーザーが初めてサインオンするときには、パスワードを変更する必要があります。ユーザーがパスワードを忘れてしまったときは、このフィールドに新規のパスワードを割り当てることができます。

システム管理者は、初期のパスワードを設定したり、既存のパスワードを変更したりできますが、ユーザーが選択したパスワードにアクセスすることはできません。

個人、顧客および仕入先

これらのフィールドは、従業員（個人）、顧客または仕入先担当の名前を入力するために使用します。このアプリケーション・ユーザー名とパスワードを使用する従業員、顧客および仕入先の姓と名をコンマで区切って入力します。「値リスト」を使用して有効な名前を選択してください。

電子メール/ファックス

このユーザーの電子メール・アドレスやファックス番号を入力します。

パスワード失効

日数

パスワード変更の間隔を示す最大日数を入力します。ここで指定した有効期限の日数が経過すると、ポップアップ・ウィンドウが表示され、アプリケーション・ユーザーにパスワードの変更を要求します。

アクセス

1つのパスワードの有効期限内に Oracle Applications にサインオンすることを許される回数を入力します。アプリケーション・ユーザーがサインオンした回数がここで指定した限度に達すると、ポップアップ・ウィンドウが表示され、アプリケーション・ユーザーにパスワードの変更を要求します。

提案：アプリケーション・ユーザーに対してパスワードを頻繁に変更するよう促すことをお勧めします。そうすることにより、Oracle Applications に権限のないアクセスが行われる心配が少なくなります。

有効日

自/至

ユーザーは、開始日以前と終了日以後は Oracle Applications にサインオンできません。開始日のデフォルト値は、当日です。終了日を指定しないと、ユーザー名は永久に有効となります。

アプリケーション・ユーザー名は監査証跡に使用されるので、Oracle Applications から削除できません。ただし、終了日を当日に設定することにより、Oracle Applications ユーザーをいつでも非アクティブ化できます。

非アクティブ化したユーザーを再度アクティブにするには、終了日を当日以降の日付に変更するか、「終了日」フィールドを消去します。

職責ブロック

職責

アプリケーション・ユーザーに割り当てようとする職責の名前を選択します。職責は、アプリケーション名と職責名の組み合わせで一意に識別されます。

セキュリティ・グループ

このフィールドは HRMS セキュリティ専用です。参照：『Oracle HRMS ユーザーズ・ガイド』の「Oracle HRMS におけるカスタマイズ、レポートおよびシステム管理」。

自 / 至

職責の情報は監査証跡に使用されるため、職責を削除することはできません。ユーザーの職責は、終了日を当日に設定することにより、いつでも非アクティブ化できます。

非アクティブ化したユーザーの職責を再度アクティブにするには、終了日を当日以降の日付に変更するか、終了日を消去します。

セキュリティ属性

セキュリティ属性は Oracle Self-Service Web Applications によって使用され、データの行（レコード）が、その行に含まれる特定のデータ（属性値）に基づいて、指定したユーザーまたは職責に対して可視となるようにします。

ユーザーに割り当てるセキュリティ属性に 1 つまたは複数の値を割り当てることができます。あるセキュリティ属性を職責とユーザーの両方に割り当て、そのユーザーにそのセキュリティ属性の値がない場合、その属性についての情報は戻されません。

たとえば、ADMINR 職責のユーザーが 1000 という CUSTOMER_ID 値の入った行を参照できるようにするには、ADMIN 職責に CUSTOMER_ID という保護セキュリティを割り当てます。その後、そのユーザーに 1000 というセキュリティ属性 CUSTOMER_ID 値を与えます。

そのユーザーが Admin 職責にログインすると、アクセスできる顧客データは、1000 という CUSTOMER_ID 値があるデータのみになります。

属性

そのユーザーからアクセスできるレコードを決定するために使用する属性を選択します。そのユーザーの職責に割り当ててある属性のいずれかから選択できます。

値

そのユーザーからアクセスできるレコードを決定するために使用する属性の値を入力します。

関連項目

- 「Oracle Applications セキュリティの概要」 (1-1)
- 「「職責」ウィンドウ」 (1-6)
- 「機能セキュリティの概要」 (1-17)
- 「職責の定義」 (1-2)

職責別ユーザー・レポート

このレポートは、特定の職責をだれが使用しているかを文書化します。このレポートは、アプリケーション・ユーザーを定義するとき、または編集するときに使用します。

レポート・パラメータ

アプリケーション名

レポートで使用する職責が属するアプリケーションの名前を選択します。

職責名

レポートで使用する職責の名前を選択します。

レポート見出し

レポート見出しはレポートを要求する対象となるアプリケーション名および職責を示します。

列見出し

ユーザー名

職責へ割り当てられたユーザーの名前。

開始日

そのユーザーについて、その職責が有効となった日付。

終了日

職責が、そのユーザーに対して無効になる、または無効になった日付。あるユーザーについて終了日が表示されない場合、そのユーザーにとってその職責は常に使用可能です。

摘要

職責へ割り当てられたユーザーについての記述。

関連項目

- 「Oracle Applications セキュリティの概要」 (1-1)
- 「「ユーザー」 ウィンドウ」 (1-9)
- 「「職責」 ウィンドウ」 (1-6)
- 「機能セキュリティの概要」 (1-17)
- 「職責の定義」 (1-2)

有効職責レポート

このレポートは、現在有効なすべての職責、個々の職責へ現在アクセスできるユーザー、およびそれらのユーザーがその職責へアクセスできる開始日と終了日を示します。

レポート・パラメータ

なし。

レポート見出し

これは、レポートの名前、レポートが実行された日時、およびページ番号を表示します。

列見出し

アプリケーション名

その職責へ関連付けられているアプリケーションの名前。

職責名

現在有効な職責の名前。

ユーザー名

現在その職責へアクセスできるユーザーの名前。

開始日

ユーザーがその職責へのアクセスを開始できる日付。

終了日

ユーザーがその職責へアクセスできる最終日付。参照：「[Oracle Applications セキュリティの概要](#)」(1-1)。

関連項目

- 「[Oracle Applications セキュリティの概要](#)」(1-1)
- 「[ユーザー](#)」ウィンドウ(1-9)
- 「[職責](#)」ウィンドウ(1-6)
- 「[職責の定義](#)」(1-2)

有効ユーザー・レポート

このレポートは、現在有効であり、しかも最低 1 つの有効な職責を持っているすべてのユーザー名を表示します。また、それらのユーザーがアクセスできるすべての職責、およびそれらのユーザーが各職責へアクセスできる開始日および終了日も表示します。

レポート・パラメータ

なし。

レポート見出し

レポート見出しは、レポートの名前、レポートが実行された日付、およびページ番号を表示します。

列見出し

ユーザー名

現在有効なユーザーの Oracle Applications 名。「ユーザー」ウィンドウで指定した開始日および終了日は、ユーザー名が現在有効であるかどうかを決定します。

アプリケーション名

その職責へ関連付けられているアプリケーションの名前。

職責名

現在有効な職責の名前。

開始日

ユーザーがその職責へのアクセスを開始できる日付。開始日は、「ユーザー」ウィンドウの「職責」ブロックでユーザーへ職責を割り当てるときに指定できます。

終了日

ユーザーがその職責へアクセスできる最終日付。開始日は、「ユーザー」ウィンドウの「職責」ブロックでユーザーへ職責を割り当てるときに指定できます。

関連項目

- [「Oracle Applications セキュリティの概要」](#) (1-1)
- [「ユーザー」ウィンドウ](#) (1-9)

「職責別レポートおよびレポート・セット」レポート

このレポートでは、どのレポート（およびその他のコンカレント・プログラム）およびレポート・セットが、個々の職責で使用可能である要求セキュリティ・グループに含まれているかを識別します。このレポートは、職責を定義するとき、または編集するときに使用します。

レポート・パラメータ

パラメータを入力しない場合、このレポートは個々の職責からアクセスできるすべてのレポートおよびレポート・セットを文書化します。

アプリケーション短縮名

使用可能なレポートおよびレポート・セットについての情報が必要な職責に関連付けられているアプリケーション名を選択します。

アプリケーション名を選択しなかった場合、このレポートは個々の職責からアクセスできるすべてのレポートおよびレポート・セットを文書化します。

職責名

使用可能なレポートおよびレポート・セットについての情報が必要な職責の名前を選択します。「職責名」に値を入力する前に、「アプリケーション短縮名」に値を入力してください。

レポート見出し

レポート見出しは、指定されたレポート・パラメータのリストを示し、レポートの内容に関する一般情報を提供します。

関連項目

- 「Oracle Applications セキュリティの概要」(1-1)
- 「「職責」ウィンドウ」(1-6)
- 「「要求グループ」ウィンドウ」(3-53)
- 「要求セキュリティ・グループの定義」(1-4)

機能セキュリティ

機能セキュリティの概要

機能セキュリティとは、アプリケーションの機能へのユーザー・アクセスを制御する機構です。

Oracle Applications の GUI ベースのアーキテクチャでは、関連するいくつかの業務機能が 1 つのフォームに集約されます。すべてのユーザーが 1 つのフォームの中の業務機能すべてにアクセスするわけではないため、Oracle Applications には、アプリケーション・ロジックの個々の部分を機能として識別する機能が用意されています。アプリケーションにより実行される処理の一部を機能として識別することにより、各機能に対してセキュリティを設定できます(つまり、職責ごとにその機能を使用するかどうかを指定できます)。

アプリケーションの開発者は各フォームを作成するときに各機能を登録します。システム管理者は、職責ごとにその機能を使用するかどうかを設定することによって、機能セキュリティを管理します。

用語

機能

機能とは、アプリケーションにより実行される個々の処理を指します。1 つの機能は一意の名前で登録され、この名前を指定することにより、この機能を職責に割り当てたり、除外することができます。

機能には、フォーム機能と非フォーム機能の 2 種類があります。明確にするために、フォーム機能を「フォーム」、非フォーム機能を「サブ機能」と呼びます。ただし両方とも、データベース内の機能のインスタンスです。

フォーム（フォーム機能）

フォーム機能（フォーム）は、Oracle Forms フォームを起動します。フォーム機能には、「ナビゲータ」ウィンドウを使用してそこにナビゲートできるという特徴があります。

サブ機能（非フォーム機能）

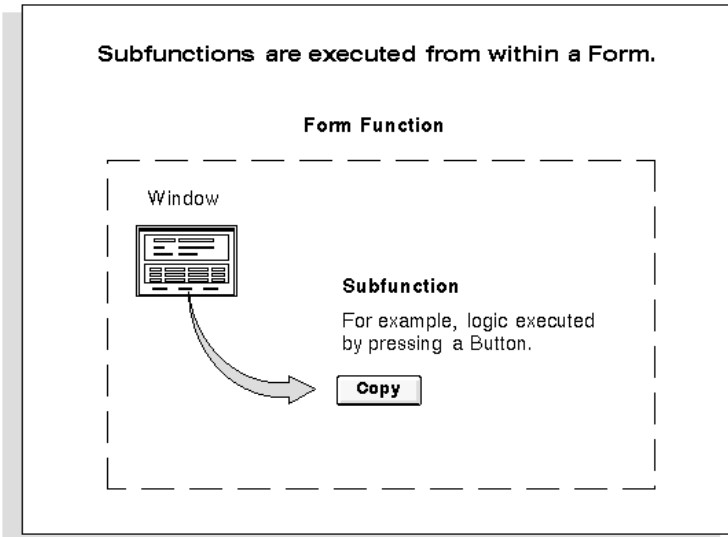
非フォーム機能（サブ機能）とは、セキュリティ設定が可能な機能として識別されるフォーム内の部分です。つまり、フォームの中で実行される機能です。

開発者はフォームを記述して、特定のサブ機能が使用可能かどうかをテストすることができます。その後、対象となる職責でサブ機能が使用可能かどうかに基づいて必要な処理を行うことができます。

サブ機能は多くの場合、フォーム上の各ボタンや他のグラフィカル要素と関連付けられます。たとえば、サブ機能が使用可能なときは、それに対応するボタンが使用可能になります。

ただし、サブ機能は、フォームの運用時にいつでもテストしたり実行したり、ユーザー・インタフェースに明示的な影響を与えるとは限りません。たとえば、サブ機能がグラフィカル要素と関連していないフォーム・プロシージャに対応する場合、そのサブ機能を利用できるかどうかはフォームのユーザーには不明です。

図 1



メニュー

メニューとは、機能および機能のメニューを階層的に配列したものです。各職責には、1つのメニューが割り当てられています。

メニュー項目

メニュー項目は、各機能または機能メニューを区別するメニューのコンポーネントです。機能と機能メニューの両方が、同じメニュー項目と対応する場合があります。たとえば1つのフォームと、そのサブ機能メニューの両方が、同じメニュー項目に属することがあります。

職責

職責は、Oracle Applications で作業するアプリケーション・ユーザーの現在の権限を定義します。アプリケーション・ユーザーがサインオンすると、次のような所定の権限が許可されている職責が選択されます。

- ユーザーがアクセス可能な機能。各機能は、その職責に割り当てられたメニューによって決定されます。
- ユーザーが実行可能なレポートなどのコンカレント・プログラム。
- フォーム、コンカレント・プログラムおよびレポートと連結しているアプリケーション・データベース・アカウント。

関連項目

- [「Oracle Applications セキュリティの概要」](#) (1-1)
- [「「フォーム機能」ウィンドウ」](#) (1-28)
- [「フォームおよびサブ機能」](#) (1-19)
- [「機能、メニューおよび「ナビゲータ」ウィンドウ」](#) (1-20)
- [「機能セキュリティの実装」](#) (1-22)
- [「機能セキュリティの動作方法」](#) (1-20)

フォームおよびサブ機能

フォームは、機能の特殊なクラスであり、次の2つの理由により、サブ機能とは異なります。

- フォームは「ナビゲータ」ウィンドウに表示されるため、そこにナビゲートできます。サブ機能は「ナビゲータ」ウィンドウには表示されないため、ナビゲートできません。
- フォームは単独で 사용할 ことができます。サブ機能は、1つのフォーム内のロジックによってのみコールできます。サブ機能を単独で 使用することはできません。

フォームは、そのすべてのプログラム・ロジックを含む全体が、常に機能として指定されます。フォームのプログラム・ロジックの一部に対してセキュリティを設定する必要がある場合、その部分をサブ機能として指定することもできます。

たとえば1つのフォームに3つのウィンドウが含まれる場合を考えます。フォーム全体がセキュリティを設定できる（職責ごとに使用するかどうかを指定できる）機能として識別されます。また、フォームの3つのウィンドウのそれぞれを機能（サブ機能）として指定して、セキュリティを個別に設定することもできます。したがって、機能セキュリティ・ルールの適用方法によっては、このフォームが割り当てられている職責であってもフォームの一部のウィンドウにはアクセスできないことがあります。

関連項目

- [「機能、メニューおよび「ナビゲータ」ウィンドウ」](#) (1-20)
- [「機能セキュリティの概要」](#) (1-17)
- [「機能セキュリティの動作方法」](#) (1-20)

機能、メニューおよび「ナビゲータ」ウィンドウ

フォーム機能、つまりフォームは、「ナビゲータ」ウィンドウを使用して選択されます。「ナビゲータ」ウィンドウ内の各フォーム名の配置は、対象となる職責に割り当てられたメニュー体系によって定義されます。

次の各タイプのメニュー項目は、「ナビゲータ」ウィンドウでは表示されません。

- サブ機能
- 項目がないメニュー
- プロンプトがないメニュー項目

「ナビゲータ」ウィンドウで表示される項目がないメニューについては、そのメニュー自体が表示されません。

関連項目

- [「機能セキュリティの概要」](#) (1-17)

機能セキュリティの動作方法

開発者による機能の登録

- 開発者は、一意の機能名を調べて対象となる職責でその機能が使用可能か確認の上で、必要な措置を実行するために、Oracle Forms コードの一部を必要とすることがあります。

- 開発者は機能を登録します。開発者はまた、値を機能に渡すパラメータを登録することもできます。たとえば、フォームでは、機能パラメータがフォームに渡された場合のみ、データ入力サポートされます。

警告：通常は、機能に渡されるパラメータのうち、Oracle Applications 製品の一部として事前定義されたパラメータは、システム管理者側では変更しないでください。システム管理者側で機能パラメータを修正できる特殊な場合については、関連するテクニカル・リファレンス・マニュアルまたは製品更新ノートに記載されています。

- 通常、開発者はアプリケーションで使用可能なすべての機能（つまりすべてのフォームと、そのセキュリティ管理可能なサブ機能）を含むメニューを定義します。一部のアプリケーションでは、開発者は特定のフォームやサブ機能を省略することにより、アプリケーションの機能を制限する追加メニューを定義できます。
- 通常、開発者が機能のメニューを定義する場合、各フォームのサブ機能をサブ機能メニュー上にまとめて、そのサブ機能メニューをフォームに関連させます。

システム管理者による機能の除外

- 各 Oracle Applications 製品には、事前定義された 1 つ以上のメニュー階層が提供されています。システム管理者は、事前定義メニュー階層を職責に割り当てることができます。職責を調整するには、システム管理者は、除外ルールを使用して、機能または機能のメニューをその職責から除外します。
- システム管理者が、事前定義メニューに除外ルールを適用しても、必要なメニューを作成できない場合は、新規のメニュー階層を定義することができます。この場合、システム管理者は、フォームとそのサブ機能の関連メニューを使用してメニュー階層を構築することをお勧めします。つまり、システム管理者は、開発者が定義したフォームとメニューとの関連は、残しておく必要があります。

現行の職責に応じて使用可能な機能

- ユーザーが、最初に職責を選択したり、変更する際は、職責のメニュー体系から取得される機能一覧が、メモリーにキャッシュされます。
- システム管理者が現行職責から除外した機能には、利用不可のマークがつけられます。
- 機能階層内のフォーム機能（すなわち、メニュー階層）は、「ナビゲータ」ウィンドウに表示されます。使用可能なサブ機能には、アプリケーションのフォームを使用して作業すると、アクセスできます。

除外した機能の表示

一部のサブ機能は、グラフィカル要素（例：ボタン）に関連付けられています。それらを除外すると次のような結果となります。

- ボタンがグレー表示になります。
- ボタンがなくなります。

それ以外のサブ機能はグラフィカル要素には関連付けられていない場合があります。この場合には、それらを除外されたことをエンド・ユーザーは認識できません。

関連項目

- [「Oracle Applications セキュリティの概要」](#) (1-1)
- [「フォーム機能」ウィンドウ](#) (1-28)
- [「フォームおよびサブ機能」](#) (1-19)
- [「機能セキュリティの概要」](#) (1-17)

機能セキュリティの実装

Oracle Applications の各製品について、そのアプリケーションの中のすべての機能を含んだメニューを伴う「フル・アクセス」職責が、1 つずつあらかじめ定義されています。さらに、アプリケーションによっては、小さい機能セット（つまりフォームおよびサブ機能の一部）を含む追加の職責があらかじめ定義されている場合もあります。

システム管理者は、特定の機能または機能メニューを除外するルールを定義することによって、特定の職責に対応する機能を制限できます。実際には、各職責に新規メニュー階層を構築するよりは、除外ルールを利用して職責のカスタマイズを行うことをお勧めします。

たとえば、ある職責をカスタマイズすることにより、その職責に含まれるフォームの機能を制限する場合、まず、そのフォームに関連付けられているサブ機能をグループ化している事前定義済みのメニューを調べます。次に、除外ルールを使用して、そのフォームのいくつかのサブ機能を職責から除外することにより、フォームの機能を制限できます。

除外ルールを適用する方法では必要な職責を作成できない場合は、事前定義されているフォーム（つまりフォーム機能）および各フォームに関連するサブ機能のメニューを使用して、その職責用のカスタム・メニューを作成できます。ただし、開発者が定義したサブ機能のメニューからフォームを切り離すことはお勧めできません。

事前定義済みメニューを使用した機能セキュリティ

「職責」フォームを使用して次のことを行います。

- メニューおよび機能を除外することによって、事前定義済みの職責の機能を制限します。
- 新しい職責を定義して、事前定義済みのメニューをその職責に割り当てます。メニューおよび機能を除外することにより、新しい職責の機能をカスタマイズします。

- いくつかの異なる職責に同じメニュー階層を割り当て、それぞれ異なる機能およびメニューを除外します。それにより、アプリケーションの機能を容易にカスタマイズすることができます。

新しいメニューを使用した機能セキュリティ

「メニュー」フォームを使用して、新しい職責で使用する機能のメニューを定義します。

- フォームおよび関連のサブ機能のメニューを使用して、新しいメニューを定義します。
- 「職責」フォームを使用して、メニュー体系を新しい職責に割り当てます。
- その職責について、特定のサブ機能を除外することにより、フォームの機能を調整します。
 - ある職責で使用しているフォームで実行するサブ機能の中から1つを除外しただけでも、新しい職責では異なる機能が使用されることになります。
 - フォームに関連付けられている事前定義済みのサブ機能のメニューに除外ルールを適用することによって、フォームの機能を容易にカスタマイズできます。

職責からの機能の除外

システム管理者は、職責に割り当てられているメニュー体系から機能またはメニューを除外できます。

- あるメニューを除外すると、そのすべてのメニュー項目、つまり、そのメニューで選択できるすべての機能および機能メニューが除外されます。
- 職責から機能1つを除外すると、その職責のメニュー体系全体からその機能のすべてのオカレンスが除外されます。

関連項目

- [「Oracle Applications セキュリティの概要」](#) (1-1)
- [「Oracle HRMS、Oracle Sales and Marketing 用特殊機能」](#) (1-26)
- [「「フォーム機能」ウィンドウ」](#) (1-28)
- [「機能セキュリティの概要」](#) (1-17)
- [「機能セキュリティの要約」](#) (1-27)
- [「新規メニュー体系の定義」](#) (1-24)

新規メニュー体系の定義

新しいメニュー体系を定義する方法

- 機能の論理的な階層リストを作成します。これで、複数の異なる職責のメニュー体系をカスタマイズするときに機能の除外が容易になります。
- ユーザーを各自のアプリケーション・フォームに導く論理的な階層メニューを作成します。

カスタム・メニュー体系を定義するための作業

- 個々の役職の職責に必要なアプリケーション機能を決定します。
- 新しいメニューを定義するときに、その項目として使用する事前定義済みメニュー、フォームおよびフォームのサブ機能を指定します。事前定義済みのメニューについて調べるには、「要求の発行」ウィンドウを使用してメニュー・レポートを印刷してください。

提案：メニュー項目に事前定義済みのメニューを使用すれば、作業が容易になります。メニュー体系を職責に割り当てた後で、機能を個別に除外できます。

- メニュー体系を計画して、メニューを設計します。
- まず最下位レベルのメニューを定義します。メニューを他のメニューの項目として選択するには、そのメニューを定義しておくことが必要です。
- メニューおよび機能を上位レベルのメニューに割り当てます。
- メニューおよび機能を最上位レベルのメニュー（ルート・メニュー）に割り当てます。
- メニュー・レポートを印刷することによりメニュー体系を文書化します。

警告：ブランクの「メニュー」フォーム（ブランク画面）から始めるようにしてください。メニューはコピーできません。メニューを別の名前で保存しても、元のメニューは上書きされてしまいます（「別名保存」の機能はありません）。

メニューの定義についての注意

メニューはスクラッチから作成します。

- メニューはコピーできません。メニュー定義を別の名前で保存することはできません（つまり「別名保存」の機能はありません）。
- 「メニュー」フォームにメニュー名が表示されているときには、そのメニューの名前を上書きする前に、「問合せ」モードになっていることを確認します。

キーボードで迅速かつ簡単に使用できるメニューの定義

- メニュー・プロンプトは、最初の文字がそれぞれ異なるように設計して、最初の文字を入力するだけで自動的に該当のフォームまたはメニューが選択されるようにします。
- メニュー・プロンプトは、最も使用頻度の高い機能が最初（小さい番号）にくるように配置します。
- メニュー定義から別のメニュー定義に項目をコピーすることはできません。

メニュー名または項目を変更するときの注意

- メニューの名前を変更しても、メニュー項目は影響を受けません。メニューの定義に新しい名前が与えられます。
- そのメニューを他のメニューが古い名前でコールした場合は、自動的に新しい（変更した）名前でそのメニューがコールされます。
- メニューを定義するか、「ルート」メニューを選択して職責を割り当てるときには、古いメニュー名は値リストには表示されません。
- 事前定義済みメニューを変更すると、そのメニューをコールする他のすべてのメニューにそのメニューの変更事項が表示されます。
- たとえば、あるフォーム機能をコールする別のプロンプトを追加することにより GL_TOP を変更した場合は、GL_TOP をコールするすべてのメニューで、GL_TOP の表示時にその追加のプロンプトが表示されます。

アップグレード時のカスタム・メニューの保存

カスタム・メニューに一意の名称を使用して、Oracle Applications のアップグレード時にカスタム・メニューを保存します。たとえば、カスタム・アプリケーションのアプリケーション短縮名で始まるメニュー名を指定できます。カスタム総勘定元帳という名称でカスタム・アプリケーションを定義します。アプリケーション短縮名は XXCGL です。この場合、カスタム・メニュー名は、XXCGL_MY_MENU のように、XXCGL で始まる名前にします。

Oracle Applications 標準メニューはアップグレード・バージョンによって上書きされることがあります。したがって、カスタム・メニューをサブメニューとして古い Oracle Applications メニューの 1 つに添付している場合には、アップグレード後にその添付メニューをもう一度作成します。もう 1 つの方法は標準 Oracle Applications メニューをサブメニューとしてカスタム・メニューに添付することです。こうしておくで、カスタム・メニューから標準メニューへのリンクはアップグレード時も保存されます。

関連項目

- 「[Oracle Applications セキュリティの概要](#)」 (1-1)
- 「[Oracle HRMS、Oracle Sales and Marketing 用特殊機能](#)」 (1-26)

- 「フォーム機能」ウィンドウ (1-28)
- 「機能セキュリティ・レポート」 (1-34)
- 「機能セキュリティの概要」 (1-17)
- 「機能セキュリティの実装」 (1-22)
- 「機能セキュリティの要約」 (1-27)

Oracle HRMS、Oracle Sales and Marketing 用特殊機能

ほとんどの Oracle Applications 製品では、オープンしているフォームをクローズしないで、「ナビゲータ」ウィンドウから複数のフォームをオープンできます。ただし、新しい職責のカスタム・メニューから Oracle Sales and Marketing フォームまたは Oracle HRMS タスク・フローにアクセスする場合は、この職責を定義するときに、その職責の最上位レベル・メニューの項目として「複数フォーム、複数セッション使用不可」機能を組み込む必要があります。

Oracle Sales and Marketing フォームは、フォームの機能名に含まれている OSM という接頭辞により識別できます。

Oracle HRMS では、ウィンドウをリンクするための方法としてタスク・フローが使用されているため、ウィンドウから次のウィンドウへと順に情報を伝達することによって、タスクが完了します。タスク・フローの一部となる Oracle HRMS フォームは、フォームの機能名の中の PER または PAY という接頭辞によって識別できます。Oracle HRMS タスク・フローを管理する方法や、あるフォームがタスク・フローの一部であるかどうかを判別する方法の詳細は、『Oracle HRMS ユーザーズ・ガイド』の「タスク・フローでのウィンドウのリンク」を参照してください。

注意：Oracle Sales and Marketing フォームも Oracle HRMS フォームも含まないメニューには、「複数フォーム、複数セッション使用不可」機能を組み込まないでください。

「複数フォーム、複数セッション使用不可」機能をメニューに組み込む手順は、次のとおりです。

- 最上位レベル・メニュー（つまり新しい職責が参照するメニュー）に「機能」メニュー項目を追加します。
- 次の「ユーザー機能名」および「機能名」を持つ機能を選択します。
 - 複数フォーム、複数セッション使用不可
 - FND_FNDSCGN_DISABLE_MULTIFORM

- 変更を保存します。

関連項目

- [「機能セキュリティの概要」](#) (1-17)
- [「機能セキュリティの実装」](#) (1-22)

機能セキュリティの要約

機能

- 機能とは Oracle Applications の中の一連のコードで、職責別に管理されるリスト内にその機能名がある場合に限り実行されます。
- システム管理者は、特定の職責から特定の機能を除外できます。
- 機能には、フォーム機能つまりフォーム、および非フォーム機能つまりサブ機能の 2 つのタイプがあります。サブ機能は、フォームの機能全体の一部であり、個別にセキュリティを設定することができます。

フォーム機能

- フォームを起動する機能。
- フォーム機能は「ナビゲータ」ウィンドウに表示されるので、そこからフォーム機能にナビゲートできます。

サブ機能

- フォーム内から実行される機能。サブ機能は、フォーム機能の中のロジックによってのみコールできます。
- サブ機能は「ナビゲータ」ウィンドウには表示されません。したがってサブ機能へはナビゲートできません。

メニュー

- メニューには、機能または別のメニュー、あるいはその両方を指すメニュー項目が含まれています。
- メニューは「ナビゲータ」ウィンドウに表示されます。
- システム管理者は、特定の職責から特定のメニューを除外できます。

関連項目

- [「Oracle Applications セキュリティの概要」](#) (1-1)
- [「Oracle HRMS、Oracle Sales and Marketing 用特殊機能」](#) (1-26)

- 「「フォーム機能」ウィンドウ」(1-28)
- 「機能セキュリティ・レポート」(1-34)
- 「機能セキュリティの概要」(1-17)
- 「機能セキュリティの実装」(1-22)
- 「新規メニュー体系の定義」(1-24)

「フォーム機能」ウィンドウ

新機能を定義します。機能とは、アプリケーションにより実行される個々の処理を指します。1つの機能は一意の名前で登録され、この名前を指定することにより、この機能を職責に割り当てたり、除外することができます。

機能には、フォーム機能と非フォーム機能の2種類があります。

明確にするために、フォーム機能は「フォーム」、非フォーム機能は「サブ機能」と呼びます。ただし両方とも、データベース内の機能のインスタンスです。

フォーム機能ブロック

機能

ユーザーに対しては、この一意の機能名は表示されません。ただし、この名前は、プログラムから機能呼び出すときに使用します。機能の命名規則に従ってください。

摘要

ユーザー機能名

機能を説明する一意の名前を入力します。この名前は、メニューに機能を割り当てるときに表示されます。この名前は、「ナビゲータ」ウィンドウの「トップ・テン・リスト」に表示されます。

タイプ

タイプとは、機能の使用方法についての自由な形式の記述です。機能のタイプは、開発者が機能の可用性をテストするときに渡されます。開発者は、機能のタイプに基づいて処理を実行するコードを作成できます。

規則によって、Oracle Applications フォーム機能は FORM タイプで登録されます。共通フォーム動作を決める少数の特別機能は UTIL タイプで登録されます。

フォーム機能は、FORM タイプで登録しなくても、有効なフォーム名 / アプリケーションを指定すると Oracle Applications はそれをフォームとして取り扱います。

フォーム

フォーム / アプリケーション

フォーム機能を定義するときには、そのフォームの名前とアプリケーションを選択します。

パラメータ

機能に渡すための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空白を使用します。

フォーム機能については、パラメータ `QUERY_ONLY=YES` を指定すると、フォームが問合せのみモードでオープンします。Oracle Application Object Library は、フォームを問合せのみモードでオープンする前にフォーム・パラメータのリストからこのパラメータを除去します。

別のフォーム名を指定することによって、該当するヘルプ・ファイル内のフォームのヘルプも検索できます。使用する構文は次のとおりです。

```
HELP_TARGET = "alternative_form_name"
```

フォーム名が指定のフォーム名で上書きされます。参照：「[コンテキスト依存ヘルプ](#)」(6-6)。

Oracle Applications フォームの中には、特定のフォームのパラメータを受け入れるようにコード化されているものがあります。たとえば、「要求実行」フォームは「要求実行」ウィンドウのタイトルを変更するときに使用する `TITLE` パラメータを受け入れます。次の構文を使用します。

```
TITLE="appl_short_name:message_name"
```

ここで `appl_shortname:message_name` は、「メッセージ・ディクショナリ」メッセージの名前です。参照：「[コードを使用した「要求の発行」ウィンドウのカスタマイズ](#)」(3-17)。

`appl_shortname:message_name` は、アプリケーション短縮名と Message Dictionary メッセージのメッセージ名です。参照：『Oracle Applications システム管理者ガイド』の「コードを使用した「要求の発行」ウィンドウのカスタマイズ」。

警告：通常は、機能に渡されるパラメータのうち、Oracle Applications 製品の一部として事前定義されたパラメータは、システム管理者側では変更しないでください。システム管理者側で機能パラメータを修正できる特殊な場合については、関連するテクニカル・リファレンス・マニュアルまたは製品更新ノートに記載されています。

Web HTML および Web ホスト

Web HTML および Web ホストの各フィールドは、機能に Oracle Self-Service Web Applications からアクセスする場合のみ必要です。Web-deployed Applications 機能の場合は、どのフィールドにも入力する必要はありません。

HTML コール

機能の URL の最後のセクションは、「HTML コール」です。「HTML コール」は、機能をアクティブ化するために使用します。その機能は、静的な Web ページでもプロシージャでもかまいません。

セキュリティ

「セキュリティ」は、Oracle Workflow から機能にアクセスするときのみに必要です。「セキュリティ」をチェックすると、ワークフロー電子メール通知の受信者は、電子メールを使用して応答できるようになります。

暗号化パラメータ

「暗号化パラメータ」をチェックすると、機能にセキュリティの層が追加され、ユーザーがブラウザのウィンドウで URL を変更しても、その機能へアクセスできなくなります。機能に暗号化機能を適用することを定義する場合は、「暗号化パラメータ」を定義する必要があります。

ホスト名

機能に必要な URL (Universal Resource Locator) またはアドレスは、「ホスト名」、「エージェント名」、「HTML コール」の 3 つのセクションからなっています。「ホスト名」は、WebServer を実行しているマシンの IP アドレスまたは別名です。

エージェント名

機能の URL の第 2 セクションは、Oracle Web エージェントです。Oracle Web エージェントは、機能を実行するときに使用するデータベースを決定します。デフォルトは、最後に使用したエージェントです。

アイコン

この機能で使用されるアイコン名を入力します。

関連項目

- [「Oracle Applications セキュリティの概要」](#) (1-1)
- [「機能セキュリティの概要」](#) (1-17)
- [「機能セキュリティの実装」](#) (1-22)

「メニュー」ウィンドウ

新規メニューを定義するか、既存のメニューを変更します。

メニューとは、機能および機能のメニューを階層的に配列したものです。各職責には、1 つのメニューが割り当てられています。

Oracle Applications の各製品について、そのアプリケーションの中のすべての機能のメニューを伴う「フル・アクセス」職責が、1 つずつあらかじめ定義されています。システム管理者は、特定の機能または機能メニューを除外するルールを定義することによって、特定の職責に対応する機能を制限できます。実際に、各職責に新規メニュー階層を構築するよりは、除外ルールを利用して職責のカスタマイズを行うことをお勧めします。

除外ルールを適用する方法では必要な職責を作成できない場合は、事前定義されているフォーム（つまりフォーム機能）および各フォームに関連するサブ機能のメニューを使用して、その職責用のカスタム・メニューを作成できます。ただし、開発者が定義したサブ機能のメニューからフォームを切り離すことはお勧めできません。

参照：

[「機能セキュリティの概要」](#) (1-17)

[「機能セキュリティの実装」](#) (1-22)

前提条件

- 「アプリケーション」ウィンドウでアプリケーションを Oracle Application Object Library に登録します。
- メニューからコールしようとするメニューのすべてを定義します。まず最下位レベルのサブメニューを定義します。サブメニューは、あらかじめ定義しておかないと、別のメニューからコールできません。

提案：メニューからサブメニューをコールすると、関連性のあるウィンドウをメニュー上で1つの見出しにまとめることができます。メニューをその他のメニュー上で再使用することが可能です。

メニュー・ブロック

メニュー項目は、そのメニューから使用可能なオプションの内容を記述します。

メニュー

メニューの目的を説明する名前を選択します。ユーザーに対しては、このメニュー名は表示されません。

ツリーの表示 ...

メニューを定義すると、「ツリーの表示 ...」ボタンを使用してメニューの階層体系を表示させることができます。参照：[「メニュー・ビューア」](#) (1-33)

ユーザー・メニュー名

職責がメニューをコールするとき、または1つのメニューが別のメニューをコールするときに、ユーザー・メニュー名を使用します。

メニュー項目ブロック

連番

メニュー項目が表示される位置を、メニューにある他の項目との相対的な関係で示す連番で入力します。このフィールドのデフォルト値は、次の整数の連番です。

連番の小さい項目は、連番の大きい項目よりも前に表示されます。

注意：連番には整数のみを使用してください。

注意：連番を変更したり、メニュー項目を頻繁に挿入および削除するときには、デフォルト値を必ず確認してください。連番が重複していたり、範囲外の番号があることがあります。

提案：メニュー項目の連番は、すでに使用されている別の連番と置き換えることはできません。一連のメニュー項目にメニュー項目を追加するときには、既存の連番を使用しないように注意して、使用する連番範囲とは重複しないことが確実な連番範囲でメニュー項目に番号を再度設定してください。

この作業を保存したら、元に戻って各項目が最終的に意図した連番を持つように番号を再度設定します。

ナビゲータ・プロンプト

メニュー項目を表示するため、ユーザーにとって使いやすくひと目でわかるプロンプトを入力します。このメニュー・プロンプトは「ナビゲータ」ウィンドウの階層リストに表示されます。

提案：パワー・ユーザーがメニュー項目を選択するときに最初の文字を入力すればいいように、先頭に一意の文字を使用したメニュー・プロンプトを入力します。

サブメニュー

別のメニューをコールして、ユーザーがそのメニューからメニュー項目を選択できるようにします。

機能

メニューに組み込む機能をコールします。フォーム機能（フォーム）は、「ナビゲータ」ウィンドウに表示されるため、そこからそのフォームにアクセスできます。それ以外の非フォーム機能（サブ機能）を使用すると、このメニューからフォーム機能全体の特定の部分にアクセスできます。

摘要

摘要は、メニュー項目をハイライト表示すると「ナビゲータ」ウィンドウの上部にあるフィールドに表示されます。

関連項目

- [「Oracle Applications セキュリティの概要」](#) (1-1)
- [「「職責」ウィンドウ」](#) (1-6)
- [「機能セキュリティ・レポート」](#) (1-34)
- [「機能セキュリティの概要」](#) (1-17)
- [「機能セキュリティの実装」](#) (1-22)
- [「職責の定義」](#) (1-2)
- [「新規メニュー体系の定義」](#) (1-24)

メニュー・ビューア

メニュー・ビューアは、サブ・メニューおよびメニューの機能を階層的に表示し、メニューおよび機能のプロパティをリストする読取り専用ウィンドウです。

「ツリーを表示」ボタンをクリックすることによって、メニュー・フォームからビューアを起動できます。ビューアは、メニュー・フォームで指定されたメニューに対して表示されます。

注意：新規メニューを作成または編集している時、変更は、メニュー・ビューアで表示する前に、データベースにコミットされる必要があります。

機能性

機能セキュリティ・レポート

機能セキュリティ・レポートを使用すると、11i メニューの体系を文書化できます。Oracle Applications ソフトウェアをアップグレードする前に、カスタマイズ済みのメニュー体系をハードコピーとして文書化するときにもこれらのレポートを使用できます。

機能セキュリティ・レポートは、機能セキュリティ機能レポート、機能セキュリティ・メニュー・レポート、機能セキュリティ・ナビゲータ・レポートで構成されます。

これらのレポートは機能セキュリティ・メニュー・レポートの要求セットから使用できます。それぞれのレポートについて、機能セキュリティを表示する対象となる職責を指定します。

機能セキュリティ機能レポート

レポートの発行時に職責を指定します。レポート結果には指定された職責でアクセス可能な機能がリストされます。

このレポートには機能セキュリティ・ルールで除外された機能は取り込まれません。

機能セキュリティ・メニュー・レポート

レポートの発行時に職責を指定します。レポート結果には、すべてのサブメニューおよび機能とともに、職責の完全メニューがリストされます。

このレポートは、除外されたすべてのメニュー項目をその除外ルールとともに表示します。

機能セキュリティ・ナビゲータ・レポート

レポートの発行時に職責を指定します。レポート結果には、指定された職責についてナビゲータに表示されるメニューがそのままリストされます。

このレポートには、機能セキュリティ・ルールで除外された機能や、ナビゲータに表示されない非フォーム機能は取り込まれません。

メニュー・レポート

このレポートは、キャラクタ・モード・メニューの体系を文書化します。このレポートを使用して、新しいメニューを定義、または既存のメニューを編集します。また、Oracle Applications ソフトウェアをアップグレードする前に、カスタマイズ済みの体系をハードコピーとして文書化するときにもこのレポートを使用できます。

注意：メニュー・レポートは、キャラクタ・モードで実行中の Oracle Applications で作成および使用されているメニューのみを詳細表示します。機能セキュリティ・メニュー・レポートを使用して、Oracle Applications 10SC で作成済のメニューの詳細を表示します。

メニュー・レポートには次の内容が出力されます。

- 指定したメニューのメニュー項目の階層リスト
- 各メニュー項目で起動される処理
- メニューが含まれているアプリケーションの名前
- メニューがコールする処理の名前

レポート・パラメータ

メイン・メニューのみ

レポートするメニューがメイン・メニューかどうかを「Yes」または「No」で指定します。「Yes」を選択すると、メイン・メニュー名のみが選択され、「メニュー名」パラメータのウィンドウに表示されます。

アプリケーション名

レポートするメニューが属しているアプリケーションの名前を選択します。

メニュー名

レポートするメニューの名前を選択します。

レポート見出し

レポート見出しとして、メニューのアプリケーションおよび名前が出力されます。

列見出し

プロンプト

このメニュー項目用にメニューに表示されるプロンプト。

摘要

このメニュー項目の説明。

処理タイプ

このメニュー項目を選択したときに起動される処理のタイプ。有効な値は次のとおりです。

メニュー サブメニューへのナビゲート。

機能 機能へのナビゲート。

アプリケーション

メニューがコールするメニューまたは機能に関連付けられているアプリケーションの名前。

処理

メニューまたは機能の名前。

関連項目

- [「Oracle Applications セキュリティの概要」](#) (1-1)

ユーザー監査およびデータ監査の概要

Oracle Applications には、2 つの監査タイプがあります。ユーザーの監査とデータベース行の変更の監査です。

ユーザー・アクティビティの監査

ユーザーの監査は次のコンポーネントによってサポートされます。

- 「サインオン: 監査レベル」 プロファイル・オプション設定
- 監査レポート

参照:

[「サインオン監査コンカレント要求レポート」 \(2-9\)](#)

[「サインオン監査フォーム・レポート」 \(2-11\)](#)

[「サインオン監査職責レポート」 \(2-13\)](#)

[「サインオン監査不成功ログイン・レポート」 \(2-15\)](#)

[「サインオン監査ユーザー・レポート」 \(2-17\)](#)

「サインオン監査」により、アプリケーションにサインオンしたユーザーを確認して、それらのユーザーの活動を追跡できます。

「サインオン監査」では、選択した監査レベルに基づいて、ユーザー名、ユーザーがシステムにアクセスした日付と時刻およびユーザーが使用する職責、フォーム、端末が記録されます。

データベース行の変更の監査

データベース行の変更の監査は次のコンポーネントによってサポートされます。

- 「ヘルプ」メニューの「このレコードについて」

- 「自動監査:有効化」プロファイル・オプション設定
- 監査フォーム - 下記を参照。

参照:

[「監査証跡データについてのレポート」](#) (2-20)

[「導入の監査」ウィンドウ](#) (2-31)

[「監査グループ」ウィンドウ](#) (2-32)

[「監査表」ウィンドウ](#) (2-35)

関連項目

- [「ユーザーのモニター」ウィンドウ](#) (2-8)
- [「監査グループ」ウィンドウ](#) (2-32)
- [「監査表」ウィンドウ](#) (2-35)
- [「導入の監査」ウィンドウ](#) (2-31)
- [「サインオン監査の設定」](#) (2-3)
- [「サインオン監査レポート」](#) (2-7)
- [「ユーザー・アクティビティの監査」](#) (2-2)
- [「リリース 11i の監査証跡の設定」](#) (2-20)
- [「リリース 11 の監査証跡表、トリガーおよびビュー」](#) (2-22)
- [「監査証跡の使用禁止と、監査データのアーカイブ」](#) (2-30)
- [「監査証跡データについてのレポート」](#) (2-20)
- [「監査情報についてのレポート」](#) (2-29)

ユーザー・アクティビティの監査

Oracle Applications には「サインオン監査」機能があり、この機能を使用すると次のことができます。

- ユーザーが行っていること、およびいつそれを行ったかを追跡する
- どのユーザーを監査し、どのようなタイプの情報を監査するかを選択する
- ユーザーが行っていることをオンラインで迅速に表示する
- アプリケーションのセキュリティをチェックする

「サインオン監査」を使用すると、ユーザーが Oracle Applications へアクセスしたユーザー名、端末および日時を記録できます。「サインオン監査」では、ユーザーが実行するコンカレント・プロセスのみでなく、ユーザーが使用する職責およびフォームも追跡できます。

主な機能

選択的監査

「サインオン監査」では、だれを監査し、どのようなタイプのユーザー情報を追跡するかを選択できます。自分の組織のニーズに合わせて、どの監査情報が必要であるかを選択して決定できます。

アプリケーション・ユーザーのモニター

「ユーザーのモニター」フォームでは、どのユーザーが Oracle Applications を使用しており、それらのユーザーが何をしているかについて、オンラインでリアルタイム情報が得られます。

どのユーザーがサインオンしているか（アプリケーション・ユーザー名およびオペレーティング・システム・ログイン名）、それらのユーザーがどの職責、フォームおよび端末を使用しているか、フォーム上でどれくらいの時間作業しているかおよびどの ORACLE プロセスを使用しているかを知ることができます。

サインオン監査レポート

サインオン監査レポートでは、ユーザーがアプリケーション内で何を行ったかについて、詳細な履歴情報が得られます。

検索する情報を絞り込むため、検索基準を指定できます。

「サインオン監査」情報をソートして、読みやすいレポートを作成することもできます。

関連項目

- 「「ユーザーのモニター」ウィンドウ」(2-8)
- 「サインオン監査の設定」(2-3)
- 「サインオン監査レポート」(2-7)
- 「ユーザー監査およびデータ監査の概要」(2-1)

サインオン監査の設定

「サインオン: 監査レベル」ユーザー・プロファイル・オプションを使用して、だれを「サインオン監査」で追跡するか、およびどのレベルで監査するかを制御できます。

ユーザーがしていることをオンラインで表示するには、「ユーザーのモニター」フォームを使用します。

詳細な監査情報が得られるサインオン監査レポートを発行するには、「要求の発行」フォームを使用します。

サインオン監査の使用可能化

「サインオン監査」を使用可能にするには、「システム・プロファイル値」フォームを使用します。ユーザー・プロファイル・レベルをユーザー、職責、アプリケーションまたはサイトのプロファイル・レベルに設定することにより、監査の有効範囲とだれを監査するかを選択します。

ユーザーは、このプロファイル・オプションを参照することも更新することもできません。

監査レベルを設定するか変更した後、あるユーザーについての新しい監査レベルは、そのユーザーが次回にオペレーティング・システムから Oracle Applications へサインオンしたときに有効になります。

監査レベルの選択

「サインオン：監査レベル」プロファイル・オプションを使用すると、Oracle Applications へサインオンしたユーザーを監査するレベルを選択できます。機能が増える順に、「なし」、「ユーザー」、「職責」および「フォーム」の4つの監査レベルがあります。

「なし」はデフォルト値であり、Oracle Applications にサインオンしたどのユーザーも監査しないことを意味します。

「ユーザー」レベルの監査では、次のことが追跡されます。

- どのユーザーがシステムにサインオンしたか
- ユーザーがログオンおよびログオフした時刻
- 使用中の端末

「職責」レベルの監査では、「ユーザー」レベルの監査機能に加えて、次のことも追跡されます。

- ユーザーが選択した職責
- ユーザーが各職責を使用した時間

「フォーム」レベルの監査では、「職責」レベルの監査機能に加えて、次のことも追跡されます。

- ユーザーが選択したフォーム
- ユーザーが各フォームを使用した時間

システム・オーバーヘッドのファクタリング

組織の「サインオン監査」の実装を計画する場合、ユーザーが Oracle Applications へアクセスするときに、それらのユーザーを詳しくモニターし監査するために必要になるシステム・オーバーヘッドの増加を考慮する必要があります。監査するユーザーの数が多いほど、また、監査レベルが高いほど、追加システム・オーバーヘッドが発生する可能性が大きくなります。

例 - ユーザー、職責およびフォームを監査する

「サインオン監査」の実装例の1つは、ユーザーのサインオン、それらのユーザーが選択した職責およびそれらのユーザーがアクセスしたフォームをすべて監査することです。

このように「サインオン監査」を実装するには、「サインオン: 監査レベル」を次のように設定します。

- フォーム監査
- 「サイト」プロファイル・レベルを設定する

例 -1 人のユーザーを除外して特定の職責を監査する

「サインオン監査」を使用する別の例は、MJONES を除いて、「人事管理者」の職責を持つすべてのユーザーを監査する組織の例です。

この例では、ユーザーがアクセスしたフォームまたはユーザーが選択した職責を考慮する必要はありません。

このように「サインオン監査」を実装するには、「サインオン: 監査レベル」を次のように設定します。

- ユーザー監査
- 「人事管理者」の職責の職責プロファイル・レベルを設定する

また、「サインオン: 監査レベル」を次のように設定することもできます。

- なし
- アプリケーション・ユーザー MJONES のユーザー・プロファイル・レベルを設定する

「アプリケーション・モニター」の使用

だれが Oracle Applications を使用して何をしているかをモニターするには、「ユーザーのモニター」フォームを使用します。システム管理者は、いつでもユーザーをモニターできます。

アプリケーション・モニターを使用すると、どのユーザーがサインオンしているか、それらのユーザーがどの職責、フォームおよび端末を使用しているか、フォーム上でどれくらいの時間作業しているか、およびどの ORACLE プロセスを使用しているかを知ることができます。

注意：「サインオン監査」の監査対象となっているユーザーのみをモニターできます。アプリケーション・モニターには、それらのユーザーについて定義した監査レベルも反映されます。

「このレコードについて」ウィンドウ

ヘルプ・メニューから「このレコードについて ...」を選択することにより、「サインオン監査」データを表示できます。

「サインオン監査」は、監査対象のユーザーが挿入するか更新したレコードの「このレコードについて」情報を自動的に関連付けることができます。この追加情報は、「対象者：表示タイプ」プロファイル・オプションを「拡張」に設定した場合に「このレコードについて」ウィンドウに表示されます。

拡張情報は、Oracle Applications セッション番号、オペレーティング・システム・ログイン名および「サインオン監査」で追跡対象となっているユーザーが行の挿入または更新に使用した端末を示します。

システム管理者は、「システム・プロファイル値」フォームを使用して「対象者：表示タイプ」を設定して、任意のユーザー、職責、アプリケーションまたはサイトに「このレコードについて」の拡張情報を表示させることができます。

「対象者：表示タイプ」プロファイル・オプション

「対象者：表示タイプ」プロファイル・オプションを使用すると、「このレコードについて」ウィンドウの表示を次の2つから選択できます。

「通常」の場合は、次の情報が表示されます。

- その行を作成したユーザーの名前
- ユーザーがその行を作成した日付
- その行が入っている表の名前
- その行を最後に更新したユーザーの名前

「拡張」の場合は、「標準」の情報のほかに、次の情報が表示されます。

- ユーザーのオペレーティング・システム・ログオン
- ユーザーの端末 ID

ユーザーは、このプロファイル・オプションを参照することも変更することもできません。

このプロファイル・オプションは、4つのどのレベルでも参照および更新できます。

失敗ログインの通知

「サインオン監査」では、ユーザー・ログインを追跡できます。ユーザーが最後にサインオンした後に、だれかがそのユーザーのアプリケーション・ユーザー名でサインオンしようとして失敗した場合には、ユーザーに警告メッセージを送ることができます。この警告メッセージは、ユーザーのサインオンの後に表示されます。

システム管理者またはユーザーは、「個別プロファイル値」フォームを使用して「サインオン:通知」ユーザー・プロファイル・オプションを「Yes」に設定することにより、この機能をアクティブにすることができます。

この通知機能を使用するために「サインオン監査」でユーザーを監査する必要はありません。

関連項目

- [「ユーザーのモニター」ウィンドウ](#) (2-8)
- [「サインオン監査レポート」](#) (2-7)
- [「ユーザー・アクティビティの監査」](#) (2-2)
- [「ユーザー監査およびデータ監査の概要」](#) (2-1)

サインオン監査レポート

標準監査レポートを印刷するには、「要求の発行」フォームを使用します。

どのユーザーがサインオンしているか、それらのユーザーがどの職責へアクセスしているか、どのフォームを使用しているか、どのコンカレント要求を発行しているか、およびだれが別のユーザーのアカウントへログオンしようとしているかなどを詳細に示したレポートを生成できます。

Oracle Applications は、次のサインオン監査レポートを提供します。

[「サインオン監査コンカレント要求レポート」](#) (2-9) (だれが、どの要求を発行したかを示します。)

[「サインオン監査フォーム・レポート」](#) (2-11) (だれが、どのフォームへアクセスしたかを示します。)

[「サインオン監査職責レポート」](#) (2-13) (だれが、どの職責へアクセスしたかを示します。)

[「サインオン監査不成功ログイン・レポート」](#) (2-15) (だれが、別のユーザーとしてサインオンしようとして失敗したかを示します。)

「[サインオン監査ユーザー・レポート](#)」(2-17) (だれが、Oracle Applications へサインオンしたかを示します。)

必要に応じてレポートを短くするため、個々のレポートについて検索基準を指定することもできます。

関連項目

- 「[ユーザーのモニター](#)」ウィンドウ (2-8)
- 「[サインオン監査の設定](#)」(2-3)
- 「[サインオン監査レポート](#)」(2-7)
- 「[ユーザー・アクティビティの監査](#)」(2-2)
- 「[ユーザー監査およびデータ監査の概要](#)」(2-1)

「ユーザーのモニター」ウィンドウ

このウィンドウを使用して、アプリケーション・ユーザーが現在何を実行しているのかをモニターします。どのユーザーがサインオンしていて、現在どの職責、フォーム（ウィンドウ）、端末を使用しているかを知ることができます。

また、ログインしてから経過した時間、現在どの Oracle プロセスを使用しているかを知ることができます。

さらに、サイト内のすべてのユーザー、特定のアプリケーションまたは職責を使用しているすべてのユーザー、または個々のユーザーをモニターすることもできます。サインオン監査をアクティブ化したユーザーだけをモニターすることができます。参照：[「ユーザー監査およびデータ監査の概要」](#) (2-1)。

前提条件

- 「システム・プロファイル・オプションの更新」ウィンドウで、「サインオン: 監査レベル」プロファイル・オプションの値を選択します。

ユーザーのモニター・ブロック

職責

ユーザーの職責は、職責監査またはフォーム監査のレベルでサインオン監査を使用可能にしたときにのみ表示されます。

フォーム

ユーザーのフォームは、フォーム監査レベルでサインオン監査を使用可能にしたときにのみ表示されます。

ログイン

ユーザーのログイン名。

時間

ユーザーがこのアプリケーションにログオンしている時間の長さ。

ORACLE プロセス

ユーザーの ORACLE プロセス。

端末名

ユーザーが作業中の端末名。

関連項目

- 「サインオン監査の設定」(2-3)
- 「サインオン監査レポート」(2-7)
- 「ユーザー・アクティビティの監査」(2-2)
- 「ユーザー監査およびデータ監査の概要」(2-1)

サインオン監査コンカレント要求レポート

このレポートを使用して、だれがどのコンカレント要求をどの職責およびフォームから要求しているのかについての情報を表示します。

注意: サインオン監査コンカレント要求レポートは、監査対象にしているユーザーについてのみ生成できます。

レポート・パラメータ

ソート・キー

レポート内の情報を、オペレーティング・システム・ログイン名、要求開始日またはアプリケーション・ユーザー名によってソートします。

ログイン名

他の検索基準を満たす特定のログイン名を検索します。このパラメータをブランクのままにした場合、レポートには他の検索基準を満たすログイン名がすべて入ります。

ユーザー名

他の検索基準を満たす特定のアプリケーション・ユーザー名を検索します。このパラメータをブランクのままにした場合は、レポートには他の検索基準を満たすアプリケーション・ユーザー名がすべて入ります。

要求開始時刻：自：/ 要求開始時刻：至：

他の検索基準を満たし、特定の期間内にある開始時間を要求したことがあるコンカレント要求を検索します。このパラメータを使用して、期間の始まりと終わりを指定します。このパラメータをブランクのままにした場合は、レポートには他の検索基準を満たす任意の日付から、このパラメータの現在の日付までの範囲内のコンカレント要求が入ります。

レポート見出し

レポートの見出しには、パラメータ値として入力した検索基準が表示されます。

列見出し

ログイン名

コンカレント要求を発行したユーザーのオペレーティング・システム・ログイン名。

要求 ID

発行したコンカレント要求のコンカレント要求 ID。「コンカレント要求」フォームを使用すると、コンカレント要求 ID の完了情報が表示されます。

コンカレント・プログラム名

ユーザーが発行したコンカレント・プログラムの名前。「コンカレント・プログラム」フォームを使用すると、コンカレント・プログラムの詳細情報が表示されます。

ユーザー名

コンカレント要求を発行したユーザーの Oracle Applications ユーザー名。「ユーザー」フォームを使用すると、アプリケーション・ユーザーの詳細情報が表示されます。参照：[「ユーザー」ウィンドウ](#) (1-9)。

職責名

ユーザーがコンカレント要求の発行時に使用した職責の名前。職責が表示されるのは、職責またはフォームのいずれかの「サインオン監査」レベルでユーザーを監査した場合に限ります。「職責」フォームを使用すると、職責についての詳細情報が表示されます。参照：「[「職責」ウィンドウ](#)」(1-6)。

フォーム名

ユーザーがコンカレント要求の発行時に使用したフォームの名前。フォーム名が表示されるのは、フォームの「サインオン監査」レベルでユーザーを監査した場合に限ります。

要求開始時刻

コンカレント要求の実行が開始された日付と時刻。

関連項目

- 「[「ユーザーのモニター」ウィンドウ](#)」(2-8)
- 「[サインオン監査の設定](#)」(2-3)
- 「[サインオン監査レポート](#)」(2-7)
- 「[ユーザー・アクティビティの監査](#)」(2-2)
- 「[ユーザー監査およびデータ監査の概要](#)」(2-1)

サインオン監査フォーム・レポート

このレポートを使用すると、どのフォームにだれがいつナビゲートしているかが表示されます。

注意：サインオン監査フォーム・レポートは、監査対象にしているユーザーについてのみ生成できます。

レポート・パラメータ

ソート・キー

ユーザーがフォームを開始または終了した時刻、ユーザーがアクセスするフォームの名前、ユーザーのオペレーティング・システム・ログイン名、ユーザーがアクセスする職責、ユーザーが使用している端末またはアプリケーション・ユーザー名によってレポート内の情報をソートします。

ログイン名

他の検索基準を満たす特定のログイン名の情報を検索します。このパラメータをブランクのままにした場合は、レポートには他の検索基準を満たすログイン名がすべて入ります。

ユーザー名

他の検索基準を満たす特定のアプリケーション・ユーザー名の情報を検索します。このパラメータをブランクのままにした場合は、レポートには他の検索基準を満たすアプリケーション・ユーザー名がすべて入ります。

端末名

他の検索基準を満たす特定の端末の情報を検索します。このパラメータをブランクのままにした場合は、レポートには他の検索基準を満たす端末名がすべて入ります。

職責名

他の検索基準を満たす特定の職責の情報を検索します。このパラメータをブランクのままにした場合は、レポートには他の検索基準を満たす職責がすべて入ります。

フォーム名

他の検索基準を満たす特定のフォームの情報を検索します。このパラメータをブランクのままにした場合は、レポートには他の検索基準を満たすフォームがすべて入ります。

有効日の開始 / 有効日の終了

ユーザーが特定の期間内にアクセスし、かつ他の検索基準を満たすフォームの情報を検索します。このパラメータを使用して、期間の始まりと終わりを指定します。これらのパラメータをブランクのままにした場合は、レポートには他の検索基準を満たす任意の日付から、このパラメータの現在の日付までにアクセスされたフォームが入ります。

レポート見出し

レポートの見出しには、パラメータ値として入力した検索基準が表示されます。

列見出し

ユーザー名

フォームにアクセスしたユーザーの Oracle Applications ユーザー名。「ユーザー」フォームを使用すると、アプリケーション・ユーザーの詳細情報が表示されます。参照：「[「ユーザー」ウィンドウ](#)」(1-9)。

ログイン名

フォームにアクセスしたユーザーのオペレーティング・システム・ログイン名。

端末名

ユーザーがフォームにアクセスするときに使用した端末のオペレーティング・システム ID。

職責名

ユーザーがフォームにアクセスするときに使用した職責の名前。職責が表示されるのは、職責またはフォームのいずれかの「サインオン監査」レベルでユーザーを監査した場合に限ります。「職責」フォームを使用すると、職責についての詳細情報が表示されます。参照：
[「職責」ウィンドウ](#) (1-6)。

有効時刻の開始 / 有効時刻の終了

ユーザーがフォームにアクセスした日時およびフォームを終了した日時。有効時刻の開始および有効時刻の終了が表示されるのは、フォームの「サインオン監査」レベルでユーザーを監査した場合に限ります。

フォーム名

ユーザーがアクセスしたフォームの名前。フォーム名が表示されるのは、フォームの「サインオン監査」レベルでユーザーを監査した場合に限ります。

関連項目

- [「ユーザーのモニター」ウィンドウ](#) (2-8)
- [「サインオン監査の設定」](#) (2-3)
- [「サインオン監査レポート」](#) (2-7)
- [「ユーザー・アクティビティの監査」](#) (2-2)
- [「ユーザー監査およびデータ監査の概要」](#) (2-1)

サインオン監査職責レポート

このレポートを使用すると、だれがどの職責をいつ選択しているのかが表示されます。

注意：サインオン監査職責レポートは、監査対象にしているユーザーについてのみ生成できます。

レポート・パラメータ

ソート・キー

ユーザーが職責を開始したまたは終了した時刻、ユーザーのオペレーティング・システム・ログイン名、職責名、ユーザーが使用している端末またはアプリケーション・ユーザー名によってレポート内の情報をソートします。

ログイン名

他の検索基準を満たす特定のログイン名の情報を検索します。このパラメータをブランクのままにした場合は、レポートには他の検索基準を満たすログイン名がすべて入ります。

ユーザー名

他の検索基準を満たす特定のアプリケーション・ユーザー名の情報を検索します。このパラメータをブランクのままにした場合は、レポートには他の検索基準を満たすアプリケーション・ユーザー名がすべて入ります。

端末名

他の検索基準を満たす特定の端末の情報を検索します。このパラメータをブランクのままにした場合は、レポートには他の検索基準を満たす端末名がすべて入ります。

職責名

他の検索基準を満たす特定の職責の情報を検索します。このパラメータをブランクのままにした場合は、レポートには他の検索基準を満たす職責がすべて入ります。

有効日の開始 / 有効日の終了

ユーザーが特定の期間内にアクセスし、かつ他の検索基準を満たす職責の情報を検索します。このパラメータを使用して、期間の始まりと終わりを指定します。これらのパラメータをブランクのままにした場合は、レポートには他の検索基準を満たす任意の日付から、このパラメータの現在の日付までにアクセスされた職責が入ります。

レポート見出し

レポートの見出しには、パラメータ値として入力した検索基準が表示されます。

列見出し

ユーザー名

フォームを選択したユーザーの Oracle Applications ユーザー名。「ユーザー」フォームを使用すると、アプリケーション・ユーザーの詳細情報が表示されます。参照：「[ユーザーウィンドウ](#)」(1-9)。

ログイン名

職責を選択したユーザーのオペレーティング・システム・ログイン名。

端末名

ユーザーが職責を選択するときに使用した端末のオペレーティング・システム ID。

職責名

ユーザーが使用した職責の名前。職責が表示されるのは、職責またはフォームのいずれかの「サインオン監査」レベルでユーザーを監査した場合に限ります。「職責」フォームを使用すると、職責についての詳細情報が表示されます。参照：「[職責ウィンドウ](#)」(1-6)。

有効時刻の開始 / 有効時刻の終了

ユーザーが職責を選択した日時および職責を終了した日時。有効時刻の開始および有効時刻の終了が表示されるのは、職責またはフォームのいずれかの「サインオン監査」レベルでユーザーを監査した場合に限ります。

関連項目

- 「[ユーザーのモニター](#)」ウィンドウ」(2-8)
- 「[サインオン監査の設定](#)」(2-3)
- 「[サインオン監査レポート](#)」(2-7)
- 「[ユーザー・アクティビティの監査](#)」(2-2)
- 「[ユーザー監査およびデータ監査の概要](#)」(2-1)

サインオン監査不成功ログイン・レポート

このレポートを使用して、別のユーザーとして Oracle Applications にサインオンしようとして失敗したユーザーを表示します。正しいユーザー名を入力しても、パスワードを誤って入力するとログインは失敗します。

サインオン監査不成功ログイン・レポートは、監査対象に関係なく、どのユーザーについても生成できます。

レポート・パラメータ

ソート・キー

ユーザーがログインしようとした時刻、ユーザーのオペレーティング・システム・ログイン名、ユーザーが使用している端末またはアプリケーション・ユーザー名によってレポート内の情報をソートします。

ログイン名

他の検索基準を満たす特定のログイン名の情報を検索します。このパラメータをブランクのままにした場合は、レポートには他の検索基準を満たすログイン名がすべて入ります。

ユーザー名

他の検索基準を満たす特定のアプリケーション・ユーザー名の情報を検索します。このパラメータをブランクのままにした場合は、レポートには他の検索基準を満たすアプリケーション・ユーザー名がすべて入ります。

端末名

レポートをできる限り簡潔にするために、他の検索基準を満たす特定の端末についての情報を検索します。このパラメータをブランクのままにした場合は、レポートには他の検索基準を満たす端末名がすべて入ります。

実施日の開始 / 実施日の終了

特定の期間内で、かつ他の検索基準を満たす不成功ログインの情報を検索します。このパラメータを使用して、期間の始まりと終わりを指定します。これらのパラメータをブランクのままにした場合は、レポートには他の検索基準を満たす任意の日付から、このパラメータの現在の日付までの不成功ログインが入ります。

レポート見出し

レポートの見出しには、パラメータ値として入力した検索基準が表示されます。

列見出し

ユーザー名

サインオンしようとして失敗したユーザーの Oracle Applications ユーザー名。「ユーザー」フォームを使用すると、アプリケーション・ユーザーの詳細情報が表示されます。参照：[「ユーザー」ウィンドウ](#) (1-9)。

ログイン名

サインオンしようとして失敗したユーザーのオペレーティング・システム・ログイン名。

端末

サインオンしようとして失敗したユーザーが使用した端末のオペレーティング・システム ID。

実施時刻

ユーザーがサインオンしようとして失敗した日付と時刻。参照：[「ユーザーのモニター」ウィンドウ](#) (2-8)。

関連項目

- [「サインオン監査の設定」](#) (2-3)
- [「サインオン監査レポート」](#) (2-7)
- [「ユーザー・アクティビティの監査」](#) (2-2)
- [「ユーザー監査およびデータ監査の概要」](#) (2-1)

サインオン監査ユーザー・レポート

このレポートを使用して、サインオンしたユーザーとその継続時間を表示します。

注意：サインオン監査ユーザー・レポートは、監査対象にしているユーザーについてのみ生成できます。

レポート・パラメータ

ソート・キー

ユーザーがアプリケーション・ユーザー名の使用を開始または終了した時刻、ユーザーのオペレーティング・システム・ログイン名、ユーザーが使用している端末またはアプリケーション・ユーザー名によってレポート内の情報をソートします。

ログイン名

レポートをできる限り簡潔にするために、他の検索基準を満たす特定のログイン名についての情報を検索します。このパラメータをブランクのままにした場合は、レポートには他の検索基準を満たすログイン名がすべて入ります。

ユーザー名

レポートをできる限り簡潔にするために、他の検索基準を満たす特定のアプリケーション・ユーザー名の情報を検索します。このパラメータをブランクのままにした場合は、レポートには他の検索基準を満たすアプリケーション・ユーザー名がすべて入ります。

端末名

レポートをできる限り簡潔にするために、他の検索基準を満たす特定の端末の情報を検索します。このパラメータをブランクのままにした場合は、レポートには他の検索基準を満たす端末名がすべて入ります。

有効日の開始 / 有効日の終了

特定の期間内に Oracle Applications にログインしたユーザーで、かつ他の検索基準を満たすユーザーの情報を検索できます。このパラメータを使用して、期間の始まりと終わりを指定します。このパラメータをブランクのままにした場合は、レポートには他の検索基準を満たす最初の日付から現在の日付までのユーザー情報が入ります。

レポート見出し

レポートの見出しには、パラメータ値として入力した検索基準が表示されます。

列見出し

セッション番号

個々のアプリケーション・ユーザー・サインオンを一意に識別する Oracle Applications セッション番号。

ユーザー名

サインオンしたユーザーの Oracle Applications ユーザー名。「ユーザー」フォームを使用すると、アプリケーション・ユーザーの詳細情報が表示されます。参照：「[「ユーザー」ウィンドウ](#)」(1-9)。

ログイン名

サインオンしたユーザーのオペレーティング・システム・ログイン名。

端末名

ユーザーがサインオンするときに使用した端末のオペレーティング・システム ID。

有効時刻の開始 / 有効時刻の終了

ユーザーが Oracle Applications にサインオンした日時および終了した日時。有効時刻の開始および有効時刻の終了が表示されるのは、ユーザーの「サインオン監査」レベルでユーザーを監査した場合に限ります。

ORACLE プロセス

ユーザーのサインオン時に使用した Oracle プロセス ID。Oracle プロセスの詳細については、データベース管理者に連絡してください。

システム処理

ユーザーのサインオン時に使用したオペレーティング・システム処理 ID。オペレーティング・システム処理 ID の詳細については、オペレーティング・システム管理者に連絡してください。

関連項目

- 「[「ユーザーのモニター」ウィンドウ](#)」(2-8)
- 「[サインオン監査の設定](#)」(2-3)
- 「[サインオン監査レポート](#)」(2-7)
- 「[ユーザー・アクティビティの監査](#)」(2-2)
- 「[ユーザー監査およびデータ監査の概要](#)」(2-1)

監査証跡データについてのレポート

監査証跡を使用すると、重要なデータに対する変更の履歴（だれが、何を、いつ変更したか）を記録できます。監査証跡により、データ行またはデータ要素が現在値にどのようにして設定されたのかを容易に判別できます。文字フィールド、数値フィールドおよび日付フィールドなど、ほとんどのタイプのフィールドについての情報を追跡できます。

フォームにデータを入力するか、フォーム内のデータを更新すると、それらのフォームの基礎となるデータベース表が変更されます。監査証跡機能は、データベース内のどの行がいつ更新されたのか、およびどのユーザーが関連のフォームを使用してログインしたかを追跡します。

監査証跡

10.4 以降の Oracle Applications リリースでは、Oracle のデータベース・トリガーに基づく構造が提供されます。監査証跡機能は、監査対象表の「シャドウ表」に変更情報を格納します。この構造は、監査データを圧縮されていない「スパース」形式で保存するもので、特定の表および表のグループ（「監査グループ」）の監査が可能になります。

関連項目

- [「監査グループ」 ウィンドウ](#) (2-32)
- [「監査表」 ウィンドウ](#) (2-35)
- [「導入の監査」 ウィンドウ](#) (2-31)
- [「ユーザー監査およびデータ監査の概要」](#) (2-1)
- [「リリース 11i の監査証跡の設定」](#) (2-20)
- [「リリース 11 の監査証跡表、トリガーおよびビュー」](#) (2-22)
- [「監査証跡の使用禁止と、監査データのアーカイブ」](#) (2-30)
- [「監査情報についてのレポート」](#) (2-29)

リリース 11i の監査証跡の設定

特定の表についてユーザーが行ったすべての変更の履歴を、格納および検索することができます。監査は、監査グループを使用して実施されます。監査グループとは、監査対象の表を機能別にグループ分けしたものです。ある表を監査するには、その表が使用可能な監査グループに含まれていることが必要です。

監査証跡を設定する手順は次のとおりです。

SYS.DBA_TABLES についての選択権限の検証

データベース管理者に、SYS.DBA_TABLES についての SELECT 権限を APPLSYS アカウトに付与するよう要請します。通常は、このステップは、インストールまたはアップグレードの一環としてすでに実施されています。

監査グループの定義

監査グループは表および列のグループで、これには、必ずしも表の中のすべての列を含める必要はありません。これによって、監査グループを使用すると、個々の表ではなくグループ単位での監査が可能になります。一般に、同じビジネス・プロセスに属する表（たとえば発注表など）をまとめてグループにします。

1 つの表は複数の監査グループに属することもできます。その場合は、表は、所属グループの中で最上位の「状態」を持つグループの使用可能レベルに従って監査されます。「使用可能」が最上位で、「使用不可 - ダンプ・データ」、「使用不可 - 無成長」、「使用不可 - 表のパーティ」の順にレベルは低くなります。

1 つの表につき最高 240 列の監査が可能になり、LONG、RAW、または LONG RAW 以外のすべてのタイプの表列の監査も可能になります。監査グループには、表の主キーを形成するすべての列を含める必要があります。これらの列は、監査グループに自動的に追加されます。監査グループにいったん追加した列を除去することはできません。参照：「[監査グループ](#) ウィンドウ」(2-32)。

監査インストールの定義

ユーザーのサイトに登録されている Oracle ID から、監査が必要なものを選択します。これを使用して、複数のアプリケーション・インストールで監査を実施することができます。監査グループに表を 1 つ追加した場合、その表のインストールのうち、監査が使用可能であるすべてのインストールについて、監査機能が自動的に使用可能になります。参照：「[導入の監査](#) ウィンドウ」(2-31)。

監査証跡表等の更新レポートの実行による、監査の使用可能化

監査証跡定義（および監査機能）は、監査証跡表等の更新レポートを実行するまでは効力を持ちません。後で定義を変更した場合は、このプログラムを再実行する必要があります。監査証跡表等の更新レポートは、標準発行（レポートの発行）フォームから実行します。

注意： 監査証跡には 2 つのデータベース接続が必要です。使用している稼働プラットフォームがこの 2 つのデータベース接続を自動的にサポートしない場合（たとえば VMS や MPE/XL など）は、環境ファイルに環境変数 `FDATDB=< データベース接続文字列 >` を追加してください。

監査証跡表等の更新レポート

このプログラムは、監査グループ内の表において、インストレーション用のデータベース・トリガーを作成します。また、このプログラムは、監査情報を入れるシャドウ表を各監査対象表について1つずつ作成します。監査定義を変更したり、監査グループに対する監査機能を使用不可にした場合は、このプログラムは、監査トリガーおよびシャドウ表を適宜削除または修正します。

さらに、このプログラムは、監査データをレポート用に検索するための特殊ビューも作成します。

関連項目

- 「監査グループ」ウィンドウ (2-32)
- 「監査表」ウィンドウ (2-35)
- 「導入の監査」ウィンドウ (2-31)
- 「ユーザー監査およびデータ監査の概要」 (2-1)
- 「リリース 11 の監査証跡表、トリガーおよびビュー」 (2-22)
- 「監査証跡の使用禁止と、監査データのアーカイブ」 (2-30)
- 「監査証跡データについてのレポート」 (2-20)
- 「監査情報についてのレポート」 (2-29)

リリース 11 の監査証跡表、トリガーおよびビュー

監査を初めて使用可能にした時点で、監査対象表に対するシャドウ表が、その監査対象表と同じ Oracle ID で自動的に作成されます。シャドウ表には監査対象の列のみが含まれています。シャドウ表内のすべての列は、監査対象表内でのそれぞれの状態に関係なく、制約を受けません。

たとえば、シャドウ表内では常に NULL を使用できます。シャドウ表内のすべての列は、監査対象表内のそれぞれに対応する列とデータ型およびサイズが同じです。

シャドウ表の名前は、元の表名の最初の 26 文字に接尾辞「_A」（監査を表す）を付加したものです。

シャドウ表の列

すべての監査証跡シャドウ表には、次のような特殊監査列が含まれています。

- AUDIT_USER_NAME (アプリケーションのユーザー ID。ただし、SQL*Plus を使用して変更が行われた場合には、Oracle ID が入ります。)

- AUDIT_TIMESTAMP (挿入が行われた日時)
- AUDIT_TRANSACTION_TYPE (挿入は I、更新は U、削除は D、最終は L、現行は C)
- AUDIT_TRUE_NULLS (VARCHAR2(250) 列。NULL から変更された列名の区切り付きリストが入ります。)
- 表の主キー。これは特殊列ではなく、監査対象表の主キーを形成するすべての列です。規則により、1 つの行が削除されるとすべての監査対象列が格納されるという点に注意してください。同様に、挿入を行うと、シャドウ表内に NULL 値の行が作成されます。主キーに対する変更には削除のマークが付けられますが、新しい主キー値も挿入されます。

たとえば、次のような表があるとします。

```
SQL> DESCRIBE AUDIT_DEMO
```

NAME	NULL?	TYPE
PRIMARY_KEY		NUMBER (5)
VALUE_ONE		VARCHAR2 (5)
VALUE_TWO		VARCHAR2 (5)
VALUE_THREE		VARCHAR2 (5)

この表のシャドウ表は次のようになります (表のすべての列を監査するものとします)。

```
SQL> DESCRIBE AUDIT_DEMO_A
```

NAME	NULL?	TYPE
AUDIT_TIMESTAMP	NOT NULL	DATE
AUDIT_TRANSACTION_TYPE	NOT NULL	VARCHAR2 (1)
AUDIT_USER_NAME	NOT NULL	VARCHAR2 (100)
AUDIT_TRUE_NULLS		VARCHAR2 (250)
PRIMARY_KEY		NUMBER
VALUE_ONE		VARCHAR2 (5)
VALUE_TWO		VARCHAR2 (5)
VALUE_THREE		VARCHAR2 (5)

トリガーおよびプロシージャの監査

監査機能が使用可能になると、監査対象表についての「実行後」イベント内に自動生成されるデータベース・トリガーにより監査が実行されます。

このトリガーは、監査する各列を比較してその値が変更されたかどうかを確認するストアド・プロシージャをコールします。値が変更されている場合は、このプロシージャは、前の(古い) 値をシャドウ表に保存します。

監査機能により、表に対する各監査対象トランザクションについて、シャドウ表内に行が1つずつ作成されます。したがって、シャドウ表の中の1つの行は、そのトランザクションで変更されたすべての列の古い値すべてを表します。

表では1つの NULL については1バイトしか使用せず、すべての無変更の値は監査証拠によってシャドウ表の中で NULL として表されるため、データは圧縮されません (" スパース " 形式)。

監査トリガー名には、監査対象の表名の最初の 26 文字に「_AI」、「_AU」、または「_AD」を付加した名前が入ります。ここで、I は挿入、U は更新、D は削除を表します。同様に、監査プロシージャ名には、表名の最初の 26 文字に「_AIP」、「_AUP」、または「_ADP」を付加した名前が使用されます。表名は、必ず最初の 26 文字が一意でなければなりません。

ビュー

シャドウ表が作成された後で、シャドウ表に対するビューが作成されるので、" スパース " 行のデータに容易にアクセスできるようになります。これらのビューは、特定の日付に対する行や列の値の問合せや、時間の経過に伴う行や列への変更の追跡などの作業を簡素化します。

ビュー名には、監査対象の表名の最初の 26 文字に「_AC#」または「_AV#」を付加した名前が入ります。ここで、C または V はビューのタイプを示し、# は番号を示します。作成サイズには制限があるため、シャドウ表の列は複数のビューに分割することが必要な場合があります。これらのビューには連番が付けられます。

個々のビューごとに、少しずつ異なる方法でデータにアクセスできます。1つのビューでは、ユーザーは特定の時刻の行の値を再構築でき (_AC)、もう1つのビューでは、値が変更された時点のデータに簡単にアクセスができます (_AV)。

前記の例に示した表の場合、_AV1 ビューおよび _AC1 ビューは次のように作成されます。

```
SQL> DESCRIBE AUDIT_DEMO_AV1
NAME                                NULL?    TYPE
-----
AUDIT_TIMESTAMP                     DATE
AUDIT_TRANSACTION_TYPE              VARCHAR2(1)
AUDIT_USER_NAME                     VARCHAR2(100)
AUDIT_TRUE_NULLS                     VARCHAR2(250)
PRIMARY_KEY                          NUMBER
VALUE_ONE                           VARCHAR2(5)
VALUE_TWO                            VARCHAR2(5)
VALUE_THREE                          VARCHAR2(5)

SQL> DESCRIBE AUDIT_DEMO_AC1
NAME                                NULL?    TYPE
-----
AUDIT_TIMESTAMP                     DATE
AUDIT_TRANSACTION_TYPE              VARCHAR2(1)
```

AUDIT_USER_NAME	VARCHAR2(100)
PRIMARY_KEY	NUMBER
VALUE_ONE	VARCHAR2(5)
VALUE_TWO	VARCHAR2(5)
VALUE_THREE	VARCHAR2(5)

表およびビューでの、データの表示例

次の例は、一連の変更の後で、元の表、シャドウ表および監査ビューにデータがどのように表示されるかを示しています（空の AUDIT_DEMO 表から始まります）。

```
SQL> INSERT INTO AUDIT_DEMO VALUES (1, 'A', 'A', 'A');
SQL> INSERT INTO AUDIT_DEMO VALUES (2, 'X', 'X', 'X');
SQL> SELECT PRIMARY_KEY KEY, VALUE_ONE VAL_1,
        VALUE_TWO VAL_2, VALUE_THREE VAL_3 FROM AUDIT_DEMO;
```

KEY	VAL_1	VAL_2	VAL_3
1	A	A	A
2	X	X	X

```
SQL> UPDATE AUDIT_DEMO SET VALUE_ONE = 'B'
        WHERE PRIMARY_KEY = 1;
```

KEY	VAL_1	VAL_2	VAL_3
1	B	A	A
2	X	X	X

```
SQL> UPDATE AUDIT_DEMO SET VALUE_TWO = 'B'
        WHERE PRIMARY_KEY = 1;
```

KEY	VAL_1	VAL_2	VAL_3
1	B	B	A
2	X	X	X

```
SQL> UPDATE AUDIT_DEMO SET VALUE_THREE = 'B'
        WHERE PRIMARY_KEY = 1;
SQL> UPDATE AUDIT_DEMO SET VALUE_ONE = 'Y'
        WHERE PRIMARY_KEY = 2;
SQL> UPDATE AUDIT_DEMO SET VALUE_ONE = NULL
        WHERE PRIMARY_KEY = 1;
SQL> UPDATE AUDIT_DEMO SET VALUE_ONE = 'C'
        WHERE PRIMARY_KEY = 1;
```

2つの挿入と6つの更新の後で、監査対象表内の最終値は次のようになります。

KEY	VAL_1	VAL_2	VAL_3
1	C	B	B
2	Y	X	X

対応するシャドウ表内の最終値は、次のようになります。シャドウ表内の各行には、監査対象行の変更前の状態が示されます。トランザクション中に行の中の値が変更されなかった場合は、そのトランザクション内の該当の値について、シャドウ表の中に NULL が記録されます。

上記の例では、シャドウ表内の最初の2つの行は、2つの監査対象行の挿入前に、それらの行にデータがなかったときの状態を表しています。2つの挿入トランザクション（タイプ I）の行については、「前の値」は NULL 値です。同様に、行1の最初の値を A ではなく B に更新すると、シャドウ表の第3行に値 A が記録されます。

```
SQL> SELECT TO_CHAR(AUDIT_TIMESTAMP, 'HH24:MI:SS') TIME,
        AUDIT_TRANSACTION_TYPE TYPE, AUDIT_USER_NAME NAME,
        PRIMARY_KEY KEY, VALUE_ONE VAL_1, VALUE_TWO VAL_2,
        VALUE_THREE VAL_3, AUDIT_TRUE_NULLS FROM AUDIT_DEMO_A;
```

TIME	TYPE	NAME	KEY	VAL_1	VAL_2	VAL_3	AUDIT_TRUE_NULLS
11:08:16	I	FND60	1				
11:08:40	I	FND60	2				
11:18:40	U	FND60	1	A			
11:20:12	U	FND60	1		A		
11:21:54	U	FND60	1			A	
11:22:15	U	FND60	2	X			
14:20:50	U	FND60	1	B			
14:21:15	U	FND60	1				NYNN

8行が選択されました。

監査対象表内の行の現在値が指定されていると、シャドウ表内の対応する行で時系列順にさかのぼることにより、行に対して加えられた変更をトレースすることができます。

上記の表の例では、2つの挿入トランザクションと6つの更新トランザクションを行ったので、シャドウ表の中にこの8つのトランザクションが示されています。最後の行で、NYNNは、第2の表列 (VALUE_ONE) の値が、未変更の値（シャドウ表内で NULL で表される）のまま、実際の NULL 値 (Y) から変更されたことを示しています。

次の2つのビューは、監査対象データをさらに検査するための方法を提供します。

ビューの中のトランザクション・タイプCの行は、データが選択されたときの行の現在値を示しています（このビューはシャドウ表と監査対象表を結合するので、現在値の行は監査対象表の現在の状態を反映します）。

_AC ビューは、データの「記入済み」バージョンを提供するため、未変更の値は NULL 値で表されず、未変更値のままで表示されます。このビューを（タイムスタンプではなく）主キーにより順序付けすると、単一の監査対象行に対応するシャドウ表内のすべての行が一括して表示され、その中でタイムスタンプにより順序付けされます。

```
SQL> SELECT TO_CHAR(AUDIT_TIMESTAMP, 'HH24:MI:SS') TIME,
        AUDIT_TRANSACTION_TYPE TYPE, AUDIT_USER_NAME NAME,
        PRIMARY_KEY KEY, VALUE_ONE VAL_1, VALUE_TWO VAL_2,
        VALUE_THREE VAL_3 FROM AUDIT_DEMO_AC1
        ORDER BY PRIMARY_KEY, AUDIT_TIMESTAMP;
```

TIME	TYPE	NAME	KEY	VAL_1	VAL_2	VAL_3
11:08:16	I	FND60	1	A	A	A
11:18:40	U	FND60	1	B	A	A
11:20:12	U	FND60	1	B	B	A
11:21:54	U	FND60	1	B	B	B
14:20:50	U	FND60	1		B	B
14:21:15	U	FND60	1	C	B	B
17:53:34	C		1	C	B	B
11:08:40	I	FND60	2	X	X	X
11:22:15	U	FND60	2	Y	X	X
17:53:34	C		2	Y	X	X

10 行が選択されました。

注意：監査対象表に対する変更が 1 秒に 1 回より多い場合は（つまり SYSDATE が提供する 1 秒に 1 回の頻度より大きい場合は）、_AC ビューで使用されている結合が原因で、このビューの中でレコードの " 不明瞭さ " (1 トランザクションあたり複数のレコード) が生じることがあります。しかし、シャドウ表自体はトランザクションを正確に追跡するので、シャドウ表を直接使用すれば、複数のトランザクションを識別することができます。

_AV1 ビューは、監査データのスパース・ビュー（タイムスタンプ順）を提供します。

```
SQL> SELECT TO_CHAR(AUDIT_TIMESTAMP, 'HH24:MI:SS') TIME,
        AUDIT_TRANSACTION_TYPE TYPE, AUDIT_USER_NAME NAME,
        PRIMARY_KEY KEY, VALUE_ONE VAL_1, VALUE_TWO VAL_2,
        VALUE_THREE VAL_3, AUDIT_TRUE_NULLS
        FROM AUDIT_DEMO_AV1;
```

TIME	TYPE	NAME	KEY	VAL_1	VAL_2	VAL_3	AUDIT_TRUE_NULLS
11:08:16	I	FND60	1				

11:08:40	I	FND60	2			
11:18:40	U	FND60	1	A		
11:20:12	U	FND60	1		A	
11:21:54	U	FND60	1			A
11:22:15	U	FND60	2	X		
14:20:50	U	FND60	1	B		
14:21:15	U	FND60	1			NYNN
17:58:31	C		1	C	B	B
17:58:31	C		2	Y	X	X

10 行が選択されました。

次に、特定の値をだれがいつ変更したのかを、ビューを使用して判別する例を示します。

```
SQL> SELECT TO_CHAR(AUDIT_TIMESTAMP, 'HH24:MI:SS') TIME,
        AUDIT_TRANSACTION_TYPE TYPE, AUDIT_USER_NAME NAME
        FROM AUDIT_DEMO_AV1
        WHERE PRIMARY_KEY = 1
        AND VALUE_ONE = 'B';
```

TIME	TYPE	NAME
-----	----	-----
14:20:50	U	FND60

同様に、だれがいつ値を NULL に変更したのかを判別できます。

```
SQL> SELECT TO_CHAR(AUDIT_TIMESTAMP, 'HH24:MI:SS') TIME,
        AUDIT_TRANSACTION_TYPE TYPE, AUDIT_USER_NAME NAME
        FROM AUDIT_DEMO_AV1
        WHERE PRIMARY_KEY = 1
        AND VALUE_ONE IS NULL
        AND SUBSTR(AUDIT_TRUE_NULLS,2,1) = 'Y';
```

TIME	TYPE	NAME
-----	----	-----
14:21:15	U	FND60

監査表の変更

ある表についての監査の開始後に、シャドウ表に列を追加することができます。しかし、シャドウ表は、列の追加の前に生じた列変更については追跡しません。追加した場合は、監査証跡表等の更新レポートを再実行して、次の操作をする必要があります。

- シャドウ表に必要な列を追加します。
- 表について監査トリガーおよびプロシージャを再生成して、追加列を監査できるようにします。

関連項目

- 「[監査グループ](#)」ウィンドウ (2-32)
- 「[監査表](#)」ウィンドウ (2-35)
- 「[導入の監査](#)」ウィンドウ (2-31)
- 「[ユーザー監査およびデータ監査の概要](#)」 (2-1)
- 「[リリース 11i の監査証跡の設定](#)」 (2-20)
- 「[監査証跡の使用禁止と、監査データのアーカイブ](#)」 (2-30)
- 「[監査証跡データについてのレポート](#)」 (2-20)
- 「[監査情報についてのレポート](#)」 (2-29)

監査情報についてのレポート

監査データについてのレポート

監査レポートは必要に応じて作成するようにしてください。監査証跡機能により、監査レポートを容易にするためのシャドウ表のビューが提供されます。このようなビューを使用するためのレポートを作成することができます。

レポート作成の迅速化を図るために、シャドウ表に対して1つまたは複数の索引を作成できます。しかし、このような索引があると、実際に取引を監査するときのパフォーマンスが低下するので、レポート作成の終了後はシャドウ表から索引を削除してください。

注意： 監査対象表の体系は製品バージョンにより異なることがあるため、監査証跡機能は、既存のシャドウ表または監査対象データのアップグレードをサポートしていません。アップグレードの前に、シャドウ表をアーカイブして、監査対象データについてのすべての必要なレポート作成を行うようにしてください。

関連項目

- 「[監査グループ](#)」ウィンドウ (2-32)
- 「[監査表](#)」ウィンドウ (2-35)
- 「[導入の監査](#)」ウィンドウ (2-31)
- 「[ユーザー監査およびデータ監査の概要](#)」 (2-1)
- 「[リリース 11i の監査証跡の設定](#)」 (2-20)

- 「リリース 11 の監査証跡表、トリガーおよびビュー」 (2-22)
- 「監査証跡の使用禁止と、監査データのアーカイブ」 (2-30)
- 「監査証跡データについてのレポート」 (2-20)

監査証跡の使用禁止と、監査データのアーカイブ

監査についてレポートしたり、監査を使用禁止にしたりすることは、いつでもできます。監査を使用禁止にするときは、次の手順を使用してください。

新しいトランザクションの監査を停止します

「使用不可 - アーカイブの準備」または「使用不可 - 監査の割込み」のいずれかを使用して、「監査証跡表等の更新」レポートを実行することにより、監査を使用禁止にします。

使用不可 - アーカイブの準備	監査対象表内のすべての行の現在値をシャドウ表にコピーしてから監査トリガーを使用禁止にします。これで、変更があっても記録されなくなります。シャドウ表は、パージする前にアーカイブするようにしてください。
使用不可 - 監査の割込	監査表内の修正された各行について、シャドウ表内に「最終」行を1つずつ格納するように、トリガーを修正します (シャドウ表内の所定の行は、更新前の監査対象行のデータを表しているという点に注意してください)。監査対象表内の行が再度変更された場合 (2 度目)、その変更は記録されません。シャドウ表は、監査対象表内の各行について1行ずつ含まれるようになるまで、徐々に大きくなります。その後は、変更があっても記録されなくなります。

監査データをアーカイブします

組織の業務要件に従って、シャドウ表内の情報をアーカイブする必要があります。

シャドウ表を消去します

監査を再起動する前に、シャドウ表を消去する必要があります。監査が使用禁止になっていたときにトランザクションがあり、シャドウ表を消去しなかった場合は、シャドウ表内のデータは無効になります。これは、トランザクションが記録されなかったために不整合が生じるからです。シャドウ表をパージするには、監査グループを「使用不可 - 表のパージ」に設定して、「監査証跡表等の更新」レポートを実行します。

使用不可 - 表のパージ	監査トリガーとビューを削除して、シャドウ表からすべてのデータを削除します。
--------------	---------------------------------------

監査を再起動します（必要な場合）

監査を再起動するには、監査グループを「要求の使用可能化」に設定して、「監査証跡表等の更新」レポートを再度実行します。

注意：ある表についての監査を「使用不可 - 表のパージ」を使用して使用禁止にして、その後で再度使用可能にすると、監査が再度使用可能になった時点で監査証跡機能はシャドウ表の内容をフラッシュします。監査を再度使用可能にするときは、その前に保存しておく必要があるシャドウ表データをアーカイブしてください。

関連項目

- 「[監査グループ](#)」ウィンドウ（2-32）
- 「[監査表](#)」ウィンドウ（2-35）
- 「[導入の監査](#)」ウィンドウ（2-31）
- 「[ユーザー監査およびデータ監査の概要](#)」（2-1）
- 「[リリース 11i の監査証跡の設定](#)」（2-20）
- 「[リリース 11 の監査証跡表、トリガーおよびビュー](#)」（2-22）
- 「[監査証跡データについてのレポート](#)」（2-20）
- 「[監査情報についてのレポート](#)」（2-29）

「導入の監査」ウィンドウ

このウィンドウを使用すると、導入における Oracle ユーザー名に対する監査証跡を使用可能にすることができます。Oracle ユーザー名により、アプリケーションの表およびデータベース・オブジェクトへのアクセス権限が与えられます。

監査を有効にするには、1 つまたは複数の監査グループを定義して、「監査証跡表等の更新」レポートを実行することが必要です。参照：[「監査証跡データについてのレポート」](#)（2-20）。

前提条件

- Oracle ユーザー名を登録します。参照：[「Oracle ユーザー・ウィンドウ」](#)（7-3）。

導入の監査ブロック

Oracle ユーザー名

監査する表を所有するユーザーの Oracle ユーザー名を選択します。

監査使用可能

「監査使用可」チェックボックスをオンにして、Oracle ユーザー名用の監査を使用可能にします。監査を有効にするには、1 つまたは複数の監査グループを定義して、「監査証跡表等の更新」レポートを実行することが必要です。

関連項目

- [「監査グループ」ウィンドウ](#) (2-32)
- [「監査表」ウィンドウ](#) (2-35)
- [「ユーザー監査およびデータ監査の概要」](#) (2-1)
- [「リリース 11i の監査証跡の設定」](#) (2-20)
- [「リリース 11 の監査証跡表、トリガーおよびビュー」](#) (2-22)
- [「監査証跡の使用禁止と、監査データのアーカイブ」](#) (2-30)
- [「監査証跡データについてのレポート」](#) (2-20)
- [「監査情報についてのレポート」](#) (2-29)

「監査グループ」ウィンドウ

このウィンドウを使用して、監査する表を選択します。監査グループを定義して表の監査を行います。監査グループは、1 つまたは複数の表から構成されます。

まず監査する表を識別して、次に「監査表」ウィンドウを使用して各表の中の監査対象の列を選択します。または、(「監査表」ウィンドウを使用して) 特定の表の中の監査する列を選択してから、監査グループを定義します (このウィンドウを使用します)。

監査グループの中の表に対して監査を使用可能または使用不可にするには、「要求の発行」ウィンドウで「監査証跡表等の更新」プログラムを実行します。後でグループの定義または監査状態を変更するときには、このプログラムを実行する必要があります。

前提条件

- 「導入の監査」ウィンドウで監査インストレーションを定義します。

注意： 監査を正常に行うためには、表とその主キー情報を事前に登録および定義しておく必要があります。監査する表がカスタム表（Oracle Applications の一部として提供されたもの以外の表）であれば、さらに次の 2 つの手順を実行します。

- Oracle Application Object Library の「表」ウィンドウ（アプリケーション開発者職責）を使用して表および主キー列を登録します。
- 「要求の発行」ウィンドウから「表の登録」コンカレント・プログラムを実行します。

監査グループ・ブロック

監査グループを識別して、そのグループに対して監査を使用可能または使用不可にします。

アプリケーション名

監査グループに関連付けるアプリケーション名を選択します。アプリケーション名とグループ名の組合せにより、監査グループを一意に識別できます。監査グループを使用して、追加のアプリケーションにある表を監査できます。

監査グループ

監査グループ名を入力します。

グループ状態

新規監査グループを定義する場合は、「要求の使用可能化」を選択します。「監査証跡表等の更新」レポートの実行時に、コンカレント・プログラムにより、監査グループの表に対応するデータベース・トリガーが作成されます。プログラムを実行すると、このフィールドでは、監査証跡機能が有効である監査グループに対して「使用可能」と表示されます。

注意： 監査グループの各表の主キー列はすべて監査用に自動的に選択されます。監査する列の選択に「監査表」ウィンドウを使用するかどうかは問題となりません。

グループの監査を使用不可にするには、次のオプションのうちから 1 つを選択してから、「監査証跡表等の更新」レポートを実行して変更を有効化します。

使用不可 - アーカイブの準備	監査対象表内のすべての行の現在値をシャドウ表にコピーしてから、監査トリガーを使用不可にします。このオプションが最も多くの領域を必要とします。その理由は、監査対象表の行ごとに最低 1 行がシャドウ表にあるためです（シャドウ表には、さらに監査対象表にある元の行に対するトランザクションごとにさらに別の行があります）。次に、表をアーカイブしてから、シャドウ表を空にします。
使用不可 - 監査の割込み	監査表の中で監査対象行が変更されるときに、トリガーを変更してシャドウ表の中の最後の 1 行を格納します。（シャドウ表の中の行は、更新前の監査対象行上のデータを表していることに注意してください）。挿入部分またはこれ以後に変更した部分は監査の対象となりません。シャドウ表はその後もゆっくりと大きくなり、その中のデータには、既存の監査ビューを使用してアクセスできません。
使用不可 - 表のパージ	監査トリガーとビューを削除して、シャドウ表からすべてのデータを削除します。

監査表ブロック

監査グループの中の監査対象のアプリケーション表を識別します。

ユーザー表

データベース表のエンド・ユーザー表名（多くの場合、表の名前と同じ名前が使用されます）を選択します。表を選択すると、その表の名前と関連アプリケーションが表示されます。

表名

このフィールドには、監査グループに含める表の実際の名前が表示されます。

アプリケーション

このフィールドには、監査グループに含める表のアプリケーション名が表示されます。

摘要

このフィールドには、監査グループに含める表の説明が表示されます。

関連項目

- 「監査表」ウィンドウ（2-35）
- 「導入の監査」ウィンドウ（2-31）
- 「ユーザー監査およびデータ監査の概要」（2-1）

- 「リリース 11i の監査証跡の設定」 (2-20)
- 「リリース 11 の監査証跡表、トリガーおよびビュー」 (2-22)
- 「監査証跡の使用禁止と、監査データのアーカイブ」 (2-30)
- 「監査証跡データについてのレポート」 (2-20)
- 「監査情報についてのレポート」 (2-29)

「監査表」ウィンドウ

このウィンドウを使用して、表から監査対象の列を選択します。

まず表の中の監査する列を識別します。次に、「監査グループ」ウィンドウを使用してその表を監査グループに含めます。または、最初に（「監査グループ」ウィンドウを使用して）監査グループを定義してから、次に表の中のどの列を監査するかを選択します（このウィンドウを使用します）。

監査グループの中の表（つまり、ここで選択した列）に対して監査を使用可能または使用不可にするには、「要求の発行」ウィンドウで「監査証跡表等の更新」プログラムを実行する必要があります。後で監査する追加の列を選択したり、定義またはグループの監査状態を変更するときには、このプログラムを実行する必要があります。

前提条件

- 「導入の監査」ウィンドウで監査インストレーションを定義します。

注意： 監査を正常に行うためには、表とその主キー情報を事前に登録および定義しておく必要があります。監査する表がカスタム表（Oracle Applications の一部として提供されたもの以外の表）であれば、さらに次の 2 つの手順を実行します。

- Oracle Application Object Library の「表」ウィンドウ（アプリケーション開発者職責）を使用して表および主キー列を登録します。
- 「要求の発行」ウィンドウから「表の登録」コンカレント・プログラムを実行します。

監査表の定義ブロック

監査するアプリケーション表を識別します。さらに続けてメニューまたはツールバーから「移動」>「次レコード」を選択すると、導入サイトに登録されているすべてのアプリケーション表の名前がアルファベット順に表示されます。

ユーザー表名

データベース表のエンド・ユーザー表名（多くの場合、表の名前と同じ名前が使用されます）を選択します。表を選択すると、その表の名前と関連アプリケーションが表示されます。

表名

このフィールドには、監査グループに含める表の実際の名前が表示されます。

アプリケーション

このフィールドには、監査グループに含める表のアプリケーション名が表示されます。

監査列ブロック

監査する列を選択します。さらに続けてメニューまたはツールバーから「移動」>「次レコード」を選択すると、導入サイトに登録されているすべてのアプリケーション表の名前がアルファベット順に表示されます。

- 一度選択すると、監査対象の列は削除できません。
- ただし、監査する列を追加することはできます。
- 監査対象の列を選択するたびに、その列が属する表を含むすべての監査グループに影響が生じます。

列名

監査するデータベース列の名前を入力します。表の主キー列の名前は自動的に入力されるので、明示的に入力する必要はありません。重複した列名を保存しようとすると、エラー・メッセージが発行されます。自動的に表示される列を問い合わせることができます。

一度選択した列は監査セットから削除できないので、注意が必要です。ただし、後で監査セットに列を追加することはできます。

列を選択すると、その列タイプと、その列がこの表の主キーの一部かどうかが表示されます。

列タイプ

このフィールドには、列が格納するデータ型が、たとえば `varchar2` のように示されます。

主キー

このフィールドには、監査対象の列が主キー列であるかどうかを示す Yes または No が示されます。

監査対象として選択しなかった主キー列があると、選択した列を保存するときに自動的に監査対象に含められます。たとえば、監査対象の表に 2 つの主キー列があり、そのうちの 1 つを監査対象として選択すると、選択した列を保存するときに 2 番目の主キー列も自動的に選択されます。

関連項目

- 「[「監査グループ」ウィンドウ](#)」(2-32)
- 「[「導入の監査」ウィンドウ](#)」(2-31)
- 「[「ユーザー監査およびデータ監査の概要](#)」(2-1)
- 「[「リリース 11i の監査証跡の設定](#)」(2-20)
- 「[「リリース 11 の監査証跡表、トリガーおよびビュー](#)」(2-22)
- 「[「監査証跡の使用禁止と、監査データのアーカイブ](#)」(2-30)
- 「[「監査証跡データについてのレポート](#)」(2-20)
- 「[「監査情報についてのレポート](#)」(2-29)

コンカレント・プログラムとコンカレント 要求の管理

コンカレント・プログラムおよびコンカレント要求の概要

コンカレント・プログラムは、他のコンカレント・プログラムやオンライン操作と同時に実行される実行ファイルであり、ハードウェアの処理能力をフル活用します。通常、コンカレント・プログラムは、仕訳の転記やレポートの作成など、長時間かかるデータ集約型のタスクです。

要求グループと要求セット

レポートとコンカレント・プログラムは、要求グループと要求セットに管理できます。

- 要求グループは、レポートまたはコンカレント・プログラムの集合です。システム管理者は、レポートとコンカレント・プログラムへのユーザー・アクセスを管理するために、レポート・グループを定義します。要求グループを作成できるのは、システム管理者のみです。
- 要求セットでは、レポートまたはコンカレント・プログラムの集合に対して、実行オプションと印刷オプションを定義します。パラメータ値を定義することもあります。要求セットを定義できるのは、エンド・ユーザーとシステム管理者です。システム管理者は、エンド・ユーザーよりも広範囲な要求セット権限を持っています。

標準要求発行と要求グループ

標準要求発行は、1つの標準フォームからすべてのレポートと他のコンカレント・プログラムを選択し、実行するための Oracle Applications の機能です。標準発行フォームは「要求の発行」と呼ばれますが、異なるタイトルが表示されるようにカスタマイズできます。

- 「要求の発行」フォームから選択できるレポートとコンカレント・プログラムは、要求セキュリティ・グループに属しています。要求セキュリティ・グループは、職責に割り当てられた要求グループです。

- カスタマイズされた「要求の発行」フォームから選択できるレポートとコンカレント・プログラムは、コードを使用する要求グループに属しています。

要約すると、要求グループを使用してレポートとコンカレント・プログラムへのアクセスを管理するには、2つの方法があります。1つはユーザーの職責に基づく方法で、もう1つはカスタマイズされた標準発行（「要求の発行」）フォームに基づく方法です。参照：[「コードを使用した「要求の発行」ウィンドウのカスタマイズ」](#)（3-17）

参照：「Oracle Applications のレポートとプログラムの実行」

ユーザー別の有効要求の制限

システム管理者は、有効（「実行中」ステータス）にできる要求の数をユーザーごとに制限できます。これにより、1人のユーザーが要求キューを独占することのないようにできます。たとえば、有効要求制限が5のユーザーが20の要求を発行した場合、同時に実行される要求は5つのみです。残りの要求は、そのユーザーの有効要求の数が5未満になるごとに、実行されます。「プロファイル・オプション」ウィンドウを使用して、「コンカレント: 有効要求制限」プロファイルを設定します。すべてのユーザーにグローバル制限を設定するには、このオプションをサイト・レベルで設定します。その後、このプロファイル・オプションをユーザー・レベルで設定することにより、ユーザーごとに制限を変更できます。

関連項目

- [「システム管理者の要求セット権限」](#)（3-10）
- [「データ・グループの概要」](#)（3-25）
- [「プログラムの要求グループ別編成」](#)（3-15）
- [「プログラムの要求セット別編成」](#)（3-3）
- [「プログラム定義のコピーと変更」](#)（3-41）

コンカレント要求の多言語サポート

リリース 11i から、ユーザーは、単一コンカレント・プログラムを複数回（毎回異なる言語で）実行させる単一コンカレント要求を発行できます。生成される出力は、言語に基づいて別のプリンタに転送されます。ユーザーは、出力言語に基づいて、完了通知も転送できます。

たとえば、ユーザーは、「請求書の印刷」プログラムの要求を発行できます。この要求により、このプログラムは毎回異なる言語で数回実行され、請求書の各セットは、別のプリンタで印刷されます。

注意：多言語要求は、要求セット内では実行できません。

その他の関連項目

『Oracle Applications 概要』

『Oracle Applications 開発者ガイド』

プログラムの要求セット別編成

要求セットは、事前定義済みの印刷オプションおよびパラメータ値を使用して複数のレポートおよびコンカレント・プログラムを実行する迅速で容易な方法です。要求セットでは、セットごとに発行されるステージ別に要求がグループ化されます。ステージの発行順は、前のステージのステータスによって決まります。

システム管理者は要求セットを使用して、レポートおよびコンカレント・プログラムへのアクセスをカスタマイズすることもできます。システム管理者は、要求セットを使用して次のことができます。

- ある職責のユーザーに、それらのユーザーの要求セキュリティ・グループ外にある選択したレポートおよびコンカレント・プログラムを実行できる能力を付与する。
- 要求および別のコンカレント・プログラムへのアクセス権を個々のユーザー単位で付与する。
- エンド・ユーザーが編集できない印刷オプションおよびパラメータ値を使用して、セット内のレポートが実行されることを保証する。

注意：多言語要求は、要求セット内では実行できません。

システム管理者には、「要求セット」ウィンドウの権限付与バージョンなど、アプリケーション・ユーザーより高い権限が与えられています。参照：「「要求セット」ウィンドウ」。

要求セットの定義

要求セットを定義し、その要求セットを「要求の発行」フォームから発行することにより、同じコンカレント要求セットを定期的に行うことができます。

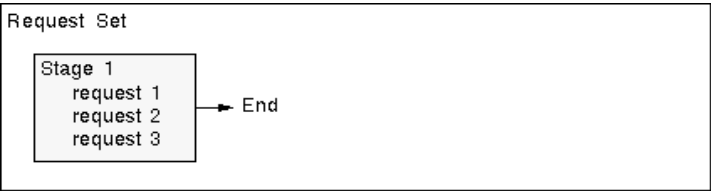
システム管理者は、定義する要求セットの中にすべての「標準要求発行」レポートまたはコンカレント・プログラムを組み込むことができます。エンド・ユーザーは、要求セットを定義する場合、そのエンド・ユーザーの職責の要求セキュリティ・グループに属するレポートおよびプログラムのみから選択ができます。

要求セットの作成と編集には、「要求セット」フォームを使用します。参照：「「要求セット」ウィンドウ」。

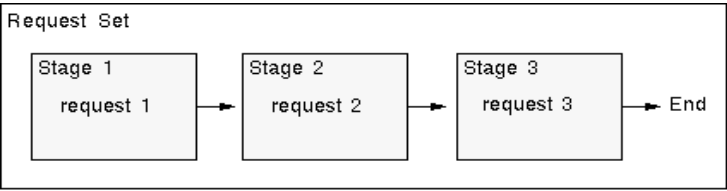
要求セット・ステージ

要求セットのステージ別編成

要求セットは、要求の実行順序を決めるため、リンクした1つ以上の「ステージ」に分割されます。各ステージは、並列で（順番にかかわらず同時に）実行させる1つ以上の要求で構成されます。たとえば、最も単純な要求セット体系では、すべての要求が単一のステージに割り当てられます。この場合は、すべての要求が並列で処理されます。

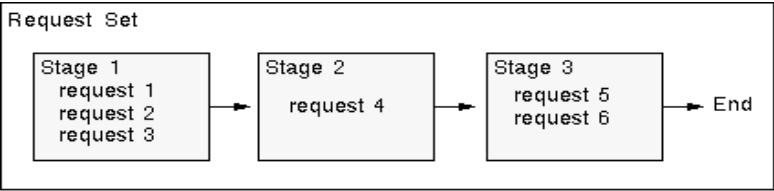


複数の要求を順次実行する場合は、要求を異なるステージに割り当ててから、そのステージを実行する順序でリンクします。



コンカレント・マネージャで一度に実行できるのは、要求セット内の1つのステージのみです。1つのステージが完了すると、次のステージが発行されます。ステージ内の要求がすべて完了するまでは、そのステージは完了したとはみなされません。

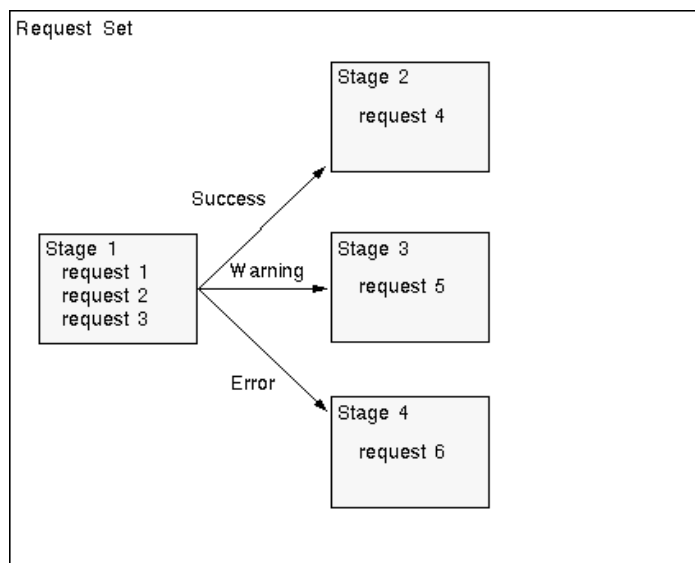
ステージを使用すると、複数の要求を並列で実行でき、その後、次のステージへ順に移動していくので便利です。この機能により、要求セットをさらに多目的かつ効率的に使用できます。



ステージのステータスの使用

要求セットやコンカレント要求と同様に、ステージも様々なステータスで完了します。各ステージは「成功」、「警告」、「エラー」のいずれかのステータスで完了します。完了ステータスを使用して要求セットを構成するには、完了ステータスに基づいて現行のステージに続くステージを定義します。たとえば、次の要求セットでは常にステージ1から実行されます。ステージ1がステータス「警告」で完了すると、「警告」リンクに進み、ステージ3が実行されます。ステージ3が完了すると、次に続くリンクがないため、このセットは終了します。

図 2



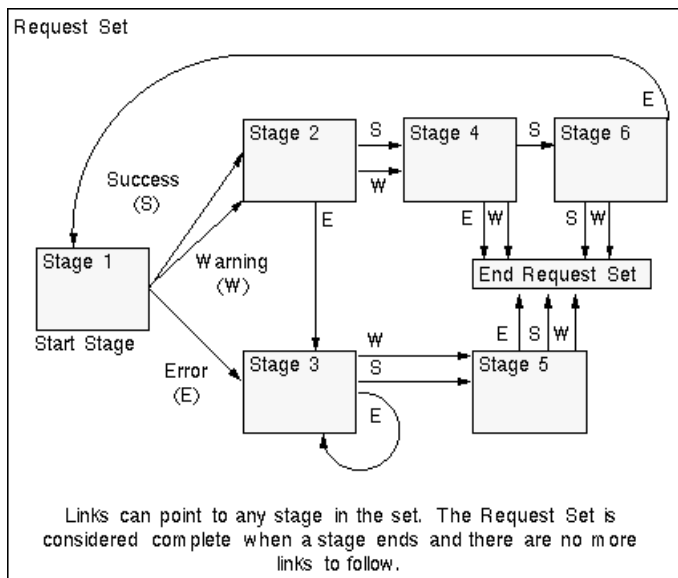
この例では、標準ステージ機能により、ステージのステータスを決めています。標準ステージ機能では、ステージのステータスを計算するのに、ステージ内の要求ステータスを使用します。ステージ内の要求がすべて「成功」ステータスで完了している場合、ステージのステータスも「成功」になります。1つ以上の要求が「エラー」で完了していると、ステージのステータスも「エラー」になります。ステージのステータスが「警告」になるのは、必ず1つ以上の要求が「警告」ステータスをもち、「エラー」ステータスの要求がない場合です。

ステージのリンク

要求セット内のステージをリンクする上での制限事項はありません。各ステージは、どのステージにもリンクできるほか、そのステージ自体にリンクすることも可能です。また、2つ以上のリンクが同一のステージを示すようにも定義できます。たとえば、ステージ1の完了

ステータスが「成功」または「警告」の場合にはステージ1をステージ2にリンクし、「エラー」ステータスの場合にはステージ3にリンクすることができます。

図 3



各完了ステータスに対し、次にリンクするステージを指定しなければ、そこで要求セットの実行が終了します。要求セットの実行は、どのステージの完了後に終了させてもかまいません。要求セット内のいずれかのステージが、別のステージへのリンクが指定されていないステータスで完了した時点で、その要求セットは実行終了となります。

ステージ評価機能

ステージの完了ステータスは、事前に定義された関数で判別されます。Oracle Applications の Standard Stage Evaluation 関数では、内蔵の要求の完了ステータスが使用されます。ステージのステータスを判別する場合は、この関数を使用します。

要求セットの完了ステータス

リンク先が定義されていないステータスでステージが完了すると、要求セットは終了します。要求セットの完了ステータスは次のいずれかの方法で決まります。

- 要求セット内で最後に実行されるステージの完了ステータスを使用する。デフォルトでは、この方法が使用されます。
- ユーザーが要求セット内の特定のステージを「重要」と定義することにより、デフォルトで決定されるステータスより優先させる。要求セット内で重要ステージが実行される

場合、そのセットの完了ステータスは最後に実行された重要ステージの完了ステータスと同じになります。この方法は、セット内の最後のステージが「クリーン・アップ」ステージで、セット全体のステータスを決めるほど重要度の高くない場合に役立ちます。

印刷要求セット

要求セット内のレポートには、個々のレポートごとに異なるプリンタを選択できます。要求セットを定義すると、レポートの出力先プリンタなどの印刷オプションが保存されるので、要求セットを実行するときに再度それらのオプションを指定する必要はありません。

注意:「コンカレント・プログラム」フォームを使用してコンカレント・プログラム用にプリンタを定義した場合、その値をユーザー・プロファイル・オプションの設定または要求セット定義によって、あるいはプログラムまたは要求セットの実行時に更新することはできません。

注意:「コンカレント・プログラム」フォームの中で、要求セット・コンカレント・プログラム（たとえば、「要求セットの買掛年令調べ」レポートなど）用にプリンタを定義しても効果はありません。そのプリンタ定義は参照されません。

コンカレント・プログラムとしての要求セット

非互換性を許可する要求セットまたは要求セット内のステージを定義すると、要求セット内の要求を入力された指示どおりに実行するコンカレント・プログラムが作成されます。

要求セットを実行するすべてのコンカレント・プログラムには、要求セット<要求セット名>というタイトルが付き、要求セット・ステージを実行するプログラムにはすべて、要求セット・ステージ<要求セットステージ名>というタイトルが付きます。「コンカレント・プログラム」フォームで、プログラム名に基づいて要求セットまたは要求セット・ステージのコンカレント・プログラムを問い合わせるには、「名称」フィールドに次のように入力します。

- 「要求セット」または「要求セット・ステージ」に続けてコンカレント・プログラム名
- すべての要求セット・プログラムについて問合せを実行するには「要求セット %」または「要求セット・ステージ %」

要求セットおよび要求セット・ステージのコンカレント・プログラムは、要求セットまたはステージの実行を記録するログ・ファイルを作成します。要求セットまたは要求セット・ステージ内の個々のレポートまたはコンカレント・プログラムも、それぞれログ・ファイルを作成します。

非互換性を許可する要求セットを実行する場合には、その要求セットを定義したコンカレント・プログラムの実行要求を発行します。要求セットのコンカレント・プログラムを使用す

ると、要求セット・ステージのコンカレント・プログラムが発行されます。要求セット・ステージのコンカレント・プログラムは、ステージ内の個々のプログラムおよびレポートの要求を発行します。要求セットのコンカレント・プログラムまたは要求セット・ステージのコンカレント・プログラムの実行要求は「親」要求であり、個々のプログラムおよびレポートの実行要求は「子」要求です。

要求セットおよび要求セットに格納されているプログラムのステータスは、「コンカレント要求」フォームを使用して確認できます。[「要求セットの要求フェーズおよびステータス」](#) (3-8) では、要求セットに関連する要求フェーズおよびステータス情報が表示されます。

要求セットの要求フェーズおよびステータス

フェーズ	ステータス	摘要
実行中	一時停止	「親」要求は、すべての「子」要求が完了するまで休止します。たとえば、要求セット・ステージはそのステージ内にあるすべてのレポートが完了するまで休止します。
	再開	同じ「親」要求から発行されたすべての要求の実行が完了しています。「親」要求は実行を再開します。

表 1

要求セットの変更

要求セットを変更できるのは、そのセットの所有者またはシステム管理者のみです。変更を行うには、「要求セット」ウィンドウで変更する要求セットを問い合せます。

注意：Oracle アプリケーション提供の要求セットに対する変更内容をアップグレード後も継続させたい場合には、アップグレードの前に、要求セットを別の名称で命名または作成しなおしてください。名称を変えずに事前定義済みの要求セットを変更すると、その変更内容は、Oracle Applications のアップグレード時に上書きされます。

関連項目

- [「コンカレント・プログラムおよびコンカレント要求の概要」](#) (3-1)
- [「システム管理者の要求セット権限」](#) (3-10)
- [「要求セット・レポート」](#) (3-15)
- [「要求セットおよび所有者」](#) (3-9)
- [「要求セットの非互換性」](#) (3-11)
- [「要求セット内のパラメータの共有」](#) (3-12)

要求セットおよび所有者

要求セットを定義または編集する場合、エンド・ユーザーの権限とシステム管理者の権限との間には大きな違いがあります。

エンド・ユーザーは自分が作成した要求セットを所有する

エンド・ユーザーは、自分の職責へ割り当てられたレポート・セキュリティ・グループの一部であるレポート、別の要求セットまたはコンカレント・プログラムを選択することによって、要求セットを作成できます。

エンド・ユーザーは、要求セットを作成した場合、自動的にその要求セットの「所有者」になります。所有権は、そのユーザーのアプリケーション・ユーザー名によって識別されます。

エンド・ユーザーは「要求セット」フォームを使用して、新しい要求セットを作成したり、自分が所有する要求セットについて問い合わせや更新を行います。エンド・ユーザーが編集できる要求セットは、自分の所有する要求セットのみです。

ここでは、エンド・ユーザーが所有する要求セットをプライベート要求セットと呼ぶことがあります。プライベート要求セットは、要求セキュリティ・グループへ自動的に追加されません。つまり、自分のプライベート要求セットへ他のユーザーが「要求の発行」ウィンドウを使用してアクセスすることはできません。ただし、システム管理者がその要求セットを要求セキュリティ・グループへ割り当てた場合は除きます。

エンド・ユーザーが所有する要求セットは、そのユーザーがどの職責の下で操作しているかに関係なく、そのユーザーからは常に使用可能です。しかし、コードを使用して要求グループ内のレポートのみを表示するようカスタマイズされた標準発行フォームには、プライベート要求セットは表示されません。

Oracle Applications へサイン・オンしたユーザーは、次のものに含まれる要求、要求セットおよびコンカレント・プログラムを実行できます。

- そのユーザーの職責の要求セキュリティ・グループ
- そのユーザーが所有する要求セット

エンド・ユーザーから見たプライベート要求セットの利点

エンド・ユーザーにとって、プライベート要求セットには主に次の2つの利点があります。

1. ユーザーは、どの職責の下で作業をしているかに関係なく、自分が所有する要求セットを常時使用できます。
2. ユーザーは、他のユーザーの選択元となる標準発行コンカレント・プログラムの一覧に要求セット選択項目を追加せずに、必要な数のみ要求セットを作成できます。

関連項目

- 「コンカレント・プログラムおよびコンカレント要求の概要」 (3-1)
- 「システム管理者の要求セット権限」 (3-10)
- 「要求セット・レポート」 (3-15)

システム管理者の要求セット権限

システム管理者の権限は、次のとおりです。

- 任意のレポートまたはコンカレント・プログラムが含まれている要求セットを作成します。
- 「要求セット」フォームを使用して、すべての要求セットについて問合せおよび編集を実行します。
- 個々の要求セットについて、非互換性ルールを許可し、定義できます。参照：「[要求セットの非互換性](#)」 (3-11)。

要求セットを定義した後、その要求セットをユーザーがすべての職責から実行または編集できるようにするには、そのユーザーを要求セットの所有者として割り当てます。所有者のいない要求セットは、どのエンド・ユーザーも編集または更新ができません。この方法により、要求セットの印刷オプションおよびレポート・パラメータを保証することができます。また、あとで要求セットを編集し、要求セットの所有権プロパティを削除したり変更したりすることもできます。

その他のユーザーも、システム管理者が要求セットをそれらのユーザーの職責の要求セキュリティ・グループへ割り当てた場合には、その要求セットを実行できます。システム管理者が要求セットを要求セキュリティ・グループへ割り当てなかった場合には、所有者のみがその要求セットを実行できます。この方法により、レポートおよびコンカレント・プログラムへのアクセス権をユーザーごとに付与できます。

要求セキュリティ・グループ、要求セットおよびレポート

システム管理者は、要求セキュリティ・グループへプライベート要求セットを含む任意の要求セットを追加できます。これにより、システム管理者はある職責のメンバーに、それらのメンバーの要求セキュリティ・グループ外にあるレポートおよびプログラムへのアクセスを許可することができます。

ユーザーの要求セキュリティ・グループに属するレポートに対する要求セットの編集権限およびレポートのビュー権限は、ユーザーの要求セキュリティ・グループに属さないレポートの場合と異なります。

- 要求セットを編集できません。
- 個々のレポートを単独で実行することはできません。要求セット全体のみを実行できます。
- ユーザーの要求セキュリティ・グループ内にある別の要求を要求セットに追加できます。

- レポートがユーザーの要求セキュリティ・グループ内にあるかどうかに関係なく、要求セットから要求を任意に削除できます。
- 要求セット内の個々のレポートに対する印刷オプションまたはパラメータは、そのレポートがユーザーの要求セキュリティ・グループ内にあれば、更新できます。
- 個々のレポートを単独で実行することはできず、要求セット全体のみを実行できます。

システム管理者から見た要求セットの利点

システム管理者にとって、要求セットには主に次の3つの利点があります。

1. 要求セットは、コンカレント・プログラムへのアクセスを個々のユーザー単位で制御する手段となります。

要求セットを定義し、それに所有者を割り当てた後、その要求セットをどの要求セキュリティ・グループにも割り当てなければ、その要求セット内のレポートおよびプログラムは所有者のみが使用できます。

2. 「所有者」フィールドをブランクのまま残すと、システム管理者は個々のプログラムおよびパラメータをエンド・ユーザーが編集も更新もできない要求セットを作成できます。

所有者がいない要求セットを編集できるのは、システム管理者のみです。

3. システム管理者は、ある職責のメンバーに、それらのメンバーの要求セキュリティ・グループ外にあるレポートおよびプログラムへのアクセスを許可することができます。

ある要求セキュリティ・グループ内にないレポートまたはプログラムを入れた要求セットを定義し、その要求セットをその要求セキュリティ・グループへ割り当てることにより、選択したレポートまたはプログラムについてユーザーに実行権限を付与できます。ただし、編集権限は付与されません。

関連項目

- 「[コンカレント・プログラムおよびコンカレント要求の概要](#)」(3-1)
- 「[要求セットおよび所有者](#)」(3-9)
- 「[要求セットの非互換性](#)」(3-11)

要求セットの非互換性

要求セットは、実際にはその要求セット内の個々のプログラムを実行するよう要求を送るコンカレント・プログラムです。非互換性ルールによって要求セットを管理すると、自分の要求セットが別のレポートまたはコンカレント・プログラムと同時に実行されないようにすることができます。これらのルールは、要求セットを作成するステージにも適用できます。

要求セット・コンカレント・プログラムについて問い合わせたり、それらのプログラムまたはステージ、あるいはその両方を自分の要求セットとは非互換としてリストに定義するには、「コンカレント・プログラム」フォームを使用します。参照:「コンカレント・プログラム」。

要求セットを実行するすべてのコンカレント・プログラムには、要求セット<要求セット名>というタイトルが付きます。「コンカレント・プログラム」フォームで、プログラムの名前に基づいて要求セットのコンカレント・プログラムを問い合わせる場合は、「名前」フィールドに次の語句を入力する必要があります。

コンカレント・プログラムの名前の前に「要求セット」

すべての要求セット・プログラムについて問合せを実行するには「要求セット %」

使用する要求セットとは非互換であるとしてプログラムをリストに登録すると、そのプログラムは、その要求セットまたは要求セット内のすべてのレポートと同じ衝突ドメイン内では同時に実行できません。参照:「[プログラム非互換性ルールの定義](#)」(3-21)。

関連項目

- 「コンカレント・プログラム」ウィンドウ
- 「[コンカレント・プログラムおよびコンカレント要求の概要](#)」(3-1)
- 「[システム管理者の要求セット権限](#)」(3-10)
- 「[プログラム非互換性ルールの定義](#)」(3-21)

要求セット内のパラメータの共有

パラメータは引数とも呼ばれ、プログラムの実行の側面を定義する値です。パラメータとその入力された値は、要求セット内の一部または全部の要求の間で共有できます。

パラメータを共有パラメータとして識別するには、パラメータにラベルを付けます。その後、要求セット内の個々のコンカレント・プログラムについて、そのプログラムのパラメータに同じラベルを割り当てることができます。要求セット内のプログラム間で、個々のプログラムのパラメータは共通の値を共有するか、もしくは受け入れます。

共有パラメータに最初に値を入力した時点で、その値がその共有パラメータ値になります。このことは、値を要求セット内のプログラムごとでなく、1回のみ入力すれば済むので便利です。

共有パラメータの動作

共有パラメータの値を選択すると、その値が以降のそのパラメータの出現時のデフォルトになります。共有パラメータの値を変更すると、その値がそれ以降のそのパラメータの出現時の新しいデフォルトになりますが、このセット内の以前の要求は影響を受けません。

すべての共有パラメータに値が入った後、ある共有パラメータの値を変更しても、それ以外の共有パラメータは影響を受けません。

注意：共有パラメータを非表示にしないでください。共有パラメータを「Display = No」（これによって値を変更できなくなります）または「Modify = No」に設定しないでください。これによって、共有パラメータが更新されても、その更新はセット内の別のレポートへ反映されず、矛盾が生じません。

図 4

Example Settings for a Shared Parameter

Parameters	#1	#2	#3	#4
Report 1	No ← 1	No	No	No
Report 2	No	Yes ← 2	Yes	Yes ← 4
Report 3	No	Yes	No ← 3	No
Report 4	No	Yes	No	No

- 1 Selecting "No" for the first report defaults "No" in subsequent requests.
- 2 Selecting "Yes" for the second report, after selecting "No" for the first report, defaults "Yes" in subsequent reports, but does not change the first report (prior reports).
- 3 Selecting "No" for the third report, after selecting "Yes" for the second report, after selecting "No" for the first report, defaults "No" in subsequent reports, but does not change the first or second reports (prior reports).
- 4 Once all the reports parameters contain values, updating a shared parameter does not update the values in either subsequent or prior reports. For example, selecting "No" for the first report and navigating through all the parameter pop-up windows provides the "No" value for all of the shared parameters. Selecting "Yes" afterwards for the second report does not update the first, third, or fourth reports.

例 - 共有パラメータ値

これまでに、「コンカレント・プログラム・レポート」および「コンカレント・プログラム詳細レポート」という2つのレポートが格納された要求セットを作成しました。これら2つのレポートとそのパラメータを次に示します。

レポート	パラメータ
コンカレント・プログラム・レポート	アプリケーション名
コンカレント・プログラム詳細レポート	アプリケーション名
	プログラム

表 2

ここでは、「アプリケーション名」パラメータを2つのレポートが共有するパラメータとして識別します。値は1回のみ、つまりセット内の最初のレポートについて「レポート・パラメータ」ウィンドウが表示され、「アプリケーション名」の入力を求めてきたときに入力するものとします。

共有パラメータを識別するため、共有パラメータに名前（この例では applname）を付け、個々のレポートの「共有パラメータ」として入力します。

図 5

Report Parameters						
Sequence	Prompt	Display	Modify	Shared Parameter	Type	Default Value
1	Application Name	Yes	Yes	applname		

図 6

Report Parameters						
Sequence	Prompt	Display	Modify	Shared Parameter	Type	Default Value
1	Application Name	Yes	Yes	applname		
2	Program	Yes	Yes			

関連項目

- 「コンカレント・プログラム」ウィンドウ
- 「コンカレント・プログラムおよびコンカレント要求の概要」(3-1)
- 「要求パラメータの動作の制御」(3-44)

要求セット・レポート

このレポートは、セットの所有者、プログラムの非互換性、プリンタと印刷形式の情報などの要求セット定義が文書化されています。このレポートは、要求セット定義を定義するとき、または編集するときに使用します。

レポート・パラメータ

なし。

レポート見出し

レポート見出しには、レポートの内容についての一般情報が出力されます。

関連項目

- 「コンカレント・プログラム・レポート」(3-53)
- 「コンカレント・プログラムおよびコンカレント要求の概要」(3-1)
- 「プログラムの要求セット別編成」(3-3)

プログラムの要求グループ別編成

ここでは、アプリケーション・プログラムとレポートを要求グループ別に編成する方法を説明します。ここでは、次のトピックを取り上げます。

- 要求セキュリティ・グループ
- 要求グループでのコードの使用
- コードを使用した「要求の発行」ウィンドウのカスタマイズ
- レポート・グループ職責レポート

要求グループの定義

要求グループを定義する場合、次のものを含めることができます。

- アプリケーションが所有するすべてのレポートおよびコンカレント・プログラム
- 個々のレポートおよびコンカレント・プログラム
- アプリケーション・ユーザーの要求セキュリティ・グループから選択できるレポートおよびコンカレント・プログラムを集めた要求セット
- 要求セット内のステージのステータスを計算するのに使用する要求セット・ステージ機能

要求グループの2つのタイプ

Oracle Applications では、次の2つのレベルで要求グループを使用します。

1. 職責レベル

要求グループが**職責**へ割り当てられた場合、その要求グループは要求セキュリティ・グループと呼ばれ、そのグループはその職責の下で操作するユーザーが「要求の発行」ウィンドウから選択できるレポート、要求セットおよびコンカレント・プログラムを定義します。

2. フォーム・レベル

要求グループが**コード**へ割り当てられた場合、そのコードを「要求の発行」ウィンドウへパラメータとして渡すことができます。そのコードは、「要求の発行」ウィンドウをコールする機能を定義するのに役立ちます。

このような「要求の発行」ウィンドウの値リストは、要求グループ内にあるレポート、要求セットおよびコンカレント・プログラムのリストです。

要求セキュリティ・グループ

要求グループが職責に割り当てられると、それは要求セキュリティ・グループになります。ある職責の下でサイン・オンしたユーザーは、その職責の要求セキュリティ・グループに含まれるレポートおよびコンカレント・プログラムを実行できます。

「要求の発行」標準発行フォームには、現行の職責の要求セキュリティ・グループ内にあるすべてのレポートおよびプログラムの一覧が表示されます。

関連項目

- 「[要求グループ](#)」ウィンドウ (3-53)
- 「[コードを使用した「要求の発行」ウィンドウのカスタマイズ](#)」 (3-17)
- 「[レポート・グループ職責レポート](#)」 (3-20)
- 「[要求グループでのコードの使用](#)」 (3-17)

要求グループでのコードの使用

通常、メニューから標準要求発行フォームをコールした場合、そのフォームで現在の職責のレポート・セキュリティ・グループに入っているレポートとコンカレント・プログラムの一覧を表示できます。

あるいは、要求グループへコードを割り当てることにより、カスタマイズした「標準発行」フォームに、その特定の要求グループに入っているコンカレント・プログラムの一覧のみを表示することもできます。要求グループ・コードは、メニューからカスタマイズした「標準発行」フォームへ渡す引数です。要約すると、次のとおりです。

- 要求グループ・コードを使用すると、コンカレント・プログラムおよびレポートへのユーザー・アクセスをフォーム・ベースの方法で制御できます。
- 要求グループへコードを割り当てることができます。
- そのコードを、メニューから「標準発行」フォームへ渡す引数として使用できます。
- 「標準発行」フォームをコールするメニューでコードを使用した場合、そのフォームには、そのコードによって識別される要求グループに入っているプログラムの一覧のみが表示されます。

関連項目

- 「要求の発行」ウィンドウのカスタマイズ (3-18)
- 「要求グループ」ウィンドウ (3-53)
- 「コードを使用した「要求の発行」ウィンドウのカスタマイズ」 (3-17)
- 「プログラムの要求グループ別編成」 (3-15)

コードを使用した「要求の発行」ウィンドウのカスタマイズ

「要求の発行」ウィンドウに別のタイトルを付けてフォームを定義することにより、あるコードを割り当てた要求グループに所属するレポートまたはコンカレント・プログラムのみをユーザーが選択できます。そのためには、「要求の発行」ウィンドウを参照するフォーム機能を登録し、その機能へ特定の引数を渡します。その後、そのフォーム機能を組み込んだメニューを作成します。参照：「[メニュー](#)」ウィンドウ (1-30)。

要求グループ・コードを引数として使用

次の表は、要求グループを「要求の発行」ウィンドウへ関連付け、そのフォームのタイトルをカスタマイズするために渡すパラメータの説明です。テキストは、「フォーム機能」フォームの「パラメータ」フィールドに入力します。

「要求の発行」ウィンドウへパラメータを渡す

後に例のあるパラメータ構文	説明
REQUEST_GROUP_CODE =" 要求グループ・コード" REQUEST_GROUP_CODE ="OE_CONC_PROGRAMS"	このパラメータは、要求グループのコードを渡すパラメータです（必須）。
REQUEST_GROUP_APPL_SHORT_NAME =" アプリケーション短縮名" REQUEST_GROUP_APPL_SHORT_NAME ="OE"	このパラメータは、要求グループに関連付けられているアプリケーションの短縮名を指定するパラメータです（必須）。
TITLE ="Application_short_name:Message_Name" TITLE ="FND:SRS_NEWTITLE"	このパラメータは、メッセージのタイトルとアプリケーション短縮名を定義するという内容のメッセージを指定するパラメータです（オプション）。
LOOKUP ="Y N" LOOKUP ="Y"	このパラメータは、TITLE パラメータがメッセージ名またはハードコード化された文字列のいずれであるかを示すパラメータです。デフォルト値は「Y」で、これは TITLE がメッセージ名であることを示します（オプション）。
表 3	

関連項目

- [「要求の発行」ウィンドウのカスタマイズ](#) (3-18)
- [「要求グループ」ウィンドウ](#) (3-53)
- [「プログラムの要求グループ別編成](#) (3-15)
- [「レポート・グループ職責レポート](#) (3-20)
- [「要求グループでのコードの使用](#) (3-17)

「要求の発行」ウィンドウのカスタマイズ

「要求の発行」ウィンドウは、いくつかの方法でカスタマイズできます。

ウィンドウ・タイトルの改名

タイトルを変更し、ウィンドウ内で使用可能な要求を反映させることができます。参照：
[「コードを使用した「要求の発行」ウィンドウのカスタマイズ](#) (3-17)。

ある要求グループから使用可能な要求の制限

指定した要求グループに属するユーザーが使用できるレポートおよびプログラムを制限できます。参照:「コードを使用した「要求の発行」ウィンドウのカスタマイズ」(3-17)。

要求の単一の要求への制限

プログラム / セット名をパラメータとして渡すことにより、単一の要求の発行についての「要求の発行」フォームをコールできます。

program/report_set 名でコールしたフォームへナビゲートすると、パラメータ・ウィンドウがポップ・アップします。発行した要求について表示される要求 ID をユーザーが確認すると、そのフォームは終了します。

要求の要求リストへの制限

プログラム / セット名をパラメータとして渡すことにより、単一のプログラム / セットについて1つまたは複数の要求の発行「要求の発行」フォームをコールできます。

このフォームへナビゲートするとパラメータ・ウィンドウがポップ・アップし、ユーザーはパラメータとして渡したプログラムについて1つまたは複数の要求を発行できます。この場合、他のプログラムについての要求は発行できません。

値セット・パラメータで使用されたパラメータの受け渡し

「要求の発行」フォームへ追加パラメータを渡し、それらのパラメータを値セットの中で参照して要求パラメータの妥当性をチェックできます。

製造「ORG」パラメータの受け渡し

ORGに関連した5つのパラメータを渡して、それらのパラメータを値セットの中で参照できます。あるいは、値セットの中で参照されるORGパラメータに値を提供する「要求の発行」フォームへナビゲートしたときにORG LOVを呼び出すこともできます。

すべての発行要求パラメータを示した完全な一覧

「要求の発行」/SRS フォームがサポートするすべてのパラメータの一覧およびそれらの使用方法に関する追加情報を次に示します。

- REQUEST_GROUP_CODE
- REQUEST_GROUP_APPL_SHORT_NAME (REQUEST_GROUP_CODE と一緒に使用する)
- CONCURRENT_PROGRAM_NAME
- PROGRAM_APPL_SHORT_NAME (CONCURRENT_PROGRAM_NAME と一緒に使用する)

- REQUEST_SET_NAME
- SET_APPL_SHORT_NAME (REQUEST_SET_NAME と一緒に使用する)
- SUBMIT_ONCE (デフォルトは「N」)
SUBMIT_ONCE は Y または N に設定できます (N がデフォルト)。
SUBMIT_ONCE は、CONCURRENT_PROGRAM_NAME または
REQUEST_SET_NAME と結合して使用します。
SUBMIT_ONCE を Y に設定した場合、フォームは「発行」ボタンをクリックすると終了します。
- TITLE
- LOOKUP (デフォルトは「N」)
- USE_ORG、ORG_ID、ORG_NAME、ORG_CODE、CHART_OF_ACCOUNTS_ID (5 つのパラメータ)
USE_ORG を「Y」に設定した場合 (デフォルトは「N」)、「要求の発行」フォームはその他の ORG パラメータが設定されているかどうかをチェックします。このパラメータを設定しない場合、「要求の発行」フォームはグローバル (GLOBAL.FND_ORG_ID、GLOBAL.FND_ORG_NAME など) からパラメータを移植しようとします。グローバルがまだ設定されていない場合、ORG LOV が表示され、パラメータとグローバルの両方は LOV から値を得ます。
値のセットは、常にグローバルでなくパラメータを参照してください。
- CHAR1、CHAR2、CHAR3、CHAR4、CHAR5
- DATE1、DATE2、DATE3、DATE4、DATE5
- NUMBER1、NUMBER2、NUMBER3、NUMBER4、NUMBER5
使用する値セットの中で、これらのパラメータを次のように参照します。
:PARAMETER.CHAR1, :PARAMETER.DATE1, :PARAMETER.NUMBER1 etc.

レポート・グループ職責レポート

このレポートは、レポートまたは要求セットへアクセスできる職責のリストを表示します。このレポートは、レポートおよび要求セットを要求セキュリティ・グループへ割り当てるか、要求セットへ所有者を割り当てることにより、レポートおよび要求セットへのアクセス権限を付与するときに使用します。

レポート・パラメータ

アプリケーション名

レポートまたは要求セットへ関連付けられているアプリケーション名を選択します。

レポート名 / 要求セット名

レポートまたは要求セットの名前を選択します。

関連項目

- 「「要求グループ」ウィンドウ」(3-53)
- 「コンカレント・プログラムおよびコンカレント要求の概要」(3-1)
- 「プログラムの要求グループ別編成」(3-15)

プログラム非互換性ルールの定義

ここでは、コンカレント・プログラムおよびレポート用の非互換性ルールを定義する方法について説明します。

非互換プログラムおよび単独実行プログラム

あるコンカレント・プログラムと他のプログラムとの間に互換性がない場合は、2つのプログラムは同じデータを同時にアクセスまたは更新できません。

コンカレント・プログラムを定義するときに、そのプログラムと非互換にするプログラムのリストを作成できます。また、そのプログラム自身と互換性がないプログラム、すなわち2つのインスタンスを同時に実行できないプログラムのリストも作成できます。

プログラムを単独実行と定義することにより、他のすべてのコンカレント・プログラムと非互換にすることもできます。

コンカレント・プログラムを、単独実行プログラムとして、または特定のコンカレント・プログラムとの互換性がないものとして定義するには、「コンカレント・プログラム」ウィンドウを使用して、そのコンカレント・プログラムの定義を編集します。参照:「コンカレント・プログラム」。

プログラムの非互換性と単独実行プログラムの定義は、衝突ドメインを使用して行います。

要求セット - 非互換可能

非互換可能な要求セットまたは要求セット・ステージを定義するときに、入力した指示に従って要求セットまたはステージ内のレポートを実行するコンカレント・プログラムを作成します。「コンカレント・プログラム」ウィンドウで、要求セットと互換性のないプログラ

ムをリストすると、それらのプログラムは、セットまたはステージ内のすべてのレポートの実行が完了するまで、起動できなくなります。

要求セットおよび要求セット・ステージ用の非互換性ルールを定義する手順は、次のとおりです。

- 要求セットの場合、「要求セット」ウィンドウで「非互換の許可」チェックボックスをチェックします。
- 要求セット・ステージの場合、「ステージ」ウィンドウで「非互換の許可」チェックボックスをチェックします。
- 「コンカレント・プログラム」フォームの「非互換プログラム」ブロックにナビゲートし、要求セットまたはステージとの間に互換性を持たせないプログラムのリストを表示します。

要求セットを実行するすべてのコンカレント・プログラムには、要求セット<要求セット名>というタイトルが付きます。一方、要求セット・ステージを実行するすべてのコンカレント・プログラムには、要求セット・ステージ<ステージ名>-要求セット<要求セット名>というタイトルが付きます。「コンカレント・プログラム」フォームで、プログラムの名前に基づいて要求セットまたはステージのコンカレント・プログラムを問い合わせる場合は、「名前」フィールドに次の語句を入力する必要があります。

- コンカレント・プログラム名の前に「要求セット」または「要求セット・ステージ」
- すべての要求セットおよびステージのプログラムについて問合せを実行するには「要求セット %」

関連項目

- 「コンカレント・プログラム」ウィンドウ
- 「データ・グループ」ウィンドウ (3-67)
- 「コンカレント・プログラムおよびコンカレント要求の概要」 (3-1)
- 「互換性がないプログラムのリストの修正」 (3-43)
- 「要求セットの非互換性」 (3-11)

コンカレント衝突ドメイン

2つのプログラムの間に互換性がないことを定義する場合は、これらのプログラムが同時にアクセスできないデータも指定する必要があります。

つまり、2つのプログラムが同じデータを同時にアクセスまたは更新できないようにするには、どこで、すなわちどのデータについてそれらのプログラムの間に互換性がないのかを知る必要があります。衝突ドメインは、2つの非互換プログラムの同時実行ができないデータを識別します。

衝突ドメイン

Oracle Applications では、データは特定のアプリケーションに属するデータベース表に格納されます。各表には、個々のレコードへのアクセスに必要な条件を決定するための情報も含まれます。このような条件は、次の1つまたは複数のデータ・グループからなります。

- 会計帳簿 - プロファイル・オプション GL_SET_OF_BOOKS に基づく
- 複数のインストレーション (MSOB と呼ばれる)
- 複数の営業単位 (プロファイル・オプション MO_OPERATING_UNIT により決定) (MULTIORG と呼ばれる)
- 複数の組織 (プロファイル・オプション INV_ORGANIZATION_ID で決定し、Manufactureing Applications で使用)
- 衝突解決ドメインとしてビジネス・グループを使用する場合のある HR
- FA 台帳を使用する場合のある FA
- その他 ...

衝突ドメインは、データのパーティション化に使用するグループ化の抽象表現です。ドメインはいくつでも定義できますが、ドメインが多すぎるとパフォーマンスが低下します。

すべてのプログラムは、発行された時点で衝突ドメインが割り当てられます。ドメインをパラメータの一部として定義すると、コンカレント・マネージャはこのドメインを使用して、非互換を解決します。ドメインがパラメータによって定義されていない場合、コンカレント・マネージャはプロファイル・オプション「コンカレント：衝突ドメイン ...」に定義された値を使用します。また、プログラム・パラメータにドメインの指定がなく、「コンカレント：衝突ドメイン ...」プロファイル・オプションも定義されていない場合は、「標準」ドメインが使用されます。標準ドメインはすべての要求に対するデフォルトです。

プロファイル・オプション「コンカレント：衝突ドメイン ...」に値が定義されておらず、プログラム・パラメータにも衝突ドメインが定義されていない場合は、すべてのプログラムで標準衝突ドメインが使用されます。

発行された各要求では、パラメータによって、アクセスするレコードが識別されます。非互換性ルールで定義されたプログラムの場合は、その他のパラメータ（衝突ドメイン・パラメータ）が使用されます。衝突ドメインは、作業するユーザーのログイン ID、会計帳簿または組織などのような変数に基づいて、自動的に設定できます。場合によっては、「要求の発行」フォームのパラメータ・フィールドで、衝突ドメイン・パラメータを選択することもできます。衝突ドメイン・パラメータが定義されると、衝突解決マネージャ (CRM) は、同一ドメイン内で非互換プログラムが同時に実行されないように、そのドメインを使用します。

非互換性ルールの適用

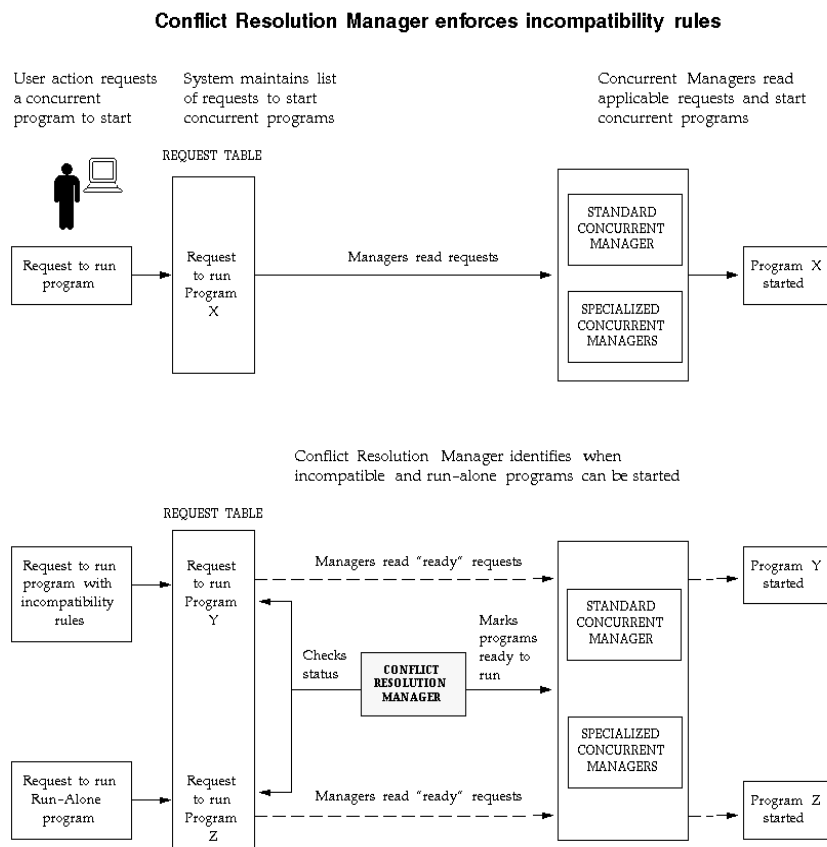
コンカレント・マネージャは、コンカレント・プログラムの実行開始要求を読み取ります。衝突解決マネージャは、コンカレント・プログラム定義を非互換性ルールに照らしてチェックします。

プログラムが単独実行と識別される場合は、コンカレント・マネージャが同じ衝突ドメイン内の他のプログラムを起動できないように衝突解決マネージャが制御します。

あるプログラムについて、そのプログラムと互換性のない他のプログラムが示されている場合は、衝突解決マネージャは、同一ドメインのすべての非互換プログラムが実行を完了するまで、そのプログラムを起動させません。

次の図は、プログラム非互換性ルールを適用するときの衝突解決マネージャの役割を示しています。

図 7



関連項目

- 「コンカレント・プログラム」ウィンドウ
- 「コンカレント・プログラムおよびコンカレント要求の概要」 (3-1)
- 「データ・グループの定義」 (3-25)
- 「プログラム定義のコピーと変更」 (3-41)
- 「互換性がないプログラムのリストの修正」 (3-43)

データ・グループの定義

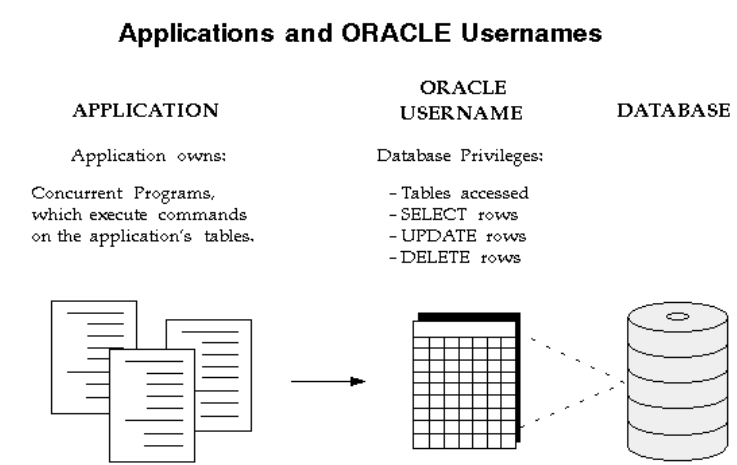
ここでは、データ・グループを定義する方法について説明します。この定義によって、アプリケーション・データベースの接続が指定されます。

データ・グループの概要

データ・グループは、Oracle Applications および各アプリケーションに割り当てられている Oracle ユーザー名のリストです。データ・グループ内のアプリケーションには、それぞれ Oracle ユーザー名を割り当てる必要があります。1つのデータ・グループの中では、アプリケーションは一度しかリストできません。

Oracle ユーザー名およびパスワードを使用することにより、Oracle データベース内の特定アプリケーションの表にアクセスできます。データ・グループ内の各 Oracle ユーザー名によって、対応するアプリケーションがアクセスできるデータベース表および表権限が定義されます。

図 8



データ・グループの目的

職責には、それぞれデータ・グループが1つずつ関連付けられています。データ・グループには2つの使用目的があります。

1. データ・グループは、それに対応する職責を選択したときに、フォームが接続される Oracle ユーザー名を識別します。
2. コンカレント・マネージャは、データ・グループを使用して、レポートまたはコンカレント・プログラム（該当の職責のユーザーが発行したもの）を所有するアプリケーションを、Oracle ユーザー名に結び付けます。

関連項目

- 「Oracle ユーザー・ウィンドウ」(7-3)
- 「「データ・グループ」ウィンドウ」(3-67)
- 「コンカレント・プログラムおよびコンカレント要求の概要」(3-1)
- 「データ・グループの使用」(3-26)
- 「データ・グループの修正」(3-28)

データ・グループの使用

複数の会計帳簿を持つデータ・グループの使用

データ・グループを使用することにより、複数の会計帳簿をサポートする Oracle Applications 製品（たとえば Oracle Payables など）の複数のインストレーションをサポートして、各会計帳簿にそれぞれ異なるアプリケーションを関連付けることができます。参照：「[図 9](#)」(3-27)。

たとえば、Oracle Payables の2つのインストレーションがあり、2つの会計帳簿がサポートされている場合には、データ・グループを使用して、特定の General Ledger 職責からどの Oracle ユーザー名の Oracle Payables にアクセスするのかを指示できます。

各アプリケーション・インストレーション（会計帳簿）についてデータ・グループを1つずつ定義します。

各アプリケーション・インストレーション（会計帳簿）について職責を1つずつ定義し、各職責に適切なデータ・グループを割り当てます。

データ・グループを使用してカスタム・アプリケーションを組み込む

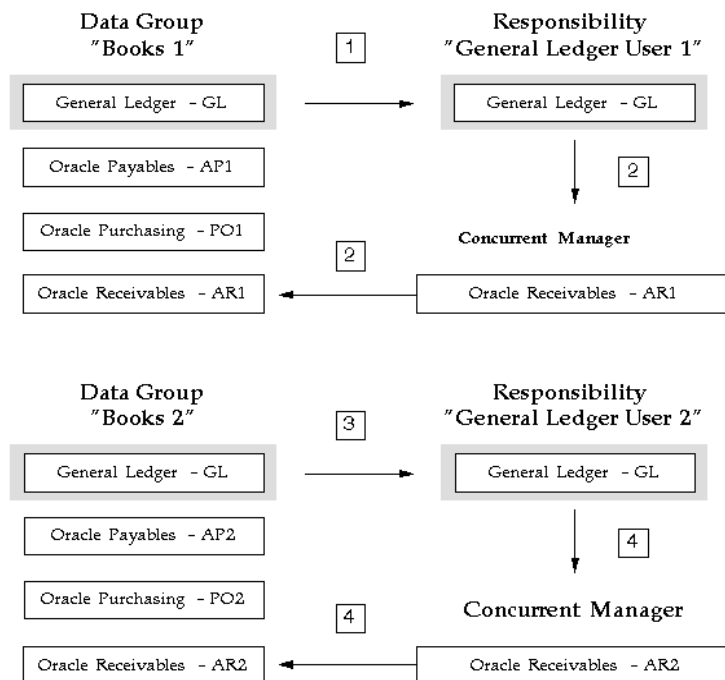
データ・グループを使用することにより、Oracle の Application Object Library を使用して開発したカスタム・アプリケーションを組み込みます。Oracle Applications にカスタム・ア

アプリケーションを組み込むには、「アプリケーション」ウィンドウを使用してそのアプリケーションを登録する必要があります。参照：「[アプリケーション・ウィンドウ](#)」(7-8)。

次に、2つのデータ・グループを使用して、Oracle Payables、Oracle Purchasing および Oracle Receivables の内の2つのインストレーションをサポートする例を示します。

図 9

Using Data Groups with multiple product installations.



- 1 Data Group Books 1 is assigned to Responsibility General Ledger User 1.
- 2 A user of responsibility General Ledger User 1 submits an Oracle Receivables report, which runs accessing data with ORACLE ID AR1 privileges.
- 3 Data Group Books 2 is assigned to Responsibility General Ledger User 2.
Note: both responsibilities attach to the same set of forms (same ORACLE ID).
- 4 A user of responsibility General Ledger User 2 submits an Oracle Receivables report, which runs accessing data with ORACLE ID AR2 privileges.

参照：

[「データ・グループの定義」 \(3-25\)](#)

[「データ・グループの修正」 \(3-28\)](#)

[「「データ・グループ」ウィンドウ」 \(3-67\)](#)

[「Oracle ユーザー・ウィンドウ」 \(7-3\)](#)

データ・グループの修正

事前定義済みの標準データ・グループ

Oracle Applications のインストールまたはアップグレード時に、インストールする各アプリケーションと Oracle ユーザー名をペアにするための標準データ・グループが定義されます（注意：これは個々の会計帳簿ごとに行われます）。

標準データ・グループ内で「アプリケーション」または「Oracle ユーザー名」用に事前定義されている値は、変更も削除もできません。しかし、Oracle ユーザー名の Tool とその記述の変更、またはアプリケーションと Oracle ユーザー名の新しいペアのグループへの追加はできます。

新しいデータ・グループの定義

Oracle Applications 用のデータ・グループはインストール・プロセスで自動的に定義されるので、ユーザーによる定義が必要になるのは、ユーザーが利用する追加のデータ・グループがある場合のみです。

あるデータ・グループをコピーし、それに新しい名前を付けることによって、新しいデータ・グループを作成できます。各アプリケーションおよびそれに割り当てられている Oracle ユーザー名の他に、その Oracle ユーザー名の Tool とその記述があれば、それらも新しいデータ・グループにコピーされます。

提案：標準データ・グループのバックアップ・コピーを作成してください。ただし、それを職責に割り当てないでください。このようにしておけば、アプリケーションにうっかり誤った Oracle ユーザー名を接続したり、アプリケーションの構成の記録を失ったりしても、初期構成に戻すことができます。

データ・グループにカスタム・アプリケーションを追加

Oracle Application Object Library を使用して開発したカスタム・アプリケーションをデータ・グループに組み込むには、以下のとおりにします。

- 「アプリケーション」フォームを使用して、そのアプリケーションを Oracle Applications に登録します。

- 「Oracle ユーザー名」 フォームを使用して、そのアプリケーションに Oracle ユーザー名を割り当てます。

Oracle ユーザー名の登録

Oracle Applications に Oracle ユーザー名を登録すると、Oracle Applications を正常に使用するのに必要な Oracle Application Object Library データベース表（フレックスフィールド表、メニュー表など）に対する権限が設定されます。参照：「[アプリケーション DBA の職務の概要](#)」（7-1）。

関連項目

- 「Oracle ユーザー・ウィンドウ」（7-3）
- 「データ・グループ ウィンドウ」（3-67）
- 「データ・グループの使用」（3-26）
- 「データ・グループの定義」（3-25）

カスタム・コンカレント・プログラム

このセクションでは、カスタム・コンカレント・プログラムについての情報をシステム管理者に提供します。カスタマイズしたコンカレント・プログラムを作成するための手順と規則について説明します。

- ログおよび出力ファイル名
- Oracle ツール・コンカレント・プログラム
- Pro*C コンカレント・プログラム
- コンカレント要求の発行（CONCSUB）

カスタム・コンカレント・プログラムの作成の詳細は、『Oracle Applications 開発者ガイド』を参照してください。

開発環境の設定の詳細は、『Oracle Applications 概要』を参照してください。

ログ・ファイル名および出力ファイル名

ログ・ファイルと出力ファイルには、ユーザーがそれらのファイルをオンラインで表示するための特定の名前および場所が必要です。

Oracle Application Object Library ルーチンの `fdpwrtr()` を使用してファイルへ書き込む場合、コンカレント・マネージャは、オペレーティング・システムの命名規則に従って自動的にファイルに名前を付けます。このファイルへの書き込み方法は、完全に移植できます。アプリ

ケーションを別のプラットフォームに移植する場合、ログ・ファイルおよび出力ファイルに別の名前を付けるためにプログラムを書き換える必要はありません。

標準名

ログ・ファイルおよび出力ファイルの標準名は、次のとおりです。

ファイル・タイプ	場所	ファイル名
ログ	デフォルト :\$<PROD>_TOP/ \$APPLLOG + 共通ディレクトリ :\$APPLCSF/\$APPLLOG	l<request ID>.req
出力	デフォルト :\$<PROD>_TOP/ \$APPLOUT + 共通ディレクトリ :\$APPLCSF/\$APPLOUT	デフォルト :<USERNAME>.<request ID> または O<request ID>.out あるいは APPCPNAM の値に応じた user.out
表 4		

この表に示す変数パラメータの値は、次のとおりです。

- <PROD>_TOP - アプリケーションの最上位環境変数。
- <Request ID> - コンカレント要求の識別番号。
- <USERNAME> - コンカレント・プロセスを要求した人のアプリケーション・ユーザー名のうちの 8 文字（大文字）まで。

Oracle ツール・コンカレント・プログラム

PL/SQL, SQL*Plus または Oracle Reports にコンカレント・プログラムを作成する場合は、「コンカレント・プログラム実行ファイル」ウィンドウの「実行ファイル」フィールドで指定した名前とまったく同じ名前を作成対象プログラムに付けます。必要に応じて拡張子も付けてください。

次の表には、これらのプログラムに使用するファイル拡張子と、プログラムの存在場所のディレクトリをリストで示します（これは、データベースに格納されている PL/SQL ストアド・プロシージャには適用されません）。このディレクトリは、カスタム・アプリケーションの最上位ディレクトリである \$<PROD>_TOP の下にあります。

Oracle Reports プログラムによって共有 PL/SQL ライブラリを使用し、カスタム・アプリケーション用に作成したライブラリを組み込む場合、そのライブラリは、カスタム・アプリケーションの最上位ディレクトリの下にある \$APPLPLS ディレクトリに置きます。

ツール	拡張子	ディレクトリ	注釈
SQL*Plus および PL/SQL	.sql	\$APPLSQL	プログラム名は大文字小文字が区別されます。また、Oracle Application Object Library で定義した実行ファイルとまったく同じ名前にする必要があります。
Oracle Reports	.rdf	\$APPLREP	Oracle Application Object Library では、最初に .rdf ファイルが検索されます。.rdf ファイルが検索されない場合は、.rex ファイルが使用されます。プログラム名は大文字小文字が区別されます。また、Oracle Application Object Library で定義した実行ファイルとまったく同じ名前にする必要があります。
SQL*Loader	.ctl	\$APPLBIN	
表 5			

Pro*C コンカレント・プログラム

コンカレント・プログラムを Pro*C で作成する場合、ディレクトリ \$FND_TOP/\$APPLUSR からスケルトン・プログラム EXMAIN.c および EXPROG.c をコピーします。それらのファイルの名前を変更し、SUBROUTINE_NAME をユーザーのサブルーチン名に一括置換します。

EXMAIN.c は、Spawned プログラムで使用されるスケルトンです。EXPROG.c はユーザーのプログラムのロジックで使用されるスケルトンです。このモジュールは Spawned プログラムまたは即時プログラムを作成するのに使用できます。即時プログラムの場合、プログラム・ライブラリに EXPROG.c のコピーを含める必要があります。プログラム・ライブラリの構築の詳細は、次を参照してください。

これらのスケルトンで作成されたプログラムを Spawned コンカレント・プログラムまたは即時コンカレント・プログラムとして使用できます。Spawned プログラムは個別のプロセスとして実行されますが、即時プログラムはコンカレント・マネージャとリンクされて実行されます。

注意：Oracle では、即時コンカレント・プログラムに関する情報を下位方向の互換性についてのみ提供しています。即時コンカレント・プログラムを新規に作成しないよう強くお勧めします。新規の Pro*C コンカレント・プログラム実行ファイルは Spawned プログラムとして定義してください。

実行ファイルの名前の指定

ユーザーのプログラムの実行ファイルに対して、「コンカレント・プログラム実行ファイル」ウィンドウの「実行ファイル」フィールドで指定した名前と同じ名前を指定します。その実行ファイルを使用しているアプリケーションの TOP ディレクトリの下で \$APPLBIN ディレクトリに入れます。

プログラム・ライブラリの構築

新しいプログラム・ライブラリをコンカレント・プログラム・ライブラリの登録フォームで登録し、このライブラリに含めるすべてのプログラムを登録します。次に、「再構築」フィールドに "Yes" と入力し、コミットします。これによって、<PROD>_TOP/\$APPLLIB\$ の下に <Library Name>.c という新規のカatalog・ファイルを作成するための要求が作成されます。この要求の完了後に、<Library Name>.c ファイルをコンパイルしてください。

prgcat.c、prglib.c などのサンプル・プログラム・ライブラリは、\$FND_TOP/\$APPLUSR の下にあります。

提案: 保守を容易にするために、コンカレント・プログラム実行ファイルを Spawned プログラムとして定義します。

C および Pro*C プログラムのコンパイル

カスタム・コードをコンパイルするための環境は、ファイル \$FND_TOP/usrxit/devenv に依存しています。このファイルを変更する場合、変更内容を有効にするにはログインし直してこのファイルを再度読み込む必要があります。

\$FND_TOP/usrxit/Makefile を使用して C または Pro*C プログラムをオブジェクト・モジュールにコンパイルします。次に、adrelink を使用してコンパイルしたプログラムをリンクします。弊社では、単独の makefile またはユーティリティを使用した実行ファイルのコンパイルやリンクはサポートしていません。

C プログラム example.c をコンパイルするには、次の構文を使用します。どの例の場合も、ユーザーのファイルがあるディレクトリからコマンドを実行する必要があります。

```
$ make -f $FND_TOP/usrxit/Makefile example.o
```

Pro*C プログラム proexamp.pc をコンパイルするには、次の構文を使用します。

```
$ make -f $FND_TOP/usrxit/Makefile proexamp.o
```

a.c、b.c、c.pc、d.pc の 4 つの C および Pro*C プログラム（これらのプログラムはすべて現在のディレクトリ内にある）をコンパイルするには、次の構文を使用します。

```
$ make -f $FND_TOP/usrxit/Makefile a.o b.o c.o d.o
```


Spawned コンカレント・プログラムをスタンドアロン・プログラムとしてリンクする

Spawned コンカレント・プログラムをスタンドアロン・プログラムとして実行する場合には、スタンドアロン実行ファイルをコンパイルする前に、次の手順を実行します。

カスタム・コンカレント・プログラムをカスタム・アプリケーションの下に定義する場合（推奨されているとおりに）、サンプル .mk ファイルを \$FND_TOP/usrxit からユーザーの \$<PROD>_TOP/\$APPLLIB ディレクトリにコピーします。コピーしたファイルに含まれている指示に従ってそのファイルを変更します。これは、スタンドアロン実行ファイルをリンクするために adrelink によって使用されるファイルです。

続いて、次のコマンドを入力します。

```
$ . $FND_TOP/fndenv
ソース・ファイルが格納されているディレクトリに移動します。

$ cd <source_directory>
$ make -f $FND_TOP/$APPLLIB/Makefile <source file>.o
```

ここでは、<source file> はプログラムが格納されているファイルの名前のことで、<directory> はソース・ファイルが格納されているディレクトリのことです。

次に、スタンドアロン実行ファイルをリンクして、リンクした実行ファイルを使用しているカスタム・アプリケーションの TOP ディレクトリの下に \$APPLBIN ディレクトリに入れることができます。

```
$ adrelink force=y "<appl_short_name> <program name>"
```

この再リンクコマンドでは、<appl_short_name> はプログラムが属するアプリケーションの短縮名で、<program name> はそのプログラム名です。

即時コンカレント・プログラムのリンク

プログラム・ライブラリを作成するには、Oracle Application Object Library のリンク手順を使用して、コンパイルしたライブラリ・カタログをプログラムのオブジェクト・ファイルにリンクします。

注意：Oracle では、即時コンカレント・プログラムに関する情報を下位方向の互換性についてのみ提供しています。即時コンカレント・プログラムを新規に作成しないよう強くお勧めします。新規の Pro*C コンカレント・プログラム実行ファイルは Spawned プログラムとして定義してください。

環境変数 \$LUSRLIB に即時コンカレント・プログラムとプログラム・ライブラリの関数を定義するモジュールが組み込まれているか確認してください。ライブラリ・カタログのオブ

ジェクト・モジュールを組み込むように \$LUSRPRG 変数を設定します。ディレクトリ \$FND_TOP/\$APPLUSR 内のファイル devenv は、変数 \$LUSRLIB および \$LUSRPRG を定義します。ファイル fndenv は devenv を実行します。

ファイル devenv および fndenv は、必要な環境変数を設定する UNIX シェル・スクリプトです。

新しい即時コンカレント・プログラム・ライブラリが期待通りに機能しない場合に備えて、そのプログラム・ライブラリをリンクする前にそのコピーを作成しておくことをお勧めします。プログラム・ライブラリをリンクするには、次のようにオペレーティング・システムからコマンドを実行します。

```
$ adrelink force=y "fnd UFNDLIBR"
```

これにより、新規のプログラム・ライブラリが UFNDLIBR として作成されます。このライブラリ名は変更できますが、新規のプログラム・ライブラリ名は 8 文字以内にする必要があります。

Pro*C コンカレント・プログラムのテスト

次の方法を使用して、プログラムをテストできます。プログラムで必要とされる各引数を渡す必要があります。パラメータを渡すには、オペレーティング・システムのプロンプトで次のように入力します。

```
$ <program name> <ORACLE username>/<ORACLE password> 0 Y ¥  
[<parameter 1> <parameter 2>... ]
```

プログラム名は、大文字で、「コンカレント・プログラム実行ファイル」ウィンドウの「実行ファイル」フィールドで入力した名前と同じ名前を入力する必要があります。0 および Y 引数は必須です。

プログラム固有のパラメータのいずれかにスペースが入っている場合は、そのパラメータを二重引用符で囲みます。パラメータにリテラル二重引用符が含まれている場合は、そのマークの前に円記号 [¥] を付けます。

ホスト言語のコンカレント・プログラム

プログラムに <name>.prog という名前を付けます。ここで、<name> は、「コンカレント実行ファイル」ウィンドウの「実行ファイル」フィールドに入力した値です。次に、実行ファイル名（拡張子なし）を使用して、\$FND_TOP/\$APPLBIN ディレクトリにある fndcpepr の記号リンクを作成します。実行ファイルおよびリンク・ファイルをアプリケーションの最上位ディレクトリの下にある \$APPLBIN ディレクトリに置きます。

たとえば、カスタム・シェル・スクリプトに CUSTOM.prog という名前を付けます。fndcpepr への記号リンクを CUSTOM という名前で作成します。この 2 つのファイルを \$APPLBIN ディレクトリに置きます。実行ファイル CUSTOM を使用してコンカレント・プログラム実行ファイルを作成します。

ホスト・プログラムのパラメータ

プログラムを実行しているコンカレント・マネージャによって、プログラム名は \$0、4 つの引数 (orauser/pwd、userid、username および request_id) は \$1 ~ \$4、プログラムに固有のパラメータは \$5 に書き込まれます。これらの各引数の長さは、最大 50 文字まで可能です。

たとえば、プログラムに 2 つのパラメータを渡す場合、最初のパラメータを指すには \$5 を使用し、2 番目のパラメータを指すには \$6 を使用します。

Oracle ユーザー・パスワードの保護

場合によっては、Oracle ユーザー名およびパスワードをホスト・プログラムに直接渡すことにセキュリティ上の問題があります。コンカレント・マネージャによってプログラムにユーザー名とパスワードが渡されないようにする場合、ユーザー名とパスワードは環境変数として渡されるようにするか、またはまったく渡されないようにコンカレント・マネージャを制御できます。

最初に、コンカレント・プログラム実行ファイルを、「コンカレント・プログラム実行ファイル」フォームでホスト・プログラムとして定義します。

ユーザー名とパスワードを環境変数として渡すようにするには、この実行ファイルを使用してコンカレント・プログラムを定義するときに「コンカレント・プログラム」ウィンドウの「実行オプション」フィールドに用語 'ENCRYPT' を入力します。'ENCRYPT' と入力すると、コンカレント・プログラムに信号が送られ、これにより、環境変数 fcp_login にユーザー名 / パスワードが渡されます。この引数 \$1 は空白のままにします。

ユーザー名とパスワードをプログラムにまったく渡す必要がない場合は、「実行オプション」フィールドに 'SECURE' と入力します。これにより、コンカレント・マネージャによってユーザー名とパスワードがプログラムに渡されることはなくなります。

成功コード

デフォルトでは、シェル・スクリプトから成功 (ステータス・コード 0) が戻ります。スクリプトがエラーを検出した場合は、UNIX の終了コマンド "exit 1" を使用して、プログラムを実行しているコンカレント・マネージャに失敗 (ステータス・コード 1) を戻します。

コンカレント要求の発行 (CONCSUB)

オペレーティング・システムから CONCSUB ユーティリティを使用してプログラムを発行することにより、コンカレント・プログラムをテストできます。

構文

次の構文で CONCSUB プログラムを実行することにより、コンカレント要求を発行してコンカレント・プログラムを実行できます。

```
$ CONCSUB <APPS username>/<APPS password> ¥
<responsibility application short name> ¥
<responsibility name> ¥
<username> ¥
[WAIT=N|Y|<n seconds>] ¥
CONCURRENT ¥
<program application short name> ¥
<program name> ¥
[PROGRAM_NAME="<description>"] ¥
[REPEAT_TIME=<resubmission time>] ¥
[REPEAT_INTERVAL= <number>] ¥
[REPEAT_INTERVAL_UNIT=< resubmission unit>] ¥
[REPEAT_INTERVAL_TYPE=< resubmission type>] ¥
[REPEAT_END=<resubmission end date and time>] ¥
[START=<date>] ¥
[IMPLICIT=< type of concurrent request>] ¥
[<parameter 1> ... <parameter n>]
```

CONCURRENT パラメータの後にスペースの入ったパラメータがある場合、そのパラメータ引数を二重引用符で囲み、さらに引用符で囲みます。Oracle Application Object Library では、引数文字列を2度解析するために、このような構文が必要となります。たとえば、この引数をプログラムに渡すには、次のようにします。

This is an example

CONCSUB によりこの引数を渡します。

```
"This is an example"
```

例

次は、CONCSUB を実行するコマンドの例です。

```
$ CONCSUB APPS/APPS ¥
SYSADMIN ¥
"System Administrator" ¥
SYSADMIN ¥
WAIT=N ¥
CONCURRENT ¥
FND ¥
FNDFMRTC ¥
PROGRAM_NAME="Register Custom Tables Weekly" ¥
REPEAT_INTERVAL=7 ¥
```

```
REPEAT_INTERVAL_UNIT="DAYS" ¥  
REPEAT_INTERVAL_TYPE="START" ¥  
START="'08-JUN-96 23:55:00"  
CGL  
APPLSYS  
ALL  
CGL
```

パラメータ

次のエントリでは、CONCSUB によりコンカレント・プログラムを発行するための必須パラメータと任意選択パラメータについて説明しています。デフォルト値を右側に示します。

ユーザー名 / パスワード	必須。この ORACLE ユーザー名およびパスワードによってプログラムで使用するデータにアクセスします。
職責アプリケーション短縮名	必須。コンカレント処理オプションを使用する職責のアプリケーション短縮名。
職責名	必須。職責の名前。職責名にスペースが含まれている場合には、その名前を二重引用符で囲みます。
ユーザー名	必須。コンカレント処理オプションを使用するアプリケーション・ユーザーの大文字のユーザー名。
WAIT	<p>任意選択。発行した要求が完了するのを待つかどうかを示すフラグ。このパラメータを省略すると、N のデフォルト値は CONCSUB に要求の完了を待たずにオペレーティング・システム・プロンプトを戻すようにさせます。</p> <p>WAIT=Y と設定して、CONCSUB が 60 秒ごとに要求ステータスをチェックし、要求が完了した時点でオペレーティング・システム・プロンプトを戻すようにさせます。また、WAIT=30 のように、秒数に整数値を入力して、指定した < 数値 > の秒ごとに要求が完了したかを CONCSUB にチェックさせることもできます。</p>

注意： WAIT=Y または WAIT=< 数値 > では、CONCSUB によってオペレーティング・システムに戻される前に要求が完了している必要があります。コンカレント・マネージャが停止していると、CONCSUB プロセスは、コマンドマネージャが起動され、要求が完了するまで無期限に待機します。

CONCURRENT	必須。プログラム固有のパラメータとオペレーティング・システム・パラメータを分けるフラグ。
------------	--

プログラム・アプリケーション短縮名	必須。コンカレント・プログラムのアプリケーション短縮名。
プログラム名	必須。プログラムの大文字の名前。コンカレント・プログラムを定義するときに「コンカレント・プログラム」ウィンドウで入力する短縮名を入力する必要があります。
PROGRAM_NAME	<p>任意選択。プログラムの付加的な名前。要求の表示フォームのプログラム・フィールドでは、これをユーザー・フレンドリな（ユーザーに分かり易い）プログラム名として表示します。CONCSUB に渡されるコンカレント・プログラム短縮名は、多くの場合、エンド・ユーザーにとって分かりにくいものとなっています。したがって、この PROGRAM_NAME パラメータを使用して、推奨されるより簡単な名前をコンカレント・プログラムに渡すことができます。PROGRAM_NAME を指定しないと、要求の表示フォームによって、「コンカレント・プログラム」ウィンドウで指定されるユーザー・フレンドリ名が表示されます。</p> <p>また、PROGRAM_NAME パラメータを使用して、同時にアクティブになっている所定のプログラムに対していくつかの要求がある可能性がある場合、一連のデータを処理するプログラムに対する要求で処理されるバッチを示すこともできます。</p>
REPEAT_TIME	任意選択。要求を再発行する時間。時間の形式は HH24:MI または HH24:MI:SS です。たとえば、REPEAT_TIME=14:30 の場合、要求が毎日午後 2 時 30 分に再発行されます。

注意： REPEAT_TIME を、任意選択パラメータ REPEAT_END および START 以外の他の再発行パラメータと一緒に使用しないでください。

REPEAT_INTERVAL	任意選択。再発行の間隔（正の整数または実数）。このパラメータと REPEAT_INTERVAL_UNIT を一緒に使用して、再発行から次の再発行までの時間を指定します。
-----------------	--

REPEAT_INTERVAL_UNIT 任意選択。再発行の間隔に使用する時間の単位。使用可能な単位として MINUTES、HOURS、DAYS または MONTHS があります。このパラメータと REPEAT_INTERVAL を一緒に使用して、再発行間の時間を指定します。たとえば、REPEAT_INTERVAL=12 および REPEAT_INTERVAL_UNIT=HOURS と設定すると、12 時間ごとに要求が再発行されます。デフォルト値は DAYS です。

注意： REPEAT_INTERVAL および REPEAT_INTERVAL_UNIT を REPEAT_TIME と一緒に使用しないでください。

REPEAT_INTERVAL_TYPE 任意選択。再発行の間隔の開始時を、その要求の要求開始時刻からに設定するか、その完了時からに設定するか。このパラメータは START または END のいずれかに設定します。デフォルト値は START です。

注意： REPEAT_INTERVAL_TYPE を使用するのには、REPEAT_INTERVAL を使用する場合があります。

REPEAT_END 任意選択。コンカレント要求の再発行を停止する日時。終了日の形式として次のいずれかを使用します。

"DD-MON-RR HH24:MI:SS" ("07-APR-01 18:32:05" の場合と同様)

または

"DD-MON-RRRR HH24:MI:SS" ("07-APR-2001 18:32:05" の場合と同様)

この日付形式にはスペースが入るため、その日付を二重引用符と引用符で囲む必要があることに注意してください。また、次のようにして、日付のみを指定することもできます。

'DD-MON-RR'

または

'DD-MON-RRRR'

START 任意選択。この形式でのプログラムの開始日時。

""DD-MON-RR HH24:MI:SS"" (""07-APR-98 18:32:05""
の場合と同様)

この日付形式にはスペースが入るため、その日付を二重引用符と引用符で囲む必要があります。開始時刻を指定しないと、プログラムが即時に発行され、次に使用可能なコンカレント・マネージャによって処理されます。デフォルト値は現在時刻です。

IMPLICIT

任意選択。このコンカレント要求を要求の表示フォームに表示するかどうか。NO、YES、ERROR または WARNING を指定します。値 IMPLICIT=NO と指定すると、その要求を要求の表示フォームに表示できます。デフォルト値は NO です。

値 IMPLICIT=YES は、システム管理者の権限を有するコンカレント要求の表示フォームのみがこの要求を表示することを意味します。その要求がユーザーにとって関心のないものである場合に、この値を使用します。

要求が失敗または警告によって完了する場合にのみその要求を表示させる場合に、IMPLICIT=ERROR または IMPLICIT=WARNING を指定します。

REPEAT_DAYS

任意選択。最後の要求開始日を起点として計算される、コンカレント要求の繰返しまでの経過日数。指定できる数値は、正の整数または実数のいずれかです。たとえば、REPEAT_DAYS=1.5 と指定すると、要求が 36 時間ごとに再発行されます。

注意： REPEAT_DAYS を任意選択パラメータ REPEAT_END や START 以外の他の再発行パラメータと一緒に使用しないでください。

提案： REPEAT_DAYS は今後のリリースでは使用されなくなります。したがって、REPEAT_DAYS の代わりに、REPEAT_INTERVAL、REPEAT_INTERVAL_TYPE および REPEAT_INTERVAL_UNIT を使用することが必要です。

parameter 1 ... parameter*n*

任意選択。プログラム固有のパラメータ。パラメータにスペースが入っている場合は、そのパラメータを二重引用符で囲んでから、さらに、引用符で囲みます。パラメータにその引数の一部として二重引用符が含まれている場合は、その二重引用符の前に円記号 [¥] を付けます。

プログラム定義のコピーと変更

次の各節では、コンカレント・プログラムの定義内容のコピーと変更の方法について説明します。

警告：既存のコンカレント・プログラムの定義に上書きしないでください。既存のプログラムを変更する場合は、プログラムをコピーし、プログラム名を変更してから、必要な変更を行ってください。

関連項目

- 「表 8」(3-49)
- 「「コンカレント・プログラム」ウィンドウ」
- 「コンカレント・プログラムおよびコンカレント要求の概要」(3-1)
- 「コンカレント・プログラムのコピーとプログラム名の変更」(3-41)
- 「コンカレント・プログラムのパラメータ」(3-43)
- 「プログラムのパラメータの変更例」(3-50)
- 「プログラムの優先順位の変更」(3-42)
- 「プログラム非互換性ルールの定義」(3-21)
- 「互換性がないプログラムのリストの修正」(3-43)

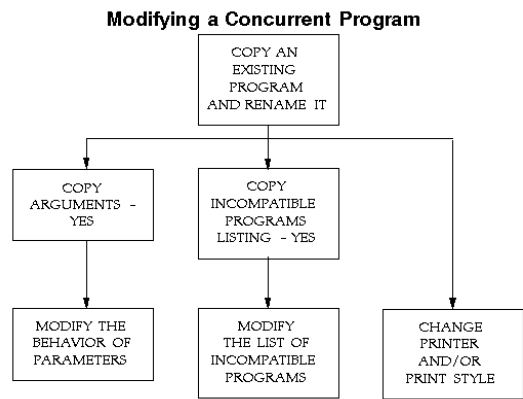
コンカレント・プログラムのコピーとプログラム名の変更

コンカレント・プログラムをコピーし、そのコピーを変更して、ニーズに適合した定義内容の新規プログラムを作成できます。次に示すプログラムの定義内容を変更することにより、コンカレント・プログラムの動作方法を変更できます。

- 非互換プログラム
- パラメータ（引数）
 - パラメータ値セット
- プリント、印刷形式など

コンカレント・プログラムの定義内容に上書きするかわりに、既存のプログラムをコピーしてプログラム名を変更することによってプログラムをカスタマイズし、ニーズに合うように新規プログラムを変更してください。次の図では、新しいコンカレント・プログラムをコピー、変更するための基本手順を示しています。

図 10



関連項目

- 「コンカレント・プログラム」ウィンドウ
- 「[コンカレント・プログラムおよびコンカレント要求の概要](#)」 (3-1)
- 「[プログラムのパラメータの変更例](#)」 (3-50)
- 「[プログラム定義のコピーと変更](#)」 (3-41)

プログラムの優先順位の変更

ユーザー・レベルではなく、プログラム・レベルで一部の要求の優先順位を制御する場合があります。

プログラムの優先順位を設定することにより、要求を発行したユーザーの優先順位ではなく、指定した優先順位でコンカレント・プログラムを実行する要求を発行できます。

たとえば、ユーザー・プロファイルの「コンカレント:優先度」の値によって決定された標準優先順位でユーザーは各種の要求を発行しますが、特定のコンカレント・プログラムに対する要求がユーザーから発行されたとき、この要求の優先順位を高くする場合があります。

その場合、プログラムに 10 の優先順位を割り当てます。ユーザーがそのプログラムの実行を要求すると、ユーザーの標準優先順位ではなく、「コンカレント・プログラム」ウィンドウで定義された高い優先順位がそのプログラムに割り当てられ、その他の要求よりも先に処理されます。優先順位の指定がないコンカレント・プログラムの実行をユーザーが要求すると、この要求ではユーザーの「コンカレント:優先度」のプロファイル値が使用されます。

関連項目

- 「「コンカレント・プログラム」ウィンドウ」
- 「プログラム定義のコピーと変更」(3-41)

互換性がないプログラムのリストの修正

コンカレント・プログラムの定義には、非互換プログラムのリストが含まれている場合があります。このリストで、あるプログラムが他のプログラムと非互換と定義されていると、これらの2つのプログラムは同じ衝突ドメイン内で同時には実行できません。参照:「[プログラム非互換性ルールの定義](#)」(3-21)。

「コンカレント・プログラム」ウィンドウの「非互換プログラム」ブロック内で、どのプログラムがコンカレント・プログラムと非互換かを調べることができます。非互換プログラムとしてリストされている各プログラムは、そのコンカレント・プログラムと同じ衝突ドメイン内では同時に実行できません。

非互換プログラム・リストの変更には、次のいずれかが可能です。

新規プログラムをリストに追加します。

「有効範囲」フィールドで次の指定が可能です。すなわち、プログラム自体を非互換にするかどうか、またプログラムとすべての子要求、つまり要求セットの一部としてそのプログラムによって開始される各コンカレント・プログラムを非互換にするかどうかの指定です。

- リストから各プログラムを削除します。

注意:「非互換プログラム」ゾーンで行われた変更内容をただちに有効にするには、「コンカレント・マネージャの管理」ウィンドウにナビゲートし、「内部コンカレント・マネージャ」の「検証」を選択してください。

関連項目

- 「「コンカレント・プログラム」ウィンドウ」
- 「「コンカレント・マネージャの管理」ウィンドウ」(4-71)
- 「プログラム定義のコピーと変更」(3-41)

コンカレント・プログラムのパラメータ

パラメータは引数とも呼ばれ、標準発行のコンカレント・プログラムに割り当てられます。プログラムを標準発行として定義するには、「コンカレント・プログラム」フォーム内の「標準発行」フィールドの値を「Yes」に設定してください。

注意：パラメータのデフォルト設定用のすべての機構（これにはその他のパラメータ、ユーザー・プロファイルなどの値の参照も含まれます）は、発行時にのみ評価されます。

コンカレント・プログラムに対応するパラメータには、値セットと動作という2つの側面があります。

パラメータ値セット パラメータに設定できる有効な値のことを指します。有効な値のセットは、値セットと呼ばれます。

パラメータ動作 アプリケーション内でのパラメータの動作を指します。たとえば、次のような動作です。

- プログラムを実行するのに対象のパラメータに値を入力する必要があるかどうか
- 対象のパラメータをエンド・ユーザーに表示するかどうか
- 対象のパラメータにデフォルト値を自動的に指定するかどうか

「値セット」の定義または変更を行うときには、まず「値セット」の目的とインプリメンテーションを十分に注意して計画してください。

「コンカレント・プログラム」フォームでパラメータを選択することによって、コンカレント・プログラムの各パラメータを見ることができます。各パラメータは、そのパラメータに対して許容される値を定義した「値セット」を持ちます。パラメータの「値セット」の名前を調べるには、「引数詳細」ブロック内の「値セット」フィールドを参照してください。

関連項目

- 「「コンカレント・プログラム」ウィンドウ」
- 「[プログラムのパラメータの変更例](#)」（3-50）
- 「[プログラム定義のコピーと変更](#)」（3-41）
- 「[要求パラメータの動作の制御](#)」（3-44）

要求パラメータの動作の制御

個別に実行中の各プログラム内のパラメータの動作は、それらのプログラムが要求セットの一部として実行される時とは異なる可能性があります。

参照：

「[表 6](#)」（3-48）

「表 7」(3-48)

「コンカレント・プログラム」フォームを使用してプログラムを定義するときに、プログラムの各パラメータの動作方法を定義します。

またプログラムが要求の一部として実行されるときには、「要求セット」フォームを使用して、プログラムの各パラメータの動作方法を定義できます。さらに、要求セット内の異なるプログラムの各パラメータに対して「共有パラメータ」のラベルを付けると、すべてのパラメータが同じ値を共有するように定義できます。参照：「[要求セット内のパラメータの共有](#)」(3-12)。

警告：新規パラメータの追加、既存パラメータの削除またはパラメータの「値セット」の変更によりコンカレント・プログラムの定義を変更すると、プログラムが実行できなくなる恐れがあります。参照：「[表 8](#)」(3-49)。

パラメータの非表示化

「コンカレント・プログラム」フォームまたは「要求セット」フォームを使用して、システム管理者はパラメータをエンド・ユーザーに表示しないように設定できます。表示されないパラメータはユーザーが変更できないため、パラメータを非表示に設定することは次のような特長があります。

- セキュリティの点で優れており、望ましいデフォルト値を確実に使用させることができます。
- プログラムの定義で有効なデフォルトのタイプと値を入力する必要があることを意味しており、またプログラムが要求セットの一部の場合は、要求セットの定義時に有効なデフォルトのタイプと値を入力する必要があることを意味します。

注意：「変更」フィールドを「No」に切り替える前に、必要なパラメータのデフォルト値を設定してください。設定しないと、このレポートを発行したときに、「要求の発行」フォームにエラーが表示されます。

パラメータを非表示と定義すると、「要求の発行」フォームを使用してプログラムを実行したとき、このパラメータは表示されません。また、「要求セット」フォームにも表示されません。

「要求セット」フォームを使用してパラメータを非表示にした場合、プログラムが要求セットの一部として実行されるときに、そのパラメータは「要求の発行」フォームには表示されません。

要求の発行後に表示されたパラメータを見る方法

コンカレント・プログラムを実行するために要求を発行した後、そのプログラムのパラメータは「コンカレント要求」フォームの「詳細」ブロックに表示されます。

パラメータが非表示に設定されているときには、そのパラメータは「コンカレント要求」フォームの「詳細」ブロックには表示されません。

これらの表示されたパラメータの値は、コンカレント・マネージャがコンカレント・プログラムに渡す値と正確に一致していますが、ユーザーが選択した表示値とは対応する場合も、対応しない場合もあります。

たとえば「要求の発行」フォーム内で、Oracle General Ledger をパラメータとして選択できますが、「コンカレント要求」フォームには対応するアプリケーションの ID が表示されます。

提案：プログラムの実行時にユーザーにエラーが出力された場合、システム管理者はコンカレント・プログラムが使用している正確な値を調べて、問題の診断に役立てることができます。

パラメータのデフォルト値の設定

パラメータのデフォルト値は、ユーザーがプログラムまたは要求セットの実行を指示するときに変更できます。

パラメータのデフォルト値は、次のように設定できます。

- 「要求セット」フォーム内の「デフォルト・タイプ」および「デフォルト値」フィールド。ここで設定された値は、「要求セット」フォーム上では変更できません。
- 「要求セット」フォーム内の「デフォルト・タイプ」および「デフォルト値」フィールド。
このデフォルト定義は、プログラムが要求セットの一部として実行されるときのみ適用されます。
- 「要求セット」フォーム内の「共有パラメータ」および「デフォルト値」フィールド。
このデフォルト定義は、プログラムが要求セットの一部として実行されるときのみ適用されます。同じ共有パラメータ・ラベルを持つすべてのパラメータは、「デフォルト値」フィールドで指定した値をデフォルト値として持ちます。

デフォルト値入力のエラー

パラメータの「デフォルト・タイプ」または「デフォルト値」が間違っていると、「要求の発行」フォームを使用してプログラムの実行が設定された時点で、ウィンドウにエラー・メッセージが表示されます。

パラメータが表示されないと、エラー・メッセージが出力されます。表示されていないフィールドは更新できません。

警告: デフォルト・タイプやデフォルト値は、パラメータの「値セット」との検証が行われないため、これらの値を入力するときは注意してください。正しくない値を入力すると、「要求の発行」フォームを使用してこの要求セットを実行したときに、それらの値はデフォルト値として表示されません。

「要求セット」内のパラメータ値の変更禁止

「要求セット」フォームにパラメータが表示され、プログラムの定義でデフォルト値が指定されていない場合、デフォルト値を定義するか共有値をパラメータで継承させることにより、ユーザーがパラメータ値を変更するのを禁止できます。

パラメータの値を表示させ、しかも「標準発行」フォームを使用して要求セットを実行するときに変更を禁止する場合は、「要求セット」内の「変更」フィールドを No に設定してください。パラメータの値は、デフォルト値または共有パラメータを使用して設定できます。

「表示」フィールドが No に設定されると、「変更」フィールドのデフォルト値は自動的に No となり、パラメータ値の更新はできなくなります。

注意: 「変更」フィールドを「No」に切り替える前に、必要なパラメータのデフォルト値を設定してください。設定しないと、このレポートを発行したときに、「要求の発行」フォームにエラーが表示されます。

変更内容を確認するための職責変更

ユーザーにパラメータを表示するかしらないかなど、パラメータの動作の変更はシステム管理者が変更をコミットした直後から有効になります。ただし一部の変更は、職責を変更するか、現在の職責を再び選択しない限り表示されません。

プログラム・パラメータの動作

パラメータ詳細	「コンカレント・プログラム」フォーム	「要求の実行」フォーム
必須	YES	パラメータには値（ユーザーが入力した値またはデフォルト値）が必要です。
表示	YES	パラメータが表示されます。
	NO	パラメータは表示されず、変更もできません。
デフォルト・タイプ & 値	YES - デフォルト・タイプおよび入力した値。	デフォルト値が表示され、その値はユーザーが変更できます。

パラメータ詳細	「コンカレント・プログラム」フォーム	「要求の実行」フォーム
	デフォルト値が入力されていません。	デフォルト値は表示されません。
表 6		

要求セット内の各パラメータの動作

パラメータ詳細	「コンカレント・プログラム」フォーム	「要求セット」フォーム	「要求の実行」フォーム
必須	YES	パラメータに値を指定する必要はありません。	パラメータには値を指定する必要があります。
表示	YES	パラメータを表示します。「表示」は「YES」に設定されました。	パラメータが表示されます。
		パラメータを表示します。「表示」は「NO」に設定されました。	パラメータは表示されません。
	NO	パラメータは表示されません。	パラメータは表示されません。
変更	n/a	YES	値は変更できません。
	n/a	NO	値は変更できません。
デフォルト・タイプ & 値	YES - 入力したデフォルト・タイプおよびデフォルト値。	デフォルト・タイプおよびデフォルト値は変更できません。	デフォルト値を変更できるのはユーザーです。
	デフォルトが入力されていません。	YES - デフォルト・タイプおよびデフォルト値は入力できます。	デフォルト値を変更できるのはユーザーです。
		NO - デフォルト・タイプおよびデフォルト値が入力されていません。	デフォルト値は表示されません。
表 7			

プログラム定義の変更に関する警告

処理	使用されたフォーム	警告
レポート・プログラムでの行数および列数の変更	「コンカレント・プログラム」- 「出力」 リージョン	詳細な出力を生成するために、いくつかのプログラムが作成されます。出力の定義を変更すると、プログラムが実行できなくなるか、または品質の低い結果が出力される恐れがあります。
印刷形式を「動的」に設定	「コンカレント・プログラム」- 「出力」 リージョン- 「形式」 フィールド	動的印刷形式では、可変な出力サイズに基づいて出力を生成するようにプログラムに通知します。動的印刷形式をサポートするには、プログラム内で特殊なコーディングが必要です。
プログラム定義での多数のパラメータの変更	「コンカレント・プログラム」- 「コンカレント・プログラム・パラメータ」 ウィンドウ	プログラムは、 x 個のパラメータ入力を前提に定義されています。新しいパラメータ ($x+1$ 個目) を入力しても、そのパラメータはプログラムによって無視されます。パラメータを削除すると、プログラムが実行できない恐れがあります。
値セットの変更	「コンカレント・プログラム」- 「コンカレント・プログラム・パラメータ」 ウィンドウ- 「検証」 リージョン- 「値セット」 フィールド	プログラムは特定のタイプと長さの値の入力を前提としています。値セットが変更されると、プログラムが動作しない恐れがあります。
トークンの変更	「コンカレント・プログラム」- 「コンカレント・プログラム・パラメータ」 ウィンドウ- 「トークン」 フィールド	プログラムは特定のタイプと長さの値の入力を前提としています。所定のトークンを受け取らなかった場合、プログラムは動作しない恐れがあります。
コンカレント実行ファイルまたはプログラムの実行方法を「即時」として定義	「コンカレント・プログラム実行ファイル」- 「実行方法」 フィールド 「コンカレント・プログラム」- 「実行ファイル」 リージョン- 「方法」 フィールド	「実行方法」が「即時」のコンカレント・プログラムは、必ずプログラム・ライブラリ FNDLIBR に登録する必要があります。プログラムをプログラム・ライブラリに登録できるのは、アプリケーション開発者です。システム管理者は登録できません。
表 8		

関連項目

- [「コンカレント・プログラムのパラメータ」](#) (3-43)
- [「コンカレント・プログラム詳細レポート」](#) (3-52)
- [「プログラムのパラメータの変更例」](#) (3-50)
- [「プログラム定義のコピーと変更」](#) (3-41)

プログラムのパラメータの変更例

コンカレント・プログラムの各パラメータの変更時期と変更方法について、次の例を考えてみます。

あるユーザーが、たとえば Oracle Bill of Materials や Oracle Purchasing スーパーバイザなど、多数のコンカレント要求を日常的に発行する場合、システム管理者はそのユーザーのコンカレント・プロセス・レコードのみを削除する効率的なページ・プログラムを作成できます。

システム管理者はこのプログラムを実行できるほか、指定された間隔で自動的に再発行することもできます。

またこのプログラムを含む「要求セット」を作成して、当該ユーザーをその要求セットの所有者として定義できます。この要求セットをどのレポート・セキュリティ・グループにも割り当てなかったならば、そのプログラムの実行は当該ユーザー（所有者）のみが可能です。この方法により、ユーザーはその所有レコードの削除に責任を負うことになります。

システム管理者の「コンカレント要求やマネージャ・データのページ」プログラムには、12のパラメータがあります。その内の3つのパラメータのみを表示させ、ユーザーには1つのパラメータの指定のみを求めるようにプログラムを変更できます。この場合、あらかじめプログラムのコピーを作成し、その名前を変更してください。参照：[「「コンカレント要求やマネージャ・データのページ」プログラム」](#) (4-16)。

次の表に、この例を実行する手順の概要を示します。

例 - プログラム・パラメータの変更

使用フォーム	タスク
コンカレント・プログラム (コンカレント・プログラム定義)	「コンカレント要求とマネージャ・データの削除」という名前の Application Object Library プログラムを問い合わせ、「コピー」を押します。「引数のコピー」と「非互換プログラムのコピー」の両方を選択します。
	コピー対象のプログラムに対して、たとえば JSMITH PURGE という新しい名前を入力します。

使用フォーム	タスク
コンカレント・プログラム	JSMITH PURGE というプログラムの各パラメータを変更するには、「パラメータ」ボタンを選択します。
「パラメータ」ウィンドウ	次の7つのパラメータを変更して、それらを非表示に設定します（ユーザー JSMITH は、プログラムのデフォルト値の参照および変更のいずれもできません）。-Oracle ID- プログラム・アプリケーション - プログラム - マネージャ・アプリケーション - マネージャ - 職責アプリケーション - 職責
	次の3つのパラメータを変更して、それらを非表示に設定します（ユーザー JSMITH は、システム管理者が設定したデフォルト値の参照および変更のいずれもできません）。各パラメータを次の（タイプ = 固定）デフォルト値に設定します。- エンティティ = 要求 - モード = 年齢 - ユーザー名 = JSMITH
	次の2つのパラメータはそのまま表示されるよう、変更しません。「モードの値」にはユーザー JSMITH が値を入力する必要があり、また「レポート」はデフォルト値の「YES」に設定されます。- モードの値 - レポート
要求セット（レポート・セット）	要求セットを作成し、その中に JSMITH の PURGE プログラムを格納します。「所有者」フィールドに JSMITH と入力します。この要求セットが割り当てられているレポート・セキュリティ・グループが存在しない場合、JSMITH の PURGE プログラムを実行できるのはユーザー JSMITH に限定されます。
「標準要求発行」プログラム・フォーム。たとえば、「レポートの実行」フォーム（レポート実行）	JSMITH の PURGE プログラムを初めて発行して実行する場合は、「再発行オプション」リージョンにナビゲートして、「インターバル」フィールドに、「5」や「日」などを入力します。
表 9	

関連項目

- 「コンカレント・プログラムのパラメータ」（3-43）
- 「コンカレント・プログラム詳細レポート」（3-52）
- 「プログラム定義のコピーと変更」（3-41）
- 「要求パラメータの動作の制御」（3-44）

コンカレント・プログラム詳細レポート

このレポートは、実行ファイルの情報、実行方法、非互換プログラムのリスト、プログラム・パラメータなどコンカレント・プログラムの定義を示します。コンカレント・プログラムがレポートを作成すると、印刷の出力と印刷の形式、および列と行の情報も出力されます。

プログラムの非互換ルールの変更など、コンカレント・プログラムの変更を検討するときには、このレポートを使用してください。

レポート・パラメータ

注意：パラメータを何も入力しなかった場合、レポートはすべてのコンカレント・プログラムに関する値を戻しますが、これは非常に長いリストになります。

アプリケーション名

ここで選択したアプリケーション名に対応付けられたコンカレント・プログラムの詳細定義情報がレポートに出力されます。

アプリケーションに対応付けられたすべてのコンカレント・プログラムに関するプログラム定義の詳細レポートを実行するには、プログラム名を選択せずに、アプリケーション名のみを選択してください。

プログラム

レポートに出力するプログラム定義の詳細を持つコンカレント・プログラムの名前を選択してください。「プログラム」に値を入力する前に、「アプリケーション名」に値を入力してください。

レポート見出し

レポートの見出しには、指定されたレポートのパラメータが表示され、そのレポートの内容に関する一般情報が示されます。

関連項目

- [「コンカレント・プログラム・レポート」](#) (3-53)
- [「プログラム非互換性ルールの定義」](#) (3-21)
- [「要求セット・レポート」](#) (3-15)

コンカレント・プログラム・レポート

このレポートは、どのコンカレント・プログラムが現在使用可能で、どのプログラムが使用禁止かを提示します。

このレポートは、ユーザーのコンカレント・プログラムに対応付けられた実行方法、引数の方法、個別実行状態、標準出力状態、要求のタイプ、および印刷形式の情報を記録するときに使用してください。

レポート・パラメータ

アプリケーション名

レポートに出力するプログラムの情報を持つコンカレント・プログラムに対応付けられたアプリケーション名を選択してください。

アプリケーション名を入力しなかった場合、レポートはすべてのコンカレント・プログラムに関する値を戻します。

レポート見出し

レポートの見出しには、指定されたレポートのパラメータが表示され、そのレポートの内容に関する一般情報が示されます。

関連項目

- 「コンカレント・プログラム」ウィンドウ
- 「[コンカレント・プログラムおよびコンカレント要求の概要](#)」(3-1)
- 「[コンカレント・プログラム詳細レポート](#)」(3-52)

「要求グループ」ウィンドウ

このウィンドウを使用して、要求グループを定義します。要求セキュリティ・グループとは、特定の職責の下で作業しているユーザーが「要求の発行」ウィンドウで選択できる、要求、要求セットおよびコンカレント・プログラムの集合です。

システム管理者：

- 職責を定義するときに、その職責に要求セキュリティ・グループを割り当てます。要求セキュリティ・グループを持たない職責は、「要求発行」ウィンドウでどのような要求も実行できません。
- 要求セキュリティ・グループには、どのような要求セットでも追加できます。要求セキュリティ・グループにプライベートな要求セットを追加すると、他のユーザーが「要求発行」ウィンドウでその要求セットを実行できます。

ユーザー：

- 「要求セット」ウィンドウで自分の要求セットを作成できます。ユーザーがプライベートな要求セットへ追加できるのは、セキュリティ・グループに割り当てた要求のみです。
- 「要求セット」ウィンドウでは、他のユーザーのプライベートな要求セットを更新できません。
- 要求セキュリティ・グループに割り当てたプライベート要求セットは削除できません。

要求グループ・ブロック

グループ

要求グループ名を使用すると、「職責」ウィンドウで要求グループを職責に割り当てられます。アプリケーション名と要求グループ名をいっしょに使用すると、要求グループを一意に識別できます。

アプリケーション

要求グループに関連付けるためのアプリケーション名を選択します。アプリケーション名と要求セキュリティ・グループ名をいっしょに使用すると、要求セキュリティ・グループを一意に識別できます。このアプリケーション名を使用しても、他のアプリケーションからこの要求グループに要求および要求セットを割り当てる妨げとはなりません。

コード

コードをこの要求グループに割り当てます。製品の中には、この要求グループ・コードを、カスタマイズされた標準発行フォームが選択できる要求を識別するためのパラメータとして使用するものもあります。参照：[「コードを使用した「要求の発行」ウィンドウのカスタマイズ」](#) (3-17)。

要求ブロック

要求グループの中の要求および要求セットを指定します。

タイプ

プログラムまたはセットを選択すると1項目のみを追加でき、アプリケーションを選択するとすべての要求をアプリケーションに含めることができます。

関連項目

- [「コードを使用した「要求の発行」ウィンドウのカスタマイズ」](#) (3-17)
- [「コンカレント・プログラムおよびコンカレント要求の概要」](#) (3-1)
- [「プログラムの要求グループ別編成」](#) (3-15)

- 「レポート・グループ職責レポート」 (3-20)
- 「要求グループでのコードの使用」 (3-17)

「コンカレント・プログラム実行ファイル」ウィンドウ

コンカレント・プログラムとともに使用するそれぞれの実行ソース・ファイルのコンカレント・プログラム実行ファイルを定義します。コンカレント・プログラム実行ファイルは、自分のソース・ファイル・ロジックを、自分および自分のユーザーがコンカレント・マネージャに発行するコンカレント要求にリンクします。

使用するオペレーティング・システムの「インストール・ガイド」に各実行方法の実行ファイルが置かれている場所が詳しく説明されています。

注意: コンカレント・マネージャ・プログラム・ライブラリに新規の即時プログラムを追加することはできません。そのかわりに作成済みコンカレント・プログラムを使用することをお勧めします。

コンカレント・プログラム実行ファイル・ブロック

アプリケーション名とプログラム名の両方を使用することにより、コンカレント・プログラム実行ファイルを一意に識別できます。

参照: 「「コンカレント・プログラム」ウィンドウ」

実行ファイル

コンカレント・プログラム実行ファイルの名前を入力します。「コンカレント・プログラム」ウィンドウでこの名前をコンカレント・プログラムに割り当てると、コンカレント・プログラムを実行ロジックに関連付けられます。

短縮名

コンカレント・プログラム実行ファイルの短縮名を入力します。

アプリケーション

コンカレント・マネージャは、このアプリケーションを使用してどのディレクトリで実行ファイルを検索するかを判断します。

実行方法

コンカレント・プログラム実行ファイルを「コンカレント・プログラム」ウィンドウで1つまたは複数のコンカレント・プログラムに割り当てると、その後は実行方法を変更できません。

次の実行方法が可能です。

ホスト	実行ファイルは、ホスト・スクリプトです。
Oracle Reports	実行ファイルは、Oracle Reports ファイルです。
PL/SQL ストアド・プロシージャ	実行ファイルは、ストアド・プロシージャです。
SQL*Loader	実行ファイルは、SQL スクリプトです。
SQL*Plus	実行ファイルは、SQL*Plus スクリプトです。
作成済	実行ファイルは、C または Pro*C のプログラムです。
即時	実行ファイルは、コンカレント・マネージャのサブルーチンとして実行するよう作成されたプログラムです。新規の即時コンカレント・プログラムを定義することはお薦めしません。そのかわりに PL/SQL ストアド・プロシージャまたは作成済み C プログラムをお薦めします。
要求セット・ステージ機能	要求セット・ステージの完了ステータスを計算する場合に使用できる PL/SQL ストアド機能です。

実行ファイル名

実行ファイルのオペレーティング・システム名を入力します。オペレーティング・システムの中には大文字小文字を区別するものがあります。したがって、ファイル名と正確に一致する名前を入力してください。

実行方法が PL/SQL ストアド・プロシージャであるか要求セット・ステージ機能である場合を除き、実行ファイル名にスペースまたはピリオド (.) を含めないでください。Oracle Applications で各実行ファイルを検索するときに使用されるパスの詳細は、『Oracle Applications インストレーション・ガイド』を参照してください。

実行ファイル名には 60 字まで使用できます。

サブルーチン名

ここに C または Pro*C プログラムのサブルーチン名を入力します。このフィールドにはスペースまたはピリオド (.) は使用しないでください。

サブルーチン・フィールドを使用するのは、Unified C API を使用する即時プログラムまたは作成済みプログラムのみです。

新規の即時コンカレント・プログラムを定義することはお薦めしません。そのかわりに PL/SQL ストアド・プロシージャまたは作成済み C プログラムをお薦めします。

「ステージ機能パラメータ」

「ステージ機能パラメータ」ボタンを押すと、要求セット・ステージ機能のパラメータを入力するウィンドウがオープンします。このボタンは、「実行方法」に要求セット・ステージ機能を選択したときにのみ使用可能になります。

コンカレント・プログラム・ウィンドウ

このウィンドウを使用して、コンカレント・プログラムを定義および変更します。

前提条件

- コンカレント・プログラムのための実行ファイルを作成します
- 「コンカレント・プログラム実行ファイル」ウィンドウを使用して、オペレーティング・システム・プログラム用のコンカレント・プログラム実行ファイルを定義します。

コンカレント・プログラム・ブロック

アプリケーション名とプログラム名の両方を使用することにより、コンカレント・プログラムを一意に識別できます。

プログラム

「要求」ウィンドウで要求を見ると、この長くてより説明的な名前が表示されます。このコンカレント・プログラムが標準要求発行を通じて実行される場合は、この名前が「要求の発行」ウィンドウに表示されます。

短縮名

短い名前を入力すると、Oracle Applications がそれを使用してコンカレント・プログラムをコンカレント・プログラム実行ファイルに関連付けます。

アプリケーション

プログラムの実行に使用する Oracle ユーザー名およびログと出力ファイルの配置場所を、プログラムのアプリケーションが決定します。

使用可能

ユーザーがこのプログラムを実行するための要求を出せるかどうか、またはコンカレント・マネージャがユーザーのプログラムを実行できるかどうかを指定します。

使用禁止となったプログラムはユーザーのリストには表示されず、どのコンカレント・マネージャのキューにも示されません。コンカレント・プログラムの情報は監査証跡に必要なので、コンカレント・プログラムを削除することはできません。

実行ファイル: 名称

自分のプログラムを実行できるコンカレント・プログラム実行ファイルを選択します。実行可能プログラムを定義するには、「コンカレント・プログラム実行ファイル」ウィンドウを使用します。1つのコンカレント・プログラム実行ファイルを使用して複数のコンカレント・プログラムを定義できます。参照: [「コンカレント・プログラム実行ファイル」ウィンドウ](#) (3-55)。

Oracle Reports のビットマップ・バージョンを含むコンカレント・プログラムを定義する場合、ORIENTATION パラメータまたはトークンを渡すことによってビットマップ・レポートの印刷の向きを管理することができます。たとえば、横長方向にレポートを生成するには、「オプション」フィールドに次のオプションを指定します。

ORIENTATION=LANDSCAPE

実行オプション値の前後に空白を入れないでください。パラメータの区切りには、シングル・スペースのみを使用してください。PORTRAIT の印刷の向きも指定できます。

PAGESIZE パラメータによって生成された出力のディメンションを制御できます。オプション・フィールドで指定された < 幅 >x< 高さ > により、レポート定義で指定された値が上書きされます。次に例を示します。

ORIENTATION=LANDSCAPE PAGESIZE=8x11.5

幅および高さの単位は、Oracle Reports の定義によって決定されます。単位は、Oracle Reports メニューの「レポート」>「グローバル・プロパティ」>「単位」のもとで設定します。

PAGESIZE パラメータで指定したサイズが、レポートの設計サイズよりも小さいと、「REP-1212」エラーが発生します。

実行ファイル: 方法

コンカレント・プログラムで使用される実行方法は、ここに表示されます。

有効な値は次のとおりです。

作成済	コンカレント・プログラムは、C または Pro*C で作成されたスタンドアロン・プログラムです。
ホスト	コンカレント・プログラムは、オペレーティング・システム用のスクリプトで作成されています。
即時	コンカレント・プログラムは、C または Pro*C で作成されたサブルーチンです。即時プログラムはコンカレント・マネージャとリンクされ、マネージャのプログラム・ライブラリに登録されている必要があります。
Oracle Reports	コンカレント・プログラムは、Oracle Reports スクリプトです。
PL/SQL ストアド・プロシージャ	コンカレント・プログラムは、PL/SQL で作成されたストアド・プロシージャです。
SQL*Loader	コンカレント・プログラムは、SQL*Loader プログラムです。

SQL*Plus

コンカレント・プログラムは、SQL*Plus または PL/SQL のスクリプトです。

要求セット・ステージ機能

要求セット・ステージの完了ステータスを計算する場合に使用できる PL/SQL ストアド機能です。

「コンカレント・プログラム実行ファイル」ウィンドウで定義された実行方法を無効にして、「作成済」と「即時」を互いに切り換えることができます。ただし、実行ファイルが選択されたときに「作成済」と「即時」のいずれかの方法が表示され、実行ファイル名とサブルーチン名の両方が「コンカレント・プログラム実行ファイル」ウィンドウで指定されている場合のみです。参照：「[コンカレント・プログラム実行ファイル](#)」ウィンドウ」（3-55）。

優先度

このプログラムに独自の優先順位を割り当てられます。コンカレント・マネージャは、ここで割り当てられた優先順位に従ってプログラムの要求を処理します。

優先順位を指定しないと、ユーザーのプロファイル・オプション「コンカレント：優先順位」が発行時に要求の優先順位を設定します。

要求 タイプ

プログラムを事前定義済みの要求タイプに関連付けるためには、その要求タイプの名前をここに入力します。要求タイプによって、コンカレント・プログラムを実行するコンカレント・マネージャを制限できます。

インクリメンタ

Oracle Applications の内部開発者専用です。増分プログラム機能は、ここに表示されます。

MLS 関数

MLS 関数がある場合には、プログラムで使用されます。

多言語コンカレント要求機能により、ユーザーは一度の要求発行で、要求を複数回（毎回異なる言語で）実行できます。このプログラムでこの機能が使用される場合には、要求で必要な言語インストールが MLS 関数によって決定されます。

参照：

『Oracle Applications 開発者ガイド』

SRS で使用

このボックスをチェックすると、ユーザーがこのプログラムを実行するための要求を「標準要求の実行」ウィンドウから依頼できます。

このボックスをチェックすると、プログラム・パラメータを登録する必要があります。ただし、それは、このウィンドウの下部にあるボタンからアクセスした「パラメータ」ウィンドウがある場合のみです。

使用禁止値の許可

「SRS で使用」ボックスをチェックしたときには、このボックスもチェックして、ユーザーが使用禁止またはすでに使用していない値をパラメータ値として入力できるようにします。

多くの値セットでは、特定の値が（列、ENABLED_FLAG および START_DATE_ACTIVE、END_DATE_ACTIVE の各列を使用して）使用可能な状態かどうかを指示する特別な表の列が使用されます。これらの値セットは、通常は使用禁止の値やすでに使用しなくなった値の問合せを可能にしますが、新しいデータにその値を入力することはできません。標準要求の実行の場合は、このことは通常ユーザーがレポートの発行をするときに、そのレポートが問合せ専用タイプのレポートであっても、使用禁止の値を入力できないことを意味します。

単独実行

プログラムが同じ論理データベース内の他のすべてのプログラムと相対的に独立した関係で実行すべきかどうかを指示します。プログラムを実行すると、同じ論理データベースにある他のすべてのプログラムの実行に影響を与えてしまう（つまり、プログラムがそれ自体を含めて論理データベース内の他のすべてのプログラムと互換性がない）ようなプログラムは、独立して実行することが必要です。

「非互換プログラム」ウィンドウには、どのような非互換のプログラムでも入力できます。

トレース使用可能

プログラムの実行時に SQL トレースをオンにします。

システム障害時再起動

このオプションは、コンカレント・マネージャがシステム障害の後で復元されたときに、このコンカレント・プログラムを自動的に再起動するよう指示します。

NLS 準拠

プログラムによって、ユーザーによるこのプログラムの要求発行が可能とされる場合には、このボックスがチェックされます。このプログラムの要求には、ユーザーが操作の際に使用している言語および地域とは異なる言語および地域が反映されます。

たとえば、英国では英国特有の日付および数字の書式を使用して英語で受注を入力し、その後、ドイツの顧客に適した日付および数字の書式を使用してドイツ語で請求書を作成することができます。

このボックスが空白の場合、ユーザーは、インストールされている言語と要求を関連付けることができます。ただし、地域は、コンカレント・マネージャ環境の地域がデフォルトとなります。

このオプションは、プログラムの開発者によってのみ設定される点にご留意ください。この機能を使用するには、NLS に準拠してプログラムを書く必要があります。参照：『Oracle Applications 開発者ガイド』。

このオプションは、プログラムの開発者によってのみ設定される点にご留意ください。この機能を使用するには、NLS に準拠してプログラムを作成する必要があります。

書式

出力の書式を次の中から選択します。

- HTML
- PCL (HP 社のプリンタ制御言語)
- PDF
- PS (Post Script)
- テキスト

注意：HTML または PDF を Oracle Report プログラムの出力タイプとして選択する場合、HTML または PDF ファイルを処理する適切なプリンタ・ドライバを使用する必要があります。

保存

このプログラムを実行するときに、プログラムの出力を自動的にオペレーティング・システム・ファイルに保管するかどうかを指示します。この値は、このプログラムに対して発行されるすべての要求のデフォルト値となります。この値を「No」に設定すると、プログラムの出力は、印刷実行後に削除されます。

これが標準要求の実行プログラムであるときには、ユーザーは「要求の発行」ウィンドウからこの値を上書きできます。

印刷

「No」を入力すると、コンカレント・プログラムの出力はプリンタに送られません。

列 / 行

このプログラムのレポート出力の列と行の最小の長さを入力します。Oracle Applications は、この情報を利用してレポートにどの印刷形式を使用できるかを判断します。

形式

選択する印刷形式は、システムとプリンタの設定によって異なります。印刷形式には次のものがあります。

- 132 列 66 行 (横長)
- 180 列 66 行 (超横長)
- 80 列 66 行 (縦長)
- 132 列 62 行 (A4)

使用できるリストは、プログラムに設定した列と行の長さに対応した形式のもののみです。

形式要

プログラムに特別の印刷形式（たとえば、小切手発行レポート）が必要なときには、このチェックボックスを選択するとその印刷形式を実行できます。

プリンタ

プログラムの出力を1台のプリンタに制限するときには、出力の送信先プリンタ名を入力します。プログラムに最小または最大の列数または行数が定義されているときには、値のリストにはプログラムの条件を満たすプリンタのみが記載されます。

ユーザーは、「要求の発行」または「要求」ウィンドウから行ったプリンタの選択を上書きすることはできません。

コピー先 ...	同じ実行ファイル、要求およびレポートの各情報を使用して別のコンカレント・プログラムを作成するためには、このボタンを選択します。非互換性およびパラメータの詳細も選択してコピーできます。
セッション管理	コンカレント・プログラムを実行するときにそのプログラムのデータベース・セッションにオプションを指定する場合は、このウィンドウを選択します。
非互換	このボタンを選択すると「非互換プログラム」ウィンドウがオープンします。
パラメータ	このボタンを選択すると「コンカレント・プログラム・パラメータ」ウィンドウがオープンします。

ウィンドウへのコピー

同じ実行ファイル、要求およびレポートの各情報を使用して別のコンカレント・プログラムを作成します。同様に、非互換性およびパラメータの詳細情報をコピーすることもできます。

関連項目

- 「コンカレント・プログラム・パラメータ」ウィンドウ
- 「[コンカレント・プログラム実行ファイル](#)」ウィンドウ (3-55)
- 「非互換プログラム」ウィンドウ

「セッション管理」ウィンドウ

コンカレント・プログラムを実行するときにそのプログラムのデータベース・セッションにオプションを指定する場合は、このウィンドウを使用します。

コンシューマ・グループ

コンカレント・プログラムに生産資源コンシューマ・グループを任意で指定します。

セグメントロールバック

コンカレント・プログラムで使用対象となるロールバック・セグメントを任意で指定します。このロールバック・セグメントは、デフォルト値の代わりに使用し、最初のコミットまでにすべて使用されます。

注意：ここでロールバック・セグメントを指定する場合は、指定したロールバック・セグメントを使用するためにコンカレント・プログラムで FND_CONCURRENT.AF_COMMIT および FND_CONCURRENT.AF_ROLLBACK という API を使用する必要があります。参照：『Oracle Applications 開発者ガイド』

最適化プログラムモード

最適化プログラムモードを任意で指定します。「ALL_ROWS」、「FIRST_ROWS」、「ルール」、「選択」のいずれかを選択できます。最適化プログラムモードを指定する場合は、多くの場合、デフォルトの原価基準の最適化プログラム（CBO）では良好な実行結果が得られず、調整が必要なカスタム・プログラムのみです。使用のプログラムが CBO 用に調整されるまでは、別の最適化プログラムモードを使用できます。

関連項目

■ 「Oracle Applications の生産資源コンシューマ・グループ」 (7-14)

コンカレント・プログラムと同時に実行すると、コンカレント・プログラムの実行の障害となるプログラムを識別します。プログラムがそのプログラム自体と非互換となるように指定できます。

アプリケーション

このフィールドのデフォルト値はコンカレント・プログラムのアプリケーションですが、有効な値であればどのようなアプリケーション名でも入力できます。

名称

コンカレント・プログラムを一意に識別するためのプログラム名およびアプリケーションを入力します。

ユーザーに分かりやすいプログラムの名前、短縮名およびプログラムの説明のリストを表示します。

スコープ

「設定」または「プログラムのみ」を入力すると、コンカレント・プログラムがこのプログラムおよびその子要求（「セット」）と非互換か、あるいはそのプログラムのみ（「プログラムのみ」）と非互換かを指定できます。

関連項目

- 「コンカレント・プログラム・パラメータ」ウィンドウ
- 「コンカレント・プログラム」ウィンドウ
- [「プログラム非互換性ルールの定義」](#) (3-21)
- [「互換性がないプログラムのリストの修正」](#) (3-43)

プログラム実行ファイルに渡すプログラム・パラメータを入力および更新します。ここで定義するプログラム・パラメータは、実行ファイル内の変数と一致していなければなりません。

衝突ドメイン・パラメータ

プログラムの衝突ドメイン値を保有するパラメータを入力します。衝突ドメイン・パラメータに関する情報については、[「コンカレント衝突ドメイン」](#) (3-22) を参照してください。

セキュリティ・グループ

このフィールドは、HRMS セキュリティ専用です。参照：『Oracle HRMS ユーザーズ・ガイド』の「Oracle HRMS でのカスタマイズ、レポート処理およびシステム管理」。

連番

プログラムがコンカレント・マネージャからパラメータ値を受け取る順序を指定するための連番を選択します。

使用可

使用禁止のパラメータが要求発行時に表示されず、実行ファイルに渡されなくなります。

情報の検証

値セット

パラメータが検証に使用する値セットの名前を入力します。選択できる値セットは、独立、表および検証未了のみです。

値セット各には、240 文字まで使用できます。

注意：日付の値セットを使用している場合、この値セットには標準日または標準日時（多言語要求機能を使用している場合）のいずれかの書式タイプが必要です。

デフォルト・タイプ

このパラメータにデフォルト値を設定するには、必要な値のタイプを識別する必要があります。有効なタイプは次のとおりです。

定数	デフォルト値には、すべてのリテラル値を使用できます。
プロファイル	デフォルト値は、「デフォルト値」フィールドで定義されたユーザー・プロファイル・オプションに指定されている現行の値です。エンドユーザー名ではなくプロファイル・オプション名を使用します。\$PROFILE\$を指定する必要はありません。
SQL 文	このデフォルト値は、「デフォルト値」フィールドで定義した SQL 文によって決定されます。
セグメント	このデフォルト値は、同じパラメータ・ウィンドウの前のセグメントに入力した値です。

デフォルト値

このパラメータには、デフォルト値を入力できます。パラメータに対するこのデフォルト値は、パラメータ・ウィンドウに入力すると自動的に表示されます。デフォルトのタイプを選択すると、デフォルト値を、定数にするかコンテキスト依存の値にするかを定めることができます。

デフォルト値は、値セットに使用できる有効な値でなければなりません。それ以外の値を使用すると、「実行要求」ウィンドウにあるパラメータ・ウィンドウを開けた際にエラー・メッセージが表示され、デフォルト値は表示されません。

各デフォルト・タイプの有効な値は次のとおりです。

定数	デフォルト値には、任意のリテラル値を入力します。
プロファイル	デフォルト値は、ここで指定するユーザー・プロファイル・オプションの現在値となります。エンド・ユーザー名ではなく、プロファイル・オプション名を入力します。
セグメント	このデフォルト値は、同じフレックスフィールド・ウィンドウの前のセグメントに入力した値です。値をコピーするセグメントの名前を入力します。
SQL 文	デフォルト値は、ここに入力する SQL 文によって決まります。SQL 文は、どのような場合でも必ず 1 行と 1 列を戻すものにする必要があります。

必須

プログラム実行ファイルに引数が必要なときは、コンカレント・プログラムにも引数が必要です。

セキュリティ可能

このパラメータの値セットによってセキュリティ・ルールの使用が禁止されているときは、このフィールドは表示されますが特に機能は果たしません。セキュリティ・ルールの使用が許されるときは、この値セットに定義されパラメータ・リストに影響を与えるセキュリティ・ルールならどれでも適用できます。

範囲

パラメータ値をこの体系の中の別のパラメータの値と比べて検証するには、「Low」または「High」のいずれかを選択します。範囲が「Low」のパラメータは、必ず範囲が「High」のパラメータよりも前に置かなければなりません（低いパラメータには、必ず高いパラメータよりも低い数値を指定します）。たとえば、「開始日」および「終了日」という2つのパラメータの使用を予定していて、ユーザーに開始日より後の終了日を入力してもらう必要があるとします。このようなときには、「開始日」の範囲を「Low」にし、「終了日」の範囲を「High」とします。この例では、パラメータ「開始日」をパラメータ「終了日」よりも前に置く必要があります。

1つのパラメータに「Low」を選択したら、同じ体系の中の別のパラメータには「High」を選択する必要があります（あるいは、その逆）。そうしないと、変更をコミットできません。

ウィンドウ情報 表示

ユーザーが「要求の発行」ウィンドウからプログラム実行の要求依頼を行ったときに、このパラメータを「パラメータ」ウィンドウに表示するかどうかを指定します。

表示されないすべてのパラメータに対してもデフォルトのタイプと値を指定する必要があります。

表示サイズ

このパラメータのフィールドの長さを文字数で入力します。ユーザーは、「要求の発行」ウィンドウの中の「パラメータ」ウィンドウにあるこのフィールドに値を入力します。

すべてのパラメータの最大サイズ（表示サイズではない）に設定された値と必要なセパレータ数（パラメータ数より1少ない）の合計が240を超えないように注意が必要です。プログラムの値の連結した長さが240を超えると、フォームによっては中のデータが切り捨てられます。

摘要サイズ

パラメータ値を記述する表示の長さを文字数で入力します。十分なスペース（最長のプロンプトおよびこのパラメータの表示サイズにさらに7を加えた合計によって決まる）がなければ、ウィンドウにはここで指定する値よりも少ない文字数の記述が表示されます。ただし、ウィンドウに表示される説明の文字数がここで指定する値を超えることはありません。

プロンプト

「要求の発行」ウィンドウの中の「パラメータ」ウィンドウに、パラメータ名ではなくこのプロンプトが表示されます。

連結摘要サイズ

パラメータ値を記述する表示の長さを文字数で入力します。ユーザーに対して、「要求の発行」フォームおよび「要求の表示」フォームの「パラメータ記述」フィールドにこのパラメータ値が表示されます。「パラメータ記述」フィールドでは、コンカレント・プログラムのすべてのパラメータ値が連結されます。

提案：各パラメータについて連結摘要サイズを設定するときには、プログラムの合計の連結摘要サイズが 80 以下となるように設定することが必要です。それは、ほとんどのビデオ画面の幅が 80 字に設定されているためです。

トークン

Oracle Reports プログラムでのパラメータについては、キーワードまたはパラメータがここに表示されます。この値については、大文字と小文字が区別されます。それ以外の種類のプログラムについては、このフィールドは入力する必要はありません。

関連項目

- 「コンカレント・プログラム」ウィンドウ
- 「非互換プログラム」ウィンドウ

「データ・グループ」ウィンドウ

このウィンドウを使用して、データ・グループを定義します。データ・グループとは、Oracle Applications と各アプリケーションに割り当てられた Oracle ユーザー名のリストです。

- カスタム・アプリケーションが Oracle Application Object Library を使用して開発されたものであれば、Oracle ユーザー名が割り当てられていて、Oracle Applications に登録されており、データ・グループに属している可能性があります。

Oracle ユーザー名を使用すると、Oracle データベース内のアプリケーション表にアクセスできます。すべてのデータ・グループには、Application Object Library の項目が自動的に含まれます。

- Oracle Applications の下でレポートまたはプログラムを実行しているコンカレント・マネージャは、データ・グループを参照して、データベース内でアプリケーション表にアクセスするときに使用する Oracle ユーザー名を識別します。

- 同期プログラムを実行する取引マネージャは、取引マネージャと同じデータ・グループを使用する職責から実行要求があったプログラムのみを実行できます。カスタム・データ・グループを作成するには、取引マネージャを使用するアプリケーションの新規取引マネージャを作成する必要があります。使用する製品のマニュアルを参照して、アプリケーションが取引マネージャを使用するかどうかを調べてください。

Oracle Applications の中の各職責は、データ・グループに割り当てられています。

Oracle Applications のインストレーションまたはアップグレードの際には、標準データ・グループが定義され、インストールされた各アプリケーションに Oracle ユーザー名を組み合わせます（注：標準データ・グループはすべての会計帳簿に対して定義されます）。標準データ・グループ内でアプリケーションまたは Oracle ユーザー名用に事前定義されている値は、変更も削除もできません。ただし、次の操作は可能です。

- アプリケーションと Oracle ユーザー名との対に関連する Tool Oracle ユーザー名および説明の変更。
- アプリケーションと Oracle ユーザー名との新規の対をグループに追加。

データ・グループ・ブロック

新規データ・グループの作成、または既存のデータ・グループの変更を行います。

標準データ・グループ内でアプリケーションまたは Oracle ユーザー名用に事前定義されている値は、変更も削除もできません。ただし、Tool Oracle ユーザー名と説明を変更したり、アプリケーションと Oracle ユーザー名の対を新規に標準グループへ追加することは可能です。

データ・グループ

データ・グループは、その名前で一意に識別できます。すでに使用されている名前を新しいデータ・グループに付けることはできません。

一度保存したデータ・グループ名は変更できません。

アプリケーション -Oracle ID 対ブロック

アプリケーションを Oracle ユーザー名と組み合わせて対を作成します。

データ・グループをコピーすると、各アプリケーション、およびアプリケーションに割り当てられた Oracle ユーザー名が自動的にこのゾーンに表示され、さらに Tool Oracle ユーザー名と摘要があればともに表示されます。すべてのデータ・グループは、自動的に Oracle Application Object Library の項目を含みます。

アプリケーション

各データ・グループの中では、アプリケーションは一度しか指定できません。

Oracle ID

アプリケーションに割り当てる ORACLE ID を選択します。データベース内の表にアクセスする際に、アプリケーションでは ORACLE ID が使用されます。各 ORACLE ID の使用により、データベース内の事前定義表セットにアクセスできます。

アプリケーションのコピー ...

このボタンを使用して既存のデータ・グループをコピーしてから、アプリケーションと Oracle ユーザー名との対を追加または削除して新規のデータ・グループを作成します。

関連項目

- [「コンカレント・プログラムおよびコンカレント要求の概要」](#) (3-1)
- [「データ・グループの使用」](#) (3-26)
- [「データ・グループの修正」](#) (3-28)
- [「データ・グループの定義」](#) (3-25)

コンカレント処理の管理

コンカレント・プロセスの概要

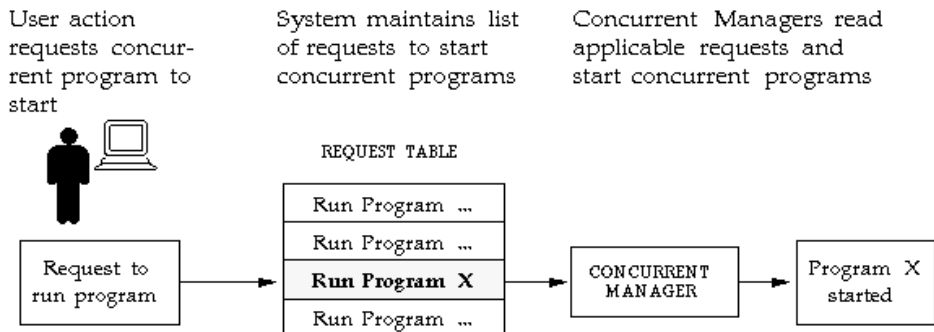
この項では、Oracle Applications によるコンカレント・プログラムの実行要求の処理およびコンカレント要求のライフ・サイクルについて説明します。

Oracle アプリケーションでは、コンカレント・プロセスによりバックグラウンドで動作するプログラムとオンライン操作が同時に実行されます。システム管理者は、プログラムの実行時期と、Oracle アプリケーションによってプログラムをバックグラウンドで実行するために使用されるオペレーティング・システム・プロセス数を管理できます。

コンカレント要求、コンカレント・プログラムおよびコンカレント・プロセス

ユーザーがレポートを実行すると、レポートの実行要求が生成されます。要求を実行するコマンドは、コンカレント要求です。レポートを生成するプログラムは、コンカレント・プログラムです。コンカレント・プログラムは、コンカレント・マネージャによって起動されます。

図 11



コンカレント・マネージャによるコンカレント・プログラムの起動

ユーザーがコンカレント・プログラムの実行を要求するたびに、その要求がデータベース表に挿入され、要求 ID で個々に識別されます。コンカレント・マネージャは、この表から要求を読み込みます。

マネージャの定義の一部は、要求の実行に振り向けることのできるオペレーティング・システム・プロセスの数です。この数を、そのマネージャのターゲット・プロセスの数と呼びます。

コンカレント・プログラムの実行

コンカレント・プログラムの実行は、実際には次の条件に基づいて開始されます。

- 起動予定の日時
- 保留されているかどうか
- 他のプログラムと非同期（ともに実行できない）かどうか
- その優先度

コンカレント要求の優先度

コンカレント要求の優先度は、アプリケーション・ユーザー名によって判別され、システム管理者が「コンカレント:優先度」ユーザー・プロファイル・オプションを使用して設定します。

最初に使用可能なコンカレント・マネージャは、その要求の優先順位を他に処理すべき要求と比較して、優先度が最も高い要求を実行します。

同じ優先度を持つ要求の中から選択される場合は、最も古い要求から順に実行されます。

親要求と子要求

通常は、複数のプログラムを要求セットとしてグループ化できます。要求セット全体を発行すると、要求 ID が作成され、そのセットの各メンバーは発行されるたびに独自の要求 ID を割り当てられます。要求セットの要求 ID によって親要求が識別され、個々のプログラムの各要求 ID によって子要求が識別されます。

関連項目

- 「コンカレント・マネージャ」ウィンドウ (4-76)
- 「コンカレント・マネージャの管理」ウィンドウ (4-71)
- 「コンカレント・プロセスのユーザー・プロファイル設定値」 (4-19)
- 「コンカレント・プロセス用のファイルおよび表の管理」 (4-14)
- 「コンカレント・マネージャの制御」 (4-50)

- 「コンカレント要求のステータスの変更」 (4-12)
- 「コンカレント要求のライフ・サイクル」 (4-6)
- 「パラレル・コンカレント・プロセスの概要」 (4-58)
- 「マネージャおよびマネージャの稼働シフトの定義」 (4-22)
- 「特定なプログラムのみを実行するためのマネージャの特殊化」 (4-33)

Oracle Enterprise Manager との統合

Oracle Applications のコンカレント・マネージャ管理インタフェースは、Oracle Enterprise Manager と統合されています。これにより、管理者によるシステム管理が容易になりました。Oracle Enterprise Manager を使用すると、システム上で使用可能なすべての Oracle Applications インスタンスを一括して管理できます。

Oracle Enterprise Manager は Java ベースのフレームワークであり、複数のコンポーネントを統合することにより、すぐれたグラフィカル・ユーザー・インタフェース (GUI) を実現しています。Oracle Enterprise Manager は、基本コンソール、エージェント、共通サービスおよびツールを組み合わせることにより、Oracle 製品の管理のための総合的な統合システム管理プラットフォームを提供します。

Enterprise Manager の詳細は、『Oracle Enterprise Manager 概説』および『Oracle Enterprise Manager 管理者ガイド』を参照してください。

Oracle Applications Manager

Oracle Applications Manager では、現行の Oracle Applications システム管理機能の中から Applications データベース管理者用の機能だけをまとめて、それらを新たに Oracle Applications Manager コンソールの形で提供しています。これらの機能には、コンカレント・マネージャおよび要求の管理機能が含まれます。コンソールの機能は、マルチウィンドウ方式の Oracle Applications フォームを補完するものであり、管理者はどのツールを使用するかを選択することができます。標準 Oracle Applications ウィンドウから発行された要求は、Oracle Applications Manager コンソールで表示することができます。同様に、コンソールで定義されたコンカレント・マネージャには、Oracle Applications フォームからアクセスすることができます。

システム管理者は、通常の機能に加えて、管理している Oracle Applications インスタンスのスナップショット要約を参照することができます。

Oracle Management Pack for Oracle Applications

Oracle Management Pack for Oracle Applications（以後 Management Pack for Oracle Applications と呼びます）は、複数の層から構成される Oracle Applications 環境のモニター、診断および能力計画の各機能を実現するために、Oracle Enterprise Manager を拡張した製品です。

Oracle Applications Manager

単一の Oracle Applications Manager コンソールから、複数の Oracle Applications インスタンスに対して多くのタスクを実行できます。たとえば、実働インスタンスおよびテスト・インスタンスに対する全コンカレント・マネージャのステータスにアクセスできます。この情報は、一度に1つのインスタンスに関して表示できます。

次の情報がコンソールから使用可能です。

- 全要求とマネージャの要約
- 全コンカレント・マネージャの詳細
- 取引マネージャの詳細
- 全コンカレント要求の詳細

コンソールからコンカレント・マネージャを開始および停止することもできます。

次のものを定義または編集できます。

- マネージャ
- 稼動シフト

コンカレント要求に関する次の情報を表示できます。

- 計画および完了オプションを含む詳細
- 解析
- マネージャ、ログおよび出力ファイル
- 統計
- 実行されていない要求に対して使用可能なマネージャ

Enterprise Manager をインストールする際、Universal Installer を使用して Oracle Applications Manager をインストールします。インストレーションおよび構成に関する情報については、Oracle Enterprise Manager および Oracle Applications Manager 文書を参照してください。

Oracle Management Pack for Oracle Applications

Oracle Management Pack for Oracle Applications を使用すると、Oracle Enterprise Manager が拡張され、多層化された Oracle Applications 環境のモニター、診断および能力計画が可能になります。Oracle Management Pack for Oracle Applications では、次のような Oracle Enterprise Manager のシステム管理機能を利用しています。

- モニター対象サービスの検出およびグラフィカル表示
- 特定のサービスのコンテキストでのコンソールからの開始ツール
- 自動化されたデータ収集サービスおよびデータ管理サービス
- インテリジェント・エージェントを利用したリモート・システムの集中モニターおよび一元管理

Oracle Management Pack for Oracle Applications を使用すると、Oracle Applications と連携動作する次のツールが拡張されます。

- Oracle Enterprise Manager コンソール

このコンソールは、コンカレント・マネージャの検出およびサーバーがダウンした場合の通知のために拡張されました。任意の Oracle Applications サブシステムのジョブも定義できるので、これにより、Oracle Applications の分散システムを一元管理できます。

- Oracle Applications Advanced Events

Oracle Applications システムの自動化されたイベント監視および問題検出のために、Oracle Applications に固有のイベントのライブラリが提供されています。さらに、イベントがトリガーをうけると自動的に実行されるように設定された調整ジョブによる自動問題解決機能も提供されています。これらの調整ジョブは、カスタム仕様で作成するか、または Oracle Management Pack for Oracle Applications に用意されている小規模の事前設定から選択します。

- Oracle Performance Manager

新しい Oracle Applications データ収集カートリッジを利用すると、Oracle Performance Manager にデータが供給されるので、これにより、Oracle Applications インスタンスが使用しているすべてのコンカレント・マネージャおよび Forms セッションに関するぼう大なリアルタイム監視図が表示されます。

- Oracle Capacity Planner

コンカレント・マネージャの実績データは、長時間にわたって収集されて Oracle Capacity Planner に供給され、生産資源の消費分析やパフォーマンス異常の検出が行われます。

■ Concurrent Processing Tuning Assistant

このユーティリティを使用すると、Oracle Concurrent Processing の要求およびコンカレント・マネージャに関する履歴処理情報を調べることができます。

Oracle Management Pack for Oracle Applications の詳細は、『Getting Started with the Oracle Management Pack for Oracle Applications』を参照してください。

コンカレント要求のライフ・サイクル

コンカレント要求は、ライフ・サイクル中に次の3つまたは4つの段階、つまりフェーズを経て処理されます。

- 保留中** 要求は実行の待機中です。
- 実行中** 要求は実行中です。
- 完了** 要求は完了しました。
- 無効** 要求は実行できません。

各フェーズ内で、要求の状態、つまりステータスは変化することがあります。次の表は、各フェーズとコンカレント要求が通る各種のステータスを示しています。

コンカレント要求のフェーズとステータス

フェーズ	ステータス	摘要
保留中	正常	要求は、次に使用可能なマネージャを待機中です。
	スタンバイ	要求を実行するプログラムには、現在実行中の他のプログラムとの互換性がありません。
	計画済み	要求は将来の日時に開始されるように予定されています。
実行中	待機中	子要求は、親要求に実行準備完了マークが付くのを待機中です。たとえば、レポート・セット内で順次に実行されるレポートは、直前のレポートが完了するまで待機する必要があります。
	正常	要求は正常に実行中です。
	一時停止	親要求は、そのすべての子要求が完了するまで一時停止します。たとえば、レポート・セットは、そのセット内のすべてのレポートが完了するまで一時停止します。
	再開	同じ親要求から発行されたすべての要求の実行が完了しています。親要求は再起動を待機しています。
	終了中	「要求詳細」ゾーンの「ステータス」フィールドで「終了」が選択されたため、実行中の要求は終了しました。

フェーズ	ステータス	摘要
完了	正常	要求は正常に完了しました。
	エラー	要求は異常終了しました。
	警告	要求は完了しましたが、警告メッセージが発行されました。たとえば、レポートは正常に生成されましたが、印刷に失敗しました。
	取消	「要求詳細」ゾーンの「ステータス」フィールドで「取消」を選択したため、「保留中」または「無効」フェーズの要求が取り消されます。
	終了済	「要求詳細」ゾーンの「ステータス」フィールドで「終了」が選択されたため、実行中の要求は終了しました。
無効	使用不可	要求を実行するプログラムが使用可能になっていません。システム管理者に連絡してください。
	保留中	「要求詳細」ゾーンの「ステータス」フィールドで「保留」が選択されたため、保留中の要求が保留されます。
	マネージャなし	要求を実行するマネージャが定義されていません。システム管理者に確認してください。
表 10		

関連項目

- 「[Report Review Agent の起動](#)」(4-9)
- 「[エンド・ユーザー・レポートおよびログ・ファイルのアクセス権限の設定](#)」(4-8)
- 「[コンカレント・プロセス用のファイルおよび表の管理](#)」(4-14)
- 「[要求のステータスおよび出力を表示する方法](#)」(4-7)

要求のステータスおよび出力を表示する方法

コンカレント要求のステータスと出力を表示するには、次のいずれかの方法を使用します。

「要求」ウィンドウを使用する

「要求」ウィンドウを使用して、コンカレント要求のステータスを表示し、要求ログ・ファイルおよびレポート出力ファイルを表示します。

システム管理者と Oracle アラート・マネージャには、エンド・ユーザーよりも高い権限を与えるために、専用の権限を備えた「要求」ウィンドウが提供されています。たとえば、その「要求」ウィンドウを使用すると、(自分自身のコンカレント要求だけでなく)すべてのコンカレント要求について、異常終了した要求も含め、ステータスとログ・ファイルを表示

できます。一部のプラットフォームでは、実行中の要求のログ・ファイルを表示することもできます。

その同じウィンドウを使用して、自分自身のレポート出力をオンラインで表示できます。ただし、別のユーザーの要求からのレポート出力を表示することはできません。

「要求」ウィンドウから、次のこともできます。

- 保留中または非アクティブの要求に対し、保留を保持したり保留を解除したりします。
- 保留中の要求を取り消したり、実行中の要求を終了したりします。
- 保留中の要求の優先度を変更します。
- マネージャ・ログ・ファイルを表示します。
- 保留中のどのような要求についても、要求を受け入れるように定義された個々のマネージャ用に、その要求がキュー内のどこにあるかを判別します。
- 「内部コンカレント・マネージャ」が無効な状態で再起動の必要がある時点を判別します。

終了済みコンカレント要求レポートを実行する

実行が終了したコンカレント要求へ関連付けられたパラメータとエラー・メッセージのリストを示すレポートを実行できます。参照：[「完了したコンカレント要求レポート」](#) (4-31)。

エンド・ユーザー・レポートおよびログ・ファイルのアクセス権限の設定

ユーザー・プロファイル・オプション「コンカレント:レポート・アクセス・レベル」により、エンド・ユーザーのレポート出力ファイルおよびログ・ファイル・アクセス権限が決定されます。システム管理者は、このプロファイル・オプションを「ユーザー」か「職責」に設定できます。

すべてのユーザーは、自分が発行した要求の結果、生成されるログとレポート出力ファイルを参照できます。

「コンカレント:レポート・アクセス・レベル」オプションを「ユーザー」レベルで「職責」に設定すると、そのユーザーはその職責に基づいて発行されるすべての要求のログとレポート出力ファイルを参照できます。

「コンカレント:レポート・アクセス・レベル」オプションを「職責」レベルで「職責」に設定すると、その職責を持つすべてのユーザーは、その職責の他のユーザーが発行したすべての要求のログとレポート出力ファイルを表示できます。

Report Review Agent の起動

Report Review Agent を使用すると、システム管理者がファイル転送サイズについて指定した制限の下で、レポート・ファイルまたはログ・ファイルの全体を一度にコピーするか、1 ページずつユーザーの PC へコピーできます。

1. Report Review Agent を設定するには、データベース管理者またはコンピュータ管理者が Net8 構成を変更する必要があります。Net8 構成変更の詳細は、リリース 11i の『Oracle Applications インストレーション・マニュアル』を参照してください。
2. Report Review Agent を起動してコンカレント・プロセス・ノード上でファイルにアクセスするには、「ファイル・サーバー：使用可能」プロファイル・オプションを「YES」に設定します。参照：「[システム・プロファイル値ウィンドウ](#)」(9-6)。
3. カスタム・エディタを使用してコンカレント出力ファイルまたはログ・ファイルを表示する場合、Report Review Agent によりクライアント上にファイルの一時コピーが作成されます。ユーザーが Oracle Applications を終了するときに、それらのファイルを自動的に削除するには、「ファイル・サーバー：一時ファイルを削除する」プロファイルを「YES」に設定します。参照：「[Reports Viewer の定義](#)」(4-10)、「[Oracle Application Object Library のプロファイル・オプション](#)」(A-1)。
4. 「ファイル・サーバー：最大転送サイズ」プロファイル・オプションを設定して、Report Review Agent によって転送されるファイルの最大許容サイズをバイト数単位で指定します。このサイズには、ユーザーが Oracle Applications レポート・ファイル・ビューアの「コピー・ファイル ...」メニュー・オプションを使用してダウンロードするファイル、およびカスタム・エディタにより自動的にダウンロードされる一時ファイルも含まれます。
このプロファイルが NULL の場合、サイズ制限はありません。参照：「[システム・プロファイル値ウィンドウ](#)」(9-6)。
5. すべてのログ・ファイル・ディレクトリと外部ファイル・ディレクトリがコンカレント・マネージャによって同じ親ディレクトリに置かれるように、ディレクトリ・ツリーを設定します。必要な環境変数の設定の詳細は、(使用しているサーバー・プラットフォームに対応する) リリース 11i の『Oracle Applications インストレーション・マニュアル』を参照してください。

つまり、すべてのログ・ファイルおよび出力ファイルを常駐させるディレクトリをサーバー上の環境変数 APPLCSF に設定することにより、すべての製品のログ・ファイルおよびレポート出力ファイルを同じ親ディレクトリに入れるようコンカレント・マネージャに指示します。すべてのパスについて、DOS で受け入れられる名前を使用してください。

注意： APPLCSF 環境変数は、使用しているサーバーに固有のものです。
APPLCSF 変数は、クライアント上では使用されません。

6. プロファイル・オプション「Applications Web エージェント」を APPS スキーマの Web Application Server Database Access Descriptor (DAD) のベース URL に設定します。例を次に示します。

`http://<WebAppServer_Machine_Name:Port>/<DAD_name>`

Reports Viewer の定義

テキスト形式のレポート・ファイルを表示する場合、デフォルトでは、Oracle Applications のレポート・ファイル・ビューアを使用します。テキスト・ファイルの表示には、ブラウザや、Microsoft Word などの他のアプリケーションを使用することもできます。デフォルトのビューアを定義するには、プロファイル・オプションを設定します。

「ビューア：テキスト」プロファイル・オプションの設定

「ビューア：テキスト」プロファイル・オプションを「ブラウザ」に設定すると、レポートは Web ブラウザに出力されるようになります。このプロファイル・オプションを空白にすると、ビューアにはレポート・ファイル・ビューアが使用されます。

このプロファイル・オプションを空白にした場合でも、レポート・ファイルまたはログ・ファイルは、次のようにするとブラウザで表示できます。まず、レポート・ファイル・ビューアでファイルを表示し、次に「ツール」メニューから「ファイルのコピー ...」を選択します。

参照：

「個別プロファイル値ウィンドウ」

「システム・プロファイル値ウィンドウ」 (9-6)

ログ・ファイルのタイプ

ログ・ファイルには、コンカレント・プログラムの実行またはコンカレント・マネージャの活動についての情報が保存されます。ログ・ファイルは、問題がある要求を確認するときに役立ちます。

ログ・ファイルは、すべての「終了済み」コンカレント要求について生成されます。

ログ・ファイルには、次の 3 つのタイプがあります。

1. 「要求ログ」ファイルは、コンカレント要求の結果として実行されているコンカレント・プログラムの実行状況を文書化したものです。すべてのコンカレント要求はログ・ファイルを生成します。
2. 「マネージャ・ログ」ファイルは、要求を実行しているコンカレント・マネージャのパフォーマンスを文書化したものです。「マネージャ・ログ」ファイルには、コンカレント・マネージャによって処理された要求のリストが登録されています。

3. 「内部コンカレント・マネージャ・ログ」ファイルは、内部コンカレント・マネージャのパフォーマンスを文書化したものです。このファイルには、内部コンカレント・マネージャの起動時にロードされたパラメータ値が表示されます。

コンカレント・プロセスがエラーで終了した場合には、問題の診断に役立つよう、ログ・ファイルを確認してください。プログラムのパフォーマンスに疑問がある場合にも、ログ・ファイルを確認できます。たとえば、レポートの実行速度が非常に遅い場合、または予期しないデータが印刷出力される場合などです。

「内部コンカレント・マネージャ・ログ」ファイルには、個々のコンカレント・マネージャが起動された時刻および個々のプロセス・モニター・セッションまたは *pmon* サイクルが開始された時刻も記録されます。個々の *pmon* サイクルでは、内部コンカレント・マネージャは個々の定義済みコンカレント・マネージャの操作が正しいかどうかを検証します。

システム管理者ログ・ファイル権限

システム管理者とエンド・ユーザーは、いずれも要求ログ・ファイルとマネージャ・ログ・ファイルをオンラインで参照できます。システム管理者のみが、内部コンカレント・マネージャのログ・ファイルを表示できます。

システム管理者は、「コンカレント要求」ウィンドウおよび「管理者コンカレント・プログラム」ウィンドウを使用して、要求ファイルおよびマネージャ・ログ・ファイルを表示できます。

ログ・ファイルへのオペレーティング・システム・アクセス

ログ・ファイルは標準オペレーティング・システム・ファイルとして、Oracle Applications のインストール時に定義されたディレクトリに格納されます。

たとえば、Oracle General Ledger のファイルは、APPLCSF 変数が設定されている場合には、\$GL_TOP/\$APPLLOG または \$APPLCSF/\$APPLLOG というパス変数を使用して位置指定されます。

Oracle Applications のログ・ファイルへアクセスするための完全なパス名は、使用しているオペレーティング・システムによって異なります。ただし、すべてのプラットフォームに共通した標準的なファイル命名規則は多数存在します。

たとえば、次のとおりです。

VMS	L64225.REQ
UNIX	l64225.req

コンカレント・マネージャ・ログ・ファイルへのオペレーティング・システム・アクセス

コンカレント・マネージャ・ログ・ファイルは、FND_TOP (Application Object Library Files) へのパス名を保存する変数)、または \$APPLTOP/\$APPLLOG の下にあるログディレクトリに保存されます。

UNIX でのコンカレント・マネージャ・ログ・ファイル命名規則は *wn.mgr* で、*n* は最高 3 桁までの番号です。

ほとんどのプラットフォームの場合、*n* は内部コンカレント・マネージャがコンカレント・マネージャへ割り当てた「コンカレント・プロセス」番号であり、内部コンカレント・マネージャのログ・ファイルから見つけることができます。

内部コンカレント・マネージャのログ・ファイル名は、オペレーティング・システムから STARTMGR コマンドを使用してコンカレント・マネージャを起動したときに指定されます。

参照:「[内部コンカレント・マネージャのオペレーティング・システムからの制御](#)」(4-53)

レポート出力ファイルへのオペレーティング・システム・アクセス

コンカレント・プログラムが生成したレポート出力ファイルは、標準的なオペレーティング・システム・ファイルとして、Oracle Applications のインストール時に定義されたディレクトリへ格納されます。

出力ファイルへのパス名

Oracle Applications ログ・ファイルへアクセスする完全なパス名は、使用するオペレーティング・システムによって異なります。しかし、すべてのプラットフォームに共通した標準的なファイル命名規則がいくつかあります。

- 個々の出力ファイル名には、コンカレント・プロセス機能によって割り当てられた一意の要求 ID が組み込まれます。

たとえば、次のとおりです。

VMS および UNIX JSMITH.64225

コンカレント要求のステータスの変更

ここでは、要求のフェーズおよびステータスを変更する方法、および保留要求や無効要求の優先順位を変更する方法について説明します。

要求のフェーズおよびステータスの変更

要求には、「保留（実行待ち）」、「実行中」、「完了」、「無効（実行不能）」の4つのフェーズがあります。各フェーズ内部での要求の状態をステータスといいます。

要求のステータスを変えると、「保留」、「実行中」または「無効」要求のフェーズを変更できます。

「保留」および「無効」の要求

「保留」および「無効」の要求は取り消すことができます。その場合、その要求のフェーズおよびステータスは、「完了 - 取消」となります。

「保留」および「無効」の要求は保留中にすることもできます。その要求のフェーズおよびステータスは「無効 - 保留中」になります。後でこの要求を選択し、保留を削除することにより、この処理を取り消すことができます。

「実行中」の要求

「実行中」の要求は終了することができます。その要求のフェーズおよびステータスは「完了 - 終了」になります。

要求のステータスの変更

「要求」ウィンドウを使用して、要求のステータスおよびその結果のフェーズを変更できます。

「保留」または「無効」要求の優先順位の変更

通常、要求は、開始時刻に従って発行順に実行するという基準で実行されます。ただし、優先順位の高い要求がある場合は開始時刻の早い要求よりも優先順位の高い要求が先に実行されます。

システム管理者は、「要求」ウィンドウを使用して、「保留」または「無効」要求の優先順位を変更することができます。

要求優先順位はアプリケーション・ユーザーに関連付けられる

ユーザーの要求の優先順位のデフォルト値は、システム管理者が各ユーザーの「コンカレント: 優先度」ユーザー・プロファイル・オプションで設定した値です。ユーザーが自分の要求の優先順位を変更することはできません。

コンカレント・プログラムについて優先順位が定義されている場合は、その優先順位の方がユーザーのプロファイル・オプションより優先されます。

- 優先順位の範囲は、1（最高）～99（最低）です。

- 標準のデフォルト順位は 50 です。
- 内部コンカレント・マネージャが発行するコンカレント・プログラムの優先順位はゼロ (0) で、これは他のすべての要求より優先されます。

提案: ある要求の優先順位を頻繁に変更する場合は、そのコンカレント・プログラムに専用の優先順位を割り当てることを検討してください。

関連項目

- 「[コンカレント要求タイプ](#)」ウィンドウ (4-84)
- 「[コンカレント・プロセスのユーザー・プロファイル設定値](#)」 (4-19)
- 「[コンカレント・プロセスの概要](#)」 (4-1)
- 「[コンカレント要求のライフ・サイクル](#)」 (4-6)
- 「[完了したコンカレント要求レポート](#)」 (4-31)

コンカレント・プロセス用のファイルおよび表の管理

この項では、オペレーティング・システムによって保存されているログ・ファイルや出力ファイルの数を維持する方法、およびコンカレント要求とコンカレント・マネージャ処理についての情報が格納されている Application Object Library データベース表の管理方法について説明します。

「コンカレント要求やマネージャ・データのパージ」プログラムを実行すると影響を受けるデータベース表は次のとおりです。

FND_CONCURRENT_REQUESTS

この表には、すべてのコンカレント要求の詳細な履歴が記載されています。

FND_RUN_REQUESTS

ユーザーがレポート・セットを登録すると、レポート・セットの中のレポートおよび各レポートのパラメータ値についての情報がこの表に格納されます。

FND_CONC_REQUEST_ARGUMENTS

この表には、コンカレント・マネージャが各プログラムの実行を開始するときに、そのプログラムに渡される引数が記録されます。

FND_DUAL

この表には、要求によりデータベース表が更新されないときの条件が記録されます。

FND_CONCURRENT_PROCESSES

この表には、Oracle Applications およびオペレーティング・システム処理についての情報が記録されます。

FND_CONC_STAT_LIST

この表には、コンカレント要求についてのランタイム・パフォーマンス統計が集計されます。

FND_CONC_STAT_SUMMARY

この表には、「コンカレント要求やマネージャ・データのパージ」プログラムによって生成されたコンカレント・プログラムのパフォーマンス統計が記録されます。「コンカレント要求やマネージャ・データのパージ」プログラムは、FND_CONC_STAT_LIST 表のデータを使って統計を算出します。

保守のための提案

MIS 部門とアプリケーション・ユーザーは、所属する組織に適したアーカイブおよびファイル保持の方針について合意しておく必要があります。ディスク・ドライブの容量が不足しないように、Oracle Applications のログ・ファイルおよび出力ファイルは、定期的に削除してください。

提案:「コンカレント要求やマネージャ・データのパージ」プログラムを一度実行して、プログラムを特定の時間間隔で自動的に発行することができます。

「コンカレント要求やマネージャ・データのパージ」プログラムを実行するタイミングについては、いくつかのガイドラインの例があります。これらのガイドラインのどれを採用するかは、自分の属するユーザー・グループの Oracle Applications の使用頻度によって決定してください。

- 通常の使用頻度であれば 30 日ごと。
- 使用頻度が高い場合は 2 週間ごと。
- AGE モードを使用する場合は、モード値を 5 に設定して、最近 5 日間のコンカレント要求データ、ログ・ファイル、レポート出力ファイルを保存します。

パージによる監査データの削除

コンカレント要求情報をパージすると、監査詳細が失われます。「サインオン監査コンカレント要求」レポートには、この監査情報を使用します。

関連項目

- 「「コンカレント要求やマネージャ・データのパージ」プログラム」(4-16)
- 「コンカレント・プロセスの概要」(4-1)
- 「コンカレント要求のライフ・サイクル」(4-6)
- 「完了したコンカレント要求レポート」(4-31)

「コンカレント要求やマネージャ・データのパージ」プログラム

次のデータやファイルを削除するには、このプログラムを使用します。

- オペレーティング・システムが保持している製品ディレクトリからの要求ログ・ファイル、コンカレント・マネージャ・ログ・ファイルおよびレポート出力ファイル。
- コンカレント要求およびコンカレント・マネージャ処理についての履歴情報を含む Application Object Library データベース表からのレコード（行）。

「コンカレント: 要求統計の収集」プロファイル・オプションを「YES」に設定している場合は、このプログラムを使用して各コンカレント・プログラムのパフォーマンス統計を計算します。

レポート・オプション

エンティティ

全て	コンカレント要求の履歴情報およびコンカレント・マネージャの履歴情報を記録しているデータベース表からレコードをパージし、さらにオペレーティング・システムから要求ログ・ファイル、マネージャ・ログ・ファイルおよびレポート出力ファイルをパージします。
マネージャ	データベース表からコンカレント・マネージャの履歴情報を記録したレコードを削除し、オペレーティング・システムからマネージャ・ログ・ファイルをパージします。
要求	データベース表からコンカレント要求の履歴情報を記録したレコードをパージし、オペレーティング・システムから要求ログ・ファイルおよびレポート出力ファイルをパージします。

モード

保存日数

コンカレント要求履歴、ログ・ファイルおよびレポート出力ファイルの保存日数を入力します。入力した数（単位：日数）より古いレコードは、ページ・プログラムによってすべて削除されます。

たとえば、「5」を入力すると、5日を超えたコンカレント要求履歴、ログ・ファイルおよびレポート出力ファイルはすべてページされます。

件数

保存対象となるコンカレント要求履歴、ログ・ファイルおよびレポート出力ファイルの（最新の）レコード数を入力します。ページ・プログラムによって、最も新しいレコードからユーザーが指定した数までのレコードが保存され、残りのレコードはすべてページされます。

たとえば、「5」を入力すると、最も新しい5つのコンカレント要求履歴レコード、要求ログ・ファイル、マネージャ・ログ・ファイル、レポート出力ファイルが保存され、残りのすべてのレコードはページされます。

モード値

モード = 保存日数のための日数またはモード = 件数のためのレコード数を定義するための値を入力します。有効な値は1～9999999です。

Oracle ID

コンカレント・プログラムが接続され、コンカレント要求レコードをページする Oracle ID、およびそれに関連するログ・ファイル、レポート出力ファイルを入力します。Oracle ID には、エンティティが「要求」または「すべて」のいずれかであれば関連項目があります。

たとえば、AP1 を入力すると、プログラムは、AP1 Oracle ID に接続するプログラムの実行要求に関連するすべての要求レコード、ログ・ファイルおよびレポート出力ファイルをページします。

ユーザー名

そのコンカレント要求レコードおよびそれに関連するログ・ファイル、レポート出力ファイルをページするアプリケーション・ユーザー名を入力します。エンティティが「要求」または「全て」のときには、ユーザー名には関連項目があります。

たとえば、JSMITH と入力すると、プログラムはユーザー JSMITH が発行した要求に関連するすべての要求レコード、ログ・ファイルおよびレポート出力ファイルをページします。

コンカレント要求レコードおよび関連するログ・ファイル、レポート出力ファイルをページしようとする職責に関連するアプリケーションを選択します。職責アプリケーションは職責

オプションとともに使用され、エンティティが「要求」または「全て」のいずれかであれば関連項目があります。

職責

コンカレント要求レコードおよび関連するログ・ファイル、レポート出力ファイルをパージしようとする職責を選択します。エンティティが「要求」または「全て」のいずれかであれば職責には関連項目があります。

たとえば、システム管理者の職責を選択すると、システム管理者の職責を持つユーザーが発行した要求と関連するすべての要求レコード、ログ・ファイルおよびレポート出力ファイルがパージされます。

プログラム・アプリケーション

コンカレント要求レコードおよび関連するログ・ファイル、レポート出力ファイルをパージしようとするアプリケーションを選択します。プログラム・アプリケーションには、エンティティが「要求」または「全て」のいずれかであれば関連項目があります。

たとえば、Oracle Payables を選択すると、Oracle Payables プログラムの実行要求に関連するすべての要求レコード、ログ・ファイルおよびレポート出力ファイルがパージされます。

プログラム

コンカレント要求レコードおよび関連するログ・ファイル、レポート出力ファイルをパージするプログラムを選択します。プログラムには、エンティティが「要求」または「全て」のときには関連項目があります。

たとえば、Program X を選択すると、ページ・プログラムが Program X の実行要求に関連するすべての要求レコード、ログ・ファイルおよびレポート出力ファイルを削除します。

マネージャ・アプリケーション

コンカレント要求レコードおよび関連するログ・ファイル、レポート出力ファイルをパージしようとするコンカレント・マネージャに関連するアプリケーションを選択します。

マネージャ・アプリケーションはマネージャ・オプションとともに使用され、エンティティを「要求」に設定したときと「マネージャ」または「全て」のいずれかに設定したときでは異なる働きをします。

- エンティティを「要求」に設定すると、マネージャ・オプションで指定されたコンカレント・マネージャが実行する要求に関連したすべての要求レコード、ログ・ファイルおよびレポート出力ファイルがパージされます。
- エンティティを「マネージャ」または「全て」に設定したときには、プログラムは上記の働き以外にマネージャ・オプションに指定されたコンカレント・マネージャに関連するすべてのマネージャ・ログ・ファイルもパージします。

マネージャ

コンカレント要求レコードおよび関連するログ・ファイル、レポート出力ファイルをパージしようとするコンカレント・マネージャを選択します。

マネージャはマネージャ・アプリケーション・オプションとともに使用され、エンティティを「要求」に設定したときと、「マネージャ」または「全て」のいずれかに設定したときでは異なる働きをします。

- エンティティを「要求」に設定すると、マネージャ・オプションで指定されたコンカレント・マネージャが実行する要求に関連したすべての要求レコード、ログ・ファイルおよびレポート出力ファイルがパージされます。
- エンティティを「マネージャ」または「全て」に設定したときには、プログラムは上記の働き以外にマネージャ・オプションに指定されたコンカレント・マネージャに関連するすべてのマネージャ・ログ・ファイルもパージします。

レポート

「コンカレント要求やマネージャ・データのパージ」プログラムがパージしたレコード数についてのレポートが必要かどうかを指定します。

- | | |
|-----|-----------------------------|
| NO | プログラムは実行されますが、レポートは生成されません。 |
| YES | プログラムが実行され、レポートが生成されます。 |

その他のパージ

FND_DUAL 表からレコードを削除するかどうかを選択します。

- | | |
|-----|-------------------------|
| NO | FND_DUAL からレコードを削除しません。 |
| YES | FND_DUAL からレコードを削除します。 |

関連項目

- 「[コンカレント・プロセスの概要](#)」 (4-1)
- 「[コンカレント要求のライフ・サイクル](#)」 (4-6)

コンカレント・プロセスのユーザー・プロファイル設定値

コンカレント要求の発行に関連したユーザー・プロファイル・オプションの設定について説明します。

コンカレント・プロセス・オプションの設定

エンド・ユーザーは、各自のコンカレント要求について複数の実行時オプションを制御できます。たとえば、要求の開始日を特定できます。

ユーザーが、要求時にこれらのオプションを明示的に入力しなかった場合は、コンカレント・プロセス・オプションのデフォルト値として、そのユーザーのユーザー・プロファイルの中の値が使用されます。

システム管理者は、「システム・プロファイル値」ウィンドウを使用して、エンド・ユーザーのユーザー・プロファイル値を設定します。システム管理者とエンド・ユーザーのどちらも、「個別プロファイル値」フォームを使用して、システム管理者自身のプロファイル値の一部を設定できます。

発行済要求のコンカレント・プロセス・オプションの変更

システム管理者またはユーザーは、「要求」ウィンドウを使用して、発行した要求のコンカレント・プロセス・オプションを、実行が開始されるまでの間に変更することができます。

- システム管理者は、どの要求のどのコンカレント・オプションでも変更できます。
- ユーザーは、各自の要求のほとんどのコンカレント・オプションを変更できます。
 エンド・ユーザーは、各自の要求の優先度を変更（または設定）することも、また、要求ログ・ファイルおよびレポート出力ファイルをオンラインで見るためのレポート・アクセス・レベルを変更（または設定）することもできません。

参照：「[ユーザー・プロファイル設定の概要](#)」(9-1)

コンカレント処理のユーザー・プロファイル・オプション

ユーザー・プロファイル・オプション	説明
コンカレント：要求の保持	「YES」を指定すると、コンカレント要求が保留になります。「NO」を指定すると、要求の優先度と開始時刻に従ってプログラムが開始されます。
コンカレント：複数タイム・ゾーン	「YES」を指定すると、クライアントを実行しているタイム・ゾーンに関係なく、要求が即時にスケジュールされます。
コンカレント：レポート・アクセス・レベル	要求の出力ファイル / ログ・ファイルのオンライン表示およびレポートの再印刷は、次のアクセス・レベルに応じて可能です。「職責」レベル：要求を発行した職責を使用しているすべてのユーザー。「ユーザー」レベル：要求を発行したユーザーのみ。
コンカレント：レポートの部数	各レポートの印刷出力部数。

ユーザー・プロファイル・オプション	説明
コンカレント：要求の優先度	通常、要求は開始時刻に従って、発行順に実行されます。優先度は要求開始時刻よりも優先されます。優先度の高い要求の方が、開始時刻の早い要求より先に開始されます。優先度の範囲は、1（最高）～99（最低）です。標準のデフォルト順位は50です。
コンカレント：要求開始時刻	要求の実行を開始できるようになる日付と時刻。開始日時が現在の日時であるか、またはそれ以前である場合は、要求はただちに実行できます。
コンカレント：出力の保存	「YES」を指定すると、コンカレント・プログラム出力が標準ファイル形式で保存されます。コンカレント・プログラムには、出力ファイルを生成しないものもあります。
コンカレント：順次要求	「YES」を指定すると、要求の開始日時に従って、一度に1つずつ（順次）要求が強制的に実行されます。「NO」を指定すると、コンカレント・プログラム間に互換性がある場合は、要求を並行して実行できます。
コンカレント：可能 TM 待ち	クライアントが、別の取引マネージャ（TM）に移って試してみる前に、特定の TM が使用可能になるまで待機する最大秒数を指定できます。
コンカレント：URL ライフタイム	このプロファイル・オプションは、要求出力の URL がシステムから削除されるまでの時間を分単位で指定するオプションです。
プリンタ	レポートの印刷先となるプリンタ。

表 11

コンカレント要求プロファイル・オプションの更新

システム管理者は、コンカレント・ユーザー・プロファイル・オプションのほとんどを、4つのどのレベルでも設定できます。4つのレベルとは、「サイト」、「アプリケーション」、「職責」および「ユーザー」です。ユーザー・プロファイル「コンカレント：レポート・アクセス・レベル」は、アプリケーション・レベルでは設定できません。

ユーザーは、ほとんどのコンカレント・プロセス・プロファイル・オプションのデフォルト値を変更できます。ただし、「コンカレント：要求の優先度」または「コンカレント：レポート・アクセス・レベル」は設定できません。

関連項目

- 「[コンカレント・プロセスの概要](#)」（4-1）

マネージャおよびマネージャの稼働シフトの定義

ここでは、コンカレント・マネージャの定義方法およびマネージャを使用可能にする時点の指定方法について説明します。

コンカレント・マネージャは、それ自体が、他のコンカレント・プログラムの実行を開始するコンカレント・プログラムです。アプリケーション・ユーザーがプログラムの実行要求を発行すると、その要求はすべての要求を登録するデータベース表に入ります。コンカレント・マネージャは、この表から要求を読み込んで、プログラムの実行を開始します。参照：[「コンカレント・マネージャ」ウィンドウ](#) (4-76)。

ここでは、マネージャをいつ使用可能にするのかを指定する方法、マネージャを使用して異なる時間帯に渡ってアプリケーション処理の作業負荷をバランスさせる方法、マネージャからコールする即時コンカレント・プログラムのライブラリを関連付ける方法について説明します。

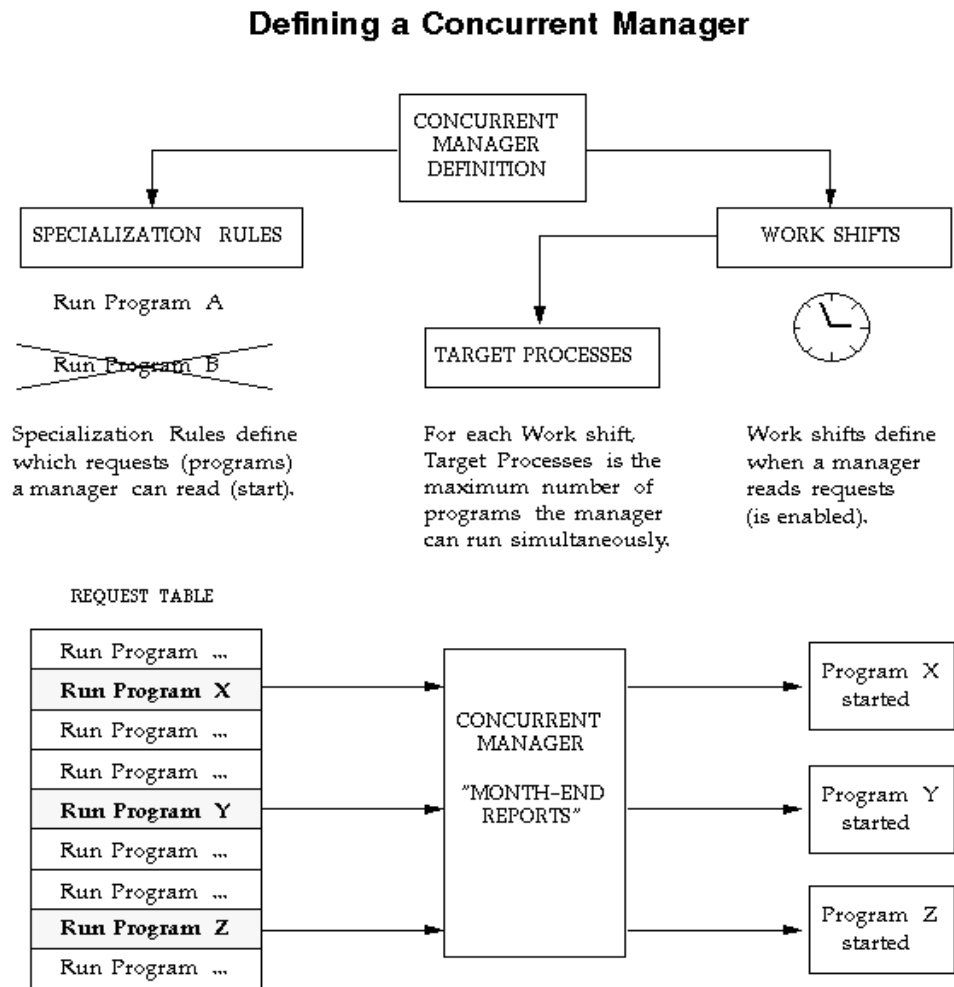
新しいマネージャの定義

コンカレント・マネージャは必要に応じていくつでも定義できます。マネージャを定義するときには、次のことを行います。

- 定義済みの即時コンカレント・プログラムのライブラリをマネージャに割り当てます。
即時コンカレント・プログラムは、コンカレント・マネージャに関連付けられているサブルーチンです。その他のすべてのコンカレント・プログラムは、実行時に独立プロセスとして生成されます。
- 稼働シフトをマネージャに割り当てます。稼働シフトは、マネージャが稼働する曜日および時間帯を決定します。
- 各稼働シフトごとに、稼働シフト時間内に要求を読み込む（プログラムを起動する）ためにマネージャが並行して実行できるオペレーティング・システム・プロセスの最大数を定義します。
- マネージャを、特定の種類の要求のみを読み込むように特殊化します。

[「図 12」](#) (4-23) はコンカレント・マネージャ定義の詳細を示しています。

図 12



プログラム・ライブラリ

生成されるプログラムに対して、コンカレント・マネージャは別のオペレーティング・システム・プロセスを開始または生成します。即時のプログラムは、コンカレント・マネージャのオペレーティング・システム・プロセスの一部として実行されます。

プログラム・ライブラリには、マネージャによりコールできる即時コンカレント・プログラムが含まれています。

即時コンカレント・プログラムは、プログラム・ライブラリに登録する必要があります。Oracle Application Object Library を使用するアプリケーション開発者は、コンカレント・プログラムをプログラム・ライブラリに登録できます。

Oracle Application Object Library FNDLIBR プログラム・ライブラリは、Oracle Applications 即時コンカレント・プログラムを格納し、標準コンカレント・マネージャに割り当てられています。FNDLIBR ライブラリをマネージャの定義に組み込むのが一般的です。

内部コンカレント・マネージャと標準コンカレント・マネージャ

Oracle System Administration には、ユーザーが使用できる定義済みのマネージャが2つあります。

- 内部コンカレント・マネージャ。これは他のすべてのマネージャを管理するマネージャです。内部コンカレント・マネージャは、個別マネージャの起動、そのステータスの検証、個別マネージャのリセットおよび停止を行います。

内部コンカレント・マネージャの定義は変更できません。

参照:「[プログラム非互換性ルールの定義](#)」(3-21)

- 標準マネージャ。標準マネージャはどのような要求もすべて受け入れます。特殊化はありません。標準マネージャは、常にアクティブです。つまり、1年365日、1日24時間、稼働しています。

警告: 標準コンカレント・マネージャの定義は変更しないでください。この定義を変更し、要求を受け入れる追加マネージャを定義していないと、一部のプログラムが実行できなくなる場合があります。標準マネージャは、いつでもどのような要求の実行にも使用可能なマネージャとして、つまり一種の安全対策として使用してください。導入サイト固有のニーズに対処するには、追加のマネージャを定義します。

取引マネージャ

従来のコンカレント・マネージャでは長時間実行のデータ集中型アプリケーション・プログラムを非同期で実行できるのに対して、取引マネージャはクライアント・マシンからの特定要求の同期処理をサポートします。クライアント・プログラムからサーバー側プログラム同期実行要求を発行すると、取引マネージャはプログラムを即時実行した後で、ステータスをクライアント・プログラムに返します。

取引マネージャは即時コンカレント・プログラムとして実施されます。実行時には、コンカレント・プロセスがこれらの多数のマネージャを起動します。コンカレント要求表をポーリングして次の処理を決定するかわりに、取引マネージャはクライアント・プログラムから信号を受け取るまで待機します。要求された取引プログラムはサーバー上で実行されます。最小限の遅延時間で実行されるため、クライアントはこの処理を意識する必要はありません。

プログラム実行終了時には、結果が完了メッセージと一連の戻り値でクライアント・プログラムに通知されます。

取引マネージャとの通信は自動で行われます。取引マネージャのメカニズムではクライアントと取引マネージャ・プロセス間の継続接続の確立は行いません。このメカニズムはサーバー・プロセスの小規模プールであり、多数のクライアントへのリアルタイム応答でのサービス供給を目的としています。

各取引マネージャはそれぞれのプログラム・ライブラリに含まれているプログラムのみを処理できます。Oracle Application Object Library を使用する Oracle Applications 開発者は、プログラム・ライブラリに取引プログラムを登録できます。

取引マネージャは、特定のデータ・グループに関連付けられていて、そのデータ・グループを使用してデータベースに接続します。取引マネージャは、同じデータ・グループに関連付けられている職責に基づいて発行された要求のみを処理できます。

カスタム・データ・グループを作成するときには、取引マネージャを使用するデータ・グループ内の各アプリケーションについて、(シード取引マネージャに関連付けられた定義済みプログラム・ライブラリを使用して) 新しい取引マネージャを定義してください。

関連項目

- 「[コンカレント・マネージャ](#)」ウィンドウ (4-76)
- 「[コンカレント・マネージャの管理](#)」ウィンドウ (4-71)
- 「[稼働シフト](#)」ウィンドウ (4-81)
- 「[マネージャ別稼働シフト・レポート](#)」 (4-32)
- 「[稼働シフト・レポート](#)」 (4-33)
- 「[稼働シフトによる処理作業負荷のバランス化](#)」 (4-28)
- 「[稼働シフトの定義](#)」 (4-25)
- 「[時間ベース・キューの使用方法](#)」 (4-29)

稼働シフトの定義

コンカレント・マネージャを定義するとき、1つ以上の稼働シフトをそのマネージャに割り当てます。稼働シフトによって、マネージャの稼働時期が決まります。稼働シフトを定義するには、「稼働シフト」フォームを使用します。

参照:

「[稼働シフト](#)」ウィンドウ (4-81)

「[マネージャ別稼働シフト・レポート](#)」 (4-32)

「稼働シフト・レポート」(4-33)

たとえば、次のような稼働シフトを定義できます。

- 8:00am ～ 5:00pm、月曜日～金曜日
- 11:00am ～ 1:00pm、(毎週) 水曜日
- 6:00pm ～ 11:59pm、2000 年 4 月 15 日

従業員の退社後、端末が空いている夜間に実行する稼働シフトを定義することができます。たとえば、次のような稼働シフトを定義できます。

- 2:00am ～ 6:00am、月曜日～金曜日

特定の曜日または特定の日付に 1 日 24 時間実行する稼働シフトを定義できます。たとえば、次のような稼働シフトを定義できます。

- 月曜日～金曜日
- (毎週) 水曜日
- 2000 年 4 月 15 日

特別の場合にのみ使用する稼働シフトを定義できます。たとえば、会社で棚卸を実施するときに使用する「棚卸」という名前の稼働シフトを定義できます。

稼働シフトの使用禁止

稼働シフトとしてある期間を定義した後、必ずしもその稼働シフトを使用しない場合は、次のような措置が可能です。

- その稼働シフトをコンカレント・マネージャに割り当てない
- 「コンカレント・マネージャ」フォームで、その稼働シフトにターゲット・プロセス数として 0 (ゼロ) を指定する
- 「コンカレント・マネージャ」フォームを使用して稼働シフト割当を削除する

稼働シフトおよび時間

稼働シフトは、午前零時から翌日の午前零時まで 1 日 24 時間実行できます。これは、24 時記法では次のように定義されます。

- 12:00am00:00:00
- 11:59:59pm23:59:59

一晩中実行する稼働シフト

24 時間時計の開始時刻および終了時刻は午前零時です。

稼働シフトの稼働時間が 24 時間実行より短く、プログラムは午前零時をまたいで継続させる場合は、次のような 2 つの稼働シフトを定義して、このマネージャに割り当ててください。

- 最初の稼働シフトは 23:59（11:59pm）に終了。
- 2 番目の稼働シフトは 00:00（12:00 am）に開始。

たとえば、ほとんどの従業員が就業していない夜間に、データ集約型プログラムを実行する場合は、このマネージャに割り当てる 2 つの稼働シフトを定義します。

- 最初の稼働シフトは、20:00（8:00pm）に始まり、23:59（11:59pm）に終了。
- 2 番目の稼働シフトは、00:00（12:00am）に始まり、05:00（5:00am）に終了。

稼働シフトのオーバーラップ - 優先度レベル

オーバーラップする稼働シフトをコンカレント・マネージャに割り当てる場合は、オーバーラップしている期間については**より限定的な時間**が指定されている稼働シフトが有効になります。たとえば、7 月 4 日の指定のある稼働シフトは、月曜日から金曜日の 9:00am から 5:00pm までの指定のある稼働シフトより優先します。

「表 12」（4-27）はオーバーラップする稼働シフトの優先度レベルを降順でリストしたものです。特定の日付および時間範囲が指定されている稼働シフトは、最高位の優先度になります。「標準」稼働シフトが最低優先度になります。

稼働シフト優先度のオーバーラップ

優先度	稼働シフトの定義	例
1	特定の日付および時間範囲の指定	2000 年 4 月 15 日 8:00am ～ 5:00pm
2	日付の指定（時間範囲の指定なし）	2000 年 4 月 15 日
3	期間および時間範囲の指定	月曜～金曜 8:00am ～ 5:00pm
4	期間の指定（時間範囲の指定なし）	月曜～金曜
5	時間範囲の指定（日付または期間の指定なし）	8:00am ～ 5:00pm
6	標準稼働シフト。日付、期間、時間の指定なし。	標準稼働シフトは、1 年 365 日、1 日 24 時間です。

表 12

優先度が同じでオーバーラップする稼働シフト

オーバーラップする稼働シフトがあり、その優先度が同じレベルの場合は、**ターゲット・プロセス数が最大の稼働シフト**が有効になります。

たとえば、曜日範囲も時間範囲も同じ 2 つの稼働シフトがあるとします。月曜日から金曜日の 9:00am から 5:00pm までの「営業日」稼働シフトがあり、そのターゲット・プロセスは 4 つです。

また、月曜日から金曜日の 11:00am から 1:00pm までの「昼食」稼働シフトがあり、そのターゲット・プロセスは 8 つです。

「昼食」稼働シフトは、11:00am ～ 1:00pm（月～金）に有効になります。この方がターゲット・プロセス数が大きいからです。

関連項目

- 「マネージャおよびマネージャの稼働シフトの定義」(4-22)
- 「稼働シフトによる処理作業負荷のバランス化」(4-28)
- 「時間ベース・キューの使用法」(4-29)

稼働シフトによる処理作業負荷のバランス化

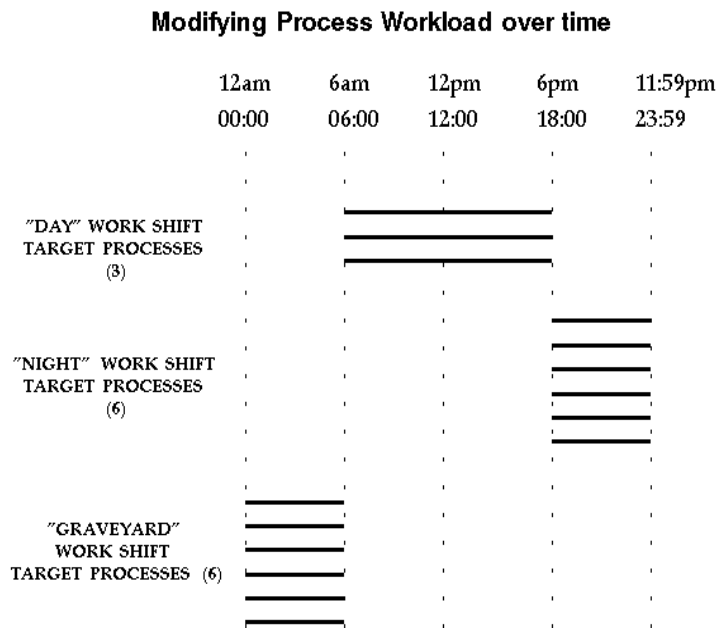
マネージャの定義には、マネージャが要求の読取りに使用できるオペレーティング・システム・プロセスの数が含まれます。マネージャは、ターゲット・プロセスとして参照されるこれらの各プロセスに対して、コンカレント・プログラムを 1 つ起動できます。

マネージャに割り当てる各稼働シフトについて、いくつかのターゲット・プロセスを定義します。

ターゲット・プロセス数の異なる稼働シフトを使用することにより、コンカレント・プロセスの作業負荷を曜日、時刻、特定の日付に従って修正できます。

下の図は、3 つの稼働シフトを使用して、3 つのプログラムを 6:00am ～ 6:00pm の間並行して実行し、6 つのプログラムを 6:00pm ～ 6:00am の間並行して実行できるようにマネージャを定義する方法を示しています。

図 13



関連項目

- 「稼働シフト」 ウィンドウ (4-81)
- 「マネージャおよびマネージャの稼働シフトの定義」 (4-22)
- 「マネージャ別稼働シフト・レポート」 (4-32)
- 「稼働シフト・レポート」 (4-33)
- 「稼働シフトの定義」 (4-25)
- 「時間ベース・キューの使用法」 (4-29)

時間ベース・キューの使用法

プログラムを実行するマネージャを、過去の典型的な実行時間に基づいて定義すると、複数の時間ベース・キューを作成できます。つまり、要求の実行に要する時間に従って要求を区分し、マネージャを特殊化できます。

そのためには、システム管理者のレポート・セキュリティ・グループの完了したコンカレント要求レポートを使用します。このレポートには、実行を完了したコンカレント・プログラムについて、実際の開始日時および実際の完了日時が記載されます。参照：「[完了したコンカレント要求レポート](#)」(4-31)。

提案：作業負荷が異なる時間帯での処理時間を決定するには、コンカレント・プログラムを異なる時間条件で（たとえば深夜、その後で昼間もう一度）実行します。

たとえば、実際の完了時刻に基づき、次の種類のプログラムを実行する各種マネージャを特殊化できます。

- 在庫棚卸し
- 満期小切手処理
- 転記
- 請求書インポート

この方法を補強するために、「オーバーフロー」マネージャを定義します。たとえば、上記のマネージャの1つ（または複数）を対象としたプログラムを受け付けることができ、稼働シフトが2:00am-4:00am（02:00-04:00）などに制限されたマネージャがオーバーフロー・マネージャの例です。長時間実行プログラムの中に「オーバーフロー」稼働シフトの開始前に起動されなかったものがあった場合は、追加マネージャが使用可能となり、そのようなプログラムを受け入れます。

そして、必須要求について定義された「例外」マネージャでさらに補強を図ります。たとえば、次のプログラムを実行できるマネージャを定義します。

- 特定の時間までに完了する必要がある特定のプログラム。「必須」マネージャは、特定プログラムについての要求のみを読み取るように特殊化することができます。
- 特定のユーザー（たとえば会社の管理部長）が発行したプログラム。1人のアプリケーション・ユーザーからの要求のみを読み取るように、マネージャを特殊化することができます。ユーザーがサインオンに使用する優先順位の高い第2のユーザー名を定義することもできます。

関連項目

- 「[コンカレント・マネージャ](#)」ウィンドウ (4-76)
- 「[コンカレント・マネージャの管理](#)」ウィンドウ (4-71)
- 「[稼働シフト](#)」ウィンドウ (4-81)
- 「[コンカレント・プロセスの概要](#)」 (4-1)
- 「[プログラムの要求タイプ別グループ化](#)」 (4-49)
- 「[マネージャおよびマネージャの稼働シフトの定義](#)」 (4-22)

- 「マネージャ別稼動シフト・レポート」 (4-32)
- 「稼動シフト・レポート」 (4-33)
- 「稼動シフトによる処理作業負荷のバランス化」 (4-28)
- 「稼動シフトの定義」 (4-25)
- 「特定なプログラムのみを実行するためのマネージャの特殊化」 (4-33)

完了したコンカレント要求レポート

このレポートは、コンカレント・プログラムの実行に実際に要した時間を表示します。このレポートは、特定のプログラムの要求のみを読み取るようにマネージャを特殊化することによって、代表的な完了時刻に基づいて要求を区分する場合に使用します。

このレポートを使用して、実行の完了したコンカレント・プログラムに関連するパラメータおよびエラー・メッセージを記録してください。

レポート・パラメータ

パラメータを入力しなかった場合は、レポートは、すべての完了したコンカレント要求について値を戻します。

プログラム・アプリケーション名

レポートに表示する完了したコンカレント要求を含むプログラムに関連付けられているアプリケーション名を選択します。

1つのアプリケーションに関連するすべての完了したコンカレント要求についてのレポートを実行するには、アプリケーション名のみを選択し、プログラム名は選択しないでください。

プログラム名

レポートに表示する完了したコンカレント要求を含むプログラムの名前を選択します。「プログラム名」に値を入力する前に、「プログラム・アプリケーション名」に値を入力する必要があります。

ユーザー名

レポートに表示する完了したコンカレント要求を出したアプリケーション・ユーザーの名前を選択します。

開始日 / 終了日

レポートの開始時刻および終了時刻を入力します。

レポート見出し

レポート見出しには、指定されたパラメータが出力され、レポートの内容についての一般情報が得られます。

関連項目

- [「コンカレント・プロセスの概要」](#) (4-1)
- [「コンカレント・プロセス用のファイルおよび表の管理」](#) (4-14)
- [「プログラムの要求タイプ別グループ化」](#) (4-49)
- [「時間ベース・キューの使用法」](#) (4-29)
- [「特定のプログラムのみを実行するためのマネージャの特殊化」](#) (4-33)

マネージャ別稼働シフト・レポート

このレポートは、各コンカレント・マネージャに割り当てられた稼働シフトを出力します。コンカレント・マネージャを定義または編集する場合は、このレポートを使用します。

レポート・パラメータ

なし。

レポート見出し

レポート見出しには、レポートの内容についての一般情報が出力されます。

関連項目

- [「「コンカレント・マネージャ」ウィンドウ」](#) (4-76)
- [「「コンカレント・マネージャの管理」ウィンドウ」](#) (4-71)
- [「コンカレント・プロセスの概要」](#) (4-1)
- [「マネージャおよびマネージャの稼働シフトの定義」](#) (4-22)
- [「稼働シフト・レポート」](#) (4-33)
- [「時間ベース・キューの使用法」](#) (4-29)
- [「特定のプログラムのみを実行するためのマネージャの特殊化」](#) (4-33)

稼働シフト・レポート

このレポートには、すべての稼働シフト定義が出力されます。コンカレント・マネージャ稼働シフトを定義または編集する場合は、このレポートを使用します。

レポート・パラメータ

なし。

レポート見出し

レポート見出しには、レポートの内容についての一般情報が出力されます。

関連項目

- 「「コンカレント・マネージャ」ウィンドウ」(4-76)
- 「「コンカレント・マネージャの管理」ウィンドウ」(4-71)
- 「「稼働シフト」ウィンドウ」(4-81)
- 「コンカレント・プロセスの概要」(4-1)
- 「マネージャおよびマネージャの稼働シフトの定義」(4-22)
- 「マネージャ別稼働シフト・レポート」(4-32)
- 「稼働シフト・レポート」(4-33)
- 「稼働シフトの定義」(4-25)

特定のプログラムのみを実行するためのマネージャの特殊化

ここでは、特定のプログラムのみを実行するようにマネージャを特殊化する方法について説明します。

特殊化ルールの概要

ユーザーがコンカレント・プログラムの実行を要求するたびに、その要求がデータベース表に挿入されます。コンカレント・マネージャはこの表から要求を読み込み、特定の要求を読み込むように定義されているマネージャがプログラムの実行を開始します。

特殊化ルールがない場合、マネージャは任意のコンカレント・プログラムを起動する要求を読み取ります。

特殊化ルールを使用すると、マネージャを特殊化して、コンカレント・プログラムを起動するための特定の種類の要求のみを読み込むことができます。たとえば、Oracle General Ledger プログラムの起動要求のみ、またはユーザー「Fred」が要求するプログラムの起動要求のみを読み込むようにできます。参照：「[コンカレント・マネージャ ウィンドウ](#)」(4-76)。

特殊化ルールの特別のタイプとして、組合せ特殊化ルールがあります。これは、複数の処理を組み合わせて1つのルールを定義します。参照：「[組合せ特殊化ルール ウィンドウ](#)」(4-82)。

関連項目

- 「[コンカレント・マネージャ ウィンドウ](#)」(4-76)
- 「[コンカレント・マネージャの制御](#)」(4-50)
- 「[組合せ特殊化ルールの定義](#)」(4-43)
- 「[特殊化ルールの定義](#)」(4-34)
- 「[例 - 特殊化ルールの使用](#)」(4-38)

特殊化ルールの定義

特殊化ルールは、処理を要求のタイプと関連付けます。処理には含むと除くの2種類があります。

- 含むは、指定されたタイプの要求のみを読み取るようにマネージャを定義する処理です。
- 除くは、指定されたタイプ以外の要求をすべて読み取るようにマネージャを定義する処理です。

コンカレント・プログラムの実行要求は、次のものに基づいて許可または禁止できます。

- 複数組織を使用している場合は、要求の「会計帳簿」（複数インストール用）または「組織」の ORACLE ID。
- プログラムの本体またはプログラムのアプリケーション。
- プログラムの要求タイプ。
- 要求を発行したユーザー。
- 複数の処理を組み合わせた組合せルール。組合せルールは、その処理を複数の要求タイプに適用します。

たとえば、組合せルールで Oracle ID からの処理を除外したり、特定のプログラムからの他の処理を除外することができます。

複数のルールの使用

ルールはそれぞれ 1 つの処理を実行します。複数のルールを使用すると、各ルールは次のように評価されます。

- 含むルールは、結合論理として OR 文を使用して一括評価されます。

たとえば、次のルールを使用した場合を考えます。

- 含む X
- 含む Y

ルールの結果によりそのマネージャでは X または Y を実行できますが、両方のプログラムを実行する必要はありません。

- 除くルールは、結合論理として AND 文を使用して一括評価されます。

たとえば、次のルールを使用した場合を考えます。

- 除く 1
- 除く 2

ルールの結果によりそのマネージャではプログラム 1 および 2 の同時実行または個別実行が禁止されます。

- 含むルールが先に評価され、続いて除くルールが評価されます。AND 文では含むルールと除くルールが同時に評価されます。たとえば (含む X または Y) AND (除く 1 および 2) のように指定できます。
- " 除く " ルールは " 含む " ルールを上書きします。

特殊化ルールの処理、その結合論理および例を次の 2 つの表に示します。参照：「[特殊化ルールの論理 - 例](#)」(4-35)。

特殊化ルールの論理 - 例

含むルール	結果
含む X	プログラム X のみ実行
含む X	プログラム X を実行
OR	... または
含むユーザー Sam	ユーザー Sam による要求を実行
	結果：プログラム X についてすべてのユーザーの要求が実行され、Sam のすべての要求が実行されます。
表 13	

除くルール	結果
除く 37	プログラム 37 は実行しない
除く 37	プログラム 37 は実行しない
AND	... および
除くユーザー Sam	ユーザー Sam による要求は実行しない
	結果：プログラム 37 に対する要求は一切実行されず、さらに Sam の要求も一切実行されません。
表 14	

含むルールおよび除くルール	結果
含むユーザー Sam	ユーザー Sam による要求のみ実行
AND	... および
除く 37	プログラム 37 は実行しない
	結果：プログラム 37 の実行要求を除いて、Sam の要求はすべて実行されます。
含む X	(プログラム X を実行
OR	... または
含むユーザー Sam	ユーザー Sam による要求を実行)
----- AND	... および
除く 37	(プログラム 37 は実行しない
AND	... および
含むユーザー Mary	ユーザー Mary による要求は実行しない)
	結果：Mary が要求したとき以外はプログラム X が実行され、Sam の要求は、プログラム 37 の実行要求以外はすべて実行されます。

「例 - 特殊化ルールのタイプ」(4-37) は特殊化ルールに関連付けられている処理タイプの例を示しています。

例 - 特殊化ルールのタイプ

ルール処理	タイプ	例	説明
含む	組合セルール	Oracle Project Accounting - Tim の予算	マネージャは、組合セルール「Tim の予算」で定義されたプログラム起動要求のみを読み取ります。
	ORACLE ID	APPS2	マネージャは、APPS2（複数インストール・スキーマ内の単一インストール）Oracle ID と関連付けられたプログラムを起動する要求のみを読み取ります。
	プログラム	Oracle Project Accounting - 販売予測	マネージャは、「販売予測」という名前のコンカレント・プログラムを起動する要求のみを読み取ります。
	要求タイプ	Oracle Inventory - 夜間レポート	マネージャは、要求タイプ「夜間レポート」に属するプログラムを起動する要求のみを読み取ります。
	ユーザー	Tim	マネージャは、アプリケーション・ユーザー「Tim」が発行したプログラム起動要求のみを読み取ります。
除く	組合セルール	Oracle Project Accounting - 月末レポート	マネージャは、組合セルール「月末レポート」で定義された要求以外のプログラム起動要求をすべて読み取ります。
	ORACLE ID	APPS2	マネージャは APPS2 Oracle ID と関連付けられた要求以外のプログラム起動要求をすべて読み取ります。
	プログラム	Application Object Library - 監査表のページ	マネージャは、「監査表のページ」という名前のプログラムについての要求を除き、すべてのプログラム起動要求を読み取ります。
	要求タイプ	Oracle Purchasing - 週末プログラム	マネージャは、要求タイプ「週末プログラム」に属するプログラム起動要求をすべて読み取ります。
	ユーザー	Margaret	マネージャは、アプリケーション・ユーザー「Margaret」が発行した要求以外のプログラム起動要求をすべて読み取ります。
表 15			

関連項目

- 「[コンカレント・マネージャ](#)」ウィンドウ (4-76)
- 「[組合せ特殊化ルール](#)」ウィンドウ (4-82)
- 「[組合セルールの使用](#)」 (4-45)
- 「[組合せ特殊化ルールの定義](#)」 (4-43)
- 「[特殊化ルールと組合セルールの違い](#)」 (4-48)
- 「[特定のプログラムのみを実行するためのマネージャの特殊化](#)」 (4-33)
- 「[例 - 特殊化ルールの使用](#)」 (4-38)

例 - 特殊化ルールの使用

次に示すのは、特殊化ルールを使用してコンカレント・マネージャが読み取ることのできる要求を定義する例です。複数のルールを使用してマネージャを特殊化する場合は、ルールとルールの間を OR および AND で結合し、複数の特殊化ルールの関連を明確にします。

含む処理および除く処理の使用

含む	Program - Oracle Assets、「名前」フィールドにデータなし
結果	マネージャはアプリケーション Oracle Assets のコンカレント・プログラム実行要求のみを読み取ります。
含む	Program - Oracle Assets、「名前」フィールドにデータなし
OR	
含む	Program - Oracle Payables、「名前」フィールドにデータなし
結果	マネージャはアプリケーション Oracle Assets またはアプリケーション Oracle Payables のコンカレント・プログラム実行要求のみを読み取ります。 複数の含む処理を使用すると、1 つのプログラム（1 回の含む処理）の範囲を超えて要求を読み取ることができるようにマネージャの機能が拡張されます。
除く	Oracle ID - APPS2
結果	マネージャは Oracle ID APPS2 に関連付けられたプログラムを除き、Oracle ID に関連付けられたすべてのコンカレント・プログラムの実行要求を読み取ります。
除く	Oracle ID - APPS2

AND

除く Program - Oracle Payables、「名前」フィールドにデータなし
結果 マネージャは Oracle ID APPS2 に関連付けられたプログラムおよびアプリケーション Oracle Payables 用のプログラムを除き、Oracle ID に関連付けられたすべてのコンカレント・プログラムの実行要求を読み取ります。

作業の単純化

複数のルールは必ずしも必要ではありません。また、ルールを減らしたり簡単にしたりもできます。次の例を見てください。

含む Program - Oracle Sales and Marketing、「名前」フィールドにデータなし

OR

含む Request Type - Sales Forecasts
結果 マネージャはアプリケーション Oracle Sales and Marketing 用のコンカレント・プログラムまたは要求タイプが Sales Forecasts になっているプログラムの実行要求のみを読み取ります。
この例では、すべてが Oracle ID OSM に関連付けられている要求タイプ Sales Forecasts に属するプログラムの場合、両方のルールが必要ではありません。この場合、2 番目の Type 含むルールは必要ありません。

除くルールは含むルールに優先

含む Program - Oracle Payables、「名前」フィールドにデータなし

AND

除く Program - Oracle Payables Invoice Aging Report
結果 マネージャはアプリケーション Oracle Payables のコンカレント・プログラムに対する要求をすべて読み取りますが、Oracle Payables のプログラム Invoice Aging Report の実行要求は読み取れません。

含む Program - Signon Audit Forms

AND

除く Request Type - Signon Audit Reports

結果 システム管理者のプログラム Signon Audit Forms が要求タイプ Signon Audit Reports に属している場合は、含むルールで特定の条件に指定されていても、マネージャはプログラム実行要求を読み取りません。除くルールは含むルールを上書きします。

特定の Oracle ID についてのみプログラムを実行するための特殊化

次の例では、特定の Oracle ID についてのみプログラムを実行するようにマネージャを特殊化できます。これは、同じ Oracle アプリケーションをインストールした複数のシステムがある場合に役立ちます。

含む Program - Oracle Payables Invoice Aging Report

AND

除く Oracle ID - APPS2

結果 プログラムが Oracle ID APPS2 に関連付けられていない場合は、マネージャは Oracle Payables のプログラム Invoice Aging Report の実行要求のみを読み取ります。除く処理は含む処理を上書きします。

ただし、Invoice Aging Report が APPS などの別の Oracle ID に対して実行されると、このマネージャはプログラム実行要求を読み取ります。これは、同じアプリケーションおよびデータ・グループを複数のシステムにインストールして作業する場合に役立ちます。

要求タイプによるプログラムの区別

マネージャを特殊化して、個別に識別して指定したプログラムを除き、ある要求タイプに属するすべてのプログラムを実行する要求を読み込むようにすることができます。

含む Request Type - Oracle General Ledger Reports

AND

除く Program - Oracle General Ledger Account Analysis

結果 Account Analysis プログラムが要求タイプ Oracle General Ledger Reports に属している場合、このマネージャは、Account Analysis プログラムを除いて、要求タイプ Oracle General Ledger Reports に属しているすべてのプログラムを実行します。

特定のプログラムの実行防止

除く処理は2回以上使用できます。たとえばマネージャが特定のアプリケーション用のコンカレント・プログラムの実行要求をすべて読み取るにもかかわらず、特定の2つのプログラムについてはマネージャで実行しないようにする場合などが挙げられます。次の操作を実行できます。

含む Program - Oracle General Ledger、「名前」フィールドにデータなし

AND

除く Program - Oracle General Ledger Consolidation Audit

AND

除く Program - Oracle General Ledger Consolidation Rules

結果 マネージャはプログラム Consolidation Audit と Consolidation Rules を除いて、アプリケーション Oracle General Ledger 用のコンカレント・プログラムの要求をすべて読み取ります。

特定の時期に特定のプログラムのみを実行するための特殊化

複数の含むルールを使用することによって、特定のプログラムのみを実行するようにマネージャを特殊化できます。その後で、マネージャの稼働シフトを定義するときに、そのプログラムの実行要求をマネージャが読み込む時点を制御できます。

含む Program - Oracle Payables Invoice Aging Report

OR

含む Program - Oracle Purchasing Receipt Accruals

結果 マネージャは、Oracle Payables Invoice Aging Report プログラムまたは Oracle Purchasing Receipt Accruals プログラムを実行する要求のみを読み取ります。

提案: この2つのレポートを夜間のみ実行するには、マネージャ実行用の稼働シフトを 2:00am ~ 6:00am (02:00 ~ 06:00) に定義できます。

提案: プログラムの実行要求を初めて発行する場合は、再発行の間隔を（たとえば1 箇月と）定義して、プログラムの実行を毎月再発行できます。

アプリケーション・ユーザーごとの特殊化

特定のユーザーからの要求のみを読み込むように、マネージャを特殊化できます。

含む ユーザー - Markus Kalkin

結果 マネージャは、アプリケーション・ユーザー Markus Kalkin が発行した要求のみを読み込みます。

含む ユーザー - Markus Kalkin

OR

含む プログラム - Oracle Inventory Process Demand Interface

OR

含む プログラム - Oracle Inventory Process Demand Histories

結果 マネージャは、ユーザー Markus Kalkin が発行した要求と、Oracle Inventory プログラム Process Demand Interface および Summarize Demand Histories の実行要求の両方を読み込みます。

提案: 特定のユーザーが発行した特定のプログラムが、実行を待っている他の要求を "飛び越す" ようにするには、上の例に示す方法でマネージャを定義し、そのユーザーについてユーザー・プロファイル・オプション「コンカレント: 優先順位」を高い優先順位に設定します（「コンカレント: 優先順位」は、ユーザーが発行する要求の優先順位を指定します）。

- マネージャを定義し、そのマネージャに付加名を与えます。
- マネージャを上 の例のように特殊化します。
- ユーザー Markus のユーザー・プロファイル・オプション「コンカレント: 優先度」を 10 に設定します。

関連項目

- 「[「コンカレント・マネージャ」ウィンドウ](#)」 (4-76)
- 「[「組合せ特殊化ルールの定義](#)」 (4-43)
- 「[特殊化ルールの定義](#)」 (4-34)
- 「[「特定のプログラムのみを実行するためのマネージャの特殊化](#)」 (4-33)

組合せ特殊化ルールの定義

組合せ特殊化ルールは複数の処理を組み合わせて 1 つのルールを生成します。処理は AND 文で結合されるため、ルールは次のように定義されます。

処理 1 AND ...
処理 2 AND ...
処理 3 AND ... so on.

複雑なルールを毎回複製するかわりに組合せルールを作成し、それを複数のマネージャで使用できます。

組合せルールの作成に使用できる処理には除くを含むの 2 種類があります。各処理は、ルール内でそれぞれ 1 行で定義します。特殊化を指定する行または個別処理を組み合わせることにより、組合せルール全体を定義します。

除く処理は含む処理を上書きします。

たとえば、アプリケーション・プログラム x の除く処理とユーザー Yvonne Jonese の含む処理を定義できます。この 2 つの処理を組み合わせると、「プログラム x の実行要求を除いて、ユーザー Yvonne Jones からすべての要求を読み込む」という組合せルールが生成されます。参照：「[「組合せ特殊化ルール」ウィンドウ](#)」(4-82)。

組合せ特殊化ルール処理およびその結合論理と例を次の表に示します。

特殊化ルールの論理 - 例

含む行の組合せルール	結果
含む <i>Program X</i>	プログラム X のみ実行
含む <i>Program X</i>	プログラム X を実行
AND	... および
含むユーザー <i>Sam</i>	ユーザー Sam による要求を実行
	結果：プログラム X について Sam の要求のみが実行されます。
表 16	

除く行の組合せルール	結果
除く <i>Program 37</i>	プログラム 37 は実行しない
除く <i>Program 37</i>	プログラム 37 は実行しない

除く行の組合せルール	結果
AND	... および
除くユーザー <i>Sam</i>	ユーザー <i>Sam</i> による要求は実行しない
	結果：プログラム 37 に対する要求は一切実行しない。また <i>Sam</i> の要求も実行しない。
表 17	

含む行および除く行の組合せルール	結果
含むユーザー <i>Sam</i>	ユーザー <i>Sam</i> による要求を実行
AND	... および
除く <i>Program 37</i>	プログラム 37 は実行しない
	結果：プログラム 37 の実行要求を除いて、 <i>Sam</i> の要求はすべて実行されます。
含む <i>Program Application General Ledger</i>	(General Ledger プログラムを実行
AND	... および
含むユーザー <i>Sam</i>	ユーザー <i>Sam</i> による要求を実行)
----- AND	... および
除く <i>Program 37</i>	(プログラム 37 は実行しない
AND	... および
除く <i>Program 38</i>	プログラム 38 は実行しない)
	結果：プログラム 37 および 38 を除いて、アプリケーション General Ledger から <i>Sam</i> のプログラム要求が実行されます。
表 18	

関連項目

- 「[コンカレント・マネージャ](#)」ウィンドウ (4-76)
- 「[組合せ特殊化ルール](#)」ウィンドウ (4-82)
- 「[組合せルールの使用](#)」(4-45)

- 「特殊化ルールと組合セルールの違い」 (4-48)
- 「特殊化ルールの定義」 (4-34)
- 「特定なプログラムのみを実行するためのマネージャの特殊化」 (4-33)

組合セルールの使用

組合セルールを使用すると、マネージャを特殊化するときの条件をより詳細に指定できます。

組合セルールは、複数の処理を組み合わせて 1 つのルールを生成します。各処理は、ルール内でそれぞれ 1 行で定義します。行または個別処理を組み合わせることにより、組合セルール全体が定義されます。

提案：組合セル特殊化ルールは、マネージャを特殊化するための多数のルールの 1 つとして使用できます。

除くと含む処理の単独使用

組合セルール内の単独の除く処理は、特殊化ルールを定義する単独除く処理と同様に動作します。どちらも、処理により指定されたプログラム以外のコンカレント・プログラムを実行するすべての要求を読み込むように、マネージャに指示します。

除く Oracle ID - APPS2

結果 マネージャは Oracle ID 「APPS」に関連付けられたプログラムを除いて、Oracle ID に関連付けられたすべてのコンカレント・プログラムの実行要求を読み取ります。

組合セルール内の単独の含む処理は、特殊化ルールを定義する単独含む処理と同様に動作します。どちらの処理も、その処理を満足する要求のみを読み込むように、マネージャに指示します。

含む Oracle ID - APPS2

結果 マネージャは Oracle ID 「APPS2」に関連付けられたコンカレント・プログラムの実行要求のみを読み取ります。

複数除く処理の使用

組合セルール内の複数の行として除く処理を使用すると、その動作は複数の特殊化ルールとして複数の除く処理を使用することと同様になります。

組合セルールに除く行を追加していくことによって、数種類の要求を除外できます。

除く Program - Oracle Sales & Marketing、「名前」フィールドにデータなし

AND

除く Program - Oracle Inventory、「名前」フィールドにデータなし

結果 マネージャはアプリケーション Oracle Sales & Marketing のプログラム要求とアプリケーション Oracle Inventory のプログラム要求を除いて、すべてのコンカレント・プログラムの実行要求を読み取ります。

複数含む処理の使用

複数の含む処理を使用して組合セルールに要件を追加するとともに、数種類の要求を除外します。

同一の処理タイプに2つの含む処理を使用することはできません。含む処理はそれぞれ特定タイプの処理に対する排他文です。たとえば、2つの異なる Oracle ID に接続されている同一プログラムについて、同じ要求を「必須」にすることはできません。

含む Program - Oracle Payables、「名前」フィールドにデータなし

AND

含む Program - Oracle Payables Confirm Receipt Batch

結果 1つのプログラム Confirm Receipt Batch がアプリケーション Oracle Payables に属している場合に限り、マネージャはそのプログラムの実行要求のみを読み取ります。

除くと含む処理の使用

除く処理と含む処理は同一の処理タイプには使用できません。含む処理はそれぞれ特定タイプの処理に対する排他文です。

たとえば、Oracle ID "APPS" に関連付けられたプログラム要求を行って、別の Oracle ID に関連付けられた要求を禁止しても意味がありません。

含むを上書きする除く

「組合セルール」内で複数行を指定すると、除く処理は常に含む処理を上書きします。

含む Program - Oracle Payables Invoice Import

AND

除く Oracle ID - APPS2

結果 マネージャは Oracle Payables Invoice Import プログラムの実行要求を読み取りますが、そのプログラムが Oracle ID「APPS2」に関連付けられているとプログラムを実行しません。除く処理は含む処理を上書きします。

特定ユーザーが発行した特定プログラムを実行するためのマネージャの特殊化

特定の 1 ユーザーが発行したときに限り、1 つの特定のプログラムを実行する要求のみを読み込むように、マネージャに指示する組合セルールを定義できます。

含む ユーザー - Sheryl

AND

含む Program - Oracle Project Accounting Distribute Usage Costs

結果 マネージャは、Distribute Usage Costs プログラムを実行するために Sheryl が発行した要求のみを読み取ります。

特定のユーザーについてマネージャが実行するプログラムの制限

特定のユーザーが発行した場合に特定のプログラムの実行要求を無視するようにマネージャに指示する組合セルールを定義できます。

含む ユーザー - Sheryl

AND

除く Program - Oracle Project Accounting Expenditure Status

結果 マネージャは、Oracle Project Accounting プログラムの Accounting Expenditure Status を実行する要求を除いて、Sheryl が発行した要求のみを読み込みます。

Oracle ID の指定によるプログラムの要求タイプからの除外

含む Request Type - Oracle Project Accounting Expenditure Reports

AND

含む Oracle ID - APPS2

AND

除く

Program - Oracle Project Accounting Expenditure Status

結果

マネージャはプログラム Expenditure Reports を除いて、Oracle ID「APPS2」について実行される Oracle Project Accounting の要求タイプ「Reports」に属しているプログラムの実行要求のみを読み取ります。

関連項目

- 「[コンカレント・マネージャ](#)」ウィンドウ」(4-76)
- 「[組合せ特殊化ルール](#)」ウィンドウ」(4-82)
- 「[組合せ特殊化ルールの定義](#)」(4-43)
- 「[特殊化ルールと組合せルールの違い](#)」(4-48)
- 「[特定のプログラムのみを実行するためのマネージャの特殊化](#)」(4-33)

特殊化ルールと組合せルールの違い

特殊化ルールと組合せ特殊化ルールとの主な違いは、複数処理を使用した場合にルールの結果にどのような影響が出るかという点にあります。

複数処理の使用

ルール	処理	複数処理の影響	他のルールとの関係
特殊化ルール	含む	追加の各含むルールを使用すると、マネージャはより多くの要求を読み取ることができる。	各ルールでは、OR 条件が設定される。OR... 含む ...
	除く	追加の各除くルールを使用すると、マネージャは除外され、読み取る要求の数が少なくなる。	各ルールでは、AND 条件が設定される。AND... 除く ...
組合せルール 特殊化行	除く	追加の各除く行を使用すると、マネージャは除外され、読み取る要求の数が少なくなる。	ルールの各行では、AND 条件が設定される。AND... 除く ...
	含む	追加の各含む行または要求を使用すると、マネージャが読み取る要求の数が少なくなる。	ルールの各行では、AND 条件が設定される。AND... 含む ...
表 19			

関連項目

- 「[コンカレント・マネージャ](#)」ウィンドウ」(4-76)
- 「[組合せ特殊化ルール](#)」ウィンドウ」(4-82)

- 「組合せルールの使用」(4-45)
- 「組合せ特殊化ルールの定義」(4-43)
- 「特定なプログラムのみを実行するためのマネージャの特殊化」(4-33)
- 「例 - 特殊化ルールの使用」(4-38)

プログラムの要求タイプ別グループ化

システム管理者は、類似のプログラムをグループにまとめることができます。これを行うには、要求タイプを定義し、それらの要求タイプをユーザーが Oracle Applications 内で要求するプログラムへ割り当てます。システム管理者は、特定の要求タイプに属するプログラムのみを実行するコンカレント・マネージャを定義できます。

要求タイプを使用してコンカレント・マネージャを特殊化すると、特定のタイプのプログラムに別のタイプのプログラムの処理が終了するのを待たせなくても済むため、Oracle Applications の処理を最適化するのに役立ちます。要求タイプを使用すると、コンカレント・マネージャの特殊化ルールを作成するときに、時間の節約になります。

要求タイプの使用

コンカレント・マネージャを要求タイプ別に特殊化するには、次の3つのステップを実行します。

1. 「コンカレント要求タイプ」フォームを使用して「要求タイプ」を定義します。
2. 「コンカレント・プログラム」フォームを使用して、「要求タイプ」を、その要求タイプのメンバーにする個々のコンカレント・プログラムへ割り当てます。
3. 「コンカレント・マネージャ」フォームを使用して、コンカレント・マネージャを特殊化するときに、その「要求タイプ」を選択します。

要求タイプの使用例

例として、次のような要求タイプを定義することができます。

クイック	実行時間が比較的短いコンカレント・プログラム用。
夜間	実行時間が長く、通常は深夜または早朝の時間帯に実行をスケジュールするコンカレント・プログラム用。
月末レポート	毎月末に実行するレポートを生成するコンカレント・プログラム用。 たとえば、毎月末に 10 本のプログラムを実行する場合、「月末レポート」という要求タイプを定義し、その要求タイプを 10 本のレポート・プログラムへ割り当てることができます。

その後、特殊化ルールを使用して、「月末レポート」タイプの要求のみを実行するコンカレント・マネージャを定義できます。こうすると、コンカレント・マネージャを定義するときに 10 本の異なるレポート・プログラムを指定せずに済みます。また、10 本のプログラムを複数のマネージャに簡単に割り当てることもできます。

関連項目

- 「[コンカレント・マネージャ](#)」ウィンドウ (4-76)
- 「[コンカレント・マネージャの管理](#)」ウィンドウ (4-71)
- 「[コンカレント・プロセスの概要](#)」 (4-1)
- 「[完了したコンカレント要求レポート](#)」 (4-31)
- 「[時間ベース・キューの使用法](#)」 (4-29)
- 「[特定のプログラムのみを実行するためのマネージャの特殊化](#)」 (4-33)

コンカレント・マネージャの制御

ここでは、コンカレント・マネージャの制御方法について説明します。

マネージャの状態

個々のマネージャは、コンカレント・プログラム起動要求を読み取り、マネージャの稼働シフトの定義、ターゲット・プロセスの数、特殊化ルールなど、特定の条件が満たされた場合に実行するプログラムを実際に起動します。

コンカレント・マネージャはいつでも起動、停止またはリセットできます。Oracle Applications は、そのためのコマンドを処理する内部コンカレント・マネージャを提供します。個々のマネージャに対してコマンドを発行するだけでなく、内部コンカレント・マネージャの状態を変更することによってすべてのマネージャを同時に制御することもできます。

個々のマネージャの起動

マネージャを、個別に再起動したり、アクティブにできます。コンカレント・マネージャを再起動すると、そのマネージャを無効にする前のコマンドは取り消され、そのマネージャの稼働シフトの開始時に内部コンカレント・マネージャがそのマネージャを起動できるようになります。

次の場合にマネージャを個別に再起動してください。

- 稼働シフト割当を修正します
- 稼働シフトのターゲットのプロセス数を修正します

- 特殊化ルールを修正します
- コンカレント・プログラムの非互換性ルールを変更します

個別マネージャの無効化

個別マネージャを停止するときには、すべての要求を強制終了させてマネージャを即時に無効化するか、マネージャが現行の要求の処理を終了してから無効にするかを選択できます。

マネージャを無効化にするを選択すると、現在実行中の要求を完了させることができます。

要求を打ち切り、個別マネージャを無効にすることを指定すると、現在実行中の要求は即時に停止し、再発行（マネージャがアクティブになったときに）できるようにマークが付きます。

Oracle Applications のコンカレント・プログラムは、打ち切られた要求が停止後に再開されたときに、データの消失も重複も起こらないように設計されています。これは、正常な停止（たとえば「コンカレント・マネージャの無効化」要求を使用した場合）でも異常終了（たとえばハードウェア障害の場合）でも同じです。

注意：マネージャを選択し、明示的に無効化にした場合は、再び選択してから明示的に有効にするまでは、無効のままです。前提条件として、内部マネージャをアクティブにしておく必要があります。

内部コンカレント・マネージャの管理

内部コンカレント・マネージャをアクティブにすると、マネージャすべて（個別に無効化されているマネージャを除く）も同時にアクティブになります。

内部コンカレント・マネージャを無効化にすると、内部コンカレント・マネージャは有効なマネージャすべてを無効化にするコマンドを発行します。このコマンドは、個別に無効化されているマネージャには効きません。

要求を打ち切り、内部コンカレント・マネージャを無効化にすると、内部コンカレント・マネージャは、他のマネージャすべてに対して要求を打ち切り、無効化にするためのコマンドを発行します。現在実行中の要求は即時に停止され、各マネージャがアクティブになったときに再発行できるようにマークが付きます。

コンカレント・マネージャのステータスの検証

内部コンカレント・マネージャは、各コンカレント・マネージャのオペレーティング・システム・プロセスを常時モニターしています。このプロセス・モニタリングを内部コンカレント・マネージャの PMON サイクルといいます。PMON サイクルの長さは、内部コンカレント・マネージャを起動する STARTMGR コマンドから渡される引数の 1 つです。

内部コンカレント・マネージャに対して、個別コンカレント・マネージャの動作状態の即時検証、またはPMONチェックの実行を指示できます。

「マネージャの管理」フォームからマネージャを制御する

コンカレント・マネージャに対してコマンドを発行するには、「コンカレント・マネージャの管理」フォームを使用します。

内部コンカレント・マネージャに対して、個々のマネージャのステータスを手動で検証したり、個々のマネージャを再起動したりするよう指示することもできます。参照：「[コンカレント・マネージャの管理](#)」ウィンドウ」(4-71)。

マネージャの制御

マネージャ	制御機能	摘要
内部マネージャ	コンカレント・マネージャの有効化	「コンカレント・マネージャの無効化」を使用して個別に無効化されたマネージャを除いて、内部マネージャおよび他のすべてのマネージャを有効にします。
	ステータスの検証（コンカレント・マネージャ）	プロセス・モニター（PMON）サイクルを手動で実行します。
	コンカレント・マネージャの無効化	内部マネージャおよび他のすべてのマネージャを無効化します。
	要求の終了およびマネージャの無効化	要求を終了して内部コンカレント・マネージャを無効化すると、実行中のすべての要求（実行中のコンカレント・プログラム）は終了し、すべてのマネージャが無効化されます。
その他のすべてのマネージャ	コンカレント・マネージャの有効化	マネージャがコンカレント稼動シフトで動作するように定義されている場合、そのマネージャは即時に起動します。「コンカレント・マネージャの無効化」および「要求の終了およびマネージャの無効化」が取り消されます。
	コンカレント・マネージャの再起動	内部マネージャは、該当マネージャの定義およびコンカレント・プログラムの非互換性についてのルールを再び読み込みます。次の場合には、マネージャの再起動が必要です。- 稼動シフト割当を変更する - ターゲット・プロセス数を変更する - 特殊化ルールを変更する - コンカレント・プログラムの非互換性を変更する

マネージャ	制御機能	摘要
	コンカレント・マネージャの無効化	マネージャを無効化します。現在実行中のすべての要求（コンカレント・プログラム）は、マネージャが停止する前に処理を完了できます。ユーザーがそのマネージャを選択して「コンカレント・マネージャの有効化」を選択するまで、マネージャは再起動しません。
	要求の終了およびマネージャの無効化	マネージャの処理対象になっている実行中のすべての要求（実行中のコンカレント・プログラム）が終了します。一度無効化されたマネージャは、ユーザーがそのマネージャを選択して「コンカレント・マネージャの有効化」を選択するまで、再起動しません。
表 20		

関連項目

- 「「コンカレント・マネージャの管理」ウィンドウ」（4-71）
- 「パラレル・コンカレント・プロセスの概要」（4-58）
- 「内部コンカレント・マネージャのオペレーティング・システムからの制御」（4-53）

内部コンカレント・マネージャのオペレーティング・システムからの制御

内部コンカレント・マネージャを制御するためにオペレーティング・システムから使用できるコマンドには、STARTMGR および CONCSUB の 2 つがあります。STARTMGR は内部コンカレント・マネージャを起動します。CONCSUB は内部コンカレント・マネージャを非アクティブ化、または異常終了させる場合、あるいは各マネージャのオペレーティング・システム・プロセスを検証する場合に使用できます。

「表 21」（4-54）は「コンカレント・マネージャの管理」フォームで表示される内部マネージャの管理状態とそれに対応するオペレーティング・システム・コマンドを比較したものです。この表に示していない引数もあります。

OS からの内部マネージャの制御

「コンカレント・マネージャの管理」フォームの場合	オペレーティング・システムの場合（一部の引数のみ）
コンカレント・マネージャの有効化	STARTMGR（構文はプラットフォームにより異なる）

「コンカレント・マネージャの管理」フォームの場合	オペレーティング・システムの場合（一部の引数のみ）
ステータスの検証（コンカレント・マネージャ）	CONCSUB FND VERIFY
コンカレント・マネージャの無効化	CONCSUB FND DEACTIVATE
要求の終了およびマネージャの無効化	CONCSUB FND ABORT
表 21	

内部コンカレント・マネージャのオペレーティング・システムからの起動

コンカレント・マネージャを起動するには、オペレーティング・システムのプロンプトから STARTMGR コマンドを使用できます。このコマンドは、内部コンカレント・マネージャを起動するコマンドで、このマネージャによってユーザー定義のコンカレント・マネージャが起動します。

各コンカレント・マネージャが「出力」ディレクトリおよび「ログ」ディレクトリに書き込めるように、この両ディレクトリに対する書込み権限を持っている必要があります。コンカレント・マネージャは各種のオプション付きで起動できます。一部のオペレーティング・システムでは、コンカレント・マネージャが停止するときに電子メールをユーザーに送信するオプションがあります。このコマンドの説明については、使用しているシステムのインストール・ガイドを参照してください。

参照：使用しているオペレーティング・システムの『Oracle Applications インストール・ガイド』

STARTMGR コマンドは次の場合に使用します。

- Oracle Applications のインストール時
- コンカレント・マネージャの停止後
- MIS がオペレーティング・システムの再起動後
- データベース管理者によるデータベースの再起動後

STARTMGR コマンドは、プラットフォームに依存します。したがって、具体的な構文については、使用プラットフォームの『Oracle Applications インストール・ガイド』を参照してください。

参照：「コンカレント・マネージャ - 付録：システム参照資料」、使用しているオペレーティング・システムの『Oracle Applications インストール・ガイド』

STARTMGR コマンドには、最大 10 個のオプション・パラメータを指定できます。

- PRINTER を除く各パラメータにはデフォルトがあります。

- STARTMGR コマンドおよび環境を修正して、独自のデフォルトを設定できます。

内部コンカレント・マネージャを起動するには、システム・プロンプトで次のコマンドを入力します。

```
$ startmgr <optional parameters>
```

パラメータは任意の順序で渡すことができます。たとえば、次のとおりです。

```
$ startmgr sysmgr="applsys/fnd" mgrname="std"
  printer="hqseq1" mailto="jsmith" restart="N"
  logfile="mgrlog" sleep="90" pmon="5" quesiz="10"
```

内部コンカレント・マネージャ起動パラメータの表示

内部コンカレント・マネージャのログ・ファイルには、STARTMGR コマンドにより実行された起動パラメータの値が表示されます。次に示すのはその一例です。パラメータ値を変更することはできません。

```
logfile=/fnddev/fnd/6.0/log/FND60.mgr   (ポート固有のパス)
PRINTER=hqunx138
mailto=appldev
restart=N
diag=N
sleep=60   (デフォルト)
pmon=20   (デフォルト)
quesiz=1   (デフォルト)
```

内部コンカレント・マネージャのオペレーティング・システムからの停止

オペレーティング・システムのプロンプトから、CONCSUB ユーティリティを使用して、SYSADMIN ユーザー名およびシステム管理者職責のもとでコンカレント要求を発行できます。

CONCSUB ユーティリティがコンカレント要求の発行と、オペレーティング・システム・プロンプトに戻ります。コンカレント要求が完了するまで待機してください。

コンカレント要求のステータスをチェックするには、「コンカレント要求」フォームを使用します。

```
CONCSUBapplsys/pwd 'Responsibility application shortname'
'Responsibility name' 'Username' [WAIT={Y|N|n}] CONCURRENT
'Program application shortname' PROGRAM
```

パラメータ

<i>appls/pwd</i>	Oracle Application Object Library データに接続するための Oracle ユーザー名およびパスワード
<i>Responsibility application shortname</i>	職責のアプリケーション短縮名。システム管理者職責の場合、アプリケーション短縮名は SYSADMIN です。
<i>Responsibility name</i>	職責の名称。システム管理者職責の場合、職責名は <i>System Administrator</i> です。
<i>Username</i>	要求の発行者のアプリケーション・ユーザー名。たとえば、SYSADMIN は、システム管理者のユーザー名です。
<i>WAIT={Y N n}</i>	<p>発行した要求の実行が完了してからオペレーティング・システムのプロンプトに戻るよう CONCSUB を設定する場合は、WAIT を Y に設定します。</p> <p>CONCSUB コマンドの待機を設定しない場合は、WAIT を N（デフォルト値）に設定します。</p> <p>CONCSUB コマンド終了までの待機時間を数値（<i>n</i> 秒：<i>n</i> は整数）で指定することもできます。</p> <p>WAIT を使用する場合、WAIT は CONCURRENT の前に入力する必要があります。</p>
<i>Program application shortname</i>	<p>プログラムのアプリケーション短縮名。</p> <p>DEACTIVATE、ABORT および VERIFY プログラムの場合、アプリケーション短縮名は FND です。</p>
<i>PROGRAM</i>	<p>「全てのマネージャの停止」コンカレント要求を発行する場合は、プログラム DEACTIVATE を使用します。</p> <p>「マネージャの停止の取消」コンカレント要求を発行する場合は、プログラム ABORT を使用します。</p> <p>「全マネージャ・ステータスの検証」コンカレント要求を発行する場合は、プログラム VERIFY を使用します。</p>

CONCSUB を使用する構文の例

```
CONCSUB <Username/Password> SYSADMIN 'System Administrator'  
SYSADMIN CONCURRENT FND DEACTIVATE
```

```
CONCSUB <Username/Password> SYSADMIN 'System Administrator'  
SYSADMIN CONCURRENT FND ABORT
```

```
CONCSUB <Username/Password> SYSADMIN 'System Administrator'  
SYSADMIN CONCURRENT FND VERIFY
```

CONCSUB によるマネージャの停止

次の時点でコンカレント・マネージャを停止するには、CONCSUB を使用します。

- MIS がオペレーティング・システムを停止する前
 - データベース管理者がデータベースを停止する前
 - コンカレント・マネージャ定義およびコンカレント・プログラム定義を有効にするとき
- この後で、STARTMGR コマンドを使用して内部コンカレント・マネージャを再起動すると、各コンカレント・マネージャが起動します。

例 - CONCSUB による毎終業時の停止

CONCSUB を使用して、一連の手順から構成されるシェル・スクリプト内部からコンカレント要求を発行するには、トークン WAIT を値 Y 付きで (WAIT=Y) 使用できます。トークン WAIT を指定すると、シェル・スクリプトは、マネージャの非アクティブ化、異常終了、ステータス検証のいずれかを実行した後に、次の手順に入ります。

参照:「内部コンカレント・マネージャのオペレーティング・システムからの制御」(4-53)

1. 特定のオペレーティング・システム用にカスタマイズされたシェル・スクリプトが起動します。
2. `CONCSUBapplsys/pwdSYSADMIN'System Administrator' SYSADMIN WAIT=Y
CONCURRENTFND DEACTIVATE`

シェル・スクリプトが CONCSUB に制御を渡すと、CONCSUB はプログラム DEACTIVATE が完了するまで待って、シェル・スクリプトに制御を戻します。
3. スクリプトは、データベースを停止するためのコマンドを発行します。
4. スクリプトは、データベースのバックアップをとるためのコマンドを発行します。
5. スクリプトは、データベースを起動するためのコマンドを発行します。
6. `$ startmgr sysmgr="applsys/fnd" mgrname="std" printer="hqseq1" mailto="jsmith"
restart="N" logfile="mgrlog" sleep="90" pmon="5" quesiz="10"`

シェル・スクリプトは制御を STARTMGR に渡し、STARTMGR が内部マネージャ（およびその他のすべてのマネージャ）を起動します。
7. シェル・スクリプトが完了します。

CONCSUB によるパスワードの非表示化

ユーザー名およびパスワードが与えられている場合は、CONCSUB ユーティリティは通常どおり動作します。

ユーザー名のみが与えられている場合（最初の引数に「/pwd」がない場合）は、ユーティリティはパスワードの入力を求めます。

ORACLE Password:

エコーはオフになります。たとえば、次のコマンドに Oracle パスワードはありません。

```
CONCSUB applsys SYSADMIN 'System Administrator' SYSADMIN
CONCURRENT FND
FNDMNRMT Y 0 20221
ORACLE Password:
Submitted request 32157 for CONCURRENT FND FNDMNRMT Y 0 20221
```

これで、最初の引数は通常通りアプリケーション・ユーザー名（たとえば SYSADMIN）になります。

パスワードは、ファイルに入れておいて、標準入力（stdin）にリダイレクトできます。

UNIX では、コマンドは次のように実行されます。

```
CONCSUB applsys SYSADMIN 'System Administrator' SYSADMIN
CONCURRENT FND
FNDMNRMT Y 0 20221 < password.file
```

ここで、password.file は、パスワードを入れてある ASCII ファイルです。シェル・スクリプトまたはバッチ・プロセスでは、この方法をできるだけ使用してください。

関連項目

- [「コンカレント・マネージャ」ウィンドウ](#) (4-76)
- [「コンカレント・マネージャの管理」ウィンドウ](#) (4-71)
- [「コンカレント・プロセスの概要」](#) (4-1)
- [「データベース・インスタンス、マネージャの位置およびファイルの配布」](#) (4-64)
- [「パラレル・コンカレント・プロセスの概要」](#) (4-58)
- [「内部コンカレント・マネージャのオペレーティング・システムからの制御」](#) (4-53)

パラレル・コンカレント・プロセスの概要

ここでは、パラレル・コンカレント・プロセスの定義、その実行環境および動作方法を説明します。

パラレル・コンカレント・プロセスについて

パラレル・コンカレント・プロセスを使用すると、クラスタ環境、大規模並列環境または同種ネットワーク環境内の複数のノード間でコンカレント・マネージャを分散できます。他のノードを待機状態にして1つのノードでコンカレント・プロセスを実行するかわりに、ハードウェア・リソースをフルに利用して、使用可能なすべてのノード間でコンカレント・プロセスを展開できます。

パラレル・コンカレント・プロセスの利点

パラレル・コンカレント・プロセスには、Oracle Applications のユーザーに対する次の利点があります。

- 高いパフォーマンス -- 複数のノードでコンカレント・プロセスを実行して、コンカレント・プロセスのスループットを改善する機能。
- フォルト・トレラント -- 1つ以上のノードが利用できないときでも、使用可能な各ノードでコンカレント・プロセスの実行を継続する機能。
- 適合性 -- プラットフォームに固有の各バッチ・キューや負荷バランス・システムを統合させて、特定のプラットフォーム上でのコンカレント・プロセスのパフォーマンスを最大限にする機能。
- 単一の制御ポイント -- 複数のノード上で実行中のコンカレント・マネージャを、クラスタ環境、大規模並列環境または同種ネットワーク環境内の任意のノードから管理する機能。

パラレル・コンカレント・プロセスの環境

パラレル・コンカレント・プロセスは、クラスタ環境、大規模並列環境および同種ネットワーク環境などの複数ノード環境で実行されます。これらの環境では、各ノードは1つ以上のプロセッサ（CPU）やその対応するメモリーで構成されます。各ノードは、その他のノードとは共有しない、それ自体のメモリーを持っています。また各ノードは、ディスクなどのリソースを共有するときを除いて、その他のノードとは独立して動作します。

パラレル・コンカレント・プロセスの場合、1つ以上のコンカレント・マネージャが、複数ノード環境内の1つ以上のノード上で実行されます。システムを構成するときに、コンカレント・マネージャの実行場所を決定してください。

システム管理者はコンカレント・マネージャの特殊化ルールを自由に組み合わせて定義し、希望する方法でそれらのルールを各ノードに適用できます。たとえば「Oracle General Ledger」の3つのコンカレント・マネージャは、3つのノード間で展開できます。また「Oracle Payables」のコンカレント・マネージャと「Oracle General Ledger」のコンカレント・マネージャは、同じノード上で同時に実行できます。

次に、パラレル・コンカレント・プロセスを実行可能な環境の例を示します。

クラスタ環境

クラスタ環境では、それぞれが1つのノードを表す複数のコンピュータが共通のディスク・プールを共有します。代表的なクラスタ環境には、IBM HACMP、VAX クラスタおよび Sequent サーバーがあります。

クラスタ環境におけるパラレル・コンカレント・プロセスの場合、1つの Oracle データベースが共通ディスク・プールに常駐し、Oracle Parallel Server の複数インスタンスがクラスタ内の複数ノード上で同時に実行されます。また複数のコンカレント・マネージャが、クラスタ内の各ノード間で分散されます。

大規模並列環境

大規模並列環境では、複数のノードが1台のコンピュータ内に収容されます。すべてのノードが、共通ディスク・プールにアクセスします。たとえば IBM SP/2 は、大規模並列コンピュータです。

大規模並列環境におけるパラレル・コンカレント・プロセスの場合、Oracle Parallel Server の個々のインスタンスは、各ノード間で分散されている複数のコンカレント・マネージャとともに、複数のノード上で同時に実行されます。

同種ネットワーク環境

同種ネットワーク環境では、同じ種類の複数のコンピュータがローカル・エリア・ネットワーク（LAN）を経由して1つのデータベース・サーバーに接続されるか、またはデータベース・サーバーのクラスタに接続されます。

たとえば簡単なネットワーク環境として、LAN 経由で1つの Sequent サーバーに接続された Sun の複数の SPARCstations で構成される場合があります。複雑なネットワーク環境としては、複数の SPARCstations が各 Sequent サーバーのクラスタに接続される場合があります。

パラレル・コンカレント・プロセスの場合、1つ以上のコンカレント・マネージャが、複数ノード環境内の1つ以上のノード上で実行されます。1つのデータベース・サーバーが ORACLE の1つのインスタンスを実行し、またデータベース・サーバーのクラスタは、ORACLE Parallel Server を使用して Oracle の複数のインスタンスを実行します。

パラレル・コンカレント・プロセスの実行方法

コンカレント・マネージャ

パラレル・コンカレント・プロセスの場合、コンカレント・マネージャを持つ各ノードは ORACLE のインスタンスを実行しているときと実行していないときがあります。ORACLE が実行されていないノードの場合、コンカレント・マネージャは Net8 経由で、ORACLE が実行されているノードに接続されます。

各コンカレント・マネージャに対して、システム管理者は1次ノードと2次ノードを割り当てます。最初に、コンカレント・マネージャはその1次ノード上で起動されます。そのノードまたはORACLE インスタンスで障害が発生した場合、コンカレント・マネージャは2次ノードに移行します。

障害が発生したノードが再び使用可能になった時点で、コンカレント・マネージャは1次ノードに移行します。プロセスの移行時に、各プロセスはその現行のノードからターゲット・ノードに移行します。

内部コンカレント・マネージャ

内部コンカレント・マネージャは、任意のノード上で実行でき、同じノードまたは他のノード上の各コンカレント・マネージャを起動および停止できます。内部コンカレント・マネージャは常に有効である必要があるため、高い耐障害性が必要です。この耐障害性を提供するため、パラレル・コンカレント・プロセスでは、内部モニター・プロセスが採用されています。

内部モニター・プロセス

内部モニター・プロセスの役割は、内部コンカレント・マネージャをモニターし、停止した場合には再起動することのみです。内部コンカレント・マネージャで障害が発生したことを検出した最初の内部モニター・プロセスは、そのマネージャをそれ自体のノード上で再起動します。

1つのノード上でアクティブ状態にできる内部モニター・プロセスは、1つに限られます。システム管理者は、システムを構成するときに、内部モニター・プロセスを持つ各ノードを決定します。また各内部モニター・プロセスに1次ノードおよび2次ノードを割り当てて、障害対策を十分に講じることができます。

内部モニター・プロセスはコンカレント・マネージャと同じように、実行時間帯を割り当て、内部コンカレント・マネージャによって起動および停止できます。

ログ・ファイルおよび出力ファイルへのアクセス

いずれかのノード上で実行される各要求からのコンカレント・ログ・ファイルや出力ファイルは、それ以外のノードからオンラインでアクセスできます。あるノード上で実行されている要求からのログ・ファイルや出力ファイルを表示する場合、ユーザーはそのノードにログオンする必要はありません。参照：「[データベース・インスタンス、マネージャの位置およびファイルの配布](#)」(4-64)。

この機能は、インストール時の設定の手順により異なります。詳細は、使用プラットフォームに関するインストラクション・マニュアルを参照してください。

プラットフォーム固有のキューおよび負荷バランス・システムの統合

一部のクラスタ・システムや大規模並列処理システムには、バッチ・プロセスをキューに入れたり、プロセス負荷を分散したりするための専用機能が用意されています（例：VMS バッ

チ・キュー、IBM LoadLeveler)。ユーザーがこれらの機能を利用して、Oracle Applications の処理のみでなく、すべての処理の管理を求めることに備えて、パラレル・コンカレント・プロセスはそれらを統合するように設計されています。したがってシステム管理者は、コンカレント・プロセスの管理を使用プラットフォームの特定の機能に合せることができます。

プラットフォーム固有のキュー・システムおよび負荷バランス・システムの統合に関する詳細は、ユーザー・プラットフォームのインストレーション・マニュアルを参照してください。

関連項目

- 「[コンカレント・マネージャ](#)」ウィンドウ (4-76)
- 「[コンカレント・マネージャの管理](#)」ウィンドウ (4-71)
- 「[ノード](#)」ウィンドウ (4-85)
- 「[コンカレント・プロセスの概要](#)」 (4-1)
- 「[データベース・インスタンス、マネージャの位置およびファイルの配布](#)」 (4-64)
- 「[パラレル・コンカレント・プロセスのインストール](#)」 (4-70)
- 「[パラレル・コンカレント・プロセスの管理](#)」 (4-62)
- 「[パラレル・コンカレント・プロセスの例](#)」 (4-65)

パラレル・コンカレント・プロセスの管理

ここでは、「システム管理」フォームを使用してパラレル・コンカレント・プロセスを管理する方法について説明します。ここには次のトピックがあり、パラレル・コンカレント・プロセスに関する各項目が説明されています。

コンカレント・マネージャの定義

「コンカレント・マネージャ」ウィンドウを使用してコンカレント・マネージャを定義してください。マネージャを定義するときは、「コンカレント・マネージャ」または「内部モニター」のいずれかのマネージャ・タイプを指定してください。また3番目のマネージャ・タイプとして、Oracle Applications で事前に定義されている内部コンカレント・マネージャ・プロセスを記述した内部コンカレント・マネージャがあります。

注意: パラレル・コンカレント・プロセスを使用する際に、マネージャ名に埋込みスペースを使用できません。マネージャの名前は1ワードで指定するか、ハイフン (manager-name の書式) または下線 (manager_name の書式) を使用して2つのワードを接続して指定してください。

各コンカレント・マネージャや各内部モニター・プロセスに1次ノードと2次ノードを割り当てるができます。また使用プラットフォームに固有のキュー管理システムが使用可能な場合は、1次システム・キュー名と2次システム・キュー名を割り当てるができます。参照:「[コンカレント・マネージャ](#)」ウィンドウ」(4-76)。

コンカレント・マネージャの管理

ターゲット・ノード

「コンカレント・マネージャの管理」フォームを使用すると、パラレル・コンカレント・プロセス環境における各コンカレント・マネージャのターゲット・ノードを表示できます。ターゲット・ノードとは、コンカレント・マネージャに関連付けられているプロセスの実行先となるノードです。

マネージャの1次ノードおよび Oracle インスタンスが使用可能なときには、ターゲット・ノードは1次ノードに設定されます。それ以外の場合、ターゲット・ノードはマネージャの2次ノードに設定されます（そのノードやその Oracle インスタンスが使用可能な場合）。プロセスの移行時に、各プロセスはその現行のノードからターゲット・ノードに移行します。

アクセス・ノード間の制御

「コンカレント・マネージャの管理」フォームを使用することにより、パラレル・コンカレント・プロセス環境における各コンカレント・マネージャのターゲット・ノードを表示することができます。あるノードでコンカレント・プロセスを制御するときには、そのノードにログ・オンする必要はありません。また、パラレル・コンカレント・プロセス環境下の任意のノードから内部コンカレント・マネージャを終了できます。

マネージャの起動

「コンカレント・マネージャの管理」フォームから内部コンカレント・マネージャに対して「起動」コマンドを発行するか、またはオペレーティング・システムのプロンプト・レベルで STARTMGR コマンドを呼び出すことによって、パラレル・コンカレント・プロセスを起動します。内部コンカレント・マネージャを起動したノードとは関係なく、内部コンカレント・マネージャはその割り当てられたノード上で起動します（ただし、リモート・プロセス起動をサポートするプラットフォームのノードから実行しているものと仮定します）。

内部コンカレント・マネージャの起動後、内部コンカレント・マネージャはすべての内部モニター・プロセスやすべてのコンカレント・マネージャを起動します。マネージャはその1次ノード上で内部モニター・プロセスおよびコンカレント・マネージャを起動しようとしませんが、1次ノードが使用できない場合には2次ノードに移ります。

マネージャの停止

「コンカレント・マネージャの管理」フォームから内部コンカレント・マネージャに Deactivate（停止）コマンドを発行することにより、パラレル・コンカレント・プロセスを停止します。すべてのコンカレント・マネージャと「内部モニター・プロセス」は、内部コンカレント・マネージャが停止する前に終了します。

マネージャ間の移行

ほとんどのプロセス移行は、1 次ノードで障害が発生したとき、またはその後回復したときに自動的に行われます。ただし、「コンカレント・マネージャ」フォームを使用してコンカレント・マネージャまたは内部モニター・プロセスに対するノード割当を手動で変更することによって、プロセス間の移行を実現できます。変更内容を有効にするには、「コンカレント・マネージャの管理」フォームから内部コンカレント・マネージャに「Verify（検証）」コマンドを発行してください。

コンカレント・プロセスの終了

「コンカレント・マネージャの管理」フォームから「終了」コマンドを発行することによって、ローカル・ノードまたはリモート・ノード上で実行中のコンカレント・プロセスを終了できます。

関連項目

- 「[「コンカレント・マネージャ」ウィンドウ](#)」（4-76）
- 「[「コンカレント・マネージャの管理」ウィンドウ](#)」（4-71）
- 「[内部コンカレント・マネージャのオペレーティング・システムからの制御](#)」（4-53）

データベース・インスタンス、マネージャの位置およびファイルの配布

次の各ページでは、パラレル・コンカレント・プロセスの設定例をいくつか示します。

パラレル・コンカレント・プロセスの場合、コンカレント・マネージャを持つ各ノードは ORACLE のインスタンスを実行しているときと実行していないときがあります。ORACLE が実行されていないノードの場合、コンカレント・マネージャは Net8 経由で、ORACLE が実行されているノードに接続されます。

コンカレント・プログラムの実行ファイルは、ログ・ファイルおよびレポート出力ファイルとともに、ノードのローカル・ディスクに格納できます。またこれらのファイルは、中央の 1 箇所に格納して、各ノードからリモートで読むこともできます。

関連項目

- 「「ノード」 ウィンドウ」 (4-85)
- 「データベース・インスタンス、マネージャの位置およびファイルの配布」 (4-64)
- 「パラレル・コンカレント・プロセスのインストール」 (4-70)
- 「パラレル・コンカレント・プロセスの概要」 (4-58)
- 「パラレル・コンカレント・プロセスの例」 (4-65)

パラレル・コンカレント・プロセスの例

「パラレル・コンカレント・プロセス - 単一データベース・インスタンス（集中および共有ログ / 出力 / 実行ファイル・システム）」 (4-66)

「パラレル・コンカレント・プロセス - Oracle Parallel Server（集中および共有ログ / 出力 / 実行ファイル・システム）」 (4-67)

「パラレル・コンカレント・プロセス - 単一データベース・インスタンス（分散ログ / 出力 / 実行ファイル）」 (4-68)

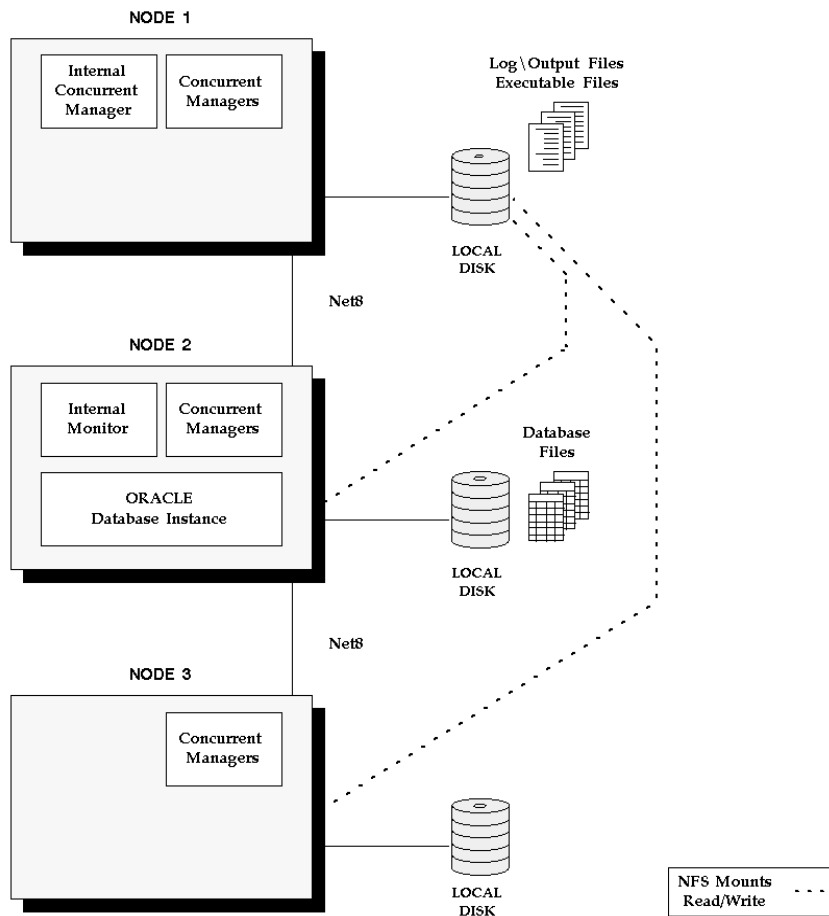
「パラレル・コンカレント・プロセス - ノード 1 がデータベース・サーバーとして動作し、ノード 2 ～ 5 がコンカレント・マネージャ・サーバーとして動作する場合」 (4-69)

関連項目

- 「パラレル・コンカレント・プロセスのインストール」 (4-70)
- 「パラレル・コンカレント・プロセスの概要」 (4-58)

パラレル・コンカレント・プロセス - 単一データベース・インスタンス（集中および共有ログ / 出力 / 実行ファイル・システム）

図 14

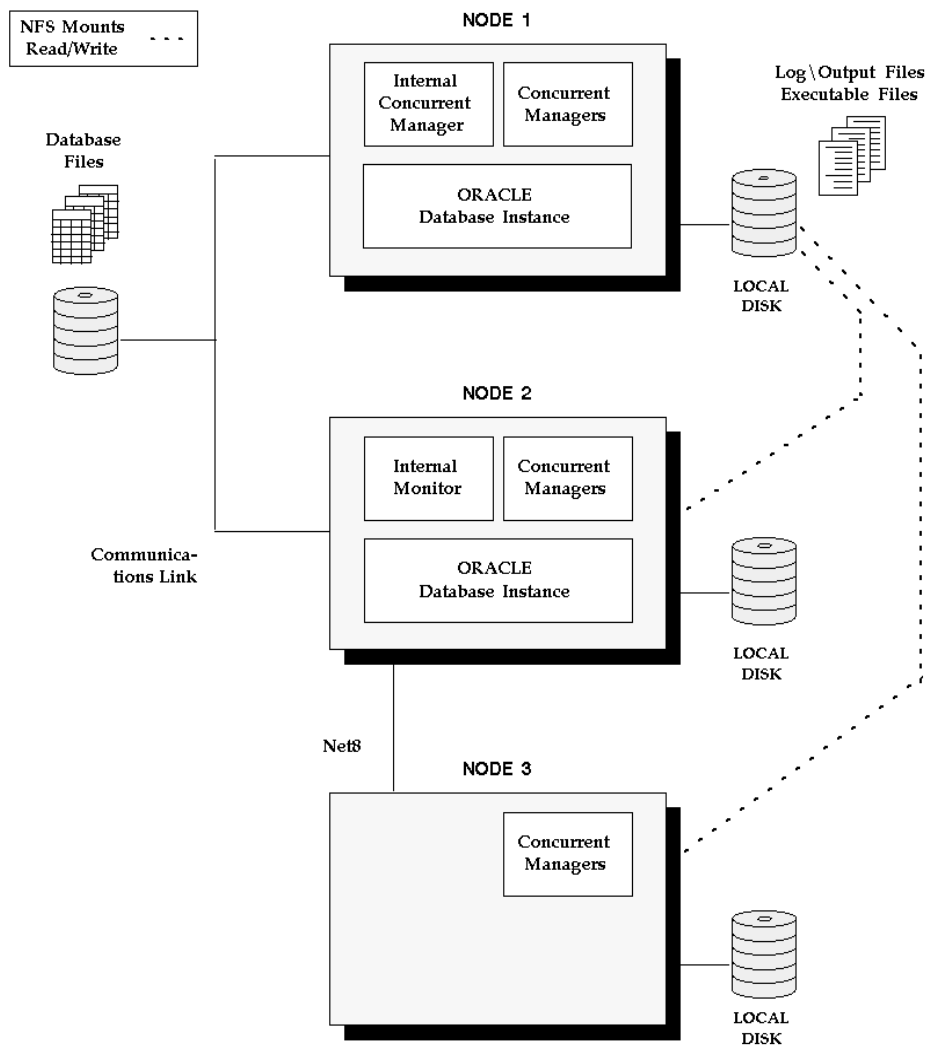


関連項目

- 「パラレル・コンカレント・プロセスの例」（4-65）

パラレル・コンカレント・プロセス - Oracle Parallel Server (集中および共有ログ / 出力 / 実行ファイル・システム)

図 15

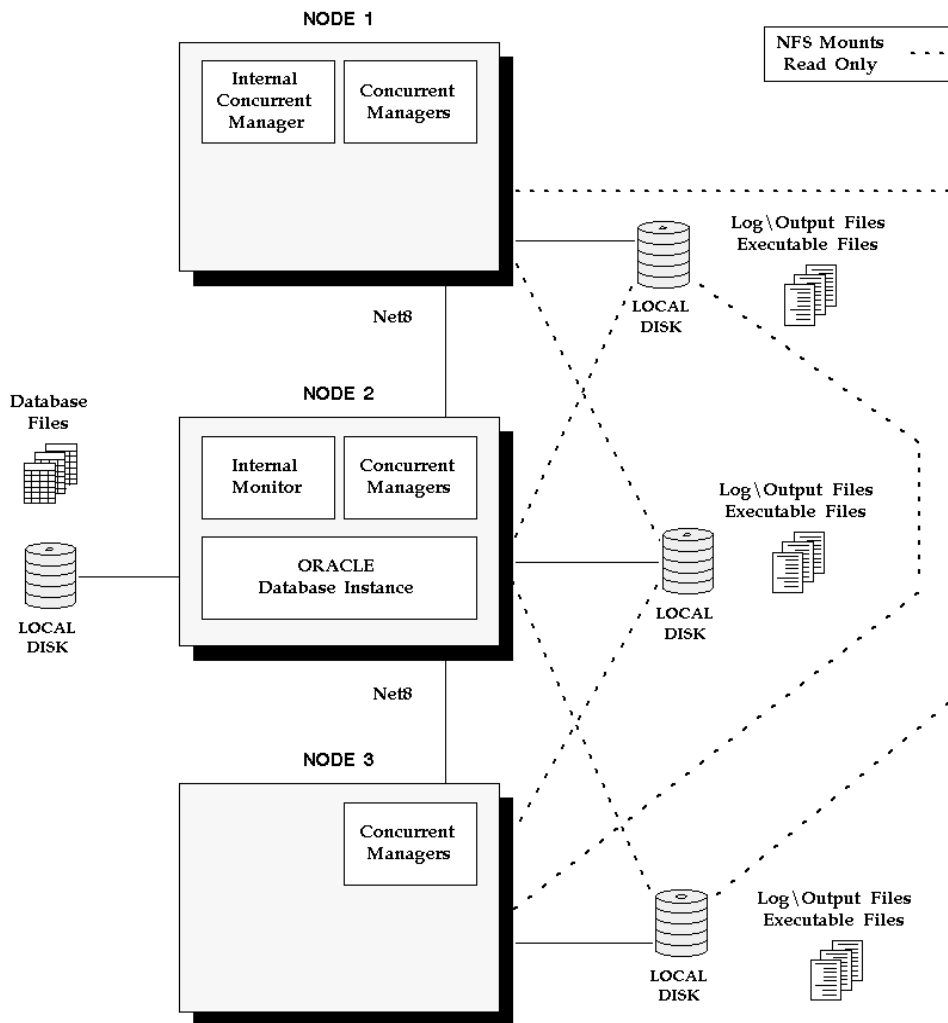


関連項目

- 「パラレル・コンカレント・プロセスの例」 (4-65)

パラレル・コンカレント・プロセス - 単一データベース・インスタンス (分散ログ / 出力 / 実行ファイル)

图 16

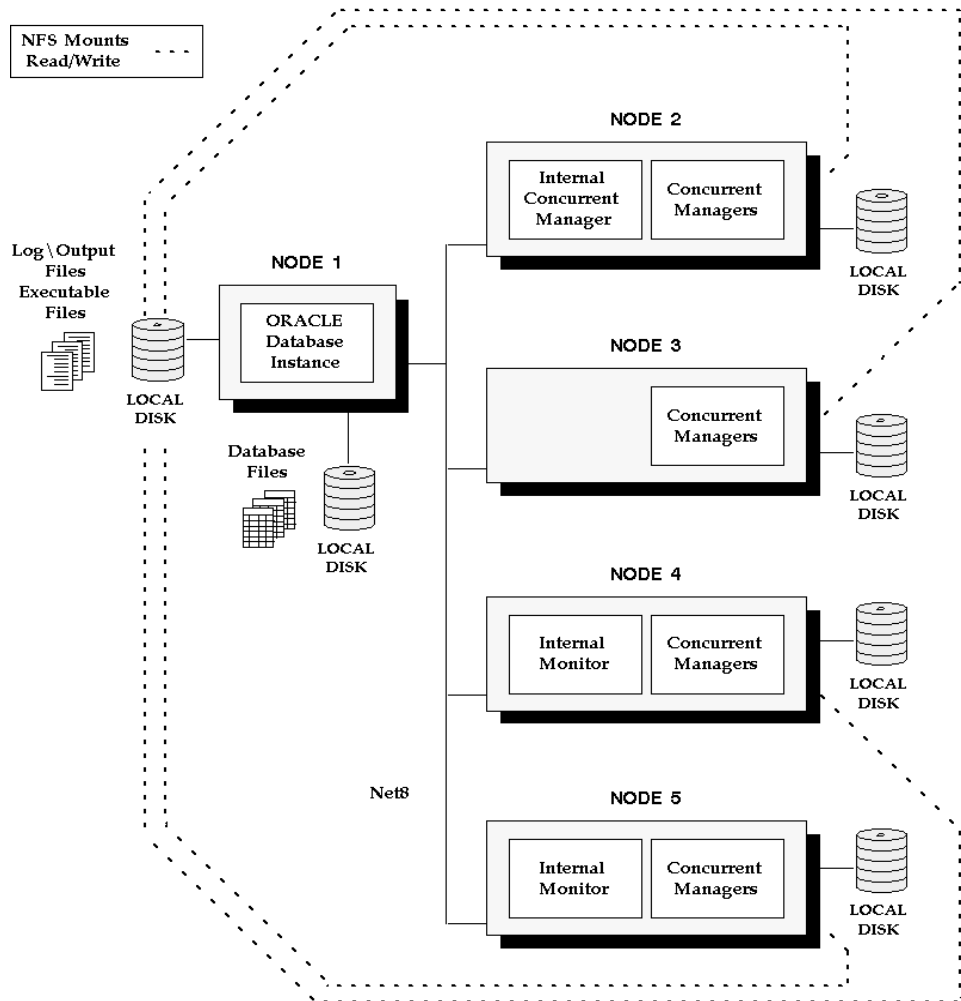


関連項目

- 「パラレル・コンカレント・プロセスの例」 (4-65)

パラレル・コンカレント・プロセス - ノード1がデータベース・サーバーとして動作し、ノード2～5がコンカレント・マネージャ・サーバーとして動作する場合

図 17



関連項目

- 「パラレル・コンカレント・プロセスの例」 (4-65)

パラレル・コンカレント・プロセスのインストール

次の各チェックリストは、パラレル・コンカレント・プロセスのインストール手順の要約を示します。次の各ステップの実行方法は、使用中のプラットフォームによって異なります。使用しているオペレーティング・システムの『Oracle Applications インストレーション・ガイド』を参照してください。

- ☐ 各ノードで **applmgr** のログインを設定します。
- ☐ 各ノード上のディレクトリ間の記号リンクを作成します。
- ☐ 必要な権限とシノニムを作成します。
- ☐ 「AutoInstall」を実行します。

現行のリリースの Oracle Applications ソフトウェアをすでにインストールしてある場合は、適切な環境変数を定義するための **adadmin** ユーティリティを使用して環境ファイルを再生成すると、パラレル・コンカレント・プロセスを設定できます。

- ☐ 必要に応じて、製品ファイルを追加ノードにインストールします。
- ☐ 各パラレル・ノード上で環境ファイルを作成します。
- ☐ 各ノード上で **afimpmon.sql** を実行して、**dcptest** を編集します。
- ☐ 各コンカレント・マネージャを定義します。

注意：パラレル・コンカレント・プロセスを使用する際に、マネージャ名に埋込みスペースを使用できません。マネージャの名前は1ワードで指定するか、ハイフン（**manager-name** の書式）または下線（**manager_name** の書式）を使用して2つのワードを接続して指定してください。

参照：「[パラレル・コンカレント・プロセスの管理](#)」（4-62）

関連項目

- 「[コンカレント・マネージャ](#)」ウィンドウ（4-76）
- 「[ノード](#)」ウィンドウ（4-85）
- 「[パラレル・コンカレント・プロセスの概要](#)」（4-58）
- 「[パラレル・コンカレント・プロセスの例](#)」（4-65）

「コンカレント・マネージャの管理」ウィンドウ

コンカレント・マネージャ（すべての取引マネージャを含む）のステータスを参照し、必要に応じて制御コマンドを発行して、マネージャのステータスを変更します。たとえば、現在有効なマネージャを無効化してその変更が確定すると、新しいステータスを表示できます。

コンカレント・マネージャ・ブロックの管理

ノード

パラレル・コンカレント・プロセス環境では、マネージャのプロセスはこのノード上で実行されるように計画されます。

コンカレント・マネージャがプラットフォーム固有のシステム・キューを使用するように定義されているときには、このフィールドにはマネージャがそのプロセスを発行するキュー名が表示されます。

プロセス現行

各マネージャ・プロセスごとに、1つのコンカレント要求を実行（1つのコンカレント・プログラムを起動）できます。通常、実際のプロセス数はターゲット・プロセス数（マネージャが実行できる最大要求数）と一致します。

ただし、要求数の不足、マネージャの非アクティブ化またはマネージャの移行が理由で実際のプロセス数がターゲット・プロセス数よりも少ないことがあります。

プロセス・ターゲット

このフィールドには、このマネージャに対して有効化できるマネージャ・プロセスの最大数が表示されます。

実行要求数 / 保留要求数

保留中の要求がある場合、通常、この数値は実際のプロセス数と同じになります。ただし、保留中の要求がないか、または要求が発行された直後の場合には、実行要求数が実際のプロセス数よりも少ない場合があります。

さらに、コンカレント・プログラムが実行中の他のプログラムと互換性がないときには、その非互換プログラムが終了するまでコンカレント・プログラムは開始されません。この場合は、保留中の要求があっても実行中の要求数は実際の処理数よりも小さいことがあります。

ステータス

ウィンドウ下部の最上行にあるボタンを使用すると、このフィールドにはマネージャの特定の処理を選択した後のマネージャのステータスが表示されます。

内部コンカレント・マネージャを制御すると、コンカレント・マネージャを個別でも、または一括でも制御できます。内部コンカレント・マネージャがマネージャをアクティブにしたときには、このフィールドはブランクのままです。

パラレル処理環境では、主ノードと2次ノード（またはシステム・キュー）が使用不可能なときには、このフィールドには「使用不可のターゲット・ノード / キュー」が表示されます。

特定のマネージャの制御

マネージャを制御するために選択できる処理は、次のとおりです。

終了 要求を終了して内部コンカレント・マネージャを無効化すると、実行中のすべての要求（実行中のコンカレント・プログラム）は終了し、すべてのマネージャが無効化されます。

事前に個別に無効化されたマネージャには影響はありません。

要求を終了して、マネージャを個別に無効化できます。マネージャの処理対象になっている実行中のすべての要求（実行中のコンカレント・プログラム）が終了します。

一度無効化したマネージャは、そのマネージャを選択して「有効化」ボタンを選択しないと再起動しません。

無効化 内部コンカレント・マネージャを無効化すると、その他のすべてのマネージャも無効化されます。事前に個別に無効化されたマネージャには影響はありません。

マネージャを個別に無効化することもできます。一度無効化したマネージャは、そのマネージャを選択して「有効化」ボタンを選択しないと再起動しません。

マネージャ（内部コンカレント・マネージャを含む）を無効化すると、現在実行中のすべての要求（コンカレント・プログラム）が完了してからマネージャが終了します。

検証 この選択肢が表示されるのは、内部コンカレント・マネージャを選択した場合のみです。

内部コンカレント・マネージャは、各コンカレント・マネージャのプロセスを定期的にモニターします。このプロセスのモニター、つまり PMON 活動を実行するには、「検証」ボタンを選択します。

このボタンを選択することによるもう一つの効果は、内部コンカレント・マネージャがコンカレント・プログラム非互換性ルールをもう一度読み取ることです。

再起動 この選択肢が表示されるのは、個々のマネージャを選択した場合のみです。コンカレント・マネージャを再起動すると、マネージャはその定義をもう一度読み取ります。

「コンカレント・マネージャ定義」フォームを使用して次のような変更を行い、さらにその変更を有効にする場合は、マネージャを再起動する必要があります。

- 稼働シフト割当を変更する
- ターゲット・プロセスの数を変更する
- パラレル・コンカレント・プロセス環境で、ノード情報またはシステム・キュー情報を変更する

有効化 内部コンカレント・マネージャを有効にすると、それ以外のすべてのマネージャも有効になります。ただし、個別に無効化したマネージャは除きます。

PC クライアントから内部コンカレント・マネージャを有効化することはできません。内部コンカレント・マネージャは、サーバー以外からは有効化できません。

内部マネージャが有効である限りは、現在無効な状態のコンカレント・マネージャも個別に有効化できます。マネージャが現在の稼働シフトで動作するように定義されている場合は、内部マネージャがそのマネージャをただちに起動します。

特定のマネージャの表示

コンカレント・マネージャの動作の詳細を表示します。

プロセス 特定のコンカレント・マネージャのプロセスの詳細を表示できます。現在有効なプロセス、移行中または終了中のプロセス、すでに終了したプロセスや無効状態のプロセスが表示されます。

要求 選択したマネージャについて、そのマネージャが処理する実行中および保留中のすべての要求を表示できます。

関連項目

- [「コンカレント・プロセス」ウィンドウ](#) (4-73)
- [「コンカレント・プロセスの概要」](#) (4-1)
- [「コンカレント・マネージャの制御」](#) (4-50)
- [「コンカレント要求のライフ・サイクル」](#) (4-6)
- [「パラレル・コンカレント・プロセスの概要」](#) (4-58)
- [「マネージャおよびマネージャの稼働シフトの定義」](#) (4-22)

「コンカレント・プロセス」ウィンドウ

ウィンドウの上部でノードと名前が表示されている特定のコンカレント・マネージャのプロセスについて、ステータス情報を表示できます。

このウィンドウを表示させると、現在アクティブ中、移行中または終了中のプロセスすべてを自動的に問い合せします。これは、すでに終了しているプロセスまたは非アクティブなプロセスと同じです。

表示される順序は、ステータス値（アクティブ、移行中、終了中、終了、非アクティブ）によって決まり、同じ状態の中ではプロセスが開始された順番によって決まります。

表示されるプロセスの数を減らすには、「要求実行」フォームから「コンカレント要求やマネージャ・データのパージ」レポートを発行することによりレコードを削除できます。レコードは、処理が開始された日からの日数を基準にして削除できます。ただし、現在アクティブなマネージャのレコードは削除できません。

ステータス

このフィールドは更新できません。有効なステータス値には次のものがあります。

有効 現在実行中のマネージャ・プロセスは、「有効」と表示されます。

無効化済 現在稼働していないマネージャ・プロセスは「無効化済」と表示されます。

これらのプロセスが無効化した理由は、システム管理者が「コンカレント・マネージャの管理」ブロックで「無効化」ボタンを選択したか、または内部コンカレント・マネージャによって、あるコンカレント・マネージャの稼働シフトの最後にそのコンカレント・マネージャが無効化されたことによります。

移行中 主ノードから2次ノードへ移行中のマネージャは、「移行中」と表示されます。

パラレル・コンカレント・プロセス環境では、コンカレント・マネージャはそれ自体に割り当てられた主ノードまたは2次ノードで実行されます。マネージャは、主ノードまたは主ノード上のデータベース・インスタンスが使用できない場合、2次ノードに移行します。マネージャは、主ノードが使用可能になると再び主ノードに移行します。

終了中 現在終了中のマネージャ・プロセスは「終了中」と表示されます。

これらのプロセスが終了した理由は、システム管理者が「コンカレント・マネージャの管理」ブロックにある「終了」ボタンを選択したか、またはユーザーが「コンカレント要求」フォームにある「終了」を選択したことによります。

終了済 すでに終了したマネージャ・プロセスは「終了済」と表示されます。

これらのプロセスが終了した理由は、システム管理者が「コンカレント・マネージャの管理」ブロックにある「終了」ボタンを選択したか、またはユーザーが「コンカレント要求」フォームにある「終了」を選択したことによります。

マネージャ識別子「コンカレント」

このフィールドには、処理を識別する個々のコンカレント・マネージャによって生成された数値が表示されます。このフィールドは更新できません。

この数値は、オペレーティング・システム・プロセス ID が使用不可能なときに参照されることがあります。

この数値を使用して、プロセスに関連するログ・ファイルを見ることができます。(これは、「コンカレント要求」フォームの「表示」フィールドから「マネージャ・ログ」を選択して見ることのできるログ・ファイルと同じものです。)

- オペレーティング・システム・レベルでは、ログ・ディレクトリ \$FND_TOP/APPLLOG に入ります。
- コンカレント・マネージャの場合は、W<number>.mgr を使用します。
- 内部モニター処理の場合は、I<number>.mgr を使用します。

マネージャ識別子「Oracle」

このフィールドには、対象のマネージャ・プロセスに関連付けられている ORACLE プロセス ID が表示されます。このフィールドは更新できません。

マネージャ識別子「システム」

このフィールドには、マネージャ・プロセスに関連付けられているオペレーティング・システムのプロセス ID が表示されます。このフィールドは更新できません。

要求識別子「実行中」

このフィールドについては、次の点に注意してください。

- 1つの要求のランタイムはふつう非常に短いため、通常、このフィールドはブランクです。
- 終了したマネージャについては、終了時に処理中の要求の ID が表示されます。

要求識別子「システム」

このフィールドには作成済みコンカレント・プロセスのオペレーティング・システム・プロセス ID が表示されます。

ログ・ファイルの表示

ログ・ファイルを表示する場合は、ウィンドウ下部付近の3つのボタンを使用します。ログ・ファイルには、問題の診断に役立つ情報が記録されています。

要求ログ

このボタンを選択すると、実行中の要求に関連付けられているプロセスのログ・ファイルを表示できます。

内部マネージャ・ログ

このボタンを選択すると、内部コンカレント・マネージャのログ・ファイルを表示できます。

マネージャ・ログ

このボタンを選択すると、要求の実行を開始したコンカレント・マネージャのログ・ファイルを表示できます。

「要求」ウィンドウ

ウィンドウの上部でノードと名前が指定されている選択済みのマネージャについて、実行中および保留状態の要求をすべて表示します。

要求診断ウィンドウ

このウィンドウには、要求が完了した場合はその時刻、完了しなかった場合はその理由を示す解析メッセージが表示されます。

関連項目

- [「コンカレント・マネージャ」ウィンドウ](#) (4-76)
- [「コンカレント・プロセスの概要」](#) (4-1)
- [「コンカレント・マネージャの制御」](#) (4-50)
- [「コンカレント要求のライフ・サイクル」](#) (4-6)
- [「パラレル・コンカレント・プロセスの概要」](#) (4-58)
- [「マネージャおよびマネージャの稼働シフトの定義」](#) (4-22)

「コンカレント・マネージャ」ウィンドウ

このウィンドウを使用すると、自分のコンカレント・マネージャを定義できます。マネージャを実行する時期と、稼働シフトをマネージャに割り当てるときにマネージャが同時に起動できるプログラムの数を決定できます。特殊化のルールを定義して、マネージャが起動できるプログラムを決めます。

「コンカレント・マネージャ」ブロック

アプリケーションとマネージャに定義する名前の組合せにより、マネージャを一意に識別できます。

アプリケーション

マネージャが他のアプリケーションに関連するプログラムを起動するときに、アプリケーション名がその障害となることはありません。マネージャが特定のアプリケーションに関連するプログラムのみを実行するように制限するためには、「特殊化ルール」ウィンドウを使用します。

タイプ

コンカレント・マネージャを一度定義すると、このフィールドの更新はできません。マネージャにはいくつかの種類があります。

コンカレント・マネージャ	コンカレント・マネージャによるコンカレント・プログラムの起動
内部モニター	内部モニターは、パラレル・コンカレント・プロセス環境で内部コンカレント・マネージャをモニターします。内部コンカレント・マネージャが異常終了すると（たとえば、ノードやデータベース・インスタンスがダウンしたため）、内部モニターが別のノードで内部コンカレント・マネージャを再起動します。
取引マネージャ	取引マネージャは、クライアント・マシンからの同期要求を処理します。

『Oracle Inventory ユーザーズ・ガイド』の「取引マネージャ」

『Oracle Inventory ユーザーズ・ガイド』の「取引マネージャの要求」

キャッシュ・サイズ（コンカレント・マネージャのみ）

マネージャによる実行対象の要求を読み込むたびにそのマネージャが記憶する要求の数を入力します。たとえば、マネージャの稼働シフトのターゲット・プロセスが1つでキャッシュ値が3の場合、マネージャは3つの要求を読み込み、その3つの要求を実行してから新しい要求を読み込みます。

要求の読取りでは、マネージャは、実行できる要求のみをキャッシュに書き込みます。たとえば、マネージャが「受注」レポートのみ実行するよう定義すると、マネージャは「受注」要求のみをキャッシュに書き込みます。

1を入力すると、コンカレント・マネージャは、別の要求を処理する準備が整うたびに対象の要求リストを参照する必要があります。

キャッシュ・サイズの数値を大きい値に設定すると、コンカレント・マネージャは要求を実行するたびに対象の要求リストを読み込む必要はなくなります。ただし、コンカレント・マネージャがその要求をキャッシュに読み込み済みの場合は、特定の要求に対する優先度の変更はコンカレント・マネージャに認識されません。さらに、新しい要求に与える優先度を高めた場合でも、バッファが空になり、コンカレント・マネージャが要求リストの参照に戻るまで、その要求は待機する必要があります。バッファ・サイズの数値を大きい値に設定した場合は、要求の待機時間が長時間になる場合があります。

サイトのニーズに対してコンカレント・マネージャの動作が最も効率的になるようキャッシュ・サイズで調整してください。ジョブの処理先の優先度が、あるマネージャに再度割り当てられる傾向がサイトの組織にある場合は、そのマネージャにかなり小さい値のバッファ・サイズを設定する必要があります。

提案：時間のかかる大きなジョブをマネージャに実行させるときには、1の値を入力します。すぐに終了する小さなジョブをマネージャに実行させるときには3または4の値を入力します。

データ・グループ（取引マネージャのみ）

取引マネージャと同じデータ・グループを使用する職責から実行要求があったプログラムのみが取引マネージャで実行されます。取引マネージャは、取引マネージャと同じデータ・グループを使用する職責から実行要求があったプログラムのみを実行します。

生産資源コンシューマ・グループ

該当マネージャの生産資源コンシューマ・グループ。生産資源コンシューマ・グループの詳細の参照先は、次のとおりです。参照：「[Oracle Applications の生産資源コンシューマ・グループ](#)」(7-14)。

並列コンカレント・プロセス詳細

ノード

パラレル・コンカレント・プロセス環境で作業を行っている場合にマネージャを特定のノードで動作させるときには、そのノード名を選択します。

主ノードを使用できる場合、それはコンカレント・マネージャで実行されるノードを指します。主ノードまたはそのノード上のデータベース・インスタンスがダウンすると、コンカレント・マネージャは2次ノードに移行します。主ノードの機能が回復すると、コンカレント・マネージャは、再び主ノードに戻ります。

ノードは、あらかじめ「ノード」フォームを使用して Oracle Applications に登録しておく必要があります。参照：「[「ノード」ウィンドウ](#)」(4-85)。

システム・キュー

パラレル・コンカレント・プロセス環境で作業を行っている場合に、マネージャが一般的なコンカレント・プロセスキュー管理ではなく、プラットフォーム固有のキュー管理システムを使用するように設定するには、そのシステムのキューまたはクラス名を指定します。たとえば、NQS Load Leveler または IBM Load Leveler のようなプラットフォーム固有のキュー管理システムからシステム・キュー名を選択できます。

この1次システム・キューは、主ノードと関連付けるためのキューです。2次システム・キューは、2次ノードと関連付けるためのシステム・キューです。

注意：マネージャが必ずプラットフォーム固有のキュー管理システムを使用するようにするには、適切なモードでコンカレント・マネージャを開始することが必要です（APPLDCP = OSQ に設定します）。使用するプラットフォームがシステム・キューとのインタフェースをサポートしているかどうかを確認するには、プラットフォームのマニュアルを参照してください。Unix 系のプラットフォームでは、該当する「Oracle Application Installation Update」を参照してください。それ以外のすべてのプラットフォームについては、該当する「Oracle Application インストール・ファイル」を参照してください。

即時コンカレント・プログラムは、アプリケーション開発者が Oracle Application Object Library を使用して、プログラム・ライブラリに登録する必要があります。

参照：『Developing Custom Extensions to Oracle Applications』

プログラム・ライブラリ

コンカレント・マネージャは、プログラム・ライブラリに登録されている即時コンカレント・プログラムのみ実行できます。コンカレント・マネージャが特殊化ルールの適用対象である場合は、それ以外の種類の実行可能コンカレント・プログラムを使用するコンカレント・プログラムも実行できます。

取引マネージャは、そのプログラム・ライブラリにリストされているプログラムのみでしか実行できません。

マネージャ動作の定義

ウィンドウの下部の 2 つのボタンを使用するとさらに別のウィンドウが開き、マネージャが動作する時期を定義したり、必要に応じて特定の種類のプログラムのみをマネージャに実行させるように指定することもできます。

稼働シフト マネージャに対して 1 つ以上の稼働シフトを割り当てて、マネージャが動作する時期を定義します。各稼働シフトごとに、マネージャが同時に実行できるプログラム数を変えることができます。

特殊化ルール マネージャに対して特殊化ルールを定義して、マネージャが読み込む要求の種類を定義します。

「稼働シフト」ウィンドウ

コンカレント・マネージャに稼働シフトを割り当てます。稼働シフトは、マネージャを使用可能にする日付と時刻を定義するものです。各稼働シフトごとに、マネージャが実行を開始するプロセス数を定義します。

稼働シフトの定義には、「稼働シフト」フォームを使用します。参照：「[稼働シフト](#)」ウィンドウ (4-81)。

プロセス

稼働シフトで同時に実行するオペレーティング・システム・プロセスの数を入力します。各プロセスは、コンカレント要求を実行できます。

たとえば、稼働シフトが 3 つのターゲット・プロセスを持つように定義すると、マネージャは 3 つまでの要求を同時に実行できます。

スリープ時間 (秒)

この稼働シフト中のマネージャのスリープ時間を入力します。スリープ時間とは、保留中のコンカレント要求（開始を待機中のコンカレント要求）のリストのチェックとチェックの間にマネージャが待機する時間を秒数で表したものです。

提案：実行要求の数が多い期間にはできるだけ短いスリープ時間を設定します。

「特殊化ルール」ウィンドウ

特定の種類の要求のみを実行するようにマネージャを特殊化します。特殊化ルールを設定しないと、マネージャはすべてのコンカレント・プログラムの起動要求を受け入れます。

含む / 除く

ルールに基づいて実行する要求を含めるか除外するかをポップリストで選択します。

タイプ

マネージャに割り当てようとする特殊化ルールのタイプを選択します。選択したルールの動作（許可または禁止）に基づいて、マネージャが要求を実行します。選択できる特殊化ルールのタイプを次に示します。

- 組合せルール

たとえば、選択した組合せルールに合致する要求のみを、マネージャが実行するようにします。また逆に、特定の組合せルールに合致する要求を実行しないようにします。

2つ以上の論理文を結合する組合せ特殊化ルールは、「組合せ特殊化ルール」フォームを使用して定義します。参照：「[組合せ特殊化ルール](#)」ウィンドウ」（4-82）。

- Oracle ID

たとえば、特定の Oracle ID によるプログラムを実行しないようにします。また逆に、コンカレント・マネージャは特定の Oracle ID によるプログラムのみを実行するようにします。

- プログラム

たとえば、選択したプログラムのみを実行しないようにします。また逆に、コンカレント・マネージャはユーザーが選択したプログラムのみを実行するようにします。また「アプリケーション」フィールドにアプリケーションを入力して「名前」フィールドに何も入力しなければ、そのプログラム・タイプを使用する特定のアプリケーションに属するすべてのプログラムを含んだり除外することもできます。

- 要求タイプ（プログラムの）

たとえば、特定の要求タイプによるプログラムを実行しないようにします。また逆に、コンカレント・マネージャはユーザーが選択した要求タイプによるプログラムのみを実行するようにします。

- ユーザー（サインオンに使用したアプリケーション・ユーザー名）

たとえば、特定のユーザーが発行したプログラムをすべて実行しないようにします。また逆に、コンカレント・マネージャは、指定ユーザーが発行したプログラムのみを実行するようにします。

関連項目

- 「「コンカレント・マネージャの管理」ウィンドウ」(4-71)
- 「「コンカレント要求タイプ」ウィンドウ」(4-84)
- 「「ノード」ウィンドウ」(4-85)
- 「「稼働シフト」ウィンドウ」(4-81)
- 「「組合せ特殊化ルール」ウィンドウ」(4-82)
- 「コンカレント・プロセスの概要」(4-1)
- 「コンカレント要求のライフ・サイクル」(4-6)
- 「パラレル・コンカレント・プロセスの概要」(4-58)
- 「マネージャおよびマネージャの稼働シフトの定義」(4-22)

「稼働シフト」ウィンドウ

コンカレント・マネージャ稼働シフトに名前を付けて定義する場合は、このウィンドウを使用します。稼働シフトを定義して、コンカレント・マネージャが動作できる時期を指定します。

各稼働シフトごとに、ある範囲の日数の期間または特定の日付を指定します。参照：「[稼働シフトの定義](#)」(4-25)。

名称

コンカレント稼働シフトには、「Week Days」、「Weeknights」または「Weekends」のように一見してすぐにわかるような名前を付けます。

自 / 至

コンカレント・シフトを開始し、終了させる時刻を入力します。時間形式は HH24:MM とします。たとえば、稼働シフト名が「Week Days」の場合は、開始時刻に「09:00」（午前 9 時）、終了時刻に「17:00」（午後 5 時）を入力します。Oracle Applications では 24 時間制が使用されることに注意してください。

週の始まりの曜日 / 週の終わりの曜日

シフトの最初と最後の日を入力します。たとえば、シフト名が「Week Days」の場合、「週の始まりの曜日」フィールドには「Monday」を入力でき、「週の終わりの曜日」フィールドには「Friday」を入力できます。「週の始まりの曜日」フィールドに値を入力した場合は、「週の終わりの曜日」フィールドにも値を入力する必要があります。この行では「日付」フィールドは使用しません。

日付

特定の日専用の稼働シフトを作成するには、ここに日付を入力します。たとえば、稼働シフトに「Memorial Day」という名前を付け、このフィールドに日付を入力して、この稼働シフトメモリアル・デーの祭日にのみ使用可能にすることができます。

特定の日に専用の稼働シフトは、特定の日付を指定していない稼働シフトを上書きします。このフィールドに値を入力する（日付を指定する）場合は、この行の「曜日」フィールドには値を入力できません。参照：「表 12」（4-27）。

関連項目

- 「コンカレント・マネージャ」ウィンドウ」（4-76）
- 「コンカレント・マネージャの管理」ウィンドウ」（4-71）
- 「コンカレント・プロセスの概要」（4-1）
- 「マネージャおよびマネージャの稼働シフトの定義」（4-22）
- 「マネージャ別稼働シフト・レポート」（4-32）
- 「稼働シフト・レポート」（4-33）

「組合せ特殊化ルール」ウィンドウ

コンカレント・マネージャが読み込むことのできる要求を識別するルールを定義します。ここで定義するルールを使用して、コンカレント・マネージャの機能を特殊化できます。

このウィンドウを使用すると、特殊化行ともいわれる複数の「含む」文および「除く」文を定義できます。また、これらの行を組み合わせると組合せルールともいわれる単独の特殊化ルールを定義することもできます。

ここで定義する組合せルールは、「コンカレント・マネージャ」ウィンドウの中の「特殊化ルール」ウィンドウで定義する個々のルールとは次の2つの点で異なります。

- 「含む」文と「除く」文は組み合わせることができます。これによって、コンカレント・プログラムを実行するために要求をきわめて具体的に識別できます。
- 組合せルール内で複数の「含む」文を使用すると、コンカレント・マネージャの対象範囲がより制限されます。

「特殊化ルール」ウィンドウ（「コンカレント・マネージャ」ウィンドウ内）で定義する個々のルールを使用すると、定義した「含む」ルールが多いほどマネージャの制約は少なくなります。

参照：「コンカレント・マネージャ」ウィンドウ」（4-76）

組合せ特殊化ルール・ブロック

アプリケーション名と組合せ特殊化ルールに対して定義する名前を一緒に使用することにより、ルールを一意に識別できます。

アプリケーション

コンカレント・マネージャが他のアプリケーションに関連するプログラムを起動するとき、アプリケーション名がその障害となることはありません。

特殊化ルール・ブロック

組合せ特殊化ルールを構成する個々のルール（文）を定義します。

- このブロック内の各ルールが1つの文を定義します。
- すべての特殊化ルールがまとまって、組合せ特殊化ルールを定義します。

含む / 除く

実行用ルールに基づく要求を含むか除くかをポップリストから選択します。

タイプ

コンカレント・マネージャに対して実行する特殊化ルールのタイプを選択します。

同一タイプによる2つの「含む」ルールを組み合わせることはできません。

- たとえば、ある Oracle ID に関連付けるプログラムを先に取り込み、別の行で別の Oracle ID に次に関連付けるプログラムを取り込むことはできません。

ルールの動作（「含む」または「除く」）に従って、マネージャは次の条件をもとにプログラムを実行できます。

- Oracle ID

たとえば、特定の Oracle ID によるプログラムを実行しないようにします。また逆に、コンカレント・マネージャは特定の Oracle ID によるプログラムのみを実行するようにします。

- プログラム

たとえば、選択したプログラムのみを実行しないようにします。また逆に、コンカレント・マネージャはユーザーが選択したプログラムのみを実行するようにします。また「アプリケーション」フィールドにアプリケーションを入力して「名前」フィールドに何も入力しなければ、そのプログラム・タイプを使用する特定のアプリケーションに属するすべてのプログラムを含んだり除外することもできます。

- 要求タイプ（プログラムの）
たとえば、特定の要求タイプによるプログラムを実行しないようにします。また逆に、コンカレント・マネージャはユーザーが選択した要求タイプによるプログラムのみを実行するようにします。
- ユーザー（サインオンに使用したアプリケーション・ユーザー名）
たとえば、特定のユーザーが発行したプログラムをすべて実行しないようにします。また逆に、コンカレント・マネージャは、指定ユーザーが発行したプログラムのみを実行するようにします。

関連項目

- [「「コンカレント・マネージャ」ウィンドウ」](#) (4-76)
- [「「コンカレント・マネージャの管理」ウィンドウ」](#) (4-71)
- [「コンカレント・プロセスの概要」](#) (4-1)
- [「プログラムの要求タイプ別グループ化」](#) (4-49)
- [「組合セルールの使用」](#) (4-45)
- [「組合せ特殊化ルールの定義」](#) (4-43)
- [「特殊化ルールと組合セルールの違い」](#) (4-48)
- [「特殊化ルールの定義」](#) (4-34)
- [「特定のプログラムのみを実行するためのマネージャの特殊化」](#) (4-33)

「コンカレント要求タイプ」ウィンドウ

このウィンドウを使用して、各プログラムに共通の要求タイプを割り当てると、いくつかのコンカレント・プログラムを1つのグループとして識別できます。

ここで定義した要求タイプをコンカレント・プログラムに割り当てするには、「コンカレント・プログラム」ウィンドウを使用します。次に、「コンカレント・マネージャ」ウィンドウでコンカレント・マネージャを定義するとき、要求タイプに応じてコンカレント・マネージャがコンカレント・プログラムを実行するかどうかを定義できます。

たとえば、要求タイプを「end-of-month reports」として定義し、その要求タイプをいくつかのコンカレント・プログラムに割り当て、次にコンカレント・マネージャが「end-of-month」要求のみを実行するように定義できます。

コンカレント要求タイプ・ブロック

定義しようとするコンカレント要求の各タイプに名前を付け、説明します。アプリケーション名と要求タイプの両方を使用することにより、コンカレント要求タイプを一意に識別できます。

他のアプリケーションに関連付けられているコンカレント・プログラムに、この要求タイプを割り当てるときにこのアプリケーション名が障害になることはありません。

関連項目

- 「[「コンカレント・マネージャ」ウィンドウ](#)」 (4-76)
- 「[「組合せ特殊化ルール」ウィンドウ](#)」 (4-82)
- 「[「コンカレント・プロセスの概要](#)」 (4-1)
- 「[「プログラムの要求タイプ別グループ化](#)」 (4-49)
- 「[「特定なプログラムのみを実行するためのマネージャの特殊化](#)」 (4-33)

「ノード」ウィンドウ

ノードは1つ以上のプロセッサおよびそれに関連するメモリーから構成されます。パラレル・コンカレント・プロセス環境では（たとえば、クラスタ、超並列処理環境、同種ネットワーク環境など）、個々のノードは、ディスクなどの資源を共有するときを除いてそれぞれが独立して動作します。

コンカレント・マネージャを別々のノードに割り当てると、コンカレント・プロセスの作業負荷を分散してスループットを向上させることができます。コンカレント・マネージャは、割り当てられたノード上で処理を実行します。

ノード・ブロック

ノード

ノードのオペレーティング・システム名を入力します。

プラットフォーム

自分のノードがあるオペレーティング・システム・プラットフォームを選択します。

ベース・パス変数

コンカレント・マネージャのログおよびこのノードの出力ファイルの位置を判断するために、導入マニュアルをよく読んで使用しているプラットフォームの正しい基本パス変数を判断してください。

関連項目

- [「コンカレント・マネージャ」ウィンドウ](#) (4-76)
- [「パラレル・コンカレント・プロセスの概要」](#) (4-58)

「ビューア・オプション」ウィンドウ

コンカレント要求の出力書式に対する MIME タイプを定義するには、このフォームを使用します。これらの MIME タイプは、レポート表示の際に使用されます。

各ファイル形式に対して、1 つ以上の MIME タイプを関連付けることができます。

ユーザーは 1 つの MIME タイプを使用して、特定書式のレポートを表示できます。たとえば、ユーザーは、Microsoft Word の全テキスト書式レポートを表示できます。特定ユーザー用にサポートされている書式の MIME タイプは、次の複数のプロファイル・オプションで設定できます。

- ビューア: HTML アプリケーション用
- ビューア: PCL 用アプリケーション
- ビューア: PDF アプリケーション用
- ビューア: ポストスクリプト・アプリケーション用
- ビューア: テキスト・アプリケーション用

この MIME タイプは、ユーザーがそのファイル書式のレポートを表示した際に、ブラウザのウィンドウに送信されます。

「ビューア・オプション」ブロック

サポートされる各ファイル書式に、1 つ以上の MIME タイプを関連付けます。「ビューア」オプションを定義することによって、各書式のファイル表示に使用可能なアプリケーションを指定できます。

ファイル形式

ファイル形式

MIME タイプ

ファイル出力に使用する MIME タイプ

関連項目

- [「Oracle Application Object Library のプロファイル・オプション」](#) (A-1)

- 「Reports Viewer の定義」 (4-10)

プリンタおよび印刷の概要

ここでは Oracle Applications が印刷命令を処理する方法を説明します。実行サマリーでは、各印刷機能、関連フォームおよび Oracle Applications 間の主な関連性を示します。実行サマリー部分の後に、詳しい説明をします。参照：「[図 18](#)」(5-3)。

管理者用要約

Oracle Applications のレポートは、Oracle Reports によって作成されます。作成されたレポートは、(オペレーティング・システムの印刷コマンドを発行したり、オペレーティング・システムの印刷コマンドを発行するカスタム印刷プログラムをコールする) コンカレント・マネージャによって、オペレーティング・システムに送られます。

Oracle Reports およびレポート生成

出力ファイル内のページ・ブレイク、キャリッジ・リターン、行送り、テキスト太字オン / オフおよびテキスト下線オン / オフ指示は、SRW ドライバ・ファイル内の値によって定義されます。

出力ファイルの印刷前または出力ファイルの印刷後に発行される改ページ・ブレイク、キャリッジ・リターンおよび行送り指示は、プリンタ・ドライバのフォームで定義される Oracle Applications のプリンタ・ドライバの初期化またはリセット文字列に指定する必要があります。

SRW ドライバおよび Oracle Applications のプリンタ・ドライバ

レポートが印刷されないとき (部数が 0 でターゲット・プリンタ・フィールドがブランクのとき)、Oracle Reports は「印刷形式」フォーム内の印刷形式で指定された SRW ドライバを使用します。

レポートを印刷するとき（部数＞0）、Oracle Reports は「プリンタ・ドライバ」フォーム内の Oracle Applications のプリンタ・ドライバで指定された SRW ドライバを使用します。

レポートのサイズは、「印刷形式」フォームを使用して定義された印刷形式内の列と行の値で決定されます。これらの値によって、SRW ドライバ・ファイル内の幅と高さの値が上書きされます。

コンカレント・マネージャによる印刷コマンドの発行 / コール

レポートが完了すると、コンカレント・マネージャは初期化文字列を出力ファイルに送ります。初期化文字列は「プリンタ・ドライバ」フォームを使用して定義されます。

コンカレント・マネージャはリセット文字列を出力ファイルに追加します。リセット文字列は「プリンタ・ドライバ」フォームを使用して定義されます。

Oracle Applications のプリンタ・ドライバは、一般的に印刷コマンドの発行または印刷プログラムのコールという2つの方法のいずれかによって実行されます。

プリンタ・ドライバの実行方法がコマンドの場合は、コンカレント・マネージャは「プリンタ・ドライバ」フォームの「引数」フィールドに入力されたオペレーティング・システムの印刷コマンド引数を発行できます。

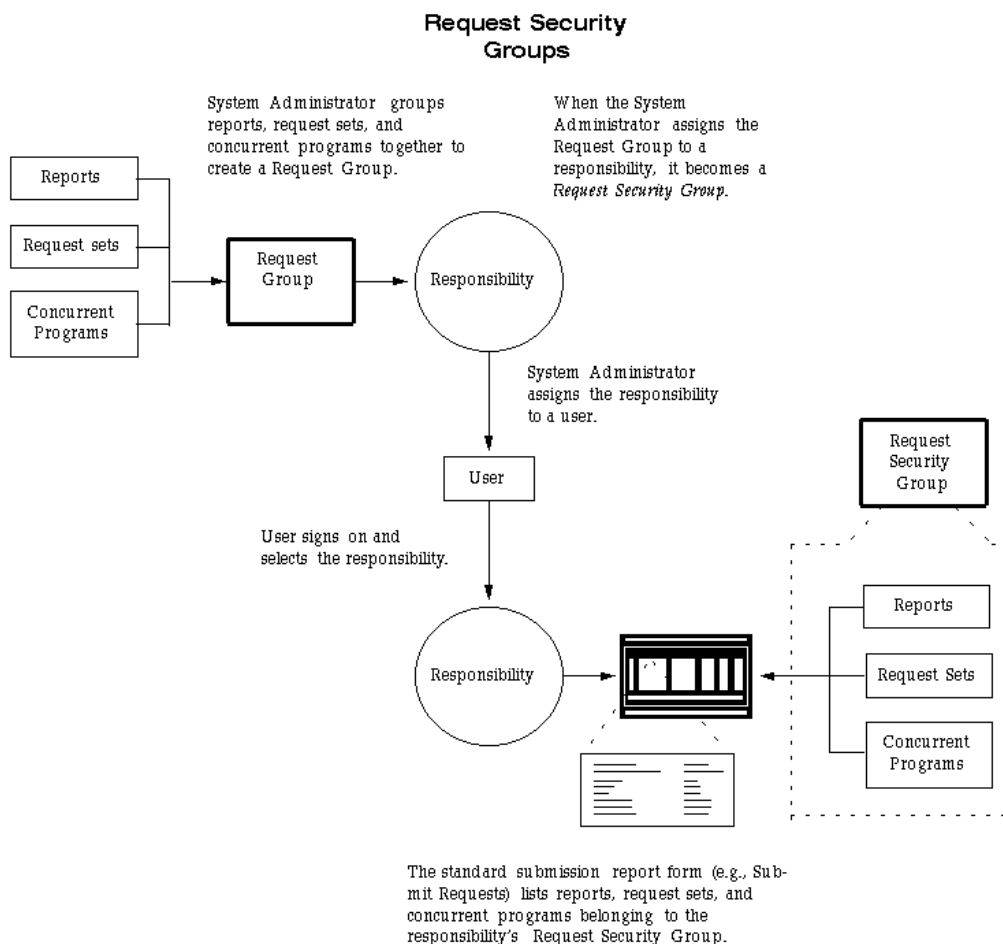
プリンタ・ドライバの実行方法がプログラムの場合は、コンカレント・マネージャは「プリンタ・ドライバ」フォームの「名前」フィールドで（パス付きで）指定されたカスタム印刷プログラムをコールできます。プログラムの引数は、フォームの「引数」フィールドに入力できます。

コンカレント・マネージャによる引数値の定義

コンカレント・マネージャは、オペレーティング・システムの印刷コマンドまたはカスタム印刷プログラムの次の4つの引数の値を定義できます。

- 印刷対象ファイルの名前
- ターゲット・プリンタのオペレーティング・システム名
- 印刷時にヘッダー・ページに印刷されるファイルのタイトル
- 印刷部数

図 18



関連項目

- 「Oracle Applications の印刷サポートのカスタマイズ」 (5-13)
- 「Oracle Reports および Oracle Applications の印刷」 (5-4)
- 「UNIX での Postscript 印刷」 (5-21)
- 「キャラクタ・モード印刷またはビットマップ印刷の設定」 (5-6)

- 「[プリンタ・タイプ、印刷形式およびプリンタ・ドライバ](#)」 (5-5)
- 「[プリンタ・タイプの定義およびプリンタの登録](#)」 (5-8)
- 「[プリンタの設定](#)」 (5-10)

Oracle Reports および Oracle Applications の印刷

Oracle Applications のレポートを実行すると、Oracle Reports により出力の生成とフォーマットが行われます。

各レポートには、そのサイズ、つまりそのレポートに含まれる列と行の数を定義する印刷形式があります。

レポートが完了すると、Oracle Applications のプリンタ・ドライバは、接続先プリンタに対してフォーマット指示を送ります。

Oracle Reports によって生成されたファイルの印刷に関するテキスト、文書およびプリンタの書式設定の指示の要約を次の表に示します。

Oracle Applications のフォーマット指示

インストラクション	説明	構造	フォーム
テキストの書式設定	太字、下線、ページ区切り	SRW ドライバ制御文字	印刷形式（コンカレント要求に無関係のプリンタ；つまり部数 =0、プリンタ・フィールドは空白）
			プリンタ・ドライバ（コンカレント要求に関連付けられているプリンタ）
文書の書式設定	レポートの幅および長さ	印刷形式 列数および行数	印刷形式
プリンタの書式設定	縦長、横長、プリンタの再設定などの印刷指示をプリンタに送る。	プリンタ・ドライバの初期設定文字列および再設定文字列	プリンタ・ドライバ
表 22			

関連項目

- 「[プリンタ・タイプ、印刷形式およびプリンタ・ドライバ](#)」 (5-5)
- 「[プリンタおよび印刷の概要](#)」 (5-1)

プリンタ・タイプ、印刷形式およびプリンタ・ドライバ

プリンタ・タイプによってプリンタが認識できるコマンドが異なります。プリンタ・タイプは、メーカーとモデルにより各プリンタを識別します。

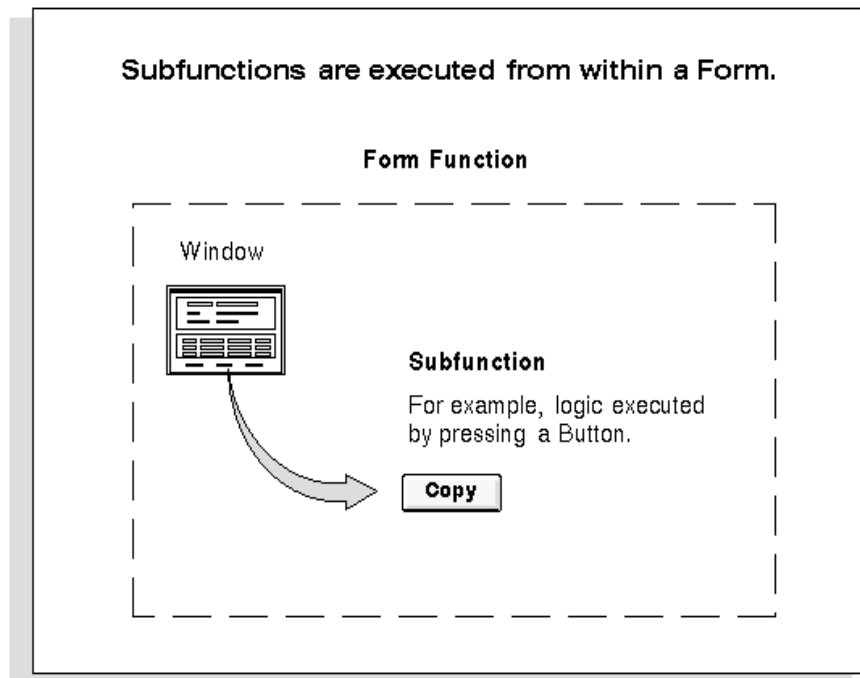
印刷形式はプリンタに対して、印刷出力の形式を指示します。プリンタ・ドライバはプリンタに対して、指定された印刷形式の出力方法を指示するコマンドを送信します。

特定の印刷形式でレポートを印刷する機能は、レポート・ファイルを送信するプリンタのタイプによって異なります。

特定のタイプのプリンタで印刷可能な印刷形式の場合は、そのプリンタ・タイプとオペレーティング・システムに固有なプリンタ・ドライバが必要です。

注意： 希望する印刷形式でレポートを印刷するときには、各プリンタに対して、印刷形式と、その形式で印刷するプリンタ・ドライバの両方を割り当てる必要があります。

図 19



プリンタ・タイプ	プリンタの種類。これはメーカー名とモデル名です。例を2つ示すと、DEC LN03 プリンタや HP Laserjet III プリンタなどです。
印刷形式	レポートの印刷方法の指定。印刷形式で指定される項目は、次のとおりです。 <ul style="list-style-type: none">- ページ当り行数- 各行の幅- 見出しページ印刷の有無
プリンタ・ドライバ	選択した「印刷形式」による印刷方法をプリンタに指示する一連のコマンド。 <ul style="list-style-type: none">- 初期化コマンドでは、印刷方向が設定される。- 再設定コマンドでは、次の印刷ジョブに対するプリンタへのインストラクションが消去される。

関連項目

- 「[「プリンタ・タイプ」ウィンドウ](#)」 (5-28)
- 「[「プリンタ・ドライバ」ウィンドウ](#)」 (5-31)
- 「[「プリンタ」ウィンドウ](#)」 (5-29)
- 「[「印刷形式」ウィンドウ](#)」 (5-30)
- 「[プリンタ・タイプの定義およびプリンタの登録](#)」 (5-8)
- 「[プリンタ・ドライバ](#)」 (5-9)
- 「[プリンタおよび印刷の概要](#)」 (5-1)
- 「[プリンタおよび印刷形式割当の階層](#)」 (5-24)
- 「[印刷形式](#)」 (5-9)

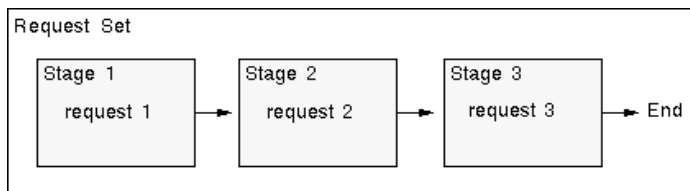
キャラクタ・モード印刷またはビットマップ印刷の設定

キャラクタ・モードの Oracle Reports のコンカレント・プログラムの実行

キャラクタ・モードの Oracle Reports プログラムは、プログラム実行要求に関連した印刷形式から、ページのサイズと方向の情報を取得します。

一部の印刷形式は事前に定義済みであり、システム管理者は必要な場合に、その他の形式を定義できます。

図 20



ビットマップの Oracle Reports コンカレント・プログラムの実行

Oracle Reports プログラムをビットマップ・モードで実行するには、「コンカレント・プログラム」フォーム内のコンカレント・プログラムの定義内容を問い合わせ、「書式」フィールドで PS を選択します。

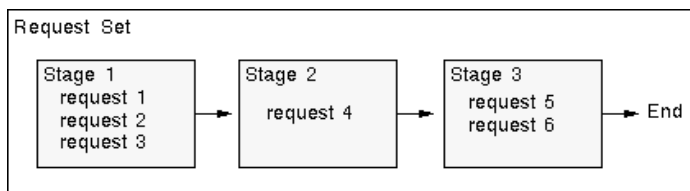
ビットマップの Oracle Reports プログラムは、プログラムの定義内容からページのサイズや印刷方向の情報を取得します（注：ビットマップ・レポートを印刷するときには、印刷形式が必要です）。

プログラムの定義内容に上書きするには、「ORIENTATION」および「PAGE SIZE」の「オプション」フィールドに値を入力してください。

複数の実行オプションを入力するときには、それぞれのオプションを 1 つの空白で分離する必要があります。各オプションの前後に複数の空白は入れないでください。例を示します。

ORIENTATION=LANDSCAPE PAGESIZE=7.5x9

図 21



「オプション」フィールドの PAGESIZE に関する注意

Oracle Reports では、レポートを定義するときにレポートの単位とサイズを、「レポート」>「グローバル・プロパティ」>「単位」の順にメニューで指定します。

ビットマップ・レポートの場合、PAGESIZE は普通 < 幅 > x < 高さ > によりインチ単位で指定しますが、これはレポート定義の内容によって異なります。

レポートの定義で指定した値に上書きするには、「コンカレント・プログラム」フォームの「オプション」フィールドに PAGESIZE パラメータを入力してください（ビットマップ・レポートの場合のみ）。例を示します。

PAGESIZE=7.5x9

PAGESIZE パラメータで指定したサイズが、レポートの設計サイズよりも小さいと、「REP-1212」エラーが発生します。

関連項目

- [「プリンタ・タイプの定義およびプリンタの登録」](#) (5-8)
- [「プリンタおよび印刷の概要」](#) (5-1)

プリンタ・タイプの定義およびプリンタの登録

プリンタを登録すると、Oracle Applications はそのプリンタを認識して、レポート・プログラムの出力をそのプリンタに送ることができます。

Oracle Applications にプリンタを登録するには、最初にプリンタ・タイプを選択して、プリンタの種類を指定してください。多数のプリンタが同一タイプのプリンタとして登録される可能性があるため、各プリンタ・タイプの定義は一回のみとなります。

プリンタを一意に識別するプリンタのオペレーティング・システム名を指定し、プリンタが属するタイプを選択することによって、Oracle Applications にプリンタを登録できます。

たとえば、Oracle Applications のユーザーが、新規に購入したプリンタで印刷できるようにするには、次の操作をしてください。

- 新しいプリンタ（プリンタ 39 など）のオペレーティング・システム名を登録し、プリンタ・タイプ（LN03 など）を選択します。
- 正しいプリンタ・タイプが定義されていない場合は、そのプリンタを登録する前に、新しいプリンタ・タイプ（LN03 など）を定義します。

関連項目

- [「印刷形式」ウィンドウ](#) (5-30)
- [「プリンタおよび印刷の概要」](#) (5-1)
- [「プリンタの設定」](#) (5-10)
- [「印刷形式」](#) (5-9)

印刷形式

印刷形式は、プリンタのページ・フォーマット、つまり列の数（ページの幅）および行の数（ページの長さ）を定義します。

各プリンタ・タイプ（各プリンタなど）には、1つ以上の印刷形式を対応させることができます。

印刷形式を使用することによって、各種プリンタでのレポート・サイズを設定できます。こうして設定した印字形態はプリンタ間で一貫しています。

- たとえば、メニュー・レポートを印刷するとき、左の余白を広くとって、孔をあけるようにする場合もあります。
- システム管理者がこの新しい印刷形式を登録し、プリンタ（タイプ）がその形式をサポートしている場合には、ユーザーはその形式を使用できます。

レポート発行時に、ユーザーはレポートを出力するときの印刷形式を選択します。

- 接続先プリンタで使用可能な形式のみが表示されます。
- 一部のコンカレント・プログラムでは、プリンタまたは印刷形式が事前に定義されていますが、それらの値は変更できません。

関連項目

- 「[印刷形式](#)」ウィンドウ（5-30）
- 「[プリンタ・ドライバ](#)」（5-9）
- 「[プリンタおよび印刷の概要](#)」（5-1）

プリンタ・ドライバ

固有のタイプのプリンタを使用して特定の形式で印刷するには、プリンタ・ドライバを定義してください。プリンタ・ドライバとは、コマンドの指定に従ってレポート出力をターゲット・プリンタに送信する機構のことです。

コンカレント・マネージャは、印刷形式と、レポートを送信するプリンタ（タイプ）に基づいて使用するドライバを決定します。

プリンタ・ドライバを定義すると、希望の形式で印刷するプリンタ・タイプに固有の情報を指定できます。

特定のタイプのプリンタを特定のプラットフォームで印刷するときには、この印刷形式に対応したプリンタ・ドライバを定義する必要があります。

プリンタ・ドライバの定義

プリンタ・ドライバの定義は、次の情報で構成されます。

ドライバ名	プリンタ・ドライバに付ける名前
プラットフォーム	このドライバに固有のプラットフォーム（存在する場合）
SRW ドライバ	SRW レポートの生成に使用される SQL*ReportWriter (SRW) ドライバの名前（存在する場合）。
プリンタ・ドライバ方法	プリンタ・ドライバを起動する方法。各ドライバは、オペレーティング・システムのコマンド、プログラムまたはサブルーチンとして起動できます。
摘要	ドライバの説明。
プログラム名	印刷処理を実行するプログラムの名前。
引数	（方法がプログラムの場合は）プログラムの標準引数、または（方法がコマンドの場合は）印刷コマンドとその引数。
初期化	プリンタを印刷形式に合せて初期設定するエスケープ・シーケンス。初期化文字列はプリンタに対して、各ページに文字を印刷する方向、つまり縦長か横長かを指示します。
再設定	印刷が完了した時点でプリンタを再設定するエスケープ・シーケンス。再設定文字列は、指示の書き込まれた黒板を消すようなもので、次に発行される一連のコマンドが誤って解釈されなくなります。

関連項目

- [「プリンタ・ドライバ」 ウィンドウ](#) (5-31)
- [「カスタム・プリンタ・ドライバの作成」](#) (5-14)
- [「プリンタおよび印刷の概要」](#) (5-1)
- [「プリンタの設定」](#) (5-10)

プリンタの設定

Oracle Applications は事前定義済みのプリンタ・タイプ、印刷形式およびプリンタ・ドライバを提供します。使用する各タイプのプリンタをサポートする印刷形式およびプリンタ・ドライバの組合せを問い合わせるには、「プリンタ・タイプ」フォームを使用します。必要に応じて、または必要であれば、事前定義済みのコンポーネントをカスタマイズします。参照：[「Oracle Applications の印刷サポートのカスタマイズ」](#) (5-13)。

注意：事前定義済みの印刷コンポーネントは、さまざまなプリンタ・タイプまたはオペレーティング・プラットフォームに合せて修正する必要がある場合があります。

プリンタ・サポートを定義するためのフォーム

プリンタ・サポートを定義するには、4つのフォームを使用します。

プリンタ・タイプ

使用しているプリンタのタイプが Oracle Applications の出荷時に登録されていないタイプの場合は、それらのプリンタ・タイプ（メーカーおよびモデル）を定義する必要があります。また、特定のプリンタ・タイプから出力する印刷形式ごとに、印刷形式およびプリンタ・ドライバの組合せをプリンタ・タイプに割り当てる必要があります。

プリンタ

Oracle Applications にプリンタを登録する場合は、プリンタをオペレーティング・システム名で指定し、それにプリンタ・タイプを割り当てます。

プリンタは、前に定義したプリンタ・タイプとしてのみ登録できます。

印刷形式

レポートを生成するため、コンカレント・マネージャは行数と桁数についての印刷形式値（つまり、PAGESIZE トークンの値）を Oracle Reports へ渡します。印刷形式は、レポートの寸法、つまり次の値を決定します。

- ページ当り行数（行数またはページ高さ）
- 各行の幅（列数またはページ幅）

プリンタ・ドライバ

プリンタ・ドライバには、プリンタの形式設定と再起動を実行する初期化文字列および再設定文字列が組み込まれています。特定のプラットフォーム上で特定のプリンタ・タイプに使用する予定の印刷形式ごとに、1つずつ定義済みのプリンタ・ドライバが必要です。

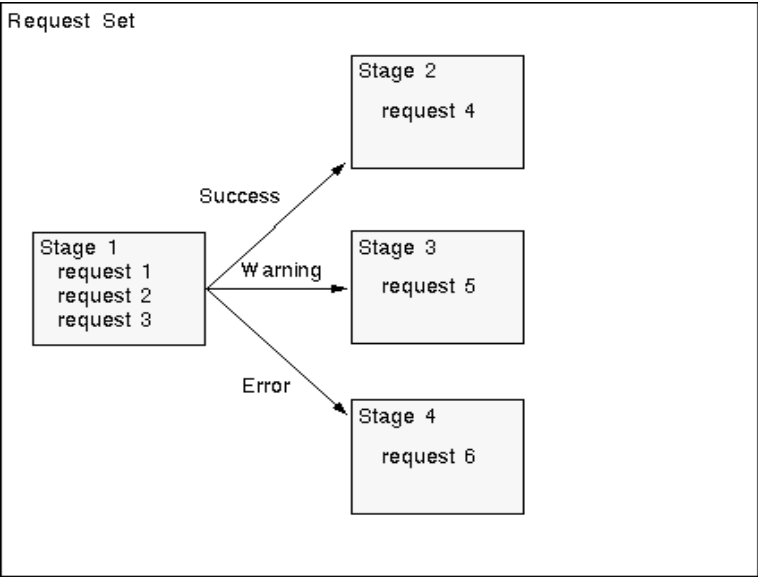
印刷設定の相互関係

次の図では、一本の線は「1つ」を示し、分岐している3つの線は「多数」を示しています。たとえば、プリンタは1つのプリンタ・タイプしか持てませんが、1つのプリンタ・タイプは多数のプリンタへ割り当てることができます。

- 多数のプリンタを、特定の1つのプリンタ・タイプとして登録できます。
- 1つのプリンタ・タイプは、複数の印刷形式をサポートできます。
- 出力に使用する1つの印刷形式ごとに、1つのプリンタ・ドライバをプリンタ・タイプへ割り当てる必要があります。

参照:「[プリンタおよび印刷の概要](#)」(5-1)

図 22



プリンタ設定情報の要求時読み込み

プリンタ設定情報、つまり「プリンタ・タイプ」定義、「印刷形式」定義および「プリンタ・ドライバ」定義は、プログラムの出力を印刷するためにその情報が最初に必要になった時点でメモリー内に読み込まれ（キャッシュされ）ます。

プリンタ設定情報を保持するキャッシュ・リージョンは、コンカレント・マネージャのプライベート・リージョンです。プリンタ設定情報は、コンカレント・マネージャが再起動されるまでメモリー内にキャッシュされたまま残ります。再起動の時点で値が消去され、新しい値がキャッシュされ（メモリー内に読み込まれ）ます。

注意：既存の「プリンタ・タイプ」、「印刷形式」または「プリンタ・ドライバ」を編集したときは、現在有効なすべてのマネージャについて、毎回コンカレント・マネージャを再起動するコマンドを発行してください（ただし、そのタイプ、形式またはドライバがまだ参照されていないか、キャッシュされていない場合は除きます）。

参照：「[コンカレント・マネージャの制御](#)」（4-50）

関連項目

- 「Oracle Applications の印刷サポートのカスタマイズ」 (5-13)
- 「[プリンタ・タイプ] ウィンドウ」 (5-28)
- 「[プリンタ・ドライバ] ウィンドウ」 (5-31)
- 「[プリンタ] ウィンドウ」 (5-29)
- 「[印刷形式] ウィンドウ」 (5-30)
- 「カスタム・プリンタ・ドライバの作成」 (5-14)
- 「プリンタ・ドライバ」 (5-9)
- 「プリンタおよび印刷の概要」 (5-1)
- 「印刷形式」 (5-9)

Oracle Applications の印刷サポートのカスタマイズ

Oracle Applications は、Oracle Reports 出力ファイルのディメンションを定義する印刷形式、選択した印刷形式の出力方法を各種プリンタ・タイプに指示するプリンタ・ドライバのみでなく、プリンタを指定できる定義済みのプリンタ・タイプを数多く提供します。

使用する各プリンタ・タイプをサポートする印刷形式とプリンタ・ドライバの組合せを問い合わせるには、「印刷形式」フォームを使用します。

注意： 事前に定義された印刷構成要素は、様々なプリンタ・タイプやオペレーティング・プラットフォームに合わせて修正が必要な場合があります。

たとえば、各印刷ページの後にブランク・ページまたは余分なページが印刷される場合は、印刷形式で定義されている行数を減らすことが必要です。また、期待どおりに解釈されず、ページ・イジェクト・コマンドが発生するようなエスケープ・シーケンスも書きなおす必要があります。

プリンタ・ドライバ定義を検証し、必要に応じてカスタマイズ

インストールの際に、使用する各プリンタ・タイプについて、プリンタ・ドライバ定義を検証してください。特に、次のドライバ定義項目が重要です。

- 初期化文字列

ページの印刷方向を検証するために、短いレポートを印刷してみます。そのレポート用にプリンタのデフォルト・フォントを変更する場合は、その情報を初期化文字列に組み込みます。

- 再設定文字列

2つの短いレポートを、それぞれ印刷方向を変えて印刷してみます。たとえば、一方は横長形式、もう一方は縦長形式で印刷して、プリンタが正しく再設定するかどうかを検証します。

- 引数

短いレポートを印刷して、オペレーティング・システムの印刷コマンドまたはカスタム印刷プログラムへの引数が正しく解釈されているかどうかを検証します。

新しい印刷形式を定義する必要がある場合は、使用する各プリンタ・タイプについて、新しい印刷形式に割り当てるプリンタ・ドライバを検証します。

Oracle Reports SRW ドライバを検証し、必要に応じてカスタマイズ

SRW ドライバ・ファイルには、Oracle Reports ファイル内での改ページ、太字オン / オフおよび下線オン / オフの属性を設定する制御文字が含まれています。これらの制御文字を正しく解釈しないプリンタ・タイプがある場合は、SRW ドライバ・ファイルをコピーして、それを修正できます。

関連項目

- [「プリンタ・ドライバ」ウィンドウ](#) (5-31)
- [「印刷形式」ウィンドウ](#) (5-30)
- [「カスタム SRW ドライバの作成」](#) (5-20)
- [「カスタム・プリンタ・ドライバの作成」](#) (5-14)
- [「スプール・ファイルの使用」](#) (5-19)
- [「プリンタおよび印刷の概要」](#) (5-1)
- [「初期化文字列および再設定文字列の使用」](#) (5-18)

カスタム・プリンタ・ドライバの作成

必要に応じて、使用しているプリンタ・タイプの初期化文字列および再設定文字列を編集します。その方法については、プリンタのユーザー・ガイドを参照してください。「初期化」フィールドと「再設定」フィールドは、「プリンタ・ドライバ」フォームにあります。

初期化文字列または再設定文字列を編集するのは、次のような場合です。

- プリンタ・タイプで別の制御文字が必要な場合。
- オペレーティング・システムおよびプラットフォームによって制御文字の意味が変わってしまう場合。

- 言語変換によって制御文字の意味が変わってしまう場合。プリンタに、別の文字セットを選択するための特殊制御文字が必要な場合。
- レポート用のプリンタのデフォルト・フォントを変更する場合（初期化文字列のみ）。

プリンタ・ドライバ起動方法

プリンタ・ドライバを起動する方法は、次の3つです。

コマンド	<p>コンカレント・マネージャは、オペレーティング・システムの印刷コマンドおよびその引数を発行できます。</p> <p>オペレーティング・システムの印刷コマンドおよびそのすべての引数は、「プリンタ・ドライバ」フォームの「引数」フィールドに入力します。</p>
プログラム	<p>コンカレント・マネージャは、カスタム印刷プログラムをコールして、そのプログラムに引数を渡すことができます。</p> <p>カスタム印刷プログラムの名前は「プリンタ・ドライバ」フォームの「ドライバ名」フィールドに、そのプログラムに渡す引数は同じフォームの「引数」フィールドに入力します。</p>
サブルーチン	<p>コンカレント・マネージャは、オペレーティング・システムを介してプリンタに印刷コマンドおよび引数を渡す事前定義済みの Oracle Applications サブルーチンをコールすることができます。</p> <p>Oracle Applications により事前定義されているサブルーチンが1つあり、その名前は「プリンタ・ドライバ」フォームの「ドライバ名」フィールドに入力します。</p> <p>ドライバ方法がサブルーチンの場合、引数フィールドは無視されます。ただし、コンカレント・マネージャは、初期化および再設定のエスケープ・シーケンスを読み取ります。</p> <p>UNIX システムのサブルーチン方法の場合、コマンド方法の場合と異なり、オペレーティング・システム・シェルは印刷コマンドと同時に起動しません。</p>

例 - プログラムによるドライバ起動方法の使用

「プログラム」ドライバの起動方法では、独自のカスタム印刷プログラムを定義できます。たとえば、次のようなカスタム・プログラムを会社で作成することが考えられます。このプログラムは、印刷コマンドを発行して実行します。最初にファイルがオープンされるため、ユーザーはそのファイルを編集して、これを第2のファイル名で保存できます。保存が終了すると、第2の（編集済み）ファイルがプリンタに送られます。このような印刷コマンドを発行する方法を、フィルタといいます。

カスタム印刷プログラムの位置

「プリンタ・ドライバ」フォームを使用してカスタム印刷プログラムをコールするには、プログラム名およびそのプログラムへのフル・パスを、「プログラム名」フィールドに入力してください。

オペレーティング・システムの PATH 環境変数によりプログラムの位置が識別できる場合（つまり \$PATH 変数名に入っている場合）は、プログラム名へのパスは必要ありません。

\$PATH 変数に相当するものがないプラットフォームの場合は、フル・パス名を使用してください。パスの長さは最高 255 文字までです。

カスタム印刷プログラムは、コンカレント・プログラムとして Oracle Application Object Library に登録されていませんが、コンカレント・プロセスの完了後にコールされます。

オペレーティング・システム・シェル・スクリプトを使用

オペレーティング・システム・シェル・スクリプトの場合、引数フィールドに正しく入力してある限り、プリンタ・ドライバ方法はコマンドとプログラムのいずれも可能です。

コマンド・シェル・プロシージャ用のスクリプト、たとえば UNIX シェルまたは VMS dcl は、次の場所に存在します。

- \$FND_TOP/\$APPLBIN.

コンカレント・マネージャが値を提供できる引数

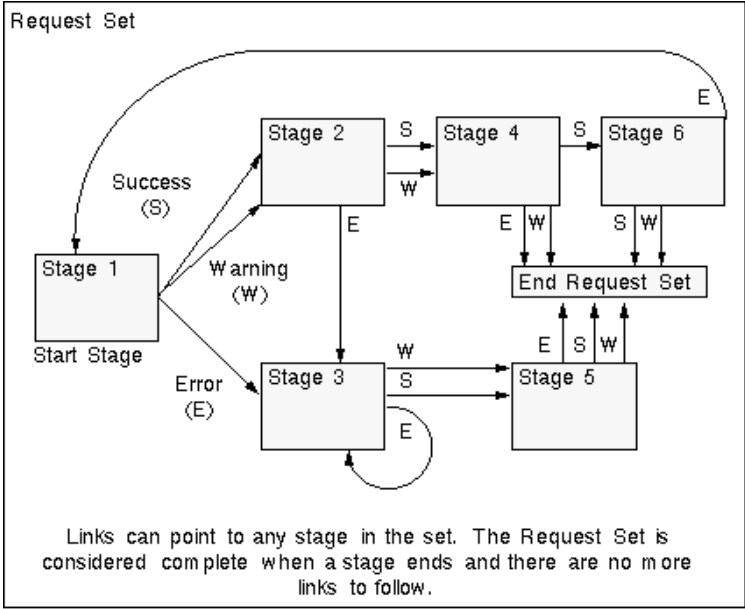
コンカレント・マネージャは、発行するオペレーティング・システム印刷コマンドへの引数、またはコールするカスタム印刷プログラムへの引数として、4 つの異なる値を提供します。これらの値を引数として使用する例を「[図 23](#)」(5-17) および「[表 23](#)」(5-17) に示します。

例 - 印刷コマンドおよび引数の入力

この例では、UNIX 印刷コマンド lp の入力は、コンカレント・マネージャが値を用意できる引数の入力と合せて行います。印刷コマンドは変化しますが、値の検索対象となるトークンは常に同じです。

印刷コマンドはオペレーティング・システムによって異なるので、『Oracle Applications インストレーション・ガイド』を参照してください。

図 23



UNIX lp 印刷コマンドに引数を渡す

引数の構文	検索されたトークンおよび値
-d\$PROFILE\$.PRINTER -d は、宛先のプリンタを呼び出す。	\$PROFILE\$.PRINTER は、要求へ関連付けられているプリンタのオペレーティング・システム名を取り出す。
-n\$PROFILE\$.CONC_COPIES -n は、印刷部数呼び出す。	\$PROFILE\$.CONC_COPIES は、プロファイル・オプション「コンカレント: レポートの印刷部数」の値が実行時に更新されない限り、この値を検索する。
-t"\$PROFILE\$.TITLE" -t は、大見出しまたは見出しのページに印刷するタイトルを呼び出す。	"\$PROFILE\$.TITLE" は、出力ファイルのタイトル（一般に「アプリケーション・ユーザー名. 要求 ID」の形式）を検索する。たとえば、ユーザー John Smith により、コンカレント要求 ID が 64225 のレポートが実行された場合、タイトルは JSMITH.64225 になります。これはオペレーティング・システムによって異なります。
\$PROFILE\$.FILENAME	\$PROFILE\$.FILENAME は、印刷するレポートのファイル名を呼び出す。検索される値は出力ファイル名で、これにはそのファイルへのパスも含まれます。
表 23	

標準入力の使用

「標準入力」を「YES」に設定すると、標準入力がプリンタ・ドライバの処理対象デバイスになるため、レポートの出力を標準入力から直接プリンタに渡すことができます。標準入力を使用する場合の例を次に2つ示します。

- UNIX で、「lpr myfile」ではなく「cat myfile | lpr」のようなパイプを実行する場合。出力ファイルは `stdin`（標準入力）に送られます。
- UNIX コマンド `lpr`。これは、ファイル名が指定されていない場合に標準入力を受け入れます。UNIX コマンド `lpr`。これは、ファイル名が指定されていない場合に標準入力を受け入れます。

「ドライバ起動方法」が「プログラム」または「サブルーチン」に設定されているときには、「標準入力」フィールドは「No」に設定してください。プログラムが標準入力を受け入れない限り、「標準入力」フィールドは常に「No」に設定してください。

注意:「標準入力」が「No」に設定されていると、コンカレント・マネージャが発行する印刷コマンドは非同期に実行されます。つまり、コンカレント・マネージャは、コマンドを発行した後でオペレーティング・システムの応答を待ちません。

関連項目

- [「Oracle Applications の印刷サポートのカスタマイズ」](#) (5-13)
- [「プリンタ・ドライバ」ウィンドウ](#) (5-31)
- [「スプール・ファイルの使用」](#) (5-19)
- [「プリンタおよび印刷の概要」](#) (5-1)
- [「初期化文字列および再設定文字列の使用」](#) (5-18)

初期化文字列および再設定文字列の使用

初期化文字列および再設定文字列を使用して、プリンタの方向および文字セット、行間を設定および再設定します。

初期化文字列および再設定文字列は、制御文字とエスケープ・シーケンスからなります。

- 制御文字は、「^」とそれに続く別の文字の組合せによって表すことができます。
- エスケープ・シーケンスは、「/e」または「\e」によって指定できます。

注意:「プリンタ・ドライバ」フォームを使用して定義するエスケープ・シーケンスには「/e」が使用されます（これは、端末定義で円記号（¥）を[メニュー]キーとして使用しているときは、フォームに円記号を入力できないためです）。リリース 11 のプリンタ・ドライバにアップグレードされている .pdf ファイル内で最初から定義されているエスケープ・シーケンスには、「¥e」が使用されます。

印刷不能文字の値は、8 進モードで表すことができます。たとえば、0x26 は「/046」で表されます。その一例として、次のようなエスケープ・シーケンスを表すとします。

```
^[^L^[16D(0x26)
```

これは次のように表すことができます。

```
/e^L/e16D/046
```

関連項目

- 「Oracle Applications の印刷サポートのカスタマイズ」(5-13)
- 「スプール・ファイルの使用」(5-19)

スプール・ファイルの使用

「スプール・ファイル」が「No」に設定されている場合は、初期化文字列および再設定文字列を入れる一時ファイルが作成され、そのファイルが印刷コマンドまたはプログラムに送られます。

印刷プログラムが独自の一時ファイルを作成する場合に限り、「スプール・ファイル」を「Yes」に設定してください。このオプションは、プログラムによるドライバ起動方法が使用され、印刷プログラムが独自の一時ファイルを作成する場合に使用することをお勧めします。

「スプール・ファイル」が「Yes」に設定されていると、コンカレント・マネージャは一時ファイルを作成しないため、一時ファイルの作成回数が減少します。

「スプール・ファイル」が「Yes」に設定されている場合は、次のように設定することをお勧めします。

- 「標準入力」を「No」にします。
- 初期化フィールドおよび再設定フィールドを NULL（ブランク）にします。

このオプションは、サブルーチンによるドライバ起動方法には適用されません。

関連項目

- 「カスタム SRW ドライバの作成」 (5-20)
- 「カスタム・プリンタ・ドライバの作成」 (5-14)
- 「プリンタおよび印刷の概要」 (5-1)

カスタム SRW ドライバの作成

SRW ドライバは、レポートの生成時に Oracle Reports により読み込まれるもので、どこに改ページを設定し、どの文字を太字または下線付きとして設定するかを宛先プリンタに指示するための制御文字を挿入します。

SRW ドライバは、Oracle Reports 出力ファイルにのみ関与します。SRW ドライバはレポートの生成時に使用されます。完成した出力ファイルをプリンタに送るときには、プリンタ・ドライバが使用されます。

SRW は、DEC LN03 プリンタ、および LN03 と同じ制御文字を認識するすべてのプリンタ用に設計されています。

SRW ドライバ・ファイルの位置および内容

SRW ドライバ・ファイルは \$FND_TOP/\$APPLREP の中に常駐しており、ファイル拡張子は「.prt」です。事前定義済みの SRW ファイル名は次のとおりです。

- A.prt
- P.prt
- L.prt
- PD.prt
- W.prt

カスタム SRW ドライバの作成

任意の SRW ドライバ・ファイルをカスタマイズすると、Oracle Reports ファイルで改ページを設定しテキストを太字または下線付きとして書式設定する制御文字を正しく解決しないプリンタ・タイプもサポートできます。

たとえば、改ページの設定をプリンタに指示する制御文字の変更が必要な場合があります。

on an LN03	on an XYZ LaserInk
new page ...	^L ^[E

Oracle Reports で改ページ、下線付きテキストまたは太字テキスト用の制御文字の書式設定を変更する場合は、次のようにします。

- .prt ファイル（SRW ドライバ）をコピーし、そのコピーを改名します。
- その新しいファイルを新しい制御文字により修正します。
- SRW ドライバ・ファイルの修正済みコピーを \$FND_TOP/\$APPLREP に入れます。
- 新しいドライバが印刷形式またはプリンタ・ドライバ定義に関連付けられている場合は、Oracle Reports はそのドライバを使用します。

注意：SRW ドライバ（.prt ファイル）は、テキストの編集を始める前にコピーし、改名してください。

SRW ドライバ - 印刷形式およびプリンタ・ドライバ

コンカレント・マネージャが Oracle Reports をコールしてレポートを実行するときに、SRW ドライバ名がパラメータとして Oracle Reports に渡されます。

SRW ドライバが必須でないのは、Oracle Reports 以外のプログラム用の形式またはプリンタ・ドライバを使用している場合もあるからです。

「印刷形式」フォームおよび「プリンタ・ドライバ」フォームに入力する SRW ドライバ名は、それぞれ少しずつ使用目的が異なります。

出力ファイルを印刷せずに Oracle Reports プログラムを実行する場合、つまりコピー数が 0 でプリンタ・フィールドがブランクである場合は、レポートの印刷形式に関連付けられている SRW ドライバが使用されます。

Oracle Reports プログラムを実行し、出力ファイルを印刷する場合、つまりコピー数が 0 より大きい場合には、レポートを印刷するプリンタのタイプにとって正しい SRW ドライバを使用するために、そのプリンタ・タイプに関連付けられている SRW ドライバを選択します。

関連項目

- 「Oracle Applications の印刷サポートのカスタマイズ」(5-13)
- 「プリンタ・ドライバ」ウィンドウ (5-31)
- 「プリンタおよび印刷の概要」(5-1)

UNIX での Postscript 印刷

一部の UNIX 環境では、UNIX の *enscript* ユーティリティを使用して印刷すると、レポート出力ファイルを Postscript 形式に変換できます。

注意：enscript を使用する前に、UNIX の文書を参照してください。
enscript の使用方法および引数は、プラットフォームに固有のものである場合があります。

コンカレント・マネージャの引数

コンカレント・マネージャは、オペレーティング・システムの印刷コマンドまたはカスタム印刷プログラムへ、引数として 4 つの異なる値を渡すことができます。コンカレント・マネージャによって渡される 4 つの値すべてを使用した例を参照してください。参照：「表 23」(5-17)。

UNIX の enscript ユーティリティ、およびコンカレント・マネージャが引数として渡す値のうちの 2 つを使用した例を参照してください。参照：「表 25」(5-23)。

enscript の引数および印刷形式

「表 24」(5-22) に、enscript 引数の例をいくつか示します。ここでは、Courier フォントを使用し、レポート出力を縦長、横長、超横長および A4 印刷形式の Postscript に変換しています。

Postscript の印刷形式および enscript の引数

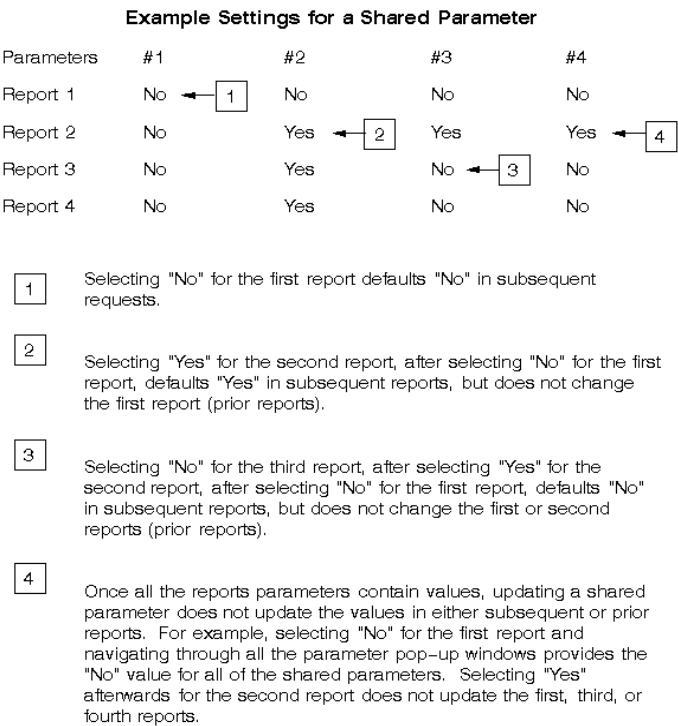
印刷形式	Enscript 引数	説明	結果
縦長	-fCourier10	フォントは Courier の 10 ポイント。	80 文字の縦長
横長	-r -fCourier8	-r を指定すると、プリンタの出力が 90 度回転し、横長モードで印刷される。フォントは Courier の 8 ポイント。	132 文字の横長
超横長	-r -fCourier6	-r を指定すると、プリンタの出力が 90 度回転し、横長モードで印刷される。フォントは Courier の 6 ポイント。	180 文字の横長
A4	-fCourier10	フォントは Courier の 10 ポイント。	132 文字の横長 (A4 用紙)

表 24

例 - enscript を使用した Postscript の印刷

この例では、「プリンタ・ドライバ」ウィンドウの「引数」フィールドに enscript コマンドおよび引数を入力し、「ドライバ方法」を「コマンド」に設定します。

図 24



例 - UNIX の enscript コマンドの使用

構文	説明
-r	enscript 引数。プリンタの出力を 90 度回転させ、横長モードで印刷する。
-fCourier8	enscript 引数。-f を指定するとフォントが選択される。この例の場合、フォントはポイント・サイズが 8 の Courier。
-B	enscript の引数。ページ見出しを省略する。
-P\$PROFILES\$.PRINTER	enscript の引数。-P は出力先となるプリンタの名前の前に付ける。コンカレント・マネージャのトークン。\$PROFILES\$.PRINTER は、対象の要求へ関連付けられているプリンタのオペレーティング・システム名を取り出す引数。
\$PROFILES\$.FILENAME	コンカレント・マネージャ・トークン。\$PROFILES\$.FILENAME は、印刷するレポートのファイル名を呼び出す引数。検索される値は出力ファイル名で、そのファイルへのパスも対象となります。

表 25

この例では、UNIX の `enscript` コマンドはコンカレント・マネージャが値を提供できる 4 つの引数のうちの 2 つと一緒に入力します。

- 引数「`$PROFILE$.CONC_COPIES`」を使用していないため、印刷するコピー数は `enscript` のデフォルト値（通常は 1）に設定されます。
- 引数「`$PROFILE$.TITLE`」を使用していないため、コンカレント・マネージャはバナー・ページまたはヘッダー・ページにレポート・タイトルを印刷するための値を提供しません。

関連項目

- 「[プリンタおよび印刷の概要](#)」(5-1)
- 「[プリンタおよび印刷形式割当の階層](#)」(5-24)
- 「[プリンタの設定](#)」(5-10)

プリンタおよび印刷形式割当の階層

プリンタおよび印刷形式を選択して、その識別情報をコンカレント・プログラムの定義に入れることができます。特定のプリンタに出力を送るようにコンカレント・プログラムを定義した場合、またはコンカレント・プログラムが特定の印刷形式で出力を生成する必要がある場合は、その設定値をユーザーが変更することはできません。また、レポート・セットのデフォルト設定やユーザー・プロファイルのデフォルト設定によって変更することもできません。

デフォルト値は、複数の方法で設定できる場合があります。これにより各種デフォルト設定値の間の階層関連を導くことができ、ここではあるデフォルトが他のデフォルトに優先する形になります。「[図 25](#)」(5-24) はレポート・プログラム実行要求の発行時におけるコンカレント・マネージャからのプリンタまたは印刷形式値の読み込み順序を示しています。

注意：コンカレント・プログラムの定義にデフォルトの印刷形式を指定したり、コンカレント・プログラムに特定の印刷形式で出力することを要求するのみでは、プリンタでその形式が使用可能になるわけではありません。印刷形式およびそれに対応するプリンタ・ドライバを、印刷に使用するプリンタ・タイプに割り当てする必要があります。

図 25

Report Parameters					
Sequence	Prompt	Display	Modify	Shared Parameter	Default Value
					Type
1	Application Name	Yes	Yes	applname	

プリンタ割当の階層

システム管理者は、コンカレント・プログラムまたはレポートの出力先を、特定のプリンタに限定できます。プログラムまたはレポートの出力先を特定のプリンタに限定すると、ユーザー・プロファイルのオプション設定は上書きされ、レポート・セットによるプリンタの選択やユーザー実行時のプリンタの選択はできなくなります。

コンカレント・プログラムの定義に存在しないプリンタがある場合は、次の表に示すようにデフォルトのプリンタ設定を入力できます。ユーザーは、実行時にどのデフォルト設定も上書きできます。

プリンタ割当階層

フォーム	説明
コンカレント・プログラム 「システム管理者」	システム管理者は、出力を常に1つの特定プリンタのみに送るように、コンカレント・プログラムを定義できます。
	この設定は、実行時、またはレポート・セット内のレポートを定義するときは変更できません。
要求セット「システム管理者」	システム管理者は、デフォルトのプリンタをレポート・セット内のレポートに割り当てることができます。
要求セット「アプリケーション・ユーザー」	ユーザーは、自分が所有するレポート・セット内のレポートに、デフォルトのプリンタを割り当てることができます。
	システム管理者は、このデフォルト設定を変更できます。
個別プロファイル値「アプリケーション・ユーザー」	ユーザーは、各自の「個別プロファイル値」フォームを使用して、自分のすべてのレポートにデフォルトのプリンタを割り当てることができます。
	この割当により、システム管理者が設定したデフォルトの「プリンタ」プロファイル・オプションは無効になります。
システム・プロファイル値 「システム管理者」	システム管理者は、導入サイト、Oracle アプリケーション、職責またはユーザーにデフォルトのプリンタを割り当てることができます。
	ユーザーは、実行時にこの設定を無効にできます。
表 26	

印刷形式割当の階層

システム管理者は、コンカレント・プログラムまたはレポートに、特定の印刷形式での出力生成を指示できます。プログラムまたはレポートに特定の印刷形式での出力生成を指示すると、レポート・セットによる印刷形式の選択、またはユーザー実行時の印刷形式の選択はできなくなります。

代替印刷形式の要件

実行方法が Oracle Reports であるすべてのコンカレント・プログラムでは、プログラムの定義時に印刷形式を選択する必要があります。その印刷形式を必須印刷形式として指定していない限り、次の2つの条件が満たされていれば、レポート・セット内の特定レポート用のデフォルトとして、またはレポートを発行する時点で、別の印刷形式を選択できます。

- その代替印刷形式が、コンカレント・プログラムの最小列数および最小行数の値（「コンカレント・プログラム」フォームで入力したもの）を満たしている場合。
- 代替形式が、宛先プリンタのプリンタ・タイプ（「プリンタ・タイプ」フォームで入力したもの）に割り当てられている場合。

印刷形式割当階層

フォーム	説明
コンカレント・プログラム「システム管理者」	システム管理者は、出力をある特定の印刷形式で生成するように、コンカレント・プログラムに指示できます。
	この設定は、実行時、またはレポート・セット内のレポートを定義するときは変更できません。
	ある印刷形式がプログラム定義に入力されていて、しかもそれが必須ではない場合は、その印刷形式は読み取られる最初のデフォルト設定となります。
要求セット「システム管理者」	システム管理者は、デフォルトの印刷形式をレポート・セット内のレポートに割り当てることができます。
要求セット「アプリケーション・ユーザー」	ユーザーは、自分が所有するレポート・セット内のレポートに、デフォルトの印刷形式を割り当てることができます。
	システム管理者は、このデフォルト設定を変更できます。
表 27	

システム管理者によるプリンタおよび印刷形式の設定

プログラム定義、プリンタおよび印刷形式

システム管理者は、「コンカレント・プログラム」フォームを使用して、指定したプリンタ（たとえば機密上安全なオフィス内のプリンタ）のみに出力ファイルを送るように、プログラムに制限を加えることができます。また、レポートに特定の印刷形式の出力生成を指示することもできます。

デフォルトのプリンタおよび印刷形式のセット内レポートへの割当

システム管理者は、「レポート・セット」フォームを使用して、レポート・セット内の各レポートについてデフォルトのプリンタを指定し、各レポートにデフォルトの印刷形式を割り当てることができます。

プロファイル・オプションを使用したデフォルト・プリンタの割当

システム管理者は、「システム・プロファイル値」ウィンドウで「プリンタ」ユーザー・プロファイル・オプションを設定することによって、導入サイト、特定の Oracle Application、特定の職責またはエンド・ユーザー用として、特定のプリンタをデフォルト・プリンタとして指定できます。

次のようにしてデフォルトのプロファイル・オプション値に上書きできます。

- 各自の「個別プロファイル値」フォームを使用して、自分の個別「プリンタ」プロファイル・オプションを設定します。
- レポートを発行するときに、他の（使用可能な）プリンタを選択します。

エンド・ユーザーによるプリンタおよび印刷形式の設定

エンド・ユーザーは次のことができます。

- 各自の「要求セット」フォームを使用して、自分のレポート・セット内のレポートについてデフォルトの印刷形式を設定。
- 「個別プロファイル値」フォームを使用して、自分用のデフォルトのプリンタを指定。
ユーザーは、システム管理者が定義したデフォルトのプロファイル・オプション設定に上書きできます。
- レポートを実行するときに、「レポートの実行」フォームを使用して使用可能なプリンタおよび印刷形式を選択。

デフォルトのプリンタまたは印刷形式が表示されたときに、他のプリンタまたは印刷形式が使用可能であれば、ユーザーはそのデフォルトを変更可能。

関連項目

- 「[プリンタ・ドライバ](#)」ウィンドウ（5-31）
- 「[プリンタ](#)」ウィンドウ（5-29）
- 「[印刷形式](#)」ウィンドウ（5-30）
- 「[プリンタ・ドライバ](#)」（5-9）
- 「[プリンタおよび印刷の概要](#)」（5-1）

- 「[プリンタの設定](#)」 (5-10)
- 「[印刷形式](#)」 (5-9)

「プリンタ・タイプ」ウィンドウ

このウィンドウを使用して、プリンタのタイプを定義し、印刷形式および対応するプリンタ・ドライバを各プリンタ・タイプに割り当てます。

プリンタのタイプを定義しておく、プリンタを特定の「タイプ」として登録することにより、印刷形式およびプリンタ・ドライバの定義を任意の台数のプリンタに割り当てられるようになります。

ユーザーがレポートを送るプリンタを選択すると、使用可能な印刷形式は通常はプリンタのタイプによって決まります。

ただし、「コンカレント・プログラム」ウィンドウは、特定の印刷形式のレポート出力を要求するように定義できます。たとえば、Oracle Reports のプログラムの中には、特定の印刷形式でないと正しく印刷できないために印刷形式を要求するものもあります。

注意: 既存の「プリンタ・タイプ」、「印刷形式」または「プリンタ・ドライバ」を編集したときは、現在アクティブなすべてのマネージャについて、コンカレント・マネージャの再起動コマンドを発行する必要があります。

参照: 「[コンカレント・マネージャの制御](#)」 (4-50)

プリンタ・タイプ・ブロック

タイプ

プリンタのタイプの名称を入力します。例に示したプリンタのタイプは、ライン・プリンタ用が「LINE」、LN03 モデル・プリンタ用が「LN03」となっているはずです。

「プリンタ」ウィンドウでプリンタの登録を行うときに、このプリンタのタイプを選択します。

プリンタ・ドライバ・ブロック

使用するプリンタに印刷形式およびプリンタ・ドライバを割り当てるときにこのブロックを使用します。

「形式」ボタンを押すと「印刷形式」ウィンドウがオープンします。

「ドライバ」ボタンを押すと、「プリンタ・ドライバ」ウィンドウがオープンします。

関連項目

- 「[プリンタ・ドライバ](#)」ウィンドウ (5-31)
- 「[プリンタ](#)」ウィンドウ (5-29)
- 「[印刷形式](#)」ウィンドウ (5-30)
- 「[プリンタ・タイプ](#)、[印刷形式](#)および[プリンタ・ドライバ](#)」 (5-5)
- 「[プリンタ・タイプの定義](#)および[プリンタの登録](#)」 (5-8)
- 「[プリンタ](#)および[印刷の概要](#)」 (5-1)
- 「[プリンタの設定](#)」 (5-10)

「プリンタ」ウィンドウ

プリンタ用オペレーティング・システム名の入力とプリンタ・タイプ（メーカーやモデル名など）の指定を行うことにより、Oracle Applications にプリンタを登録します。

次の点に注意してください。

- レポートの印刷を行う前に、Oracle Applications にプリンタを必ず登録します。
- 登録するプリンタのタイプは、必ずあらかじめ定義しておきます。プリンタのタイプの定義には「[プリンタ・タイプ](#)」ウィンドウを使用します。

「プリンタ」ユーザー・プロファイル・オプションを設定すると、ユーザーがレポートの発行を行うためのデフォルト・プリンタを指定できます。

プリンタ・ブロック

タイプ

使用するプリンタのタイプ（メーカー、モデル名など）を選択します。レポートの中には、特定のタイプのプリンタでないと正しく印刷できないものがあります。

プリンタのタイプを選択するためには、あらかじめ定義しておく必要があります。「[プリンタ・タイプ](#)」ボタンを押すと、ウィンドウがオープンしてプリンタのタイプを定義できます。

関連項目

- 「[プリンタ・タイプ](#)」ウィンドウ (5-28)
- 「[プリンタ・ドライバ](#)」ウィンドウ (5-31)
- 「[印刷形式](#)」ウィンドウ (5-30)

- 「[プリンタ・タイプの定義およびプリンタの登録](#)」(5-8)
- 「[プリンタおよび印刷の概要](#)」(5-1)
- 「[プリンタの設定](#)」(5-10)

「印刷形式」ウィンドウ

このウィンドウを使用して、印刷形式を定義します。印刷形式とは、レポートをどのような形式で印刷するかを指定するものです。たとえば、印刷形式で次の項目を指定できます。

- 各ページの行数
- 各行の幅
- ページの向き（縦長か横長か、など）

Oracle Applications レポートは出荷時の標準的な印刷形式に対応しています。次の印刷形式が定義されています。

- 縦長
- 横長
- 超横長
- A4
- 動的縦長

すべてのレポートですべての印刷形式を使用できるわけではありません。追加の印刷形式を定義してレポートをカスタマイズすることもできます。

一度定義した印刷形式は削除できません。

印刷形式ブロック

印刷形式を定義します。名称とユーザー名を組み合わせることで、印刷形式を固有のものとして識別できます。

注意：既存の「プリンタ・タイプ」、「印刷形式」または「プリンタ・ドライバ」を編集したときは、現在アクティブなすべてのマネージャについて、必ずコンカレント・マネージャの再起動コマンドを発行する必要があります。

参照：「[コンカレント・マネージャの制御](#)」(4-50)

順序

このウィンドウで問合せを実行するときには、印刷形式の表示順序を決めるための番号を入力します。負の番号はゼロよりも前に表示され、正の番号はゼロよりも後ろに表示されます。

名称

複数の印刷形式があるときには、その名称（ユーザー名ではない）がアルファベット順にリスト・ウィンドウに表示されます。

印刷形式の名称は更新できません。

ユーザー名

このユーザー名は、このウィンドウ以外の場所には表示されません。

ヘッダー抑制フラグ

レポート要求者とその時間を示すヘッダー・ページをレポートで印刷できます。このヘッダー・ページを印刷しない形式を定義するには、「ヘッダー抑制」チェックボックスをチェックします。

たとえば、小切手を印刷するときにヘッダー・ページを抑制すると、小切手が上書きされるのを防止することができ、小切手番号の正しい順序を維持できます。

ドライバ

Oracle Reports で生成されたアプリケーション・レポートを印刷するときにコールする Oracle Reports (SRW) ドライバの名称を入力します。このフィールドは、Oracle Reports によって生成されたアプリケーション・レポートのみに使用されます。

関連項目

- [「プリンタ・タイプ」ウィンドウ](#) (5-28)
- [「プリンタ・ドライバ」ウィンドウ](#) (5-31)
- [「プリンタ」ウィンドウ](#) (5-29)
- [「プリンタ・タイプ、印刷形式およびプリンタ・ドライバ」](#) (5-5)
- [「プリンタおよび印刷の概要」](#) (5-1)
- [「印刷形式」](#) (5-9)

「プリンタ・ドライバ」ウィンドウ

このウィンドウを使用して、プリンタ・ドライバおよびプリンタ・コマンドを定義します。

注意：既存の「プリンタ・タイプ」、「印刷形式」または「プリンタ・ドライバ」を編集したときは、現在アクティブなすべてのマネージャについて、必ずコンカレント・マネージャの再起動コマンドを発行する必要があります。

Oracle Applications には出荷時に次の印刷形式に対応したプリンタ・ドライバが付属しています。

- 縦長
- 横長
- 超横長
- A4
- 動的縦長

次のプリンタに対応するプリンタ・ドライバが提供されています。

- Apple
- DEC LN03
- HP Laserjet II、HP Laserjet III、HP Laserjet 4
- HP line printer、HP 256X line printer
- EPOCH
- EPSON FX1050 および DMTX1
- QMS PS 825/925

使用するプリンタが上記以外の場合は、必要なプリンタ・ドライバまたは印刷形式を別途定義してください。

プリンタ・ドライバ・ブロック

参照：「[コンカレント・マネージャの制御](#)」（4-50）

名称

各プラットフォームのプリンタ・ドライバには固有の名前を付ける必要があります。

ユーザー名

このユーザー名は、Oracle Applications から参照されるため、各プラットフォームで固有のものであることが必要です。

SRW ドライバ

プリンタ・ドライバから起動される Oracle Reports (SRW) プリンタ・ドライバがあればその名称を入力します。この情報を必要とするのは、Oracle Reports プログラムのみです。

ファイルへの絶対パスまたはファイル名のみを入力します。ファイル名のみを入力すると、Oracle Applications はそのファイルがディレクトリ \$FND_TOP/\$APPLREP にあるものと見なしします。

「ドライバ方法」リージョン

コマンド	プリンタ・ドライバは、オペレーティング・システム・シェルの内部で実行されます。UNIX の <code>lpr</code> コマンドがその例です。
プログラム	プリンタ・ドライバは、オペレーティング・システム・シェルを経由せずに直接プログラムとして実行されます。 <ul style="list-style-type: none">- 印刷用の C のスタンドアロン・プログラムがその例です。- この方法はコマンド方法よりも実行速度は速いのですが、MS-DOS の PRINT のようなシェル・コマンドにはアクセスできません。
サブルーチン	プリンタ・ドライバは、あらかじめ定義された Oracle Applications のルーチンを実行します。 <ul style="list-style-type: none">- VMS プラットフォームでコールされる <code>SYS\$PRINT</code> ルーチンがその例です。- サブルーチンは、各プラットフォームに固有であり、コンカレント・マネージャからのシステム・コールで直接起動します。

「ドライバ方法」パラメータ・リージョン

スプール・ファイル

プリンタ・ドライバで印刷用のファイル・コピーを作成するかどうかを指定します。ドライバ・メソッドが「プログラム」に設定されているときには、このチェックボックスを選択すると、印刷プログラムではそれ自体のスプール・ファイルが作成されます。

- スプール・ファイルの例は、UNIX の `lpr` コマンドで、このコマンドは `-s` オプションを指定しないと、ファイルのコピーを作成します。

標準入力

プリンタ・ドライバが標準入力からの情報を受け付けるかどうかを指定します。ドライバ・メソッドが「プログラム」に設定されているときには、このチェックボックスは選択しません。プログラムが標準入力を受け付けないときには、このチェックボックスはチェックしないでください。

- UNIX のコマンド `lpr` がその例で、ファイル名が指定されないと標準入力を受け付けます。

プログラム名

次の名称を選択します。

- ドライバ・メソッドが「プログラム」であるときにドライバが起動するプログラム
- ドライバ・メソッドが「サブルーチン」であるときにドライバが起動するサブルーチン

「引数」リージョン

初期化

プリンタに送信する初期化文字列を入力します。この文字列はプリンタ・ドライバが印刷を開始するために必要です。

再設定

印刷完了時にプリンタを待機状態に戻す再設定文字列を入力します。

関連項目

- [「カスタム・プリンタ・ドライバの作成」](#) (5-14)
- [「プリンタ・タイプ、印刷形式およびプリンタ・ドライバ」](#) (5-5)
- [「プリンタ・ドライバ」](#) (5-9)
- [「プリンタおよび印刷の概要」](#) (5-1)

Oracle Applications ヘルプ

Oracle Applications ヘルプのカスタマイズ

Oracle Applications ヘルプ・ファイルは HTML 書式であるため、Netscape Communicator など市販の Web ブラウザ / エディタを使用して容易に変更することができます。独自に作成した HTML および GIF ファイルを、ヘルプ・システムに追加することも可能です。

注意： Oracle Applications が新しいリリースになったとき、またはパッチが配布されたときは、独自に変更した内容をすべての更新されたヘルプ・ファイルに再度適用しない限り、最新の情報にアクセスすることはできません。また、オラクル社では、Oracle Applications ヘルプ・ファイルの各リリース間での変更を示す手段を提供していません。

「Oracle Applications ヘルプのカスタマイズ」には、次のようなトピックがあります。

- [「ヘルプ・ファイルのダウンロードおよびアップロード」](#) (6-1)
- [「ヘルプ・ファイルのリンク」](#) (6-3)
- [「検索インデックスの更新」](#) (6-7)
- [「ヘルプ・ナビゲーション・ツリーのカスタマイズ」](#) (6-7)
- [「グローバル環境でのヘルプのカスタマイズ」](#) (6-8)

ヘルプ・ファイルのダウンロードおよびアップロード

Oracle Applications ヘルプ・ファイルは、データベースに格納されます。次の方法が、これらのファイルの検索およびカスタマイズ中にそれらを差し替えるために提供されています。

編集するためのヘルプ・ファイルのダウンロード

編集するためにヘルプ・ファイルをダウンロードするには、そのファイルを Web ブラウザで開いた後、HTML ファイルとして保存するだけです。

注意: @ 符号がファイル名の最初の文字である場合は、それを削除してください。@ で始まるファイル名は、ヘルプ・システムでエラーの元となる場合があります。

新規または変更されたヘルプ・ファイルのアップロード

Oracle Applications では、新規または変更されたヘルプ・ファイルのアップロード用に FNDGFU と命名されたコマンド行ユーティリティが提供されています。

FNDGFU は、\$FND_TOP/bin ディレクトリにあります。このディレクトリをパスに挿入すると、FNDGFU を簡単にコールできます。

ヘルプ・ファイルのアップロードに際し、FNDGFU では、次の引数が取得されます。

```
FNDGFU <apps/pwd> 0 Y PROGRAM_NAME=FND_HELP
PROGRAM_TAG=<application>:<custom_level> CONTENT_TYPE=<mime_type>
LANGUAGE=<language_code> <filenames>
```

- | | |
|------------------------------|---|
| <apps/pwd> | APPS スキーマのユーザー名とパスワードです。特定のデータベースを指定するには、@ 符号とデータベース SID (@database) を追加します。 |
| <application> | アプリケーションの短縮名です。 |
| <custom_level> | ファイルのカスタマイズ・レベルです。カスタマイズされたヘルプ・ファイルに対しては、100 以上の番号を使用してください。事前にアップロードしたファイルを置換するには、新規ファイルのアップロード時と同じカスタマイズ・レベルを使用します。事前にアップロードされたファイルをデータベースから削除せずに上書きするには、より上位のカスタマイズ・レベルを使用します。 |
| <mime_type> | ファイルの MIME タイプです。 |
| <language_code> | ファイルの言語コードです。 |

注意: この引数は、オプションです。省略すると、言語は、デフォルトで userenv('LANG') に設定されます。

<filenames> アップロードするファイルのスペースで区切られたリストまたは現在のディレクトリのグローバル・ファイル名です。

注意：すべての引数を単一コマンド行に入力します。これらは、ここで個別行に表示され、次の例では、表示媒体に応じて表示されます。

例 1

```
FNDGUF apps/apps@devdb 0 Y PROGRAM_NAME=FND_HELP
PROGRAM_TAG=GL:100 CONTENT_TYPE=text/html LANGUAGE=US *.htm
```

- apps/apps@devdb へ接続します。
- アップロードされたファイルを Oracle General Ledger（GL）ヘルプの一部として識別します。
- 更新されたファイルのカスタマイズ・レベルを 100 として識別します。
- テキスト /html として MIME タイプを識別します。
- 言語を US English（US）として識別します。
- 現在のディレクトリにある全 .htm ファイルをアップロードします（UNIX の場合）。

例 2

```
FNDGUF apps/apps@custdb 0 Y PROGRAM_NAME=FND_HELP
PROGRAM_TAG=FND:100 CONTENT_TYPE=image/gif *.gif
```

- apps/apps@custdb へ接続します。
- アップロードされたファイルを Application Object Library（FND）ヘルプの一部として識別します。
- 更新されたファイルのカスタマイズ・レベルを 100 として識別します。
- イメージ /gif として MIME タイプを識別します。
- 言語を識別しません。言語は、`userenv('LANG')` にデフォルトで設定されます。
- 現在のディレクトリにある全 .gif ファイルをアップロードします（UNIX の場合）。

ヘルプ・ファイルのリンク

Oracle Applications ヘルプ・システムでは、ファイルの名前が変更されたり、ファイルが分割された場合もリンクの動作を維持するハイパーテキスト・リンクの特別構文がサポートされます。以下で詳細に説明する特別構文の例を次に示します。

Oracle Applications Help ウィジェットの詳細は、「ウィジェットの全容」を参照してください。

Oracle Applications ヘルプ・ファイルでは、この構文が使用されます。これはカスタム・ヘルプ・ファイルでも使用できます。また、ファイル名に基づいた従来のハイパーテキスト・リンクを常に使用することもできます。

ヘルプ・ファイルのリンクには、次のトピックも含まれます。

- 「[特別リンク構文](#)」(6-4)
- 「[クロス・アプリケーション・リンク](#)」(6-5)
- 「[関連トピック・リンク](#)」(6-6)
- 「[コンテキスト依存ヘルプ](#)」(6-6)

特別リンク構文

Oracle Applications ヘルプ・ファイルのリンクでは、特定のファイル名ではなく、ファイルに含まれる名前付きアンカーの1つが示されます。Oracle Applications ヘルプ・システムでは、リンクがネゴシエーションされるたびに、ファイルのアンカー名が動的に解決されます。

各ファイルに含まれるアンカー名に関する情報は、アップロードの際、自動的にヘルプ・システムに挿入されます。作成者は、リンクの重複を防ぐために、アンカー名が必ずアプリケーションのヘルプ・ファイルにおいて一意となるようにします。これにより、ファイル名の変更時に、それらのリンクの破損の心配をする必要はありません。

従来の HTML の名前付きアンカー

名前付きアンカーは、次の HTML タグの種類を示します。

名前付きアンカーは、HTML ファイルの本体の任意の場所に置かれ、通常、該当するファイルへの内部リンクに使用されます。ポンド記号 (#) は、アンカー名を示すリンクのアンカー名の前に置かれます。

たとえば、ユーザーが `widghets` というアンカー名のあるセクションへジャンプできるようにするには、HTML を次のように使用します。

ウィジェットの詳細は、「ウィジェットの全容」を参照してください。

． ． ．

ウィジェットの全容

内部文書リンク・サポートのための拡張

Oracle Applications ヘルプ・ファイルでは、この従来の HTML 構文が拡張され、ヘルプ・ファイル内だけでなく、ヘルプ・ファイル間にもリンクが作成されます。特定の名前付きアンカーを含むファイルにリンクするには、アンカー名の前に符号 (@) を置きます。このアンカーが表示されるファイル内の正確な場所にリンクするには、従来の HTML で追加するように、後にアンカー名が続くポンド記号を追加します。この結果、特別構文は次のようになります。

「リンク・テキスト」

たとえば、上記のような widgets アンカーを含むファイルにリンクするには、このアンカーの発生する場所で、HTML を次のように使用します。

Oracle Applications Help ウィジェットの詳細は、「[ウィジェットの全容](#)」(6-5) を参照してください。

このアンカーを含むファイルの最上部にリンクする場合には、ポンド記号のセグメント #widgets は省略できます。

Oracle Applications ヘルプ・ファイルのリンクでは、ポンド記号セグメントが省略されることは、めったにありません。これは、トピックの配置がファイル内またはファイル間で変更された場合も、他のファイルからのこれらのトピックへのリンクは常に適切なファイルおよびトピックの発生するファイル内の正確な点へと移動することを示します。

注意：アンカー名を区別するときに、ケース（大 / 小文字）を使用しないでください。多くの Web ブラウザとは異なり、Oracle Applications ヘルプ・システムでは、アンカー名の大 / 小文字は区別しません。

クロス・アプリケーション・リンク

Oracle Applications ヘルプ・システムでは、URL に関する限り、特定のアプリケーションに関連付けられたすべてのヘルプ・ファイルは、同じディレクトリに存在します。他のアプリケーションに関連付けられたヘルプ・ファイルは、アプリケーションの短縮名を採って命名されたディレクトリに存在します。これらのアプリケーション・ディレクトリのレベルはすべて、ヘルプ・システムと同じです。

別のアプリケーションに関連付けられるヘルプ・ファイルへと移動するリンクを作成するには、全ヘルプ・アプリケーション・ディレクトリの親レベルへと移動する関連リンクを作成し、Oracle Applications の特別リンク構文を完結する前に、該当するアプリケーション・ディレクトリへと移動します。この結果、クロス・アプリケーション・リンク構文は次のようになります。

「リンク・テキスト」

たとえば、上記を説明するウィジェットの全容トピックが、Oracle Payables ヘルプ・トピックであり、Oracle General Ledger ヘルプ・ファイルからそのトピックへリンクする場合には、次のようなリンクを使用します。この場合、AP は Oracle Payables の短縮名です。

Oracle Payables のウィジェットの詳細は、「[Oracle Applications ヘルプ](#)」(6-1) を参照してください。

このように使用された場合、アプリケーション短縮名の大 / 小文字は区別されません。

注意：これらのアプリケーション・ヘルプ・ディレクトリは、URL で使用された際に Oracle Applications ヘルプ・システムにより認識される仮想ディレクトリです。すべてのファイルは、アプリケーション短縮名が付いた状態で実際にデータベースで保存されます。アプリケーション短縮名は、それらに関連付けられる多くの属性の 1 つです。

注意：Oracle Payables の正式な短縮名は、SQLAP です。これが、Oracle Applications ヘルプ・システムで使用される仮想ディレクトリ用に、AP と短縮されました。同様に、Oracle General Ledger の公式短縮名 SQLGL は GL に、Oracle Assets の短縮名 OFA は FA に短縮されました。これらは、例外です。

関連トピック・リンク

Oracle Applications ヘルプ・システムのリンクは、単一ターゲットに限定されません。次の構文を使用して、複数のトピックおよびファイルにリンク先を指定できます。

「[Oracle Applications ヘルプ](#)」(6-1) 関連トピック

ユーザーがリンクをネゴシエーションするときに、関連トピックというヘッダーのあるページが表示されます。このページには、これらのアンカー名に対応するページ・タイトルのリストが含まれ、各タイトルは、該当するファイルへリンクされます。

クロス・アプリケーション・リンクを含めるには、アプリケーション短縮名およびコロンをアンカー名の前に追加します。

「[Oracle Applications ヘルプ](#)」(6-1) 関連トピック

コンテキスト依存ヘルプ

Oracle Applications で「ヘルプ」を要求すると、現在のウィンドウに対応するトピックがオープンされます。「レポート・パラメータ」ウィンドウで「ヘルプ」を要求すると、そのレポートの説明用にヘルプ・ファイルがオープンされます。

Oracle Applications ヘルプ・ファイルには、これらのコンテキスト依存リンクを使用可能にする特別なアンカー名が含まれます。ウィンドウからヘルプを呼び出すと、Oracle Applications では、次のように組み合わせられたフォーム名およびウィンドウ名に基づいてアンカー名が検索されます。

「フォーム機能」ウィンドウの「パラメータ」フィールドに `HELP_TARGET` パラメータを指定することにより、アンカー名の `form_name` 部分を上書きすることもできます。参照：
[「フォーム機能」ウィンドウ](#) (1-28)。

「レポート・パラメータ」ウィンドウからヘルプを呼び出す場合は、Oracle Applications では、次のように構成されたアンカー名が検索されます。

検索インデックスの更新

Oracle interMedia Text を使用すると、Oracle Applications ヘルプ・システムによって提供される検索機能が使用できるようになります。カスタマイズ済みの文書をアップロード後に検索インデックスを再構築する場合は、`aflobbld.sql` というスクリプトを実行します。これにより、ユーザーが実行するすべての検索で、カスタマイズ済みの文書が対象になります。

interMedia の索引を再度作成するには、次のコマンド行を使用します。

```
sqlplus <apps/pwd> @$fnd/sql/aflobbld.sql
```

ここで

<apps/pwd> は、APPS スキーマのユーザー名およびパスワードです。特定のデータベースを指定するには、@ 記号およびデータベース SID (`@database`) を付加します。

例

```
sqlplus apps/apps@devdb @$fnd/sql/aflobbld.sql
```

- このコマンドを実行すると、Oracle SQL*PLUS によって `apps/apps@devdb` に接続されます。
- さらに、Oracle Applications ヘルプの検索インデックスが再度作成されます。

ヘルプ・ナビゲーション・ツリーのカスタマイズ

ヘルプ・ナビゲーション・ツリーをカスタマイズするには、Help Builder アプレットを使用します。

ヘルプ・ナビゲーション・ツリーは、Oracle Applications ヘルプ・システムのナビゲーション・コントロールの基本動作をつかさどる階層データ構造です。これは、ヘルプが表示された場合に、ブラウザ・ウィンドウの左側フレームに表示されるコントロールです。

Help Builder にアクセスするには、Oracle Self-Service Web Applications から、「システム管理」 - 「Help Builder」の順にナビゲートします。

グローバル環境でのヘルプのカスタマイズ

Oracle Applications のヘルプ・システムには、様々な言語に翻訳され、様々な国や地域に対応してローカライズされたファイルが格納されています。企業環境が複数の言語や文化的な境界にまたがっている場合、または Oracle Human Resources を使用している場合は、ヘルプ・ファイルのカスタマイズ時に次の情報が適用される場合があります。

「グローバル環境でのヘルプのカスタマイズ」のトピックは、次のとおりです。

- [「異言語間でのリンク」](#) (6-8)
- [「ローカライズされたヘルプ・ファイルのリンク」](#) (6-9)
- [「コンテキスト依存ヘルプのローカライズ」](#) (6-10)

異言語間でのリンク

Oracle Applications ヘルプの URL で使用される仮想ディレクトリ階層の 1 つ上位の階層は、アプリケーション間のリンクを作成するのに使用されるアプリケーション・ディレクトリです。2 つ上位の階層は、言語ディレクトリで、これは、言語間リンクを作成する場合に使用できます。

異なる言語のヘルプ・ファイルへ移動するリンクを作成するには、次のリンク構文を使用します。

[「Oracle Applications ヘルプ」](#) (6-1) リンク・テキスト

たとえば、Oracle Payables ヘルプのフランス語版の「ウィジェットについて」へリンクするには、次のリンクを使用します。ここで、AP は Oracle Payables の短縮名、F はフランス語のコードを表します。

フランス語のウィジェットの詳細は、「Qu'est-ce qu'un widget?」を参照してください。

言語コードをこの方式で使用する場合は、大文字小文字が区別されます。

注意：異なる言語へのリンクをたどった後は、リンクを逆にたどって元の言語に戻るまで、ユーザーはリンク先の言語環境のままです。リンクは、ヘルプ・ファイル内のリンクでも、ナビゲーション・ツリーからのリンクでも構いません。後者では、常にユーザーの元の言語環境が維持されます。

ローカライズされたヘルプ・ファイルのリンク

アンカー名の末尾にオプションのローカライゼーション・コードを追加すると、同じトピックを扱うファイルを区別できますが、これは、異なる国または地域に対してのみ該当します。ローカライゼーション・コードは検証されず、長さが 100 文字未満の任意の文字列が可能です。

ローカライズ済みのアンカーには、次の構文を使用します。

指定のローカライゼーション環境へリンク

一方、ローカライズ済みアンカーへのリンクは、ちょうど通常の Oracle Applications ヘルプのリンクと同様に作成されます。次の例では、ユーザーを現行のローカライゼーションの内側にとどまらせることを前提としています。

「Oracle Applications ヘルプ」(6-1) 現行のローカライゼーションの内側でのリンク

ユーザーを異なるローカライゼーション環境に移動する場合は、次の構文を使用します。

「Oracle Applications ヘルプ」(6-1) 指定のローカライゼーション環境へリンク

リンクの場合、ローカライゼーション・コードでは、ファイルの相対的な URL の言語部分が変更され、ファイルのアンカー名は変更されません。

ローカライズ済みのアンカーが付いているファイルがヘルプ・システムに存在しない場合、ローカライズ済みのリンクが付加されると、ローカライズされていないアンカーの付いたファイルが表示されます。こうすると、複数のローカル・ヘルプ環境で、ローカライズ済みのファイルが汎用ファイルと共存できます。

注意：ローカライズ済みのアンカーの付いたファイルが別の言語に存在する場合、そのローカライズ済みアンカーが現行の言語で検出できないと、そのファイルが表示されます。これが起こらないようにするには、言語ごとに別個のローカライゼーション・コードを使用してください。

例

UK 地域のウィジェットが UK 以外の国のウィジェットと異なる場合、次に示すアンカーを使用すると、UK 固有のファイルをヘルプ・システムで識別できます。

このファイルを UK のローカライゼーション環境の外側からリンクする場合は、次の構文を使用します。

UK のウィジェットの詳細は、「Sticky Widgets」を参照してください。

UK でのウィジェットが他の国のウィジェットとは異なる箇所でも数回停止し、UK でローカライズされたアンカーが付いているファイルがヘルプ・システムから消去された場合、このリンクは中断することなく、汎用のウィジェット・ヘルプ・ファイルへの移動を開始します。

コンテキスト依存ヘルプのローカライズ

HELP_LOCALIZATION_CODE プロファイル・オプションを適切なレベルで設定して、ローカライズ済みヘルプ・ファイルへのコンテキスト依存呼出しを使用可能にします。詳細は、「ヘルプ・ローカライゼーション・コード」を参照してください。

コンテキスト依存ヘルプのローカライズ済みアンカー名のフォームは、次のとおりです。

[「Oracle Applications ヘルプ」](#) (6-1)

アプリケーション DBA の職務

アプリケーション DBA の職務の概要

アプリケーション・データベース管理者 (DBA) は、Oracle Applications システム管理者と Oracle データベース管理者を兼務します。

Oracle Applications 製品のデータベース権限は、それぞれの Oracle ユーザー名によって決まります。Oracle ユーザー名の作成は Oracle データベース管理者が行い、その Oracle ユーザー名としての登録はシステム管理者が行います。

Oracle ユーザー名

Oracle ユーザー名によって、ユーザーは Oracle データベースの許可ユーザーとして識別されます。

- 各 Oracle ユーザー名は、データベース管理者が割り当てるデータベース・ユーザー名およびパスワードからなります。
- 各 Oracle ユーザー名で、Oracle データベース内の一連のデータにアクセスできます。
- 通常、各 Oracle アプリケーションはそれぞれ専用の Oracle ユーザー名があり、アプリケーション固有のデータがその中に入っています。つまり、アプリケーションが所有する表およびその他のデータベース・オブジェクトは、Oracle ユーザー名によりアクセスされます。

データベース・ユーザー名とパスワードは Oracle データベースに接続するものであり、アプリケーション・ユーザー名およびパスワードは Oracle Applications にアクセスするものであるという点に注意してください。

Oracle データベースは Oracle Applications 製品を介してアクセスするものであり、そのアプリケーションの Oracle ユーザー名により、アクセス権限が付与されます。

関連項目

- 「Oracle ユーザー・ウィンドウ」 (7-3)
- 「Oracle ユーザー名の登録」 (7-2)
- 「アプリケーション・ウィンドウ」 (7-8)

Oracle ユーザー名の登録

Oracle Applications の Oracle ユーザー名はインストレーション・プロセスで自動的に登録されるので、追加の Oracle ユーザー名が必要になった場合に限り、「Oracle ユーザー」ウィンドウを使用してユーザー名を登録します。

Oracle Applications に Oracle ユーザー名を登録する必要があるのは、次の場合です。

- Oracle Application Object Library を使用してカスタム・アプリケーションを作成する場合。
- 追加の Oracle ユーザー名を Oracle Applications 製品に関連付ける必要がある場合。

注意： Oracle ユーザー名を登録する前に、データベース管理者が、Oracle データベースに接続する Oracle ユーザー名を作成しておく必要があります。その上で、「Oracle ユーザー」ウィンドウを使用して、Oracle ユーザー名を登録します。

「Oracle ユーザー」ウィンドウを使用して新しい Oracle ユーザー名を登録すると、アプリケーションの実行に必要な Oracle Application Object Library データベース表に対して必要な権限を設定するためのコンカレント要求が発行されます。これらのデータベース表には、ユーザーが Oracle Application Object Library の機能（メニューおよびフレックスフィールドなど）にアクセスできるようにするための情報が入っています。

Oracle ユーザー名の登録

Oracle Application Object Library を使用して作成したカスタム・アプリケーションに関連付けられている Oracle ユーザー名は、Oracle Application Object Library をアップグレードするたびに再登録する必要があります。

すでに登録済みの Oracle ユーザー名が Oracle Application Object Library データベース表に対して持っている権限を変更した場合、

- Oracle Applications は、Oracle Application Object Library データベース表に対して権限を作成または再作成するコンカレント要求を発行します。
- 変更を有効にするには、このコンカレント要求を正常に完了させることが必要です。

Oracle ユーザー名を「制限付き」として登録

Oracle Applications では、Oracle ユーザー名を「制限付き」Oracle ユーザー名として登録できます。制限付き Oracle ユーザー名の場合は、ユーザーは Oracle Application Object Library 表内のデータを変更することはできません。

データベース管理者は、ユーザーが他の Oracle Applications 表内のデータを変更できないように、Oracle ユーザー名を設定できます。

Oracle ユーザー名を制限付きとして登録するには、「Oracle ユーザー」ウィンドウを使用します。

- Oracle ユーザー名を制限付きとして登録すると、Oracle Application Object Library データベース表に対して読取り専用権限を設定するコンカレント要求が発行されます。
- 制限付き Oracle ユーザー名にアクセスする職責を持つユーザーには、Oracle Application Object Library データベース表に対して読取り専用権限しかありません。したがって、これらのユーザーは、メニューやフレックスフィールドなどの Oracle Application Object Library 機能に関連したデータを挿入または更新、削除することはできません。

データ・グループの定義

データ・グループは、Oracle ユーザー名を Oracle Applications 製品に割り当てるもので、有効なアプリケーションと Oracle ユーザー名のペアのリストを含んでいます。

Oracle Applications 用のデータ・グループはインストール・プロセスで自動的に定義されるので、定義が必要になるのは、利用する追加のデータ・グループがある場合のみです。参照：「[データ・グループの定義](#)」(3-25)。参照：「[「データ・グループ」ウィンドウ](#)」(3-67)。

関連項目

- 「[Oracle ユーザー・ウィンドウ](#)」(7-3)
- 「[アプリケーション DBA の職務の概要](#)」(7-1)

Oracle ユーザー・ウィンドウ

Oracle ユーザー名を Oracle Applications に登録します。Oracle ユーザー名は、Oracle データベースにアクセス権限を与えます。

インストール・プロセスで Oracle ユーザー名は自動的に登録されるので、Oracle ユーザー名を登録する必要はありません。ただし、Oracle Application Object Library を使用してカスタム・アプリケーションを作成するとき、または追加の Oracle ユーザー名を Oracle Applications に関連付けるときを除きます。

Oracle ユーザー名を登録して使用可能にするときには、Oracle ユーザー名と Oracle Application Object Library のデータベース表の間に必要なデータベース権限を設定するためにコンカレント要求を発行します。

Oracle ユーザー名を「制限付きの」Oracle ユーザー名として登録するには、Oracle Application Object Library 表に対する読取り専用の権限を設定するためのコンカレント要求を発行します。「使用可能」に設定した Oracle ユーザー名には、これらの表にアクセスするためのすべての権限があります。「使用禁止」に設定した Oracle ユーザー名には、これらの表にアクセスする権限はありません。

Oracle ユーザー名の登録とその使用可能設定を行わない場合、または登録されている Oracle ユーザー名を無効にした場合、ユーザーはメニューやフレックスフィールドなどの Oracle Application Object Library 機能を使用できなくなります。

インストール・プロセスで登録された Oracle ユーザー名は、パスワード以外は変更しないでください。

既存の Oracle パスワードへの変更を登録するには、Oracle Applications でパスワード変更を登録した「直後に」、データベースへのパスワードの変更を行います。Oracle Applications でパスワードの変更を行い、データベースでその変更をインプリメントするまでは、この Oracle ユーザー名を使用する職責はデータベースに接続できません。

パスワードは『Oracle 8i Server SQL 言語リファレンス・マニュアル』に記載されているパスワード作成ガイドラインに従ったものでなければなりません。パスワードに文字以外の値を使用する場合には、データベース上でそのパスワードを変更する際にパスワードの前後を引用符で囲む必要があります。

警告 : Oracle Application Object Library 表を含んでいる applsys Oracle ユーザー名へのパスワードの変更と他の Oracle ユーザー名へのパスワードの変更を同時に行うことはできません。

パスワードの変更と保存が終了したら、ただちに Oracle Applications からログアウトし、データベースで applsys パスワードを変更し、その後他の操作を行う前に再度サインオンします。applsys パスワードの変更中は、他のユーザーに Oracle Applications へのログオンを禁止してください。

APPS アカウントのパスワード

applsys パスワードは APPS アカウント (APPS, APPS2, APPS3) のパスワードと同一でなければなりません。同一パスワードを使用することにより、複数の会計帳簿が正しく処理されます。

前提条件

- アプリケーションにふさわしい Oracle ユーザー名を作成します (通常、この作業はデータベース管理者によって実行されます)。Oracle ユーザー名には「セッションの作成」権限が含まれていなければなりません。
- または、既存の Oracle ユーザー名パスワードに対して行おうとする変更を調整します。Oracle Applications にパスワードの変更を登録した直後にデータベースでパスワードを変更する必要があります。

注意 : Oracle Applications での変更を登録し、それをデータベースにインプリメントするまでは、Oracle ユーザー名を使用する職責はデータベースに接続できません。

Oracle ユーザー・ブロック

パスワード

Oracle ユーザー名のパスワードを入力します。入力したパスワードは表示されません。既存の Oracle パスワードへの変更を登録するには、Oracle Applications でパスワード変更を登録した直後に、データベースへのパスワードの変更を行います。

Oracle Applications でパスワードの変更を登録し、データベースにその変更をインプリメントするまでは、この Oracle ユーザー名を使用する職責はデータベースに接続できません。

警告 : パスワードを Oracle Application Object Library 表を含む applsys Oracle ユーザー名に変更する場合は、このパスワードを同時に他の Oracle ユーザー名には変更しないようにしてください。

パスワードの変更と保存が終了したら、ただちに Oracle Applications からログアウトし、データベースで applsys パスワードを変更し、その後他の操作を行う前に再度サインオンします。applsys パスワードの変更中は、他のユーザーに Oracle Applications へのログオンを禁止してください。

権限

この Oracle ユーザー名に付与する、Oracle Application Object Library のデータベース表に対する権限タイプを入力します。Oracle Application Object Library 表には、メニュー、ヘルプ・テキスト、フレックスフィールドなどの Oracle Application Object Library の機能に関する説明が含まれています。これらの表にアクセスする権限がないと、このような機能を使用できません。

このフィールドのデフォルト値は「使用可能」です。

使用可能	使用可能に設定した Oracle ユーザー名には、Oracle Application Object Library のデータベース表に対するすべての権限（挿入、問合せ、更新、削除）が含まれます。
制限付き	制限付き Oracle ユーザー名には、Oracle Application Object Library のデータベース表に対して問合せ権限のみがあります。この Oracle ユーザー名では、Oracle Application Object Library のデータを見ることはできませんが、情報の挿入、更新、削除はできません。
使用不可	使用不可に設定した Oracle ユーザー名には、Oracle Application Object Library のデータベース表に対する権限がありません。この Oracle ユーザー名は、Oracle Application Object Library に対して情報の挿入、問合せ、更新、削除を行えず、Oracle Application Object Library の機能を利用できません。

インストール時に構成された Oracle ユーザー名に関連して別の 2 つの権限タイプが表示されます。ただし、これらの権限タイプは値リストからは選択できません。

制限なし	インストレーション・プロセスで Oracle ユーザー名を制限なしの権限を付けて登録すると、すべてのユーザーが「アプリケーション・サインオン・セキュリティ」フォームにアクセスでき、そこで Oracle Applications の有効なユーザー名とパスワードを入力します。
Applsys	インストレーション処理中に Oracle Application Object Library の Oracle ユーザー名が Applsys 権限とともに登録されています。

参照：

[「Oracle Applications セキュリティの概要」\(1-1\)](#)

使用しているオペレーティング・システム用の『Oracle Applications インストレーション・マニュアル』

インストール・グループ

Oracle ユーザー名に関連付けられたインストレーション・グループの値を入力します。インストール・グループ番号は連続した整数でなければなりません。このとき 1 は第 1 会計帳簿（または製品インストールの第 1 セット）、2 は第 2 会計帳簿、3 は第 3 会計帳簿などのようになります。インストール・グループ番号の 0 は、インストレーションを 1 回行うだけで済む製品を表します。

注意： インストレーション・プロセスはカスタム・アプリケーション用の Oracle ユーザー名 ("スキーマ" とともいう) に影響を及ぼさないため、この値はあくまでも参照用であり、現在では使用されていません。

参照：使用しているオペレーティング・システム用の Oracle Applications 導入マニュアル

関連項目

- 「Oracle ユーザー名の登録」 (7-2)
- 「アプリケーション DBA の職務の概要」 (7-1)

「コンカレント衝突ドメイン」ウィンドウ

コンカレント衝突ドメインは、互換性のないコンカレント・プログラム同士が、関連する情報を使用して同時に実行されないようにします。

たとえば、衝突ドメインは、数の範囲である場合があります。2つのコンカレント・プログラムで同じ数の範囲が使用されると、これらのプログラムは非互換となる場合があります。ただし、異なる数の範囲を使用する場合には、衝突しません。

コンカレント・マネージャはコンカレント衝突ドメインを使用して、どのコンカレント・プログラム同士を同時に実行できないかを判別します。たとえば、次のとおりです。

- コンカレント・プログラム A がコンカレント・プログラム B と互換性がないと定義されると、同じコンカレント衝突ドメインを使用して A と B を同時に実行することはできません。
- たとえば、プログラム A と B が実行を依頼されたコンカレント衝突ドメインの Standard に割り当てられている場合、プログラム A と B は同時には実行されません。

▶ 衝突ドメインの定義

1. 固有の「ドメイン」名を入力します。ここに入力した名前は、「要求の発行」ウィンドウでパラメータの値として使用される場合があります。
2. ドメインの一意の「短縮名」を入力します。「短縮名」は、8文字以下にしてください。
3. オプションとして、ドメインに摘要を付けることもできます。

関連項目

- 「Oracle ユーザー・ウィンドウ」 (7-3)
- 「Oracle ユーザー名」 (7-1)
- 「アプリケーション DBA の職務の概要」 (7-1)
- 「アプリケーション・ウィンドウ」 (7-8)

アプリケーション・ウィンドウ

カスタム・アプリケーションを定義するときには、Oracle Applications にいくつかの情報を提供します。アプリケーション名、アプリケーション略称、アプリケーションのベースパスおよびアプリケーションの概要を Oracle Application Object Library に登録する必要があります。Oracle Application Object Library は、その情報を使用して、職責やフォームなどのアプリケーション・オブジェクトを、そのアプリケーションに属するものとして識別します。カスタム・アプリケーションのこの識別機能により、Oracle Applications は、アップグレードのときにアプリケーション・オブジェクトとカスタマイズ内容を保存することができます。アプリケーション・ベースパスは、ユーザーのカスタム・アプリケーションに関連するファイルがどこにあるかを Oracle Application Object Library に通知します。

カスタム・アプリケーションを使用して、カスタム・メニューおよびコンカレント・プログラム、カスタム職責など多くのカスタム・コンポーネントの名前を指定できます。一部のオブジェクトでは、そのオブジェクト名のアプリケーション部分が Oracle Applications の各アプリケーションを識別するのみですが、その他のコンポーネントでは、ここで選択したアプリケーションがカスタム・オブジェクトの機能に影響を及ぼします。

前提条件

- ☐ アプリケーションのベースパスに変換される環境変数を定義します（使用しているオペレーティング・システムの『Oracle Applications Installation Guide』を参照してください）。

アプリケーション・ブロック

カスタム・アプリケーションを登録するときは、そのアプリケーションの参照時に Oracle が識別するための情報を指定します。アプリケーションの名前は変更可能ですが、変更するとアプリケーション名をハードコードしてあるアプリケーション・コードを変更する必要があります。たとえば、アプリケーション名がハードコードされているメニューを通じてプログラム引数を渡すときには、そのプログラム引数も変更する必要があります。

注意：他の人が開発したアプリケーションは、変更によりどのような影響が出るか予想し難いため、名前を変更しないでください。Oracle Applications のアプリケーション名は変更しないでください。これらのアプリケーションには、アプリケーション名へのハードコード参照が含まれている場合があります。

アプリケーション

アプリケーション・ユーザーに対して表示されるリストには、分かりやすい名前が使用されます。

短縮名

Oracle Applications は、アプリケーションの短縮名を使用してフォーム、メニュー、コンカレント・プログラム、その他のアプリケーション・コンポーネントを識別します。短縮名は、ユーザーに対して表示されるときに隠れたフィールドに格納されます。

短縮名には空白を使用しないでください。フォームからコンカレント・プロセスを要求するときおよびメニューからサブルーチンを起動するときにアプリケーション短縮名を使用します。

提案：短縮名には 50 字まで使用できますが、4 文字か 5 文字で設定する方が、その短縮名を使用するアプリケーションの保守やルーチンのコールが容易になります。カスタム・アプリケーション短縮名が将来、Oracle Applications の短縮名と競合する危険を減らすため、カスタム・アプリケーション短縮名を「XX」で始めることをお勧めします。

ベースパス

アプリケーションのディレクトリ・ツリーの一番上のディレクトリを表す環境変数名を入力します。Oracle Applications は、外部ファイルに常駐する処理を定義するときに、ベースパスの下にある特定のディレクトリでアプリケーションの実行ファイルおよびスクリプトを検索します。

一般的には、複数のアプリケーションが同一のディレクトリに書き込まないように、アプリケーションのベースパスはアプリケーションごとに変えてください。

ただし、カスタム職責、メニューおよびその他のコンポーネントに名前を付けるときのみを使用するカスタム・アプリケーションを定義することもあります。そのようなときには、そのアプリケーションと同じフォームを使用する Oracle アプリケーションのベースパスを使用できます。たとえば、Custom_GL アプリケーションを定義するときに、カスタム・アプリケーションに GL_TOP ベースパスを使用できます。

参照：開発環境（『Oracle Applications インストレーション・マニュアル』）

ネットワーク・テスト・ウィンドウ

Oracle Applications によるネットワークのパフォーマンスを評価するには、「ネットワーク・テスト」ウィンドウを使用します。利用可能な待ち時間とバンド幅を把握することによって、パフォーマンスを最大まで引き上げるためにマシン設定を計画したり修正したりできます。

ネットワーク・テストは待ち時間テストとバンド幅テストで構成されます。待ち時間とは、1 つのパケットがクライアント側アプリケーションとサーバーの間を 1 往復するのにかかる時間を指します。バンド幅テストでは、ネットワークがサーバーからクライアントに 1 秒間に転送できるバイト数 (bytes per second:bps) を調べるためにデータ転送レートがチェックされます。

毎回のテスト実行で指定する条件を記述したノートを使用できます。

▶ ネットワーク上の待ち時間をテストする手順は、次のとおりです。

それぞれのテスト試行について試行回数と反復回数を指定します。

1 回の処理で 1 つのパケットがクライアント・アプリケーションからサーバーに転送され、その後そのパケットが返送されます。試行は指定された反復回数で構成されます。1 回の試行で発生したデータ往復の合計時間を反復回数で割ると、平均待ち時間が得られます。これが試行結果です。

5 回の試行回数と 100 回の反復回数がそれぞれデフォルトで設定されています。

「テスト実行」ボタンを選択してテストを実行します。

▶ ネットワーク上のバンド幅をテストする手順は、次のとおりです。

それぞれのテスト試行について試行回数と反復回数を指定します。1 回の処理間で数キロバイトのデータがクライアントからサーバーに転送され、その後そのデータが返送されます。このフォームでデータ転送の平均レートが測定されます。

5 回の試行回数と 10 回の反復回数がそれぞれデフォルトで設定されています。

「テスト実行」ボタンを選択してテストを実行します。

テスト結果の評価

待ち時間テストとバンド幅テストの結果がともに「結果」ブロックに表示されます。

待ち時間結果では、PC クライアントからサーバーへの 1 回分の往復の平均の往復時間が表示されます。

バンド幅結果では、それぞれの試行について平均データ転送レートが KB/ 秒単位で表示されます。

比較用として Redwood Shores の開発本部で完了したテストの結果がサンプル・データ・フィールドに表示されます。これらのテストは理想的な条件下で実施されたため、実際のテスト結果と一致することはほとんどありません。

あるテスト結果がその他の試行結果と著しく異なる場合は、その情報は破棄してください。

データの消去

過去のテスト結果をデータベースから消去するには、「旧テスト・データの消去」ボタンを使用します。


フォルダの管理

デフォルトのフォルダ定義を特定のユーザーまたは職責に割り当てることによってフォルダを管理します。新規所有者へのフォルダ定義の割当て、パブリック (だれからもアクセス可能)

にするフォルダ定義の決定、およびフォルダ自動問合せの動作の設定などのフォルダ定義を管理します。

「デフォルト・フォルダの検索」ウィンドウでは、フォルダまたはフォルダ割当の検索方法に応じてさまざまなタスクを実行できます。

前提条件

 デフォルト・フォルダを作成。参照:「フォルダ内のデータ表示のカスタマイズ」。

▶ フォルダを職責に割り当てる手順は、次のとおりです。

1. 「デフォルト・フォルダの検索」ウィンドウにナビゲートします。職責別デフォルト・フォルダ割当を使用すると、デフォルト・フォルダに割り当てる職責が表示されます。
2. それぞれの職責についてデフォルト・フォルダを割り当てることができます。この職責を持つユーザーがこのフォルダ・ブロックにナビゲートすると、ユーザー・レベルのデフォルトで上書きされていないかぎり、指定済みのデフォルト・フォルダが表示されます。

「フォルダ」フィールドにデフォルト・フォルダ名を入力します。そのフォルダが属するフォルダ・セットの名前が自動的に設定されます。

フォルダ名がわからない場合は先にフォルダ・セットを入力してから、そのセットに属するフォルダを参照してください。

フォルダ・セットのデフォルト・フォルダ定義を保存すると、そのフォルダ・セットは値リストには表示されなくなります。

フォルダ・セット: フォルダ・セットはそれぞれ特定のフォルダ・ブロックに関係付けられています。このためユーザーまたは職責には各フォルダ・セット内にデフォルト・フォルダを1つ設定できます。フォルダ・セット名はブロック内に表示されるレコードを記述するのが一般的ですが、ブロックには関係付けられた複数のフォルダ・セットが登録されていることがあります。

▶ フォルダをユーザーに割り当てる手順は、次のとおりです。

1. 「デフォルト・フォルダの検索」ウィンドウにナビゲートします。「ユーザー別デフォルト・フォルダ割当」を使用すると、的確なユーザーのリストが表示されます。
2. それぞれの職責についてデフォルト・フォルダを割り当てることができます。ユーザーがこのフォルダ・ブロックに移動すると、指定済みのデフォルト・フォルダが表示されます。

「フォルダ」フィールドにデフォルト・フォルダ名を入力します。そのフォルダが属するフォルダ・セットの名前が自動的に設定されます。

フォルダ名がわからない場合は先にフォルダ・セットを入力してから、そのセットに属するフォルダを参照してください。

フォルダ・セットのデフォルト・フォルダ定義を保存すると、そのフォルダ・セットは値リストには表示されなくなります。

フォルダ・セット: フォルダ・セットはそれぞれ特定のフォルダ・ブロックに関係付けられています。このためユーザーまたは職責には各フォルダ・セット内にデフォルト・フォルダを1つ設定できます。フォルダ・セット名はブロック内に表示されるレコードを記述するのが一般的ですが、ブロックには関係付けられた複数のフォルダ・セットが登録されていることがあります。

ソース・タイプ: ユーザーまたは職責のいずれかです。このウィンドウに入力されたレコードによって、「ユーザー」のソース・タイプが使用されます。現在のユーザーの職責に定義済みのデフォルト・フォルダが設定されている場合には、そのデフォルト・フォルダは「職責」のソース・タイプとともにリスト表示されます。

「ユーザー」のデフォルト値によって「職責」のデフォルト値が上書きされます。このウィンドウでは「職責」デフォルト・フォルダは削除できません。

職責: このデフォルト・フォルダ定義を使用する職責です。

▶ 所有権をフォルダに割り当てる手順は、次のとおりです。

1. 「デフォルト・フォルダの検索」ウィンドウにナビゲートします。「フォルダ」を使用すると、フォルダに関する一般情報が表示されます。
2. 所有権の変更が必要なフォルダを選択します。
3. 「所有者の変更」で指定したフォルダの新規所有者を入力するか、「所有者」フィールドの値を変更することによって個別フォルダの所有者を変更します。

フォルダ・セット: フォルダ・セットはそれぞれ特定のフォルダ・ブロックに関係付けられています。このためユーザーまたは職責には各フォルダ・セット内にデフォルト・フォルダを1つ設定できます。フォルダ・セット名はブロック内に表示されるレコードを記述するのが一般的ですが、ブロックには複数のフォルダ・セットが登録されていることがあります。

パブリック: このフォルダ定義をパブリックにするかどうか、つまり所有者以外のユーザーがこのフォルダを使用できるかどうかを設定するものです。このフィールドを使用して、フォルダ定義を共有可能にするかどうかを指定してください。

ユーザーのデフォルト: ユーザーまたは職責でこのフォルダ定義をデフォルトとして使用するかどうかを設定するものです。この定義がデフォルト定義の場合に「デフォルト割当」を使用すると、その定義がデフォルト・フォルダ定義になっているユーザーおよび職責が表示されます。

デフォルト・アサイメント: このフォルダ定義をデフォルトとして使用するユーザーおよび職責です。

▶ フォルダ定義を削除する手順は、次のとおりです。

1. 「デフォルト・フォルダの検索」ウィンドウにナビゲートします。「フォルダ」を使用すると、フォルダに関する一般情報が表示されます。
2. 複数のフォルダの問合せを実行した場合は、削除するフォルダを選択します。
3. フォルダを削除します。フォルダを削除すると、そのフォルダに対するフォルダ定義がユーザーおよび職責のデフォルト割当とともにすべて削除されます。

「言語」ウィンドウ

「言語」ウィンドウは、Oracle Applications で使用できる言語について、情報を検討し変更するために使用します。

言語ブロック

各レコードには、基本言語（英語の場合は「en」など）、地方語が話されている地域コード（米国なら「US」など）、地方語の短縮名（「usaeng」など）、および地方語の完全な名称（「英語（アメリカ）」など）が入っています。また、各レコードに内部言語コード、地域コード、ISO（国際標準化機構）の言語コードと地域コード、地方語のコード・セット、および地方語のステータス標識も含まれています。

通常、製品に添付されているシード・データを更新する必要はありませんが、ユーザーは「翻訳」ウィンドウに「言語の摘要」を表示する方法を変更できます。

摘要

ユーザーは「言語」の摘要を更新して、「変換」ウィンドウに表示されるフィールド名を変更できます。

「地域」ウィンドウ

「地域」ウィンドウは、Oracle Applications で使用する国の値について、情報を検討し変更するために使用します。

地域ブロック

各レコードには、2つの英大文字からなる地域コード（たとえば「US」）、地域の短縮名（「アメリカ」など）、NLS コード、ISO 数字エンティティ・コード、「代替」地域コード、および EU 国別コード、ならびにたとえば「アメリカ合衆国」などの長い説明（「摘要」）が入っています。

通常、製品に添付されているシード・データを更新する必要はありませんが、ユーザーはアプリケーション全体の「値リスト」ウィンドウでの国の表示方法を変更できます。

摘要

「地域」の摘要を更新して、Oracle Application 製品の「値リスト」に表示する地域値を変更することができます。

Oracle Applications の生産資源コンシューマ・グループ

Oracle8i のデータベース生産資源マネージャは、データベース・ユーザーおよびアプリケーションにあるリソースの割当ておよび管理に使用されます。

生産資源コンシューマ・グループおよびリソース・プランでは、ユーザー間で処理中の生産資源の分割方法を指定する方法が提供されます。生産資源コンシューマ・グループでは、同様の生産資源使用要件を持つユーザーのセットが定義されます。総合生産資源計画では、生産資源がどのように生産資源コンシューマ・グループ間で配分されるかが指定されます。

Oracle Applications により、システム管理者は、個々の Oracle Applications ユーザーを生産資源コンシューマ・グループに割り当てることができます。また、コンカレント・プログラムおよびコンカレント・マネージャを、生産資源コンシューマ・グループに割り当てすることもできます。

注意：これらの生産資源コンシューマ・グループは、CPU リソースに対してのみ適用されます。

追加の情報については、『Oracle8i 概要』および『Oracle8i 管理者ガイド』を参照してください。

生産資源コンシューマ・グループの割当て

システム管理者は、特定のユーザーに対してユーザー・プロファイル・オプション「FND: 生産資源コンシューマ・グループ」の値を設定することによって、生産資源コンシューマ・グループにユーザーを割り当てることができます。ユーザーはこのプロファイル・オプションを表示できますが、更新することはできません。参照：「ユーザー・プロファイル」。

システム管理者は、「コンカレント・プログラムの定義」フォームの「パラメータ」ウィンドウで、リソース・コンシューマ・グループにコンカレント・プログラムを割り当てることができます。参照：「コンカレント・プログラム・パラメータ・ウィンドウ」

システム管理者は、「コンカレント・マネージャの定義」フォームでリソース・コンシューマ・グループにコンカレント・マネージャを割り当てることができます。参照：「[コンカレント・マネージャ ウィンドウ](#)」(4-76)

生産資源コンシューマ・グループ割当ての階層

衝突は、単一セッションに関連付けられた生産資源コンシューマ・グループ間で発生します。たとえば、1つの生産資源コンシューマ・グループに割り当てられたコンカレント・マネージャは、別のグループに割り当てられたコンカレント・プログラムを実行する場合があります。同様の状況は、ユーザーが、自分とは別の生産資源コンシューマ・グループを持つ取引マネージャによって管理される取引を実行する際に発生します。この衝突を解決するために、Oracle Applications では階層が使用されます。

コンカレント・プログラムの場合、まずそのプログラムに割り当てられた生産資源コンシューマ・グループがあるかどうかチェックされ、ある場合には、それが使用されます。ない場合には、プログラムを実行しているコンカレント・マネージャがチェックされ、その生産資源コンシューマ・グループが使用されます。コンカレント・マネージャが生産資源コンシューマ・グループに割り当てられていない場合、デフォルト・グループ `Default_Consumer_Group` が使用されます。

取引マネージャが取引プログラムを実行している場合、プログラムに割り当てられている生産資源コンシューマ・グループが（ある場合は、）再度チェックされます。ない場合は、取引マネージャがチェックされます。割り当てられた生産資源コンシューマ・グループが取引マネージャにない場合、取引を開始したセッションのユーザーのプロファイル・オプション値がチェックされます。定義された生産資源コンシューマ・グループがない場合には、デフォルトの生産資源コンシューマ・グループが使用されます。

フォームを実行するユーザーの場合、まずそのユーザーのプロファイル・オプション値がチェックされ、定義されている場合には、それが使用されます。定義されていない場合、デフォルトの生産資源コンシューマ・グループが使用されます。

Oracle Applications におけるコスト・ベース最適化

Oracle Applications とコスト・ベース最適化

Oracle Applications リリース 11i では、SQL 文の最も効率的な実行方法を選択するために、コスト・ベース最適化手法を使用します。この手法により最も効率的な実行計画を決定するために、最適化プログラムは、使用可能なアクセス経路を検討すると共に、対象の SQL 文がアクセスするスキーマ・オブジェクトの統計に基いた情報をファクタリングします。

コスト・ベース最適化を効率的に使用するためには、データベース統計を常に最新の情報にしておく必要があります。Oracle Applications が提供する FND_STATS パッケージには、これらの統計の収集に役立つプロシージャがあります。FND_STATS パッケージは、DBMS_STATS パッケージを使用して統計を収集します。

また、最適化のテストのために、セッション・レベル・パラメータの一部を操作することもできます。これらのパラメータは、システム・プロファイル・オプション「初期化 SQL 文 - カスタム」で制御できます。このプロファイル・オプションのアプリケーション・ユーザー・セッションのための値に、FND_CTL プロシージャを使用してください。

コスト・ベース最適化の詳細は、『Oracle8i 概要』および『Oracle 8i Tuning』を参照してください。

CBO のコンカレント・プログラム

Oracle Applications では、アプリケーション・データベース・オブジェクトの統計を収集するために、FND_STATS パッケージを使用するコンカレント・プログラムが提供されています。参照：「[FND_STATS パッケージ](#)」(8-3) FND_STATS パッケージ：「[FND_STATS パッケージ](#)」(8-3) ページ。

DBMS_STATS については、『Oracle 8i Tuning』を参照してください。

統計の収集と保守には、次のコンカレント・プログラムを使用できます。

- 表統計の収集
- 表統計のバックアップ
- 表統計の復元
- スキーマ統計の収集
- 列統計の収集
- 全列統計の収集
- 全インデックス列の統計の分析

表統計の収集

表統計の収集プログラムは、指定された表の表統計を収集します。このプログラムは、新規の統計を収集する前に、既存の統計のバックアップを FND_STATTAB 表に保存します。また、デフォルトの設定では、関連する索引統計も収集します。

このコンカレント・プログラムは、できるだけ多くの作業を並列化しようとします。ユーザーが分析対象のテーブル上での選択権限を持つ場合に限り、処理が並列化されます。backup_flag の値が「BACKUP」である場合は、新規の統計を収集する前に、既存の統計がすべてエクスポートされます。既存の統計は、export_table_stats を使用してエクスポートされます。エクスポートされたデータは、FND_STATTAB に保存されます。backup_flag の値が「BACKUP」以外の値である場合は、表統計はエクスポートされません。

このコンカレント・プログラムが使用するプロシージャの詳細は、「GATHER_TABLE_STATS」を参照してください。

バックアップ表統計

このコンカレント・プログラムは、指定された表の統計を FND_STATTAB 表に保存します。また、デフォルトの設定では、関連する索引統計および列統計もバックアップします。

統計に対して、表統計の復元プログラムとともに使用できる識別子を割り当てることができます。さまざまな統計識別子を指定して、統計をバックアップすることができます。デフォルトの識別子は「BACKUP」です。指定する識別子により、バックアップのさまざまなバージョンを保管することができます。

表統計の復元

このコンカレント・プログラムは、指定された統計識別子に基づいて、以前にバックアップされた表統計を復元します。デフォルトの統計識別子は「BACKUP」です。

指定された表と関連する索引統計および列統計もインポートされます。

スキーマ統計の収集

このコンカレント・プログラムは、指定されたスキーマ・レベル統計を収集します。

このプログラムは、統計を収集する前に、既存の統計のバックアップも作成します。こうすることにより、新規の統計を収集した後にデータベースの速度が低下した場合は、システムを以前のステータスに復元することができます。このバックアップで使用される統計 ID は「NULL」です。

このプログラムは、スキーマ・レベル統計を収集した後で、指定された列のヒストグラムを FND_HISTOGRAM_COLS 表内に作成します。

また、このプログラムは、FND_EXCLUDE_TABLE_STATS 表で指定されたすべての INTERFACE 表のデフォルト統計をエクスポートします。

このコンカレント・プログラムは、定期的に行うようにしてください。

列統計の収集

全列統計の収集

このコンカレント・プログラムは、FND_HISTOGRAM_COLS 表に基づいて、指定されたスキーマのすべての列統計を収集します。

全インデックス列の統計を分析

このコンカレント・プログラムは、指定されたスキーマ内にあるすべての表の全インデックス列を分析します。

FND_STATS パッケージ

FND_STATS パッケージは、Oracle Applications データベース・オブジェクトの統計を収集するためのプロシージャを提供します。また、現在の統計を表 (FND_STATTAB) に保存したり、その統計を復元したりするためのプロシージャも提供します。このパッケージを利用すると、統計（一部の統計に限る）を並列処理で収集するのが容易になります。このパッケージは、DBMS_STATS パッケージをコールします。

FND_STATS パッケージには、さまざまなタイプのオブジェクトの統計を収集するのに要した時間を記録するために、FND_STATS_HIST 表にデータをインポートするプロシージャもあります。

DBMS_STATS の詳細は、『Oracle 8i チューニング』および『Oracle 8i パッケージ・プロシージャ・リファレンス』を参照してください。

CREATE_STAT_TABLE プロシージャ

このプロシージャは、統計を保管するための表 `stattab` を、`ownname` のスキーマ内に作成します。

このプロシージャは、統計を保管するための表 `fndstattab` を、`APPLSYS` スキーマ内に作成します。この表へのアクセスには、このパッケージ内のプロシージャだけを使用してください。

次の例の最初のプロシージャは、デフォルト表 `fndstattab` を、`FND` 特定スキーマ内に作成します。2 番目のプロシージャは、スキーマ名、表名および表領域名のパラメータを使用します。

構文

```
FND_STATS.CREATE_STAT_TABLE ;

FND_STATS.CREATE_STAT_TABLE (
    schemaname IN VARCHAR2,
    tablename  IN VARCHAR2,
    tblspcname IN VARCHAR2);
```

パラメータ

schemaname	スキーマの名称。
tablename	表の名称。
tblspcname	統計表を作成する表領域。指定を省略すると、統計表はユーザーのデフォルト表領域に作成されます。

BACKUP_TABLE_STATS

このプロシージャは、特定の表の統計を、表 `fndstattab` 内に保存します。`CASCADE` を「`TRUE`」に設定すると、指定された表に関連するすべての索引統計と列統計も保存されます。別の統計識別子 (`statid`) を指定すると、統計を異なるバージョンで保存することができます。

構文

```
FND_STATS.BACKUP_TABLE_STATS (
    errbuf      OUT VARCHAR2,
    retcode     OUT VARCHAR2,
    schemaname  VARCHAR2,
    tablename   VARCHAR2,
    statid      VARCHAR2 DEFAULT 'BACKUP',
    partname    VARCHAR2 DEFAULT NULL,
```

```

        cascade      BOOLEAN  DEFAULT TRUE);

FND_STATS.BACKUP_TABLE_STATS (
    schemaname  VARCHAR2,
    tabname     VARCHAR2,
    statid      VARCHAR2 DEFAULT 'BACKUP',
    partname    VARCHAR2 DEFAULT NULL,
    cascade     BOOLEAN  DEFAULT TRUE);

```

パラメータ

errbuf	コンカレント・プロセスとして実行する場合に必要です。
retcode	コンカレント・プロセスとして実行する場合に必要です。
schemaname	スキーマの名称。
tabname	表の名称。
statid	fnd_stattab 内部で統計と関連付けられる識別子（任意指定）。
partname	表パーティションの名称。表が複数のパーティションに分割されていて、 partname が NULL である場合は、グローバル統計とパーティション表統計の両方がエクスポートされます。
cascade	「TRUE」に設定すると、この表の列統計と索引統計もエクスポートされます。

BACKUP_SCHEMA_STATS プロシージャ

このプロシージャは、特定のスキーマの統計を、表 fnd_stattab 内に保存します。別の statid を指定すると、統計を異なるバージョンで保存することができます。

構文

```

FND_STATS.BACKUP_TABLE_STATS (
    schemaname  VARCHAR2,
    statid      VARCHAR2 DEFAULT NULL
);

```

パラメータ

schemaname	スキーマの名称。ALL は、すべての Oracle Applications スキーマを意味します。
statid	fnd_stattab 内部で統計と関連付けられる識別子（任意指定）。

RESTORE_SCHEMA_STATS プロシージャ

このプロシージャは、`schemaname` により識別されるスキーマ内のすべてのオブジェクトの統計を、指定された `statid` の表 `fnd_stattab` から取り出してから、それらの統計をディクショナリに保存します。

構文

```
FND_STATS.RESTORE_SCHEMA_STATS (  
    schemaname  VARCHAR2,  
    statid      VARCHAR2 DEFAULT NULL  
);
```

パラメータ

schemaname スキーマの名称。ALL は、すべての Oracle Applications スキーマを意味します。

statid `fnd_stattab` 内部で統計と関連付けられる識別子（任意指定）。`fnd_stattab`

RESTORE_TABLE_STATS プロシージャ

このプロシージャは、特定の表の統計を、指定された `statid`(任意指定) の表 `fnd_stattab` から取り出してから、それらの統計をディクショナリに保存します。CASCADE を「TRUE」に設定すると、指定された表に関連するすべての索引統計と列統計もインポートされます。

構文

```
FND_STATS.RESTORE_TABLE_STATS (  
    errbuf  VARCHAR2,  
    retcode VARCHAR2,  
    ownname VARCHAR2,  
    tabname VARCHAR2,  
    statid  VARCHAR2 DEFAULT NULL,  
    partname VARCHAR2 DEFAULT NULL,  
    cascade BOOLEAN  DEFAULT TRUE,  
);
```

```
FND_STATS.RESTORE_TABLE_STATS (  
    ownname  VARCHAR2,  
    tabname  VARCHAR2,  
    statid   VARCHAR2 DEFAULT NULL,  
    partname VARCHAR2 DEFAULT NULL,  
    cascade  BOOLEAN  DEFAULT TRUE,  
);
```


パラメータ

errbuf	コンカレント・プロセスとして実行する場合に必要です。
retcode	コンカレント・プロセスとして実行する場合に必要です。
ownname	スキーマの名称。
tabname	表の名称。
statid	fnd_stattab 内部で統計と関連付けられる識別子（任意指定）。
partname	表パーティションの名称。表が複数のパーティションに分割されていて、 partname が NULL である場合は、グローバル統計とパーティション表統計の両方がエクスポートされます。
cascade	「TRUE」に設定すると、この表の列統計と索引統計もエクスポートされます。

RESTORE_COLUMN_STATS プロシージャ

このプロシージャは、特定の列の統計を、指定された statid（任意指定）の表 fnd_stattab から取り出してから、それらの統計をディクショナリに保存します。このプロシージャには、2つのバージョンがあります。1つのバージョンでは、所有者 / 表および列についての特定の入力値を使用します。もう1つのバージョンでは、FND_HISTOGRAM_COLS 表で指定されたすべての列の統計を復元します。

構文

```
FND_STATS.RESTORE_COLUMN_STATS (
    ownname  VARCHAR2,
    tabname  VARCHAR2,
    colname  VARCHAR2,
    partname VARCHAR2 DEFAULT NULL,
    statid   VARCHAR2 DEFAULT NULL
);

FND_STATS.RESTORE_COLUMN_STATS (
    statid   VARCHAR2 DEFAULT NULL
);
```

パラメータ

ownname	スキーマの名称。
tabname	表の名称。

colname	列の名称。stattab 内部で統計と関連付けられる識別子（任意指定）。
partname	表パーティションの名称。表が複数のパーティションに分割されていて、partname が NULL である場合は、グローバル統計とパーティション表統計の両方がエクスポートされます。
statid	fndstattab 内部で統計と関連付けられる識別子（任意指定）。

GATHER_SCHEMA_STATS プロシージャ

このプロシージャは、特定のスキーマ内のすべてのオブジェクトの統計を収集します。

構文

```
FND_STATS.GATHER_SCHEMA_STATS (  
    schemaname      VARCHAR2,  
    estimate_percent NUMBER  DEFAULT NULL,  
    degree          NUMBER  DEFAULT NULL,  
    internal_flag   NUMBER  DEFAULT NULL,  
    Errors          OUT  Error_Out  
);
```

```
FND_STATS.GATHER_SCHEMA_STATS (  
    errbuf          OUT  VARCHAR2,  
    retcode         OUT  VARCHAR2,  
    schemaname      VARCHAR2,  
    estimate_percent NUMBER  DEFAULT NULL,  
    degree          NUMBER  DEFAULT NULL,  
);
```

このプロシージャには出力パラメータがあります。SQL*Plus プロンプトから直接使用することはできません。SQL ラッパーの例を、次に示します。

例

設定 サーバー出力オン

```
BEGIN  
  
    DECLARE  
        Error FND_STATS.Error_Out;  
  
    BEGIN  
  
        FND_STATS.GATHER_SCHEMA_STATS('&schema', &percent,  
                                       &degree, '&flag', Error);  
        FOR i IN 0..FND_STATS.MAX_ERRORS_PRINTED LOOP  
  
            Error(i) が null の場合終了;
```

```

        dbms_output.put_line('Error
                               #' || i || ' ' || Error(i));

    END LOOP;

END;

```

パラメータ

errbuf	コンカレント・プロセスとして実行する場合に必要です。
retcode	コンカレント・プロセスとして実行する場合に必要です。
schemaname	分析するスキーマ。ALL は、すべての Oracle Applications スキーマを意味します。
estimate_percent	概算手法で使用する行の割合。NULL は、「計算を実行する」ことを意味します。有効範囲は [00,99] です。
degree	並列性の度合。NULL は、表デフォルト値を使用することを意味します。
internal_flag	値「NOBACKUP」を指定すると、GATHER_SCHEMA_STATS プロシージャで現在の統計のバックアップは作成されません。こうすると、GATHER_SCHEMA_STATS プロシージャの処理が短時間で完了します。「INTERNAL」は、DBMS_STATS ではなく ANALYZE コマンドが使用されることを意味します。
errors	出力エラー。

例外

ORA-20000: スキーマが見つからないか、または必要な権限がありません。

ORA-20001: 入力値が不正です。

GATHER_SCHEMA_STATISTICS プロシージャ

このプロシージャは、SQL プロンプトから GATHER_SCHEMA_STATS を使用するためのより単純なコールを提供します。このプロシージャは、単一のスキーマのスキーマ統計を収集する場合にのみ使用してください。パラメータ **schemaname** は、「ALL」にしないでください。このプロシージャは、GATHER_SCHEMA_STATS を内部的に呼び出します。このプロシージャと GATHER_SCHEMA_STATS との違いは、GATHER_SCHEMA_STATISTICS には出力パラメータがないため、SQL プロンプトからのコールが容易であるという点です。

構文

```

FND_STATS.GATHER_SCHEMA_STATISTICS (

```

```

schemaname      VARCHAR2,
estimate_percent NUMBER DEFAULT NULL,
degree          NUMBER  DEFAULT NULL,
internal_flag    NUMBER  DEFAULT NULL,
);

```

パラメータ

schemaname	分析するスキーマ。
estimate_percent	概算手法で使用する行の割合。NULL は、計算を実行することを意味します。有効範囲は [00,99] です。
degree	並列性の度合。NULL は、表デフォルト値を使用することを意味します。
internal_flag	値「NOBACKUP」を指定すると、GATHER_SCHEMA_STATS プロシージャで現在の統計のバックアップは作成されません。こうすると、GATHER_SCHEMA_STATS プロシージャの実行が短時間で完了します。「INTERNAL」は、DBMS_STATS ではなく ANALYZE コマンドが使用されることを意味します。

GATHER_INDEX_STATS プロシージャ

このプロシージャは、索引統計を収集します。実行結果は、ANALYZE INDEX [ownname.]indname [PARTITION partname] COMPUTE STATISTICS | ESTIMATE STATISTICS SAMPLE estimate_percent PERCENT を実行した場合と同じです。

並列処理では実行されません。

backup_flag の値が「BACKUP」である場合は、統計を収集する前に export_table_stats プロシージャが実行されます。エクスポートされたデータは、fnd_stattab に保存されます。backup_flag の値が「BACKUP」ではない場合は、export_table_stats は実行されません。

構文

```

FND_STATS.GATHER_INDEX_STATS (
    ownname      VARCHAR2,
    indname      VARCHAR2,
    percent      NUMBER DEFAULT NULL,
    partname     VARCHAR2 DEFAULT NULL,
    backup_flag  VARCHAR2 DEFAULT NULL
);

```

パラメータ

ownname	分析する索引のスキーマ。
indname	索引の名称。
percent	概算手法で使用する行の割合。NULL は、「計算を実行する」ことを意味します。有効範囲は [00,99] です。
partname	パーティション名。
backup_flag	値「BACKUP」は、統計を収集する前に export_table_stats プロシージャが実行されることを示します。デフォルト値は NULL です。

GATHER_TABLE_STATS プロシージャ

このプロシージャは、表統計と列統計（および索引統計）を収集します。このプロシージャは、できるだけ多くの作業を並列化しようとします。ユーザーが分析対象のテーブル上での選択権限を持つ場合に限り、処理が並列化されます。

backup_flag の値が「BACKUP」である場合は、統計を収集する前に export_table_stats プロシージャが実行されます。エクスポートされたデータは、fnd_stattab に保存されます。backup_flag の値が「BACKUP」ではない場合は、export_table_stats は実行されません。

構文

```
FND_STATS.GATHER_TABLE_STATS (
    errbuf          OUT  VARCHAR2,
    retcode         OUT  VARCHAR2,
    ownname         VARCHAR2,
    tabname         VARCHAR2,
    percent         NUMBER DEFAULT NULL,
    degree          NUMBER  DEFAULT NULL,
    partname        VARCHAR2 DEFAULT NULL,
    backup_flag     VARCHAR2 DEFAULT NULL,
    cascade         BOOLEAN DEFAULT TRUE,
    tmode           VARCHAR2 DEFAULT 'NORMAL'
);
```

```
FND_STATS.GATHER_TABLE_STATS (
    ownname         VARCHAR2,
    tabname         VARCHAR2,
    percent         NUMBER DEFAULT NULL,
    degree          NUMBER  DEFAULT NULL,
    partname        VARCHAR2 DEFAULT NULL,
    backup_flag     VARCHAR2 DEFAULT NULL,
    cascade         BOOLEAN DEFAULT TRUE,
```

```
tmode          VARCHAR2 DEFAULT 'NORMAL'
);
```

パラメータ

errbuf	コンカレント・プロセスとして実行する場合に必要です。
retcode	コンカレント・プロセスとして実行する場合に必要です。
ownname	表の所有者。
tabname	表の名称。
percent	概算手法で使用する行の割合。NULL は、「計算を実行する」ことを意味します。有効範囲は [00,99] です。
degree	並列性の度合。NULL は、表デフォルト値を使用することを意味します。
partname	パーティション名。
backup_flag	値「BACKUP」は、統計を収集する前に <code>export_table_stats</code> プロシージャが実行されることを示します。デフォルト値は NULL です。
cascade	この表の索引についての統計を収集します。索引統計の収集は、並列化されません。このオプションを使用した実行結果は、表の各索引に対して <code>gather_index_stats</code> プロシージャを実行した場合と同じです。

GATHER_COLUMN_STATS プロシージャ

このプロシージャは、列統計を収集します。プロシージャの 1 つのバージョンでは、SEED データ表の `fnd_histogram_cols` で指定された列の列統計を収集します。もう 1 つのプロシージャでは、指定された列の列統計を収集します。

構文

```
FND_STATS.GATHER_COLUMN_STATS (
    appl_id          NUMBER DEFAULT NULL,
    percent          NUMBER DEFAULT NULL,
    degree           NUMBER  DEFAULT NULL,
    backup_flag      VARCHAR2 DEFAULT NULL,
    Errors           OUT  Error_Out
);
```

```
FND_STATS.GATHER_COLUMN_STATS (
    ownname          VARCHAR2,
    tabname          VARCHAR2,
    colname          VARCHAR2,
    percent          NUMBER DEFAULT NULL,
    degree           NUMBER  DEFAULT NULL,
```

```
hsize          NUMBER    DEFAULT 254,  
backup_flag    VARCHAR2  DEFAULT NULL,  
partname       VARCHAR2  DEFAULT NULL  
);  
  
FND_STATS.GATHER_COLUMN_STATS (  
  errbuf       OUT  VARCHAR2,  
  retcode      OUT  VARCHAR2,  
  ownname      VARCHAR2,  
  tabname      VARCHAR2,  
  colname      VARCHAR2,  
  percent      NUMBER  DEFAULT NULL,  
  degree       NUMBER  DEFAULT NULL,  
  hsize        NUMBER  DEFAULT 254,  
  backup_flag  VARCHAR2  DEFAULT NULL,  
  partname     VARCHAR2  DEFAULT NULL  
);
```

パラメータ

errbuf	コンカレント・プロセスとして実行する場合に必要です。
retcode	コンカレント・プロセスとして実行する場合に必要です。
appl_id	アプリケーション ID。
ownname	表の所有者。
colname	列名。
tabname	表名。
percent	概算手法で使用する行の割合。NULL は、計算を実行することを意味します。有効範囲は [00,99] です。
degree	並列性の度合。NULL は、表デフォルト値を使用することを意味します。
hsize	ヒストグラム内のバケット数。
backup_flag	値「BACKUP」は、統計を収集する前に export_table_stats プロシージャが実行されることを示します。デフォルト値は NULL です。
partname	パーティション名。
errors	エラー出力。

GATHER_ALL_COLUMN_STATS プロシージャ

このプロシージャは、FND_HISTOGRAM_COLS 表に基いて、指定されたスキーマの列統計を収集します。

構文

```
FND_STATS.GATHER_ALL_COLUMN_STATS (  
    ownname          VARCHAR2,  
    percent          NUMBER DEFAULT NULL,  
    degree           NUMBER DEFAULT NULL  
);
```

```
FND_STATS.GATHER_ALL_COLUMN_STATS (  
    errbuf           OUT  VARCHAR2,  
    retcode          OUT  VARCHAR2,  
    ownname          VARCHAR2,  
    percent          NUMBER DEFAULT NULL,  
    degree           NUMBER DEFAULT NULL  
);
```

パラメータ

errbuf	コンカレント・プロセスとして実行する場合に必要です。
retcode	コンカレント・プロセスとして実行する場合に必要です。
ownname	表の所有者。ALL は、すべての Applications スキーマを意味します。
percent	概算手法で使用する行の割合。NULL は、計算を実行することを意味します。有効範囲は [00,99] です。
degree	並列性の度合。NULL は、表デフォルト値を使用することを意味します。

ANALYZE_ALL_COLUMNS プロシージャ

このプロシージャは、指定されたスキーマ内にあるすべての表の全インデックス列の列統計を分析します。

構文

```
FND_STATS.ANALYZE_ALL_COLUMNS (  
    ownname          VARCHAR2,  
    percent          NUMBER DEFAULT NULL,  
    hsize            NUMBER DEFAULT 254  
);
```



```

FND_STATS.ANALYZE_ALL_COLUMNS (
    errbuf          OUT  VARCHAR2,
    retcode         OUT  VARCHAR2,
    ownname         VARCHAR2,
    percent         NUMBER DEFAULT NULL,
    hsize           NUMBER DEFAULT 254
);

```

パラメータ

errbuf	コンカレント・プロセスとして実行する場合に必要です。
retcode	コンカレント・プロセスとして実行する場合に必要です。
ownname	表の所有者。ALL は、すべての Applications スキーマを意味します。
percent	概算手法で使用する行の割合。NULL は、計算を実行することを意味します。有効範囲は [00,99] です。
hsize	ヒストグラム内のバケット数。

LOAD_XCLUD_STATS プロシージャ

このプロシージャは、SEED データ表の FND_EXCLUDE_TABLE_STATS で指定されたデフォルト統計をロードします。このプロシージャには、2つのバージョンがあります。1つのバージョンでは、特定のスキーマのすべての表の統計をロードします。もう1つのバージョンでは、指定されたスキーマの指定された表の統計をロードします。

構文

```

FND_STATS.LOAD_XCLUD_STATS (
    schemaname      VARCHAR2
);

FND_STATS.LOAD_XCLUD_STATS (
    schemaname      VARCHAR2,
    tablename       VARCHAR2
);

```

パラメータ

schemaname	スキーマの名称。
tablename	表の名称。

FND_CTL

FND_CTL には、FND_SESS_CTL プロシージャが 1 つだけ格納されています。このプロシージャを使用すると、セッション・レベル・パラメータ（一部のパラメータに限る）を容易に操作することができます。このデータベース・プロシージャは、システム・プロファイル・オプション「初期化 SQL 文 - カスタム」から起動され、アプリケーション・ユーザー・セッションのセッション・レベル・パラメータを制御します。参照：「初期化 SQL 文 - カスタム・プロファイル・オプション」。

FND_SESS_CTL のパラメータ

OLPT_OPT_MODE	すべての OLTP セッション（非コンカレント・マネージャ・セッション）の最適化プログラム・モード。 有効な値は、ALL_ROWS、FIRST_ROWS、CHOOSE および RULE です。
CONC_OPT MODE	すべてのコンカレント・セッションの最適化プログラム・モード。 有効な値は、ALL_ROWS、FIRST_ROWS、CHOOSE および RULE です。
TRACE_OPT	SQL_TRACE のオプション。 有効な値は、TRUE および FALSE です。
TIMESTAT	TIMED_STATISTICS のオプション。TRUE または FALSE に設定します。
LOGMODE	表 FND_TRACE_LOG にセッションのログを記録するかどうかを指定します。 有効な値は、LOG または NULL です。
EVENT_STMT	その他のイベント設定またはセッション変更コマンド。有効な SQL 文だけを値として指定できます。 たとえば、バインド値を取得するには、イベント 10047 を次のように設定します。 'ALTER SESSION SET EVENTS =' ' ' 10046 TRACE NAME CONTEXT FOREVER, LEVEL 4' ' '

例

次の例では、FND_SESS_CTL およびシステム・プロファイル・オプション「初期化 SQL 文 - カスタム」を使用して、セッション・パラメータを制御します。

次の文は、OLTPセッションには値 `FIRST_ROWS` を、コンカレント・セッションには値 `ALL_ROWS` を指定します。トレース・オプションはオンに設定します。

```
BEGIN FND_CTL.FND_SESS_CTL('FIRST_ROWS',
    'ALL_ROWS', 'TRUE', 'TRUE', '', ''); END;
```

次の文の実行結果は上記の文と同じですが、イベント文が含まれています。

```
BEGIN FND_CTL.FND_SESS_CTL('FIRST_ROWS', 'ALL_ROWS',
    'TRUE', 'TRUE', '', 'ALTER SESSION SET EVENTS
    ='||''''||' 10046 TRACE NAME CONTEXT FOREVER,
    LEVEL 4 '||'''''); END;
```

次の文は、パラメータ `complex_view_merging` を設定します。

```
BEGIN FND_CTL.FND_SESS_CTL('','','','','',
    'ALTER SESSION SET "_complex_view_merging" =
    true'); END;
```

次の文は、ログの記録をオンに設定します。このオプションのログ表は、`FND_TRACE_LOG` です。

```
BEGIN FND_CTL.FND_SESS_CTL('','','','','LOG', '');
    END;
```

`FND_TRACE_LOG` は、LOG モードがオンに設定された場合に、LOG 情報を格納するトランザクション表です。このテーブルは定期的にページする必要があります。

ユーザー・プロファイル

ユーザー・プロファイル設定の概要

ユーザー・プロファイルとは、ユーザー・アプリケーションの表示形態や動作内容に影響を及ぼす一連の変更可能オプションのことです。システム管理者は、各ユーザー・プロファイル・オプションを希望する値に設定することにより、Oracle Applications の運用方法を制御できます。ユーザー・プロファイル・オプションは、「サイト」、「アプリケーション」、「職責」および「ユーザー」の4つのレベルで設定できます。

主な機能

会計帳簿

「職責」、「アプリケーション」、または「サイト」に会計帳簿を割り当てることにより、セキュリティ管理を充実させることができます。会計帳簿とは、共通の科目コード、カレンダー、および機能通貨を共有する Oracle Applications 内の会社またはグループ会社で使用される会計帳簿類の集合です。

関連項目

- 「Oracle Application Object Library のプロファイル・オプション」 (A-1)
- 「システム・プロファイル値ウィンドウ」 (9-6)
- 「プロファイル・オプションをパラメータまたはセグメントのデフォルト値として使用する場合」 (9-3)
- 「ユーザー・プロファイル・オプションの設定」 (9-2)
- 「ユーザー・プロファイル・オプションの例」 (9-4)
- 「ユーザー・プロファイル・オプション値レポート」

ユーザー・プロファイル・オプションの設定

システム管理者は「システム・プロファイル値」ウィンドウを使用して、ユーザーのプロファイル・オプションを設定します。ユーザー・プロファイル・オプションの値を変更した場合、その変更内容は、ユーザーが次にログ・オンしたとき、または職責を変更したときにただちに有効になります。参照:「[システム・プロファイル値ウィンドウ](#)」(9-6)。

ユーザー・プロファイルを設定するときには、システム管理者は Oracle Applications に対して、「ユーザー」、「職責」、「アプリケーション」、または「サイト」を記述した標準情報（プリンタなど）を提供します。ユーザー・プロファイル・オプションの値は各プロファイル・レベルで設定できます。

サイト	インストール・サイトのすべてのユーザーに適用されるオプション設定。
アプリケーション	アプリケーションに関係する職責を指定されたすべてのユーザーに適用されるオプション設定。
職責	指定された職責で現在サインオンしているすべてのユーザーに適用されるオプション設定。
ユーザー	アプリケーション・ユーザー名で識別される個々のユーザーに適用されるオプション設定。

各レベルで設定された値は、各ユーザーのプロファイル・オプションの実行時値として設定されます。オプションの実行時値は、そのオプションの最上位レベルの設定になります。

プロファイル・オプションが複数のレベルで設定できるときには、サイト・レベルの優先順位が最も低く、次にアプリケーション・レベル、その次が職責レベルで、最も高い優先順位を持つのはユーザー・レベルです。たとえばサイト・レベルで入力された値は、その他のレベルの値が入力されると上書きされます。ユーザー・レベルで入力された値が最も高い優先順位を持ち、その他の各レベルで入力された値はこの値で上書きされます。

たとえば、あるユーザーのプリンタ・オプションが、サイト・レベルおよび職責レベルでのみ設定されていると仮定します。そのユーザーがログオンすると、職責レベルの設定がここでは最も高いレベルなので、プリンタ・オプションの値は職責レベルで設定された値になります。

提案: システム管理者は、Oracle Applications のインストール後、サイト・レベルのオプション値を設定してから他の3つのレベルでプロファイル・オプションを設定してください。サイト・レベルで指定されたオプションは、同じオプションがその他のレベルで指定されるまでは、デフォルトとして有効です。

アプリケーション・ユーザーは「個別プロファイル値」ウィンドウを使用して、ユーザー自身の個別プロファイル・オプションをユーザー・レベルで設定できます。ユーザーには表示されないプロファイル・オプションもあります。また、表示されていてもエンド・ユーザーには更新できないプロファイル・オプションもあります。

関連項目

- [「Oracle Application Object Library のプロファイル・オプション」](#) (A-1)
- [「システム・プロファイル値ウィンドウ」](#) (9-6)
- [「プロファイル・オプションをパラメータまたはセグメントのデフォルト値として使用する場合」](#) (9-3)
- [「ユーザー・プロファイル・オプションの例」](#) (9-4)
- [「ユーザー・プロファイル・オプション値レポート」](#)
- [「ユーザー・プロファイル設定の概要」](#) (9-1)

プロファイル・オプションをパラメータまたはセグメントのデフォルト値として使用する場合

プロファイル・オプションの設定値を、コンカレント・プログラムのパラメータまたはフレックスフィールドのセグメントのデフォルト値として使用することができます。[「プロファイル・オプションをデフォルト値として入力するときのフォーム」](#) (9-3) には、デフォルト値となる設定値のプロファイル・オプションを入力するために使用する、フォームのリストが示されています。

プロファイル・オプションの設定値をデフォルト値として使用するには、フォームの「デフォルト・タイプ」フィールドにナビゲートして、プロファイルを選択します。次にプロファイル・オプションの内部名を「デフォルト値」フィールドに入力します。

プロファイル・オプションをデフォルト値として入力するときのフォーム

フォーム	ウィンドウ	フィールド
コンカレント・プログラム	パラメータ	「パラメータ詳細」 リージョン - デフォルト・タイプ / デフォルト値
要求セット	レポート・パラメータ	デフォルト・タイプ / デフォルト値
キー・フレックスフィールド・セグメント	セグメント	「検証情報」 リージョン - デフォルト・タイプ / デフォルト値
付加フレックスフィールド・セグメント	セグメント	「検証情報」 リージョン - デフォルト・タイプ / デフォルト値
表 28		

関連項目

- [「Oracle Application Object Library のプロファイル・オプション」](#) (A-1)

- 「システム・プロファイル値ウィンドウ」 (9-6)
- 「ユーザー・プロファイル・オプションの設定」 (9-2)
- 「ユーザー・プロファイル・オプションの例」 (9-4)
- 「ユーザー・プロファイル設定の概要」 (9-1)

ユーザー・プロファイル・オプションの例

例 #1

「購買」部門が最近プリンタを購入し、その部門のすべてのレポートをその新しいプリンタで印刷する場合。Oracle Payables の「プリンタ」プロファイル・オプションを変更して、新しく購入したプリンタを反映するのみで十分です。

提案: 例 #2 では、デフォルトのプロファイル・オプションの重要性が強調されています。Oracle Payables のアプリケーション・ユーザーまたは Oracle Payables に対応付けられた職責が、プリンタ・プロファイル・オプションに対して指定された値をすでに持っている場合、アプリケーション・レベルで設定した値はその値で上書きされます。ユーザー・プロファイル・オプションは、最初にサイト・レベルで設定し、必要に応じて次の階層のレベルで設定するようにしてください。ユーザー・プロファイル・オプションがあるレベルで設定されていないと、それより 1 つ低いレベルで設定されたユーザー・プロファイル・オプションがデフォルト設定として有効になります。

例 #2

さらにシステム管理者は GL 会計帳簿 ID のプロファイル・オプションを使用して、会計帳簿を職責、アプリケーション、またはサイト・レベルに割り当てることによって、Oracle Applications 内でセキュリティを管理できます。会計帳簿を職責に割り当てることによって、この職責でアクセスできる各フォームや機能を管理するのみでなく、特定の会計帳簿も管理できます。

会計帳簿の定義方法の詳細は、Oracle Applications 製品のリファレンス・ガイドを参照してください。

関連項目

- 「Oracle Application Object Library のプロファイル・オプション」 (A-1)
- 「システム・プロファイル値ウィンドウ」 (9-6)
- 「プロファイル・オプションをパラメータまたはセグメントのデフォルト値として使用する場合」 (9-3)
- 「ユーザー・プロファイル・オプションの設定」 (9-2)

- [「ユーザー・プロファイル設定の概要」](#) (9-1)

ユーザー・プロファイル・オプション値レポート

このレポートは、ユーザー・プロファイル・オプションの設定内容を示します。いくつかの職責またはユーザー、または異なるアプリケーションに対して異なるプロファイル・オプションの値を定義するときには、このレポートを使用してください。

レポート・パラメータ

プロファイル・オプション名

報告する値を持つプロファイル・オプション名を選択してください。プロファイル・オプション名を選択しないと、このレポートにはすべてのプロファイル・オプションが記載されます。

ユーザー名

報告するプロファイル・オプションの値を持つユーザーの名前を選択してください。

アプリケーション短縮名

報告するプロファイル・オプションの値を持つアプリケーションの名前を選択してください。

職責名

報告するプロファイル・オプションの値を持つ職責の名前を選択してください。

レポート見出し

レポートの見出しには、指定されたレポート・パラメータが表示され、そのレポートの内容に関する一般情報が示されます。

関連項目

- [「Oracle Application Object Library のプロファイル・オプション」](#) (A-1)
- [「システム・プロファイル値ウィンドウ」](#) (9-6)
- [「プロファイル・オプションをパラメータまたはセグメントのデフォルト値として使用する場合」](#) (9-3)
- [「ユーザー・プロファイル・オプションの設定」](#) (9-2)
- [「ユーザー・プロファイル設定の概要」](#) (9-1)

システム・プロフィール値ウィンドウ

このウィンドウを使用して、プロフィール・オプションの値を表示し、設定します。

プロフィール・オプションの表示と設定は、サイト、アプリケーション、職責、ユーザーの各レベルで行えます。この設定は、ユーザーがサインオンするか職責を変更するとただちに有効となります。参照：[「ユーザー・プロフィール設定の概要」](#) (9-1)。

プロフィール値ブロック

1つまたは複数のレベルでプロフィール・オプションの値を設定します。各値は、その左側に設定された値を無効にします。たとえばユーザー値を設定すると職責値が変更されます。この結果、アプリケーション値の設定も変わり、今度はサイト値の設定が変わります。

プロフィール・オプション値が無効になっている場合は、値のかわりにそのフィールドの LOV を参照します。

プロフィール

このフィールドにはプロフィール・オプションの名前が表示されます。

サイト

このフィールドにはインストレーション・サイトのすべてのユーザーの現在の値が設定されていれば、その値が表示されます。

アプリケーション

「プロフィール値の検索」ブロックで指定されたアプリケーションが所有している職責の下で作業を行っているすべてのユーザーの現在の値が設定されていれば、その値がこのフィールドに表示されます。

職責

このフィールドには、「プロフィール値の検索」ブロックで識別された職責の下で作業を行っているすべてのユーザーの現在の値が設定されていれば、その値が表示されます。

ユーザー

このフィールドには、「プロフィール値の検索」ブロックで指定されたアプリケーション・ユーザーのための現在の値が設定されていれば、その値が表示されます。

提案：アプリケーションのインストール後に要求されたオプションがあれば、それに対してサイト・レベルのデフォルト値を設定する必要があります。4つのレベルのいずれかで割当を行わなかったプロフィール・オプションは、デフォルト値がなく、フォームを使用したり、レポートを実行したり、コンカレント要求を実行したときにエラーの原因となります。

「システム・プロファイル値の検索」ブロック

プロファイル・オプション値を表示および設定する1つまたは複数のレベルを指定します。

表示

次の4つのレベルでインストールしたプロファイル・オプションに設定されている値を見ることができます。

- サイト：インストール・サイトのすべてのユーザーに影響します。
- アプリケーション：特定のアプリケーションが所有する職責の下で作業しているすべてのユーザーに影響します。
- 職責：特定の職責の下で作業しているすべてのユーザーに影響します。
- ユーザー：一意のアプリケーション・ユーザーに影響します。

Oracle Order Entry に対して、たとえば「OE」などの特定の文字列を持ったプロファイル・オプションの値をすべて検索できます。また、現在、値が設定されているプロファイル・オプションのみを表示することもできます。

サイト

インストール・サイトでプロファイル・オプションの値を表示するには「サイト」チェックボックスをチェックします。

アプリケーション

アプリケーションが所有する職責のプロファイル・オプション値を表示するには、アプリケーションを選択します。

職責

特定の職責のプロファイル・オプション値を表示するには職責を選択します。

ユーザー

特定のユーザーのプロファイル・オプション値を表示するには、アプリケーション・ユーザーを選択します。

プロファイル

値を表示しようとするプロファイル・オプションの名前を入力します。文字列とワイルドカード記号 (%) を使用して、プロファイル・オプションを検索できます。たとえば、「Conc%」と入力して「検索」ボタンを押すと、「Concurrent:」という接頭辞を持ったすべてのプロファイル・オプションを検索できます。

値のないプロファイル

すべてのプロファイル（現在設定値を持たないプロファイルを含む）を表示するかどうかを指定します。このチェックボックスを選択しないと、現在値が設定されているプロファイルのみが検索対象となります。

検索

指定した1つまたは複数のレベルで、すべてのプロファイル・オプションを表示するか検索対象のプロファイル・オプションを表示するには、「検索」ボタンを選択します。

関連項目

- [「Oracle Application Object Library のプロファイル・オプション」](#) (A-1)
- [「ユーザー・プロファイル・オプション値レポート」](#)
- [「ユーザー・プロファイル設定の概要」](#) (9-1)

プロセス・ナビゲーションの管理

プロセス・ナビゲータ処理の作成

プロセス・ナビゲータに表示されるプロセスの作成やカスタマイズには、Oracle Workflow Builder を使用する必要があります。プロセス・ナビゲータのプロセスを新規に作成する方法を次に示します。

プロセス・ナビゲータのプロセスの用語およびコンポーネントと、それらを定義している Oracle Workflow Builder の対応するコンポーネントを次の表に示します。

プロセス・ナビゲータの コンポーネント	摘要	Oracle Workflow Builder コンポーネントの管理
処理	プロセス・ナビゲータに表示されるダイアグラム。	処理アクティビティと処理ダイアグラム
処理摘要	表示された処理の摘要。	処理アクティビティ
ステップ	プロセス内のアイコン。ダブルクリックすると Oracle Applications フォームが直接表示されます。	通知アクティビティ
処理摘要	選択された処理ステップの摘要。	メッセージ
ステップに関連付けられたフォーム。	プロセス・ナビゲータ処理のステップをダブルクリックすると表示される Oracle Applications フォームです。	「フォーム」タイプのメッセージ属性

表 29

注意：次の手順では、Oracle Workflow Builder の大部分の機能について説明するわけではありません。プロセス・ナビゲータのプロセスを作成するための手順の説明です。Oracle Workflow Builder は、ワークフローのプロセスを設計するためのツールです。Workflow のプロセスは、承認プロセスに沿った文書の転送から Oracle Applications の設定まで多くの種類があります。参照：Oracle Workflow。

参照：「[プロセス・ナビゲータ処理の作成](#)」（10-2）

プロセス・ナビゲータ処理の作成

プロセス・ナビゲータの処理を新規に作成するには、まず、Oracle Workflow Builder の必要なコンポーネントを作成する必要があります。作成するコンポーネントでプロセス定義が構成されます。構成されたプロセス定義は、データベースまたはフラット・ファイルに保存されます。次に、プロセス・ナビゲータによってデータベースのプロセス定義が読み込まれ、プロセスとプロセス情報が表示され、関連する Oracle Applications フォームにアクセスできるようになります。

▶ プロセス・ナビゲータ処理の新規作成

1. Oracle Workflow Builder をオープンします。
2. 項目タイプを作成します。項目タイプとは、作成するプロセスと関連するすべてのコンポーネントを格納したりポジトリです。参照：項目を作成する場合：『Oracle Workflow ガイド』
3. 内部名が USER_NAME であるロール・タイプの品目属性を作成します。参照：項目またはアクティビティ属性を定義する場合：『Oracle Workflow ガイド』

注意：< 動詞 > フォーム・タイトル > の書式を使用して、メッセージの表示名を入力します。フォーム・タイトルにすでに動詞がある場合、表示名としてフォーム・タイトルを使用します。フォーム・タイトルに動詞が含まれていない場合、次の動詞のどれかを使用することを検討してください。

定義 (Define) / 割当 (Assign) / 実行 (Run) / ロード (Load) / 変換 (Convert) /
オープン (Open) / 設定 (Set) / 生成 (Generate) / 検討 (Review)

4. プロセス・ナビゲータの処理ステップで行われるタスクを記述するメッセージを作成します。参照：メッセージを作成する場合：『Oracle Workflow ガイド』

5. メッセージのフォーム・タイプを作成します。シードされた処理では、通常これらのメッセージ属性にオープン・フォームの内部名が割り当てられます。ただし、これは必須ではありません。参照: フォーム機能名の検索 [タグが必要]。関連項目: メッセージ属性を定義する場合: 『Oracle Workflow ガイド』
6. プロセス・ナビゲータの処理ステップで行われるタスクを記述するメッセージを作成します。参照: メッセージを作成する場合: 『Oracle Workflow ガイド』
7. プロセス・ナビゲータの処理を表すプロセス・アクティビティを作成します。参照: プロセス・アクティビティを作成する場合: 『Oracle Workflow ガイド』

注意: プロセスの表示名を入力します。この名前は、プロセス・ナビゲータのプロセス・リストに表示されます。プロセスの命名規則により、機能を表す名前の後に " プロセス " という語をつける必要があります。プロセスの説明を入力します。説明は、ユーザーがプロセス・ナビゲータのプロセスを選択した際に表示されます。Oracle Workflow Builder リリース 2.5 については、説明は、240 文字までに制限されています。

8. プロセス・ダイアグラムを作成します。プロセス・アクティビティを作成すると、それに関連したプロセス・ダイアグラムを作成できます。プロセス・ダイアグラムは、プロセスをプロセス・ナビゲータに表示する際に表示されます。参照: プロセスのダイアグラミング: 『Oracle Workflow ガイド』

注意: プロセス・ナビゲータのプロセス・ダイアグラムに含む通知アクティビティの「実行者タイプ」は、品目属性 USER_NAME に設定する必要があります。

9. 変更を保存します。データベースに作業結果を保存すると、実際には、変更されたカレント・データ・ストアにあるものすべてが保存されます。作業結果をフラット・ファイルに保存すると、実際には、カレント・データ・ストアにあるものすべてがファイルに保存されます。

注意: プロセス・ナビゲータのプロセスを新規に作成する場合、ワークフロー・プロセス定義のコピーをフラット・ファイルに保存してから、フラット・ファイルをソース管理システムでチェックすることにより、作業中のプロセス定義を保守することをお勧めします。その後、データベース内のプロセス定義を変更するには、フラット・ファイルを読み込んでからデータベースに直接保存します。データベースに保存されているプロセス定義をソース管理されたバージョンとして使用しないでください。データベースにアクセスできる人がプロセス定義を変更するおそれがあるためです。

10. プロセスへのアクセスを可能にします。

プロセスへのアクセスを可能にする

ナビゲータからプロセスにアクセスする前に、次の2つのステップを完了する必要があります。「フォーム機能」ウィンドウでプロセスに機能を新規に作成します。次に、作成した機能を「メニュー」ウィンドウの職責トップ・メニューに追加することにより、プロセスを職責に追加します。

▶ プロセスの機能の作成

1. システム管理者として「フォーム機能」ウィンドウにナビゲートします（「アプリケーション」>「機能」）。
2. 次の書式を使用して、プロセスの機能名を入力します。

`<app>_<processname>`

`<app>` は、アプリケーション短縮名です。`<processname>` は、プロセス・アクティビティを作成したときに入力した内部名です。

3. ユーザー機能名を入力します。ここで入力した名前はナビゲータに表示されます。
4. 機能タイプとして "PROCESS" と入力します。
5. タブ付リージョン「フォーム」において、次の書式を使用して「パラメータ」フィールドに値を入力します。

`<itemtype>:<processname>`

6. 作業内容を保存します。プロセス機能の作成には、他のフィールドは必要ありません。
参照：「[「フォーム機能」ウィンドウ](#)」（1-28）

▶ メニューへの機能の追加

ナビゲータのプロセスにユーザーがアクセスするためには、ユーザーの職責によって参照されるメニューにプロセスが追加されている必要があります。特定の職責によって参照されるメニューを調べるには、「職責」ウィンドウを使用します（「セキュリティ」>「職責」>「定義」）。

1. システム管理者として「メニュー」ウィンドウにナビゲートします（「アプリケーション」>「メニュー機能」）。
2. 「検索」ウィンドウを使用して、目的のメニューにアクセスします。
3. 新規の行で、値リストを使用して、「機能」フィールドでプロセスに対して作成した機能を選択します。他のフィールドには一切入力しないでください。「順序」フィールド

は、自動的に入力されます。「ナビゲータ・プロンプト」フィールドと「サブメニュー」フィールドは未入力であればなりません。

4. 作業内容を保存します。

参照: 「[メニュー](#)」ウィンドウ (1-30)

データベースからのシードされたプロセスへのアクセス

シードされたプロセスにデータベースからアクセスするには、初めに Oracle Applications リリース 11.0.1（製品版）をインストールし、Oracle Workflow Builder リリース 2.0.3 以上をクライアント PC にインストールする必要があります。

▶ シード・プロセスにアクセスする手順は、次のとおりです。

1. クライアントから Oracle Workflow Builder を実行します。
2. 「ファイル」メニューから「オープン」を選択します。
3. 「データベース」を選択します。
 - 「ユーザー」には、データベースの「FNDNAM」を入力します。
 - 「パスワード」には、データベースの「FNDNAM」パスワードを入力します。
 - 「接続」には、データベースの別名を指定します。この別名は、クライアントの次のディレクトリにある tnsnames.ora ファイルに入力される必要があります。

ローカル・ドライブ (つまり、"C") の orant¥network¥admin

注意: Windows 95 を使用している場合、前述のディレクトリ構造の「orant」は「orawin」になります。

4. 「項目タイプの表示」ウィンドウで、表示するシードされたプロセスに関連する項目タイプを選択します。1 つ以上の項目タイプを選択する場合には、コントロール・キーを押したまま項目タイプを選択します。「表示」を選択してから、「OK」を選択します。

▶ フォーム機能名の検索

1. Oracle Applications にログオンしてから、目的のフォームにナビゲートします。
2. 「ヘルプ」メニューから「Oracle Applications について」を選択します。「フォーム情報」まで下にスクロールしてから、フォーム名を記録します。
3. 「Implementation System Administration」の職責で Oracle Applications にログインしてから、「アプリケーション」>「フォーム」にナビゲートします。「フォーム」ウィンドウ

で、記録しておいたフォーム名に対する問合せを「フォーム」フィールドに対して実行します。

4. 問合せが完了したら、「ユーザー・フォーム名」フィールドの値を記録します。
5. 「フォーム」ウィンドウをクローズしてから、「アプリケーション」>「機能」にナビゲートします。「機能」ウィンドウで、記録しておいた「ユーザー・フォーム名」の値に対する問合せを「フォーム」フィールドに対して実行します。
6. 「機能」フィールドに戻される値は、プロセス・ナビゲータの処理ステップをフォームに関連付ける必要があるフォームの機能名です。

文書連番とは

文書連番とは、Oracle Applications 製品が生成する文書に対する一意の採番です。Oracle Applications を使用しながら取引を開始するには、フォームを介してデータを入力し、文書（請求書など）を生成します。文書連番は、取引を作成したアプリケーション（たとえば Oracle Receivables）および生成された元の文書（たとえば請求書番号 1234）を識別する監査証跡を生成します。

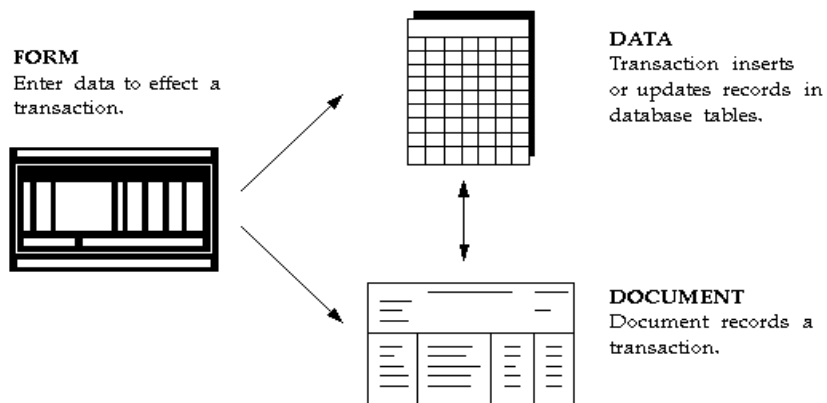
文書連番は、完全性を確保するためのものです。たとえば、文書連番を使用すると、失敗した取引も含めてすべての取引を勘定に入れることができます。

文書連番は監査証跡を提供することもできます。たとえば、文書連番は、総勘定元帳から補助元帳、および最初に科目残高に影響を及ぼした文書（たとえば請求書番号 1234）まで、監査証跡を提供できます。

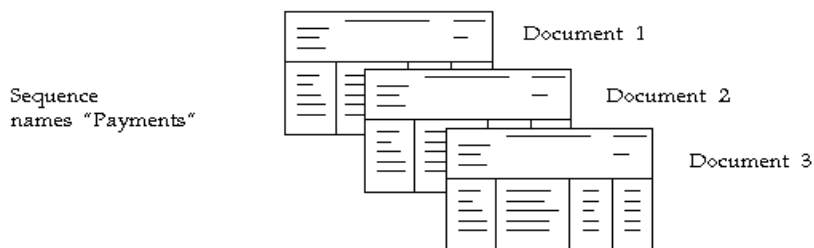
文書連番は監査データを生成するので、文書が削除されても、その文書の監査レコードは残ります。

図 26

Transactions generate documents. For example, recording payments generates payment documents.



A sequence defines how documents are numbered; what the first number in the sequence is, and whether numbers are generated automatically or entered manually.



関連項目

- 「[文書カテゴリの定義](#)」 (11-5)
- 「[文書採番と文書入力](#)」 (11-7)
- 「[文書連番の割当](#)」 (11-6)
- 「[文書連番の定義](#)」 (11-3)

文書連番の定義

連番を定義するには、連番名およびその連番を "所有" するアプリケーションを選択します。

- 連番は、それを所有するアプリケーションに属するデータベース表に格納されている文書に採番します。
- 連番用の監査レコードは、アプリケーションの監査表に格納されます。この表の名前は、アプリケーション短縮名 `_DOC_SEQUENCE_AUDIT` です。たとえば、Oracle Payables が所有する連番用の監査表は、`AP_DOC_SEQUENCE_AUDIT` です。

注意：データベース管理者は、連番を使用する職責（連番を使用してフォームにアクセスする職責）に関連付けられているすべての Oracle ユーザー名について、アプリケーションの監査表に対するアクセス権限を付与する必要があります。

連番が使用可能になる期間の開始日および終了日を設定できます。開始日のデフォルトは現在の日付です。終了日のデフォルトは無期限であるため、その連番定義は満了しません。

自動的に文書に連番を採番するのか、それともユーザーが手動で番号を入力するのかを選択できます。

自動採番、無欠番採番、手動採番

自動採番では、各文書の生成時にそれぞれに一意の番号が割り当てられます。自動採番では、作成の日時の順に番号が付けられます。

無欠番採番の場合も各文書について自動的に一意の番号が生成されますが、番号の割当ての前に、文書が正常に生成されたかどうかを確認する点が異なります。したがって、無欠番採番では、文書作成の不完全または失敗が原因で連番に欠番を生じることはありません。

注意：無欠番採番はシステムのパフォーマンスに影響を及ぼす恐れがあるので、無欠番採番を使用するのは、それが必須である場合のみにしてください。

手動採番では、ユーザーが、各文書の生成の前にそれぞれに一意の番号を割り当てる必要があります。手動採番の場合は、数値の順序および完全性は要求されません。ユーザーは、連番値を入力するときに、番号を飛ばしたり省略したりできます。

自動採番 - 初期値およびメッセージ表示

連番を定義して文書の自動採番を行う手順は、次のとおりです。

- 連番の初期値を入力します。デフォルトは 1 です。

- 文書の生成時に、連番名および連番値（文書番号）をユーザーに知らせるメッセージを表示するかどうかを選択します。

次に、連番定義の例を2つ示します。1つは自動採番で、もう1つは手動採番です。

例 - 自動連番と手動連番

「文書連番」フォームのフィールド	例 1 自動採番による順序	例 2 手動採番による順序
（順序）名称	自動支払	修正
（所有）アプリケーション	ORACLE PAYABLES - 連番により、Oracle Payables データベース表に格納済文書に採番できます。	ORACLE RECEIVABLES - 連番により、Oracle Receivables データベース表に格納されている文書に採番できます。
有効 - 自	現在日 & 時間 （デフォルト値）	OCT-01-94 ユーザーは、1994 年 10 月 1 日まで連番 " 修正 " を使用不可と定義しています。
有効 - 至	フィールドはブランクにします。連番は永久に有効となります。	DEC-31-94 ユーザーは、連番「修正」を 1994 年 12 月 31 日より後には有効とならないように定義しています。
（採番）タイプ	自動 - 一意の番号が、自動的に連番で生成されます。無欠番採番 - 文書作成が失敗した場合のロールバックのために、番号の省略や欠落は生じません。	手動 - 取引が完了し、文書が生成される前に、ユーザーが一意の番号を入力する必要があります。ユーザーは、番号を飛ばすあるいは省略することができません。
初期値	1 （デフォルト値）ユーザーは、独自の初期値（たとえば 5700 など）を入力できます。	ただし、採番タイプが「手動」の場合は、使用できません。
メッセージ	YES - 自動採番される文書が作成される時に、連番名および連番値（文書番号）を示すメッセージが表示されます。	採番タイプが「手動」の場合は、使用できません。
表 30		

関連項目

- 「文書カテゴリの定義」 (11-5)
- 「文書連番とは」 (11-1)

文書カテゴリの定義

文書カテゴリは、文書を論理グループに編成します。

- 文書カテゴリ（文書タイプともいう）とは、連番によりどの文書に番号を割り当てるかを定義するために使用するルール of 1 つです。
- 各カテゴリにそれぞれ異なる連番を割り当てることにより、各文書カテゴリごとに別々に採番することができます。

文書カテゴリは、ユーザーが入力した取引により発生した文書を格納するデータベース表を識別します。

- カテゴリに連番を割り当てると、その連番により、特定の表に格納されている文書に番号が付けられます。

カテゴリを使用して、文書をより厳密に類別します。たとえば、売掛 / 未収金請求書は、次のようにいくつかのカテゴリに細分できます。

- チャージバック
- 前受 / 預り金
- 保証金
- デビット・メモ
- クレジット・メモ
- 売上請求書
- 顧客サービス請求書

同様に、買掛請求書または購入請求書も、次のようにいくつかのカテゴリに細分できます。

- 標準
- 経費精算書
- 前渡金
- 利息
- クレジット・メモ
- デビット・メモ

関連項目

- [「文書連番とは」](#) (11-1)
- [「文書連番の割当」](#) (11-6)
- [「文書連番の定義」](#) (11-3)

文書連番の割当

文書につける連番を割り当てる前に、どの文書に番号をつけるのかを定義する必要があります。

連番と割当

連番の定義は一連の文書への連番割当とは異なります。

- 連番の定義は、文書の番号を自動的に生成するのか、ユーザーが手動で入力するのかを決定します。
- 連番の割当、つまり連番が割り当てられる文書は、「連番割当」フォームで定義します。

割り当てられた連番により採番するための文書の定義

特定の連番名に割り当てるための特定の文書を定義する 4 つのルールを組合せを指定します。

そうすると、各文書定義に、それぞれ異なる（採番用）連番を割り当てることができます。

次の 4 つのルールの組合せは、選択した連番によりどの文書に番号を割り当てるのかを定義します。

アプリケーション 採番する文書を生成するアプリケーションを選択します。

たとえば、売上請求書に採番するのであれば、Oracle Receivables を選択します。

カテゴリ

文書の論理サブセットを識別する文書カテゴリを選択します。

たとえば、Oracle Receivables ですべての請求書に採番する必要がない場合には、売上請求書のカテゴリのみに採番することを選択できます。

カテゴリでは、Oracle Application を使用して、入力した取引（生成された文書）を格納する表が識別されます。

文書を定義するために選択できるカテゴリの値は、選択するアプリケーションによって異なります。

会計帳簿

採番する文書の影響を受ける業務用の勘定体系を選択します。このルールは、オプションとして、文書フレックスフィールドを介して使用可能にすることができます。

方法

文書を入力する方法（自動または手動）を選択します。このルールは、オプションとして、文書フレックスフィールドを介して使用可能にすることができます。

外部プログラムなどのコンカレント・プロセスが、Oracle Application に取引データを入力するように設定されている場合は、自動を選択します。

フォームを使用して手動でアプリケーションに文書を入力する場合は、手動を選択します。

文書定義への連番の割当

個々の一意の文書定義について、アクティブにできる連番割当はそれぞれ1つのみです。文書定義はアプリケーション、カテゴリ、そしてオプションの文書フレックスフィールド・セグメントの会計帳簿および方法で構成されます。

注意：文書定義に連番を割り当てるときには、個々のアクティブな連番はそれぞれ1つのアプリケーションとカテゴリの一意の組合せ（つまりアプリケーション表）にしか割り当てることができません。

アクティブな割当およびアクティブな連番

アクティブな連番割当には転記日の終了日はありません。つまり、このような割当の終了日が現在日より前になることはありません。

- アクティブな連番割当では、終了日が存在しないか、または終了日が現在日より前の日付ではありません。
- 連番割当およびその有効期間は、「連番割当」フォームで定義します。

連番定義もアクティブであることが必要です。つまり、連番定義の終了日（その割当の終了日に対置するものとして）は、現在日より前にはできません。

- 連番定義およびその有効期間は、「文書連番」フォームで定義します。

文書連番を定義するときには、連番の名前を指定します。さらに、次のようにして、その連番により各文書に採番する方法を定義します。

- 番号を順番に自動生成するか、それともユーザーが手動で入力するのかを選択します。
- 連番内に初期値または最初の番号を入力します。

関連項目

- [「文書カテゴリの定義」](#) (11-5)
- [「文書採番と文書入力」](#) (11-7)
- [「文書連番とは」](#) (11-1)
- [「文書連番の定義」](#) (11-3)

文書採番と文書入力

特定の連番で使用する文書採番のタイプ（自動または手動）と、文書入力の方法（同じく自動または手動）とを混同しないようにしてください。

関連項目

- [「文書カテゴリの定義」](#) (11-5)
- [「文書連番とは」](#) (11-1)
- [「文書連番の割当」](#) (11-6)
- [「文書連番の定義」](#) (11-3)

文書連番ウィンドウ

新規の文書連番に名前を付け、その文書連番が各文書に番号を付ける方法を定義します。

文書連番は、Oracle Applications 製品が生成した文書（たとえば、Oracle Receivables で作成した請求書）に固有の番号を付けます。「連番割当」ウィンドウを使用して、定義したルールに合致する文書のみに番号が付けられるように連番を割り当てます。

文書連番によって、ユーザーが作成するすべての文書が必ず認識されるようにできます。参照：[「連番割当」ウィンドウ](#) (11-11)。

文書連番ブロック

文書連番の名前、番号付けスキームのタイプ、有効期日、初期値を定義します。

名前

一度入力した連番名は変更できません。

アプリケーション

連番に関連するアプリケーションは、一度選択すると変更できません。

連番用の監査レコードは、アプリケーションの監査表に格納されます。この表の名前は、アプリケーション短縮名 `_DOC_SEQUENCE_AUDIT` です。たとえば、Oracle Payables が所有する連番用の監査表は、`AP_DOC_SEQUENCE_AUDIT` です。

有効: 自 / 有効: 至

文書連番が有効となる日付と無効となる日付を入力します。「開始日」フィールドにはデフォルトで自動的に当日の日付が入ります。連番を定義した後は、開始日を変更できません。「終了日」フィールドに何も入力しないと、文書連番は永久に有効となります。終了日を指定して連番を定義した後は、終了日は変更できません。終了日が定義されておらず、連

番のアクティブな割当もないのであれば、終了日に当日を入力するとその連番を使用禁止にできます。一度使用禁止にした連番は、再度アクティブにすることはできません。

タイプ

文書に番号を付ける連番タイプは、一度定義すると変更できません。

自動 作成される個々の文書には、作成の日付と時刻に従って順番に固有の番号が付けられます。

手動 手動採番では、ユーザーが、各文書の生成の前にそれぞれの番号を割り当てる必要があります。

番号には一意の値を入力します。ただし、数字の順序や欠番の有無は関係ありません。

注意:「ユーザー対応自動」タイプは現在サポートされていません。今後 Oracle Applications のバージョンで考慮される予定です。

警告:「差異なし」タイプは、特定のローカライゼーションのコンテキストでのみ有効です。このタイプはシステムのパフォーマンスに影響を及ぼす恐れがあるので、選択する前に海外サポート窓口にご相談ください。

メッセージ

各文書でユーザーに対して連番名と値（番号）を示すメッセージ（画面の下部のメッセージ行内）を表示するには、「メッセージ」チェックボックスをチェックします。

このチェックボックスは、自動採番による連番にのみ適用されます。メッセージは表示されるフォームに示されるのみで、要求のログ・ファイルには記録されません。

連番を定義した後では、メッセージの選択は変更できません。

初期値

連番の中の最初の文書の値を入力します。このフィールドが適用されるのは、自動採番または連続採番の文書連番の場合のみです。文書連番の最大値は 1.0e+27 です。

このフィールドに何も入力しないと、最初の文書には「1」の値が自動的に割り当てられます。

連番を定義した後では、初期値は変更できません。

関連項目

- 「[文書カテゴリ](#)」ウィンドウ（11-10）
- 「[連番割当](#)」ウィンドウ（11-11）

- 「文書連番とは」(11-1)

「文書カテゴリ」ウィンドウ

文書をいくつかの論理グループに分割して、それぞれに異なった連番を割り当てて番号を付けるために、文書にカテゴリを定義します。

文書連番は、連番が割り当てられる個々の文書に一意の番号を付けます。

- 「連番割当」フォームを使用して、定義したルールに合致する文書にのみ番号を付ける連番を割り当てます。
- いずれかのフォームにタイトルとして使用されている文書のカテゴリ、またはタイプは、連番がどの文書に番号を割り当ててくるかを定義するルールの1つです。

各カテゴリは、ユーザーが生成したトランザクションによって作成された文書を格納するための表を識別します。

- 連番をカテゴリに割り当てると、表に格納されている文書にその文書連番が番号を付けます。

文書カテゴリ・ブロック

文書カテゴリに名前を付け、そのカテゴリに表に関連付けます。

このブロックを入力すると、Oracle は自動的に既存の文書カテゴリを問合せます。

アプリケーション

一度カテゴリを定義すると、アプリケーションの選択を変更することはできません。カテゴリに割り当てられるのは、選択したアプリケーションに属する表のみです。

コード

カテゴリ・コードはアプリケーションの中で固有である必要があります。カテゴリの定義を行った後は、コードを変更できません。

名称

名前は必要に応じて変更できます。たとえば、カテゴリ名がすでに定義されていても、それを分かりやすい値に変更できます。

摘要

摘要は、必要に応じて変更できます。たとえば、カテゴリの摘要がすでに定義されていても、それを分かりやすい値に変更できます。

表名

カテゴリ別に識別する文書の保管用表名を選択します。

- 順次採番機能は、レポートの完全性をチェックしたり新たにレポートを生成するときに、表中のそのカテゴリの文書を検索します。
- 選択できるのは、そのカテゴリに関連するアプリケーションに属する表のみです。
- カテゴリの定義を行った後では、表の選択を変更できません。

関連項目

- 「[連番割当](#) ウィンドウ」 (11-11)
- 「[文書連番とは](#)」 (11-1)
- 「[文書連番ウィンドウ](#)」 (11-8)

「連番割当」ウィンドウ

文書連番を付ける文書を定義し、この文書連番定義に割り当てます。

文書連番は、Oracle Applications 製品が生成した文書（たとえば、Oracle Receivables で作成した請求書）に付けられます。

文書は、それを作成するときに使用したアプリケーションおよびカテゴリ（文書が格納される表）別に定義されます。文書を定義するためのオプション・ルール（会計帳簿と文書入力方法）を使用可能にすると、追加のフィールドが表示されます。

文書の定義を入力し、それに順番を付けるほかに、必要に応じて割当ての有効期間を入力できます。

前提条件

- 「文書連番」ウィンドウを使用して文書連番を定義します。参照：[「文書連番ウィンドウ」](#) (11-8)。

連番割当ブロック

文書を作成したアプリケーションと文書のカテゴリ（文書が格納される表）によって文書を指定します。文書の定義には、文書が影響を与える会計帳簿および文書を入力するための方法を含めることができます。

文書定義の入力が終了したら、文書に割り当てる連番を選択し、必要に応じて割当ての有効期間を入力します。

アプリケーション、カテゴリ、会計帳簿および入力方法の固有の組合せ1つ1つに割り当てることのできるアクティブな連番は1つのみです。会計帳簿および入力方法の2つの基準は任意で、文書フレックスフィールドに設定されます。

ただし、アプリケーションとカテゴリが同じであれば、1つの順序、1つの番号付けスキーム、初期値をアプリケーション、カテゴリ、会計帳簿および入力方法の複数の組合せに割り当てることもできます。

アプリケーション 採番する文書を生成するアプリケーションを選択します。

たとえば、売上請求書に採番する場合は、Oracle Receivables を選択します。

カテゴリ 文書の論理サブセットを識別する文書カテゴリを選択します。

たとえば、Oracle Receivables ですべての請求書に採番する必要がない場合には、売上請求書のカテゴリのみに採番することを選択できます。

割当リージョン

割当ての有効期日は連番の開始日と終了日の間の日付でなくてはならないので、使用可能な連番のリストは割当てに指定された開始日と終了日によって決まります。

開始日 / 終了日

文書の定義への連番割当てが有効となる日付と無効となる日付を入力します。「開始日」フィールドは、自動的に当日の日付がデフォルト値として入りますが、連番割当てを定義した後は開始日の変更はできません。

「終了日」フィールドに何も入力しないと、連番割当ては永久に有効となります。一度終了日を入力し、連番割当てを定義すると、その終了日は変更できなくなります。

終了日が定義されておらず、連番のアクティブな割当てもないのであれば、当日を終了日として入力するとその連番割当てを使用禁止にできます。一度使用禁止にした連番割当ては、再度アクティブにすることはできません。

連番

文書定義に割り当てる連番を選択します。連番のアプリケーションと文書のアプリケーションは同一でなければなりません。

一度連番割当てを定義すると、その連番名は変更できなくなります。

連番割当てを使用禁止にしてその文書定義（文書フレックスフィールド組合せ）に新規の連番を割り当てるには、まず終了日を入力して現行の連番割当てを使用禁止にし、次に新規の割当てのための新規のレコード（行）を作成します。

文書フレックスフィールド

文書フレックスフィールドにはセグメントがないことも、1つか2つのセグメントで構成されていることもあります。

会計帳簿	採番する文書の影響を受ける業務の勘定体系を選択します。
方法	<p>文書を入力する方法を「自動」または「手動」から選択します。</p> <p>コンカレント・プロセス（たとえば、外部プログラム）により取引データを Oracle Application に入れ、文書を作成する場合は、「自動」を選択します。</p> <p>フォームを使用して手動でアプリケーションに文書を入力する場合は、「手動」を選択します。</p>

一度定義した文書フレックスフィールド定義は変更できません。文書フレックスフィールドに追加のセグメントを定義することはできません。

注意：この付加フレックスフィールドを使用可能にするには、「付加フレックスフィールド・セグメント」ウィンドウを使用します。アプリケーション「Application Object Library」とタイトルの「文書フレックスフィールド」を選択します。必ずフレックスフィールドを解除してください。次に「セグメント」ウィンドウに移動してセグメントを使用可能にします。フレックスフィールドを設定したらそのフレックスフィールドを確認し、新規の定義を保存してコンパイルします。

参照：『Oracle Applications フレックスフィールド・ガイド』の「付加フレックスフィールド体系の定義」。

関連項目

- 「[文書カテゴリ ウィンドウ](#)」 (11-10)
- 「[文書連番とは](#)」 (11-1)
- 「[文書連番ウィンドウ](#)」 (11-8)

Oracle Application Object Library の プロファイル・オプション

Oracle Application Object Library のプロファイル・オプション

この項では、Oracle Application Object Library の各プロファイル・オプションのリストを示します。これらのオプションは、Oracle Application のすべての製品で使用できます。ここでは、各プロファイル・オプションごとに、そのプロファイルの設定が Oracle Application Object Library によってどのように使用されるかを簡単に説明しています。

勘定科目ジェネレータ：デバッグ・モード

このプロファイル・オプションを設定すると、勘定科目ジェネレータ機能の Oracle Workflow プロセス・モードが制御されます。このプロファイル・オプションは、パフォーマンス向上のため、通常は「NO」に設定します。勘定科目ジェネレータの導入状態をテストしていて、Oracle ワークフロー・モニターを使用して結果を確認している場合は、このプロファイル・オプションを「YES」に設定してください。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は、ACCOUNT_GENERATOR:DEBUG_MODE です。

アプリケーション Web エージェント

APPS スキーマの WebServer DAD のベース URL を指定します。このプロファイル・オプションは、インストール処理時に、システム管理者が設定します。

独自の URL を入力するには、次の構文を使用します。

```
http://<WebServer.Machine_Name>/<DAD_name>/
```

ユーザーは、このプロファイル・オプションを参照できますが、更新することはできません。このプロファイル・オプションは、全レベルで参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は、APPS_WEB_AGENT です。

アプリケーション・ヘルプ Web エージェント

アプリケーション・ヘルプ Web エージェントはオプションであり、アプリケーション・ヘルプ・システムの WebServer DAD のベース URL がアプリケーション Web エージェントの場合と異なる場合は、そのベース URL を表示します。システム管理者は、このプロファイル・オプションをインストール処理時に設定します。

構文は、アプリケーション Web エージェントの場合と同じです。

ユーザーは、このプロファイル・オプションを参照できますが、更新はできません。

このプロファイル・オプションは、全レベルで参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は、HELP_WEB_AGENT です。

アプリケーション Web 認証サーバー

Oracle Self-Service Web Applications の認証に使用する Web サーバー。

ユーザーは、このプロファイル・オプションを参照できますが、更新することはできません。

このプロファイル・オプションは、サイト・レベルで参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	NO	NO
職責	NO	NO
ユーザー	NO	NO

このプロファイル・オプションの内部名は、WEB_AUTHENTICATION_SERVER です。

添付ファイル・ディレクトリ

ファイル・タイプ添付データが格納されているディレクトリ。システム管理者は、このプロファイル・オプションをインストール処理時に設定します。

ユーザーは、このプロファイル・オプションを参照できますが、更新することはできません。

このプロファイル・オプションは、全レベルで参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は、ATTACHMENT_FILE_DIRECTORY です。

自動監査：有効化

監査証跡のオンまたはオフ（YES または NO）を切り替えることができます。デフォルトの設定は NO（オフ）です。

フォームにデータを入力するか、またはフォーム内のデータを更新すると、表示し使用するフォームの基礎データベース表が変更されます。

監査証跡では、データベース表内のどの行がいつ更新され、どのユーザーがフォームを使用してログインしたかが追跡されます。

- いくつかの更新を追跡して、データベース表の変更を文書化するための監査データの証跡を確立します。
- 監査証跡は、Oracle Application Object Library を使用する開発者がフォーム単位で利用できる機能です。
- 監査証跡をサポートするすべてのフォームを監査セットといいます。
- すべてのフォームが、監査証跡をサポートできるわけではありません。
- 特定のフォームについて監査証跡を使用可能または使用不可にするには、Oracle Application Object Library のアプリケーション開発者の職責にアクセスする必要があります。

ユーザーは、このプロファイル・オプションを参照することも更新することもできません。

このプロファイル・オプションは、サイト・レベルおよびアプリケーション・レベルで参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	NO	NO
ユーザー	NO	NO

このプロファイル・オプションの内部名は AUDITTRAIL:ACTIVATE です。

BIS/AOL: デバッグ・ログ・ディレクトリ

BIS デバッグ・ログ・ファイルのディレクトリ。

ユーザーは、このプロファイル・オプションを参照および変更できます。

システム管理者は、このプロファイル・オプションをサイト・レベルで参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	NO	NO
職責	NO	NO
ユーザー	NO	NO

このプロファイル・オプションの内部名は、BIS_DEBUG_LOG_DIRECTORY です。

コンカレント : 有効要求限度

同時に実行できる要求の数を、ユーザーごとに、またはサイトのすべてのユーザーについて制限できます。限度を指定しなかった場合、制限はありません。

ユーザーは、このプロファイル・オプションを参照することも更新することもできません。

このプロファイル・オプションは、4つのどのレベルでも参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	NO
職責	YES	NO
ユーザー	YES	YES

このプロファイル・オプションの内部名は、CONC_REQUEST_LIMIT です。

コンカレント : URL の添付

このオプションを「YES」に設定すると、要求完了通知に URL が添付されます。ユーザーが要求を発行し、「完了オプションの定義」リージョンに通知対象者を指定すると、要求の完了時に、指定の全員へ通知が送信されます。このプロファイル・オプションを YES に設定した場合、通知に URL が付加され、それによって通知対象者は要求の結果をオンラインで表示できます。

このプロファイル・オプションを更新できるのは、システム管理者のみです。

ユーザーは、このプロファイル・オプションを参照できますが、更新することはできません。

このプロファイル・オプションは、どのレベルでも参照できますが、更新できるのは「サイトレベルのみです。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	NO
職責	YES	NO
ユーザー	YES	NO

このプロファイル・オプションの内部名は、CONC_ATTACH_URL です。

コンカレント : 衝突ドメイン

データに衝突ドメインを指定します。衝突ドメインでは、2つの非互換プログラムの同時実行ができないデータが識別されます。

ユーザーは、このプロファイル・オプションを参照できますが、更新することはできません。

このプロファイル・オプションは、4つのどのレベルでも参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は CONC_CD_ID です。

コンカレント : 要求の統計の収集

実行時のコンカレント・プロセスについて統計を収集するには、このプロファイル・オプションを「YES」に設定します。

統計結果を確認するには、「コンカレント・マネージャや要求のデータ等の削除」プログラムを実行して未加工データを処理し、計算済みの統計結果を FND_CONC_STAT_SUMMARY 表に書き込む必要があります。その後、SQL*PLUS を使用してこの表からデータを取り出すか、または「要求」ウィンドウから「解析」ウィンドウを使用してレポート単位でデータを取り出すことができます。

ユーザーは、このプロファイル・オプションを参照することも更新することもできません。

このプロファイル・オプションは、どのレベルでも参照できますが、更新できるのは「サイト」レベルのみです。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	NO
職責	YES	NO
ユーザー	YES	NO

このプロファイル・オプションの内部名は、CONC_REQUEST_STAT です。

コンカレント : デバッグ・フラグ

オラクル社のサポート担当者は、翻訳マネージャをデバッグするためにこのプロファイル・オプションにアクセスする場合があります。デバッグ以外の場合は、このプロファイル・オプションは NULL に設定してください。

ユーザーは、このプロファイル・オプションを参照することも更新することもできません。

このプロファイル・オプションは、4 つのどのレベルでも参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は CONC_DEBUG です。

コンカレント : コピー用ディレクトリ

出力ファイルの任意選択コピー用ディレクトリ。

ユーザーは、このプロファイル・オプションを参照および更新できます。

このプロファイル・オプションは、4 つのどのレベルでも参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES

職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は CONC_OUTPUT_COPY_PATH です。

コンカレント：要求の保留

コンカレント要求を発行する場合、それらの要求を自動的に保留状態にできます。

デフォルト値は「NO」です。コンカレント・マネージャは、個々の要求に指定された優先順位および開始時刻に従って要求を実行します。

この値を変更しても、すでに発行済みの要求には影響はありません。

「YES」は、コンカレント要求およびレポートを自動的に保留状態にすることを意味します。要求の保留状態を解除する手順は次のとおりです。

- 「要求」ウィンドウにナビゲートして要求を1つ選択します。
- 「要求管理」タブ付きリージョンを選択します。
- 「保留」チェックボックスをオフにします。

ユーザーは、このプロファイル・オプションを参照および更新できます。

このプロファイル・オプションは、4つのどのレベルでも参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は、CONC_HOLD です。

コンカレント：複数タイム・ゾーン

「YES」を選択すると、要求発行で使用する「計画開始日」のデフォルト値は 'Sysdate-1' に設定されます。Sysdate-1 に設定すると、クライアント・セッションを実行している時間ゾーンとは無関係に、発行された要求が即時スケジューリングされます。クライアントのセッションをコンカレント・マネージャのセッションとは異なる時間ゾーンで実行する場合は、このプロファイル・オプションを使用してください。

ユーザーは、このプロファイル・オプションを参照することも更新することもできません。

このプロファイル・オプションは、4つのどのレベルでも参照でき、「サイト」レベルで更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	NO
職責	YES	NO
ユーザー	YES	NO

このプロファイル・オプションの内部名は、CONC_MULTI_TZ です。

コンカレント : PMON 方法

PMON とは、プロセス・モニターのことを指します。内部コンカレント・マネージャは、個々のコンカレント・マネージャのプロセスをモニタリングして、各マネージャが実行中であるかどうかを検証します。内部コンカレント・マネージャは、次の2つの方法のいずれかを使用して、個々のマネージャのプロセスをモニタリングします。

RDBMS (デフォルト) RDBMS とは、リレーショナル・データベース管理システムの意味です。内部コンカレント・マネージャは、個々のマネージャが発行した値を保持しているデータベース表を読み取ります。あるマネージャが実行を停止すると、そのマネージャのデータベース表の値が変化します。

OS OS とは、オペレーティング・システムのことです。内部コンカレント・マネージャは、各マネージャのシステム・プロセス ID ごとにシグナルをオペレーティング・システムに送り、そのプロセスが完全かどうかを検証します。

注意 : OS 手法は、ハードウェア・プラットフォームまたはオペレーティング・システムでこの方法が必須の場合に限り使用してください。

このプロファイル・オプションの設定を変更するには、applsyst アカウントから SQL スクリプトを実行する必要があります。このスクリプトは afimpmon.sql というタイトルで、Application Object Library 製品ディレクトリの sql ディレクトリに入っています。たとえば UNIX では、パスは \$FND_TOP/sql/afimpmon.sql のようになります。

追加情報 : 使用しているオペレーティング・システム用の『Oracle Applications インストール・マニュアル』

注意：「コンカレント:PMON 方法」プロファイル・オプションの設定を変更するときは、事前にオラクル社カスタマ・サポートにご連絡ください。

ユーザーは、このプロファイル・オプションを参照することも更新することもできません。このプロファイル・オプションは、「システム・プロファイル・オプション」フォームからは参照も変更もできません。

レベル	可視	更新可能
サイト	NO	NO
アプリケーション	NO	NO
職責	NO	NO
ユーザー	NO	NO

このプロファイル・オプションの内部名は、CONC_PMON_METHOD です。

コンカレント: レポート・アクセス・レベル

コンカレント・プログラムが生成した出力ファイルおよびログ・ファイルをレポートするためのアクセス権限を決定します。このプロファイル・オプションは、システム管理者がユーザーまたは職責に設定できます。

「コンカレント: レポート・アクセス・レベル」プロファイル・オプションを「ユーザー」に設定すると、次の機能が有効になります。

- 要求の完了レポート出力をオンラインで表示できます。
- 要求の診断ログ・ファイルをオンラインで表示できます。(システム管理者もこの権限を持っています)。
- 「コンカレント: 出力の保存」プロファイル・オプションが「YES」に設定されている場合は、完了レポートを再印刷できます。
- 職責を変更しても、オンラインで検証できるレポートとログ・ファイルは変化しません。

「コンカレント: レポート・アクセス・レベル」プロファイル・オプションを「職責」に設定すると、アクセスできるレポートおよび診断ログ・ファイルは現行の職責により決定されます。

- 職責を変更すると、オンラインで検証できるレポートとログ・ファイルも、そのユーザーの新規の職責に合わせて変更されます。個別レポートを発行したユーザーは、常にそのレポートの出力とログ・ファイルを表示できますが、現行の職責のユーザーが発行したレポートとログ・ファイルを表示することもできます。

ユーザーは、このプロファイル・オプションを参照できますが、更新はできません。

このプロファイル・オプションは、サイト・レベル、職責レベルおよびユーザー・レベルで参照および更新することができます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	NO	NO
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は CONC_REPORT_ACCESS_LEVEL です。

コンカレント：コピーのレポート

個々のコンカレント要求用に印刷する出力の部数を設定できます。デフォルトは1に設定されます。

- この値を変更しても、すでに発行済みの要求に影響はありません。

ユーザーは、このプロファイル・オプションを参照および更新できます。

このプロファイル・オプションは、4つのどのレベルでも参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は CONC_COPIES です。

コンカレント：要求の優先度

コンカレント要求に対するデフォルトの優先順位を表示します。要求の優先順位を変更できるのは、システム管理者のみです。

要求は、通常、「発行順に実行」の原則に基づいて、開始時刻に従って実行されます。優先順位は要求開始時刻よりも優先されます。優先順位の高い要求の方が、開始時刻の早い要求より先に開始されます。

優先順位の範囲は、1（最高）～99（最低）です。標準のデフォルト順位は50です。

ユーザーは、このプロファイル・オプションを参照できますが、更新はできません。

このプロファイル・オプションは、4つのどのレベルでも参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は、CONC_PRIORITY です。

コンカレント：要求の開始時刻

要求の実行を開始する日付および時刻を設定できます。

- 開始の日付および時刻が現在の日付および時刻と同じかそれ以前の場合、要求はただちに実行できる状態になります。
- 未来の時刻（例：2002 年 6 月 12 日午後 3 時 45 分）に要求を開始する場合は、プロファイル・オプション値として 2002/06/12 15:45:00 と入力します。

注意：複数言語コンカレント要求機能を使用する場合、この値は必ず標準形式（YYYY/MM/DD HH24:MI:SS）にする必要があります。

- 日付と時刻の両方を指定する必要があります。
- この値を変更しても、すでに発行済みの要求に影響はありません。
- ユーザーは、要求発行時にこの開始時刻を上書きできます。このプロファイル・オプションをブランクのままにしておくこともできます。その場合、要求を発行すると、開始時刻を入力するよう要求されます。

ユーザーは、このプロファイル・オプションを参照および更新できます。

このプロファイル・オプションは、4つのどのレベルでも参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES

職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は、CONC_REQUEST_STAT です。

コンカレント：出力の保存

「コンカレント：出力の保存」プロファイルは、あるコンカレント・プログラムのデフォルトの動作を、その出力ファイルの保存または削除のどちらに適用するかを指定する場合に使用します。このプロファイルの対象となるコンカレント・プログラムは、Oracle Applicationsのキャラクタ・モード・バージョンで作成されたプログラムと、「出力の保存」に NULL 値に設定されているプログラムのみです。

- 「YES」を選択すると、要求出力が保存されます。
- 一部のコンカレント要求では、出力ファイルは生成されません。
- 要求出力を保存すると、要求を再印刷することができます。これは、要求が「エラー」ステータスで終了した場合、たとえば、要求は正常に実行されたにもかかわらず、プリンタが誤動作した場合などに役立ちます。
- この値を変更しても、すでに発行済みの要求には影響はありません。

ユーザーは、このプロファイル・オプションを参照および更新できます。

このプロファイル・オプションは、4つのどのレベルでも参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は、CONC_SAVE_OUTPUT です。

コンカレント：順次要求

要求のプログラムが互に互換の場合、要求を各要求の開始日時に従って一度に1つずつ（順次）実行するか、または要求を同時に実行することができます。

- 同じデータベース表に同時にアクセスしたときに各コンカレント・プログラムが取り出す値が不正確になる場合、それらのプログラムには互換性はありません。

- コンカレント・プログラムが互いに互換性がないと定義されている場合は、これらのプログラムは同時に実行できません。

「YES」を指定すると、要求は同時に実行されません。各要求は、発行された順に順次実行されます。

「NO」を指定すると、要求のコンカレント・プログラムに互換性がある場合はそれらの要求を同時に実行できます。

この値を変更しても、すでに発行済みの要求には影響はありません。

ユーザーは、このプロファイル・オプションを参照および更新できます。

このプロファイル・オプションは、4つのどのレベルでも参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は CONC_SINGLE_THREAD です。

コンカレント : 各要求発行後の要求要約の表示

この新規プロファイル・オプションを使用すると、要求を発行するたびに「要求要約」を表示するか、要求発行画面を保持するかを選択できます。

デフォルトは「YES」です。「YES」を選択すると、要求を発行するたびに「要求要約」画面が表示されます。

「NO」を選択すると、決定ウィンドウがオープンして、別の要求を発行するかどうかの確認が求められます。別の要求の発行を選択すると、発行ウィンドウに戻ります。発行ウィンドウは消去されないで、同じ要求のコピーを一部変更するだけで、簡単に要求を発行できます。

ユーザーは、このプロファイル・オプションを参照および更新できます。

このプロファイル・オプションは、4つのどのレベルでも参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES

職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は CONC_REQ_SUMMARY です。

コンカレント : 使用可能な TM を待機

クライアントが、別の翻訳マネージャ (TM) に移って試してみる前に、特定の TM が使用可能になるまで待機する最大秒数を指定できます。

ユーザーは、このプロファイル・オプションを参照および更新できます。

このプロファイル・オプションは、サイト・レベルおよびアプリケーション・レベルで参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	NO	NO
ユーザー	NO	NO

このプロファイル・オプションの内部名は、CONC_TOKEN_TIMEOUT です。

コンカレント : URL ライフタイム

このプロファイル・オプションに数値を入力すると、要求出力用の URL が維持される時間の長さを分単位で指定できます。この指定期間が過ぎると、この URL はシステムから削除されます。このプロファイル・オプションの対象は、ユーザーが「要求の発行」ウィンドウまたは「要求セットの発行」ウィンドウの通知フィールドに値を入力した要求に対して作成された URL のみです。

注意：「コンカレント・マネージャや要求のデータ等の削除」プログラムが実行されると、要求出力の URL は、URL のライフタイム期限が満了していない場合でも、すべて削除されます。

ユーザーは、このプロファイル・オプションを参照および更新できます。

このプロファイル・オプションは、すべてのレベルで参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は、CONC_URL_LIFETIME です。

通貨：混合通貨精度

「混合通貨精度」は、複数の異なる通貨を表す数値を表示するときに、小数点以下何桁まで表示するかを指定するために使用します。

- 通常、通貨を表す数値は右揃えで表示されます。
- 各通貨にはそれぞれ精度値があり、それによって表示される小数点以下の桁数が決まります。米国ドルの場合、精度のデフォルトは2なので、345.70 のように表示されます。
- 「混合通貨精度」は、表示する通貨の最大精度値と同じか、それより大きくなるように設定してください。

たとえば、米国ドル（精度 =2）、日本円（精度 =0）およびバーレーン・ディナール（精度 =3）を表示する行についてのレポートを作成する場合は、「混合通貨精度 =3」を設定します。

ユーザーは、このプロファイル・オプションを参照および更新できます。

このプロファイル・オプションは、4つのどのレベルでも参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は、CURRENCY:MIXED_PRECISION です。

通貨：負の書式

通貨の負の数値の識別には、数種類の書式を使用できます。デフォルトの識別子は通貨金額の前に付くハイフン (-) で、"-xxx" のようになります。次の記号も選択できます。

山カッコ<>< xxx >
後書きハイフン-xxx -
カッコ () (xxx)
大カッコ[] [xxx]

Oracle Applications リリース 11 で負の数の書式として "(xxx)" または "[xxx]" を使用すると、負の数は "<xxx>" と表示されます。

ユーザーは、このプロファイル・オプションを参照および更新できます。

このプロファイル・オプションは、4つのどのレベルでも参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は、CURRENCY:NEGATIVE_FORMAT です。

注意:「通貨:負の書式」は通貨表示のみに影響します。通貨以外の負の数値の前には、ここで選択したオプションとは無関係にハイフンが付きます。

通貨: 正の書式

通貨の正の数値の識別には、数種類の書式を使用できます。デフォルトの条件では、特別な識別子はありません。

ユーザーは、このプロファイル・オプションを参照および更新できます。

このプロファイル・オプションは、4つのどのレベルでも参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は、CURRENCY:POSITIVE_FORMAT です。

通貨：千単位区切り文字

3 桁区切りを使用して、通貨金額を千単位で区切ることができます。たとえば、百万は 1,000,000 のように表示されます。

ユーザーは、このプロファイル・オプションを参照および更新できます。

このプロファイル・オプションは、4 つのどのレベルでも参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は、CURRENCY:THOUSANDS_SEPARATOR です。

データベース・インスタンス

有効な two_task 接続文字列を使用すると、デフォルトの two_task を上書きできます。このプロファイルは、Oracle Parallel Server 専用に設計されています。これにより、それぞれの職責およびユーザーを、サーバーの特定のノードに接続させることができます。

ユーザーは、このプロファイル・オプションを参照できますが、更新はできません。

このプロファイル・オプションは、4 つのどのレベルでも参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は INSTANCE_PATH です。

デフォルトの国

これは、すべての所在地ゾーンについての「国」フィールドのデフォルトのソースであり、フレキシブル所在地書式機能およびフレキシブル銀行体系機能、税金登録番号および納税者ID検証ルーチンで使用されます。

このプロファイルは、「国および地域の保守」フォームのリストに登録された任意の有効な国に設定でき、ユーザーごとに異なる値に設定できます。

ユーザーは、このプロファイル・オプションを参照および更新できます。

このプロファイル・オプションは、4つのどのレベルでも参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は、DEFAULT_COUNTRY です。

フレックスフィールド：自動スキップ

フレックスフィールドにデータを入力するときのキーストローク数を減らすために、あるセグメントに完全な有効値を入力すると即座に次のセグメントに自動的にスキップするように設定できます。

- 「YES」を指定した場合、セグメントに有効な値を入力すると、自動的に次のセグメントに移動します。
- 「NO」を指定した場合、セグメントに有効な値を入力した後に次のセグメントに移動するには [Tab] を押す必要があります。

注意：セグメントの有効値の文字数とセグメントの文字数が一致しない場合、一部のセグメントでは、次に移動するために [Tab] を使用する必要があります。たとえば、5文字までの値が入力可能なフレックスフィールドのセグメント内でセグメントの有効値が4文字の場合、自動スキップでは次のセグメントに移動できません。

ユーザーは、このプロファイル・オプションを参照および更新できます。

このプロファイル・オプションは、4つのどのレベルでも参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は、FLEXFIELDS:AUTOSKIP です。

フレックスフィールド : BiDi 方向

このプロファイル・オプションは、セム語で稼動している Oracle Applications でのフレックスフィールド・ウィンドウの外観を制御します。指定できる値には、「左から右へ」および「右から左へ」があります。特定のインストレーションでこのプロファイル・オプションを定義していない場合のデフォルト値は、「右から左へ」です。この場合、ウィンドウは通常の「左から右」の方向に表示され、テキストおよびレイアウトは、セム語環境の「右から左」の性質に適応して逆方向に表示されます。

ユーザーは、このプロファイル・オプションを参照および更新できます。

このプロファイル・オプションは、4 つのどのレベルでも参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は、FLEXFIELDS:BiDi_DIRECTION です。

フレックスフィールド : LOV 警告限度

「フレックスフィールド : LOV 警告限度」は、値リストを取り出すときの効率を高めるために使用します。

LOV に大量のデータが格納されている場合、特に縮小基準を指定していないと、LOV の実行には非常に長い時間がかかることがあります。ユーザーは、リスト全体からの値の取出しを続行するかどうかを尋ねられる前に、このプロファイル・オプションに戻す対象の行数を設定します。

ユーザーは、このプロファイル・オプションを参照および更新できます。

このプロファイル・オプションは、4つのどのレベルでも参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は、QUICKPICK_ROWS_BEFORE_WARN です。

フレックスフィールド：付加ウィンドウのオープン

カスタマイズした付加フレックスフィールドにナビゲートしたときに、付加フレックスフィールド・ウィンドウを自動的にオープンするかどうかを制御できます。

- 「YES」を指定すると、カスタマイズした付加フレックスフィールドにナビゲートしたときに、付加フレックスフィールド・ウィンドウが自動的にオープンします。
- 「NO」を指定した場合、カスタマイズした付加フレックスフィールドにナビゲートする際には、「編集」メニューから「**フィールドの編集**」を選択するか、または値リストを使用して付加フレックスフィールド・ウィンドウをオープンする必要があります。

ユーザーは、このプロファイル・オプションを参照および更新できます。

このプロファイル・オプションは、4つのどのレベルでも参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は、FLEXFIELDS:OPEN_DESCR_WINDOW です。

注意：このプロファイル・オプションはフォルダ内の付加フレックスフィールドには適用されません。

フレックスフィールド：オープン・キー・ウィンドウ

キー・フレックスフィールドにナビゲートしたときに、キー・フレックスフィールド・ウィンドウを自動的にオープンするかどうかを制御できます。

- 「YES」を指定すると、キー・フレックスフィールドにナビゲートしたときに、キー・フレックスフィールド・ウィンドウが自動的にオープンします。
- 「NO」を指定した場合、キー・フレックスフィールドにナビゲートするときには、「編集」メニューから「**フィールドの編集**」を選択するか、または値リストを使用してキー・フレックスフィールド・ウィンドウをオープンする必要があります。

ユーザーは、このプロファイル・オプションを参照および更新できます。

このプロファイル・オプションは、4つのどのレベルでも参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は、FLEXFIELDS:OPEN_KEY_WINDOW です。

フレックスフィールド：共有表ロック

このプロファイル・オプションは将来のリリース用に予約されています。このプロファイル・オプションの値は変更しないでください。

このプロファイル・オプションの内部名は、FLEXFIELDS:SHARED_TABLE_LOCK です。

フレックスフィールド：短縮入力

フレックスフィールドにフレックスフィールド短縮入力 that 定義されている場合は、略称別名を使用して、フレックスフィールド内の一部またはすべてのセグメントに値を自動的に入力できます。

使用不可

略称別名が定義されているかどうかに関係なく、そのユーザーはフレックスフィールドに「短縮入力」を使用できません。

新規入力のみ

「短縮入力」は、ほとんどの外部キー・フォームで新しいレコードの入力に使用できます。組合せフォーム、既存レコードの更新または問合せの入力には使用できません。

問合せおよび新規入力	「短縮入力」は、新しいレコードの入力または問合せの入力に使用できます。既存のレコードの更新には使用できません。
全入力	「短縮入力」は、新しいレコードの入力または古いレコードの更新に使用できます。問合せの入力には使用できません。
常に	「短縮入力」は、略称別名が定義されているフレックスフィールドの挿入、更新、問合せに使用できます。

ユーザーは、このプロファイル・オプションを参照および更新できます。
このプロファイル・オプションは、4つのどのレベルでも参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は、FLEXFIELDS.SHORTHAND_ENTRY です。

フレックスフィールド：全体値の表示

ある別名が、フレックスフィールド内のすべてのセグメントについての有効な値を定義しており、「フレックスフィールド：短縮入力」が使用可能になっている場合は、その別名を入力したときにフレックスフィールド・ウィンドウは表示されません。
「YES」を選択すると、カーソルが最終セグメントにある状態でフレックスフィールド・ウィンドウが全画面表示されます。

ユーザーは、このプロファイル・オプションを参照および更新できます。
このプロファイル・オプションは、4つのどのレベルでも参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は、FLEXFIELDS.SHOW_FULL_VALUE です。

フレックスフィールド：サーバー上で検証

キー・フレックスフィールドのサーバー側 PL/SQL フレックスフィールド検証を使用可能にするには、このプロファイル・オプションを「YES」に設定します。これを設定すると、入力されたセグメント組合せの検証に必要なネットワーク上の通信量が減少するため、広域ネットワーク上でキー・フレックスフィールドを使用するときのパフォーマンスが改善されます。

ただし、クライアント側の検証を使った方が検証のパフォーマンスがよい場合があります。その場合は、このプロファイル・オプションを「NO」に設定します。

ユーザーは、このプロファイル・オプションを参照および更新できます。

このプロファイル・オプションは、4つのどのレベルでも参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は、FLEXFIELDS_VALIDATE_ON_SERVER です。

FND: デバッグ・ログ・ファイル名

ログ / トレース・サービスで使用するデバッグ・メッセージを収録しているファイルのファイル名。このプロファイル・オプションの値が NULL の場合、ログ / トレース・サービスはオフになります。

ユーザーは、このプロファイル・オプションを参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は、AFLOG_FILENAME です。

FND: デバッグ・ログ・レベル

ログ / トレース・サービスでは、デバッグ・メッセージのフィルタ処理がその優先順位レベルに応じて実行されます。デバッグ / トレース・サービスには、5つのレベルがあります。優先順位の高い方から順に、エラー、例外、イベント、プロシージャおよび文です。デバッグ・ログ・レベルは、ユーザーがメッセージを参照する場合の最低レベルです。指定できるプロファイル・オプションの値は、NULL（オフの意味）および前述の5つの優先順位レベルです。たとえば、「FND: デバッグ・ログ・レベル」プロファイルを「イベント」に設定すると、ファイルには、プログラマが「イベント」、「例外」または「エラー」というマークを付けたメッセージが書き込まれます。

ユーザーは、このプロファイル・オプションを参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は、AFLOG_LEVEL です。

FND: デバッグ・ログ・モジュール

ログ / トレース・サービスでは、デバッグ・メッセージのフィルタ処理がそのモジュールに応じて実行されます。モジュール名は、すべてのアプリケーションおよびコーディング言語に対して一意です。このプロファイル・オプションにあるモジュールを指定すると、そのモジュールに対するメッセージのみがログ・ファイルに書き込まれます。このプロファイル・オプションを空白にしておくと、すべてのモジュールに対するメッセージがログ・ファイルに書き込まれます。

ユーザーは、このプロファイル・オプションを参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は、AFLOG_MODULE です。

FND: リソース・コンシューマ・グループ

リソース・コンシューマ・グループは、Oracle8i Database Resource Manager で使用し、データベース・ユーザー間およびアプリケーション間で CPU リソースの割振りを行う機能です。各フォーム・セッションは、リソース・コンシューマ・グループに割り当てられます。システム管理者は、各ユーザーに、すべてのユーザー・フォーム・セッションおよび翻訳のリソース・コンシューマ・グループを割り当てることができます。ある処理にリソース・コンシューマ・グループが存在しない場合、そのシステムにはデフォルトのグループである Default_Consumer_Group が使用されます。

ユーザーは、このプロファイル・オプションを参照できますが、更新はできません。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は、FND_RESOURCE_CONSUMER_GROUP です。

FND: ディレクトリ上書

「FND: ディレクトリ上書」プロファイル・オプションは、Work Directory 機能で使します。「FND: ディレクトリ上書」の値は、自分の代替ファイルが書き込まれているディレクトリにしてください。通常、このプロファイル・オプションは、「ユーザー」レベルのみに設定します。

ユーザーは、このプロファイル・オプションを参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は、APPLWRK です。

フォルダ : カスタマイズの許可

システム管理者は、ユーザーがフォルダ・ブロック内でフォルダ定義レイアウトを作成またはカスタマイズできるかどうかを制御します。

- 「YES」を指定すると、フォルダ定義の作成またはカスタマイズができます。つまり、「フォルダ」メニュー全体がフォルダ・ブロック内で使用可能になります。
- 「NO」を指定した場合は、フォルダ・ブロック内で既存のフォルダ定義のオープンのみができます。つまり、「フォルダ」メニューの「オープン」オプションのみが使用可能になります。

ユーザーは、このプロファイル・オプションを参照できますが、更新はできません。

レベル	可視	更新可能
サイト	NO	NO
アプリケーション	NO	NO
職責	NO	NO
ユーザー	YES	YES

このプロファイル・オプションの内部名は、FLEXVIEW:CUSTOMIZATION です。

ゲートウェイ・ユーザー ID

ゲートウェイ・アカウントに対する Oracle ログイン。これは、環境変数 GWYUID と同じです。たとえば、applsypub/pub などです。

ユーザーは、このプロファイル・オプションを参照できますが、更新することはできません。

このプロファイル・オプションは、すべてのレベルから参照できますが、サイト・レベルでのみ更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	NO
職責	YES	NO
ユーザー	YES	NO

このプロファイル・オプションの内部名は、GWYUID です。

ヘルプ・ローカライゼーション・コード

システム管理者がこのプロファイル・オプションを設定すると、ローカライズ済みのコンテンツ依存ヘルプ・ファイルが優先されます。

ユーザーは、このプロファイル・オプションを参照できますが、更新はできません。

このプロファイル・オプションは、すべてのレベルで参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は `HELP_LOCALIZATION_CODE` です。

ヘルプ・ツリー・ルート

このプロファイル・オプションを設定すると、コンテキスト依存ヘルプが起動したときに、ナビゲーション・フレームに表示されるツリーが決まります。

ユーザーは、このプロファイル・オプションを参照できますが、更新はできません。

このプロファイル・オプションは、すべてのレベルで参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は、`HELP_TREE_ROOT` です。

診断メニュー・エントリを隠す

このプロファイル・オプションを設定すると、ユーザーが「ヘルプ」メニューから「診断」メニュー項目にアクセスできるかどうかを設定されます。YES に設定すると、「診断」メニュー項目は非表示になります。NO に設定すると、「診断」メニュー項目は表示されます。

ユーザーは、このプロファイル・オプションを参照することも更新することもできません。

このプロファイル・オプションは、すべてのレベルで参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は、FND_HIDE_DIAGNOSTICS です。

ICX: Discoverer ランチャ、フォーム・ランチャおよびレポート・ランチャ

これらのプロファイル・オプションは、Oracle Self Service Web Applications の個別ホーム・ページで使します。

これらの各プロファイル・オプションのサイト・レベルの値を、各アプリケーションを起動するベース URL に設定します。このプロファイル・オプションの値は、アプリケーションの起動に十分な値にしてください。ただし、個別のホーム・ページで提供される場合がある追加パラメータは指定しないでください。

ユーザーは、このプロファイル・オプションを参照できますが、更新はできません。

このプロファイル・オプションは、すべてのレベルで参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

これらのプロファイル・オプションの内部名は、ICX_DISCOVERER_LAUNCHER、ICX_FORMS_LAUNCHER および ICX_REPORT_LAUNCHER です。

添付の指示

このプロファイル・オプションを使用すると、レコードを問い合わせるときの添付の指示を（パフォーマンスを考慮して）オフにできます。

ユーザーは、このプロファイル・オプションを参照および更新できます。

このプロファイル・オプションは、4つのどのレベルでも参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は、ATCHMT_SET_INDICATOR です。

初期化 SQL 文 - カスタム

このプロファイル・オプションを使用すると、データベース・セッションの起動時に実行されるサイト固有の初期化コード（最適化プログラムの設定値など）を追加できます。このプロファイル・オプションの値は、必ず有効な SQL 文である必要があります。

システム管理者は、このプロファイル・オプションを任意のレベルに設定できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は、FND_INIT_SQL です。

初期化 SQL 文 - Oracle

このプロファイル・オプションは、アプリケーション固有のコードを追加する場合に使用します。値は、各データベース・セッションの起動時に実行される有効な SQL 文（または、複数の文の場合は PL/SQL ブロック）です。

このプロファイル・オプションの値は、シード・データとして送られ、更新することはできません。

注意：このプロファイル・オプションの値は変更しないでください。カスタム初期化コードを追加する場合は、「初期化 SQL 文 - カスタム」を使用してください。

このプロファイル・オプションは、アプリケーション・レベルのみで設定します。この初期化コードの実行対象は、該当のアプリケーションが所有する職責のみです。

レベル	可視	更新可能
サイト	YES	NO
アプリケーション	YES	NO
職責	YES	NO
ユーザー	YES	NO

このプロファイル・オプションの内部名は、FND_APPS_INIT_SQL です。

最大ページ長

レポートのページ当り行数の最大値を指定します。

ユーザーは、このプロファイル・オプションを参照および更新できます。

このプロファイル・オプションは、4つのどのレベルでも参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は、MAX_PAGE_LENGTH です。

MO: 営業単位

職責でログオンする営業単位を決定します。

ユーザーは、このプロファイル・オプションを参照および更新できます。

このプロファイル・オプションは、職責レベルでのみ参照および更新できます。

レベル	可視	更新可能
サイト	NO	NO
アプリケーション	NO	NO
職責	YES	YES
ユーザー	NO	NO

このプロファイル・オプションの内部名は、ORG_ID です。

従業員：インストール済

「従業員：インストール済」が使用可能になっている場合、システム管理者は、アプリケーションのユーザー名およびパスワードを従業員名にリンクすることができます。

- 「個人」フィールドは、「アプリケーション・ユーザーの定義」フォーム（**業務セキュリティユーザー**）で使用できます。

Oracle Purchasing はこの機能を使用して、部門の従業員を Oracle Applications ユーザーに関連付けます。

このプロファイル・オプションはインストレーション・プロセスで使用可能になります。「従業員：インストール済」の値は変更できません。

ユーザーは、このプロファイル・オプションを参照することも更新することもできません。

このプロファイル・オプションはサイト・レベルで参照できますが、更新はできません。

レベル	可視	更新可能
サイト	YES	NO
アプリケーション	NO	NO
職責	NO	NO
ユーザー	NO	NO

このプロファイル・オプションの内部名は、PER_EMPLOYEE:INSTALLED です。

プリンタ

レポートを印刷するプリンタを選択できます。プリンタを選択できない場合は、システム管理者に連絡してください。プリンタは、Oracle Applications に登録しておく必要があります。

ユーザーは、このプロファイル・オプションを参照および更新できます。

このプロファイル・オプションは、4つのどのレベルでも参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は、PRINTER です。

RRA: 一時ファイルの削除

カスタム・エディタを使用してコンカレント出力ファイルまたはログ・ファイルを表示する場合、Report Review Agent によりクライアント上にファイルの一時コピーが作成されます。ユーザーが Oracle Applications を終了するときにそれらのファイルを自動的に削除するには、このプロファイルを「YES」に設定します。

このプロファイル・オプションを更新できるのは、システム管理者のみです。

このプロファイル・オプションは、4つのどのレベルでも参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は、FS_DELETE です。

RRA: 使用可能

Report Review Agent を使用してコンカレント・プロセスノード上のファイルにアクセスするには、このユーザー・プロファイルを「YES」に設定します。

このプロファイル・オプションを更新できるのは、システム管理者のみです。

このプロファイル・オプションは、4つのどのレベルでも参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は、FS_ENABLED です。

RRA: サービス・プレフィクス

この新規プロファイル・オプションを使用すると、Report Review Agent に割り当てられているデフォルトのサービス名プレフィクス (FNDFS_) を上書きできます。新しいプレフィクスを Report Review Agent に割り当てると、Applications の共有実行ファイルのインスタンスを複数所有する必要はなくなります。

このオプションで有効な値は 9 文字以下で、英数字とアンダースコアのみ使用します。デフォルト値の「FNDFS_」のように、入力する値の最後の文字にアンダースコアを使用することをお勧めします。

ユーザー・レベルでは、このプロファイル・オプションを参照も更新もできません。

このプロファイル・オプションを参照および更新できるのは、サイト・レベルのみです。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	NO	NO
職責	NO	NO
ユーザー	NO	NO

このプロファイル・オプションの内部名は FS_SVC_PREFIX です。

注意：リリース 4.0 より前のバージョンの GLDI では、「RRA: サービス・プレフィクス」プロファイルはサポートされません。したがって、プロファイル・オプションに入力した値にかかわらず、デフォルトのプレフィクス「FNDFS_」が使用されます。このため、アプリケーションの実行ファイルに GLDI がアクセスできるようにするには、1 つ以上の Report Review Agents でデフォルトのプレフィクスを使用し続ける必要があります。

RRA: 最大転送サイズ

Report Review Agent が転送するファイルの最大許容サイズをバイト数で指定します。これらのファイルには、ユーザーが Oracle Applications レポート・ファイル・ビューアの「コピー・ファイル ...」メニュー・オプションを使用してダウンロードするファイル、およびカスタム・エディタにより自動的にダウンロードされる "一時" ファイルも含まれます。たとえば、サイズを 64K に設定する場合には 65536 を入力します。このプロファイルが NULL の場合、サイズ制限はありません。

このプロファイル・オプションを更新できるのは、システム管理者のみです。

このプロファイル・オプションは、4 つのどのレベルでも参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
User	Yes	Yes

このプロファイル・オプションの内部名は、FS_MAX_TRANS です。

セキュリティ・グループ使用可能

HRMS セキュリティをインプリメントしている場合は、このプロファイル・オプションを「YES」に設定します。それ以外の場合には、このプロファイル・オプションは「NO」に設定します。セキュリティ・グループ機能を使用できるのは、HRMS セキュリティのみです。

このプロファイル・オプションを更新できるのは、システム管理者のみです。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	NO	NO
ユーザー	NO	NO

このプロファイル・オプションの内部名は、ENABLE_SECURITY_GROUPS です。

順次採番

「順次採番」により、Oracle 会計管理製品のフォームで作成した文書に番号が割り当てられます。たとえば、請求書を作成するフォームでは、各請求書に順次番号が付けられます。

順次採番は、文書が転記されたか、または文書が消失したかをチェックする手段となります。アプリケーション内のすべてのフォームが、順次採番のサポート用に選択できるとは限りません。

「順次採番」には、次のプロファイル・オプション設定値があります。

常に使用 文書に順序が存在しない場合には、文書を入力できません。

未使用 常に文書を入力できます。

一部使用 順序が存在しない場合、警告が表示されますが、文書の入力はできます。

ユーザーは、このプロファイル・オプションを参照できますが、更新はできません。

このプロファイル・オプションは、サイト・レベル、アプリケーション・レベルおよび職責レベルで参照および更新できます。

注意：会計帳簿ごとに順次採番の管理が必要な場合は、'職責'レベルを使用してください。その必要がない場合は、'サイト'レベルまたは'アプリケーション'レベルのどちらかを使用してこのオプションを設定することをお勧めします。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	NO	NO

このプロファイル・オプションの内部名は、UNIQUE:SEQ_NUMBERS です。

セッション ID

この実行時プロファイル・オプションには、作成された最後のデータベースのセッション ID 番号が入っています。

ユーザーは、このプロファイル・オプションを参照できますが、更新はできません。

このプロファイル・オプションは、「システム・プロファイル・オプション」フォームからは参照も変更もできません。

レベル	可視	更新可能
サイト	NO	NO
アプリケーション	NO	NO
職責	NO	NO
ユーザー	NO	NO

このプロファイル・オプションの内部名は、DB_SESSION_ID です。

サインオン：監査レベル

「サインオン：監査レベル」を使用すると、Oracle Applications にサインオンするユーザーをどのレベルで監査するかを選択できます。機能が増える順に、「なし」、「ユーザー」、「職責」および「フォーム」の4つの監査レベルがあります。

「なし」はデフォルト値であり、Oracle Applications へサインオンしたどのユーザーも監査しないことを意味します。

「ユーザー」レベルの監査では、次のことが追跡されます。

- どのユーザーがシステムへサインオンしたか
- ユーザーがログオンおよびログオフした時刻
- 使用中の端末

「職責」レベルの監査では、「ユーザー」レベルの監査機能が実行され、かつ次のことも追跡されます。

- ユーザーが選択する職責
- ユーザーが各職責を使用した時間

「フォーム」レベルの監査では、「職責」レベルの監査機能が実行され、かつ次のことも追跡されます。

- ユーザーが選択するフォーム
- ユーザーが各フォームを使用した時間
- すべてのレベルで参照および更新を実行できるシステム管理者

ユーザーは、このプロファイル・オプションを参照することも更新することもできません。

このプロファイル・オプションは、4つのどのレベルでも参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は、SIGNONAUDIT:LEVEL です。

サインオン：通知

「YES」を選択すると、ログイン時に次のことを示すメッセージが表示されます。

- 前回のセッション以降に失敗したコンカレント要求の有無
- 自分のユーザー名と不正なパスワードを使用して他人が Oracle Applications にログインしようとした回数
- 自分のユーザー・プロファイル内で識別されているデフォルト・プリンタが登録解除されるか未指定になった時点

ユーザーは、このプロファイル・オプションを参照および更新できます。

このプロファイル・オプションは、4つのどのレベルでも参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は、SIGNONAUDIT:NOTIFY です。

他人にわかりにくいサインオン・パスワード

「他人にわかりにくいサインオン・パスワード」プロファイル・オプションは、他人にわかりにくいパスワードを選択するためのルールを設定します。パスワードは、次のルールに従っている場合に他人にわかりにくいとみなされます。

- 最低でも1個の文字と、1個の数字を持つパスワード。
- ユーザー名を含まないパスワード。

- 文字の繰り返しを含まないパスワード。

ユーザーは、このプロファイル・オプションを参照できますが、更新はできません。

このプロファイル・オプションは、4つのどのレベルでも表示および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は SIGNON_PASSWORD_HARD_TO_GUESS です。

サインオン・パスワード長

「サインオン・パスワード長」には、アプリケーションのサインオン・パスワードの長さの最小値を設定します。値を入力しないと、デフォルトにより最小値として5が設定されます。

ユーザーは、このプロファイル・オプションを参照できますが、更新することはできません。

このプロファイル・オプションは、4つのどのレベルでも参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は SIGNON_PASSWORD_LENGTH です。

サイト名

「サイト名」は、Oracle Applications のインストレーションを識別します。インストレーション・プロセスでは、これは「サイト名の指定なし」に設定されます。

インストレーションの後で、「サイト名」の値を指定してください。

「サイト名」は、MDI ウィンドウのタイトルに表示されます。インストール時にタイトルにたとえば、"テスト"や"製品"などの追加情報が表示されるようにする場合は、ここで情報を追加できます。

ユーザーは、このプロファイル・オプションを参照することも更新することもできません。

このプロファイル・オプションは、ユーザー・レベルで参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	NO	NO
職責	NO	NO
ユーザー	NO	NO

このプロファイル・オプションの内部名は、SITENAME です。

ソケット・リスナー有効化

このプロファイル・オプションは、FormsClient Controller（ソケット・リスナー）がサインオン・フォームで起動したかどうかを示すフラグです。

ユーザーは、このプロファイル・オプションを参照できますが、更新することはできません。

このプロファイル・オプションは、4つのどのレベルでも参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は、SOCKET_LISTENER_ACTIVATED です。

ストアドプロシージャ・ログディレクトリ

ログディレクトリを指定すると、Oracle 7.3 で使用したストアドプロシージャを使用してログファイルの生成および保存ができます。また、このログディレクトリは、データベースの init.ora ファイルで設定する必要もあります。

たとえば、「ストアド・プロシージャ・ログ・ディレクトリ」が /rladev/rla/1.1/log で「ストアド・プロシージャ出力ディレクトリ」が /rladev/rla/1.1/out の場合には、(前述のディレクトリに書き込むストアド・プロシージャが保存されているデータベースの) init.ora ファイルに次のとおり入力してください。

```
UTL_FILE_DIR = /rladev/rla/1.1/log,  
/rladev/rla/1.1/out
```

ユーザーは、このプロファイル・オプションを参照することも更新することもできません。
このプロファイル・オプションは、ユーザー・レベルで参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	NO	NO
職責	NO	NO
ユーザー	NO	NO

このプロファイル・オプションの内部名は、UTL_FILE_LOG です。

ストアドプロシージャ出力ディレクトリ

出力ディレクトリを指定すると、Oracle 7.3 で使用したストアドプロシージャを使用して出力ファイルの生成および保存ができます。また、この出力ディレクトリをデータベースの init.ora ファイルで設定する必要もあります。

たとえば、「ストアドプロシージャ・ログディレクトリ」が /rladev/rla/1.1/log で「ストアドプロシージャ出力ディレクトリ」が /rladev/rla/1.1/out の場合は、これらのディレクトリに書き込むストアドプロシージャが格納されているデータベースの init.ora ファイルに次のとおり入力してください。

```
UTL_FILE_DIR = /rladev/rla/1.1/log,  
/rladev/rla/1.1/out
```

ユーザーは、このプロファイル・オプションを参照することも更新することもできません。
このプロファイル・オプションは、サイト・レベルで参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	NO	NO

職責	NO	NO
ユーザー	NO	NO

このプロファイル・オプションの内部名は、UTL_FILE_OUT です。

TCF: HOST

「TCF:PORT」プロファイルとこのプロファイルを組み合わせて設定すると、TCF Server のネットワーク位置が明確になります。TCF Server では、いくつかの関連サーバー・ロジックを実行してデータベースへアクセスできるようにすると、Oracle Applications の UI の様々な部分がサポートされます。

ほとんどの構成では、これらのプロファイルは、TCF Server が起動すると同時に、TCF Server の管理ユーティリティである 'ServerControl' によって設定されます。ServerControl を使用すると、これらの2つのプロファイル（TCF:HOST、TCF:PORT）はサイト・レベルで設定されます。

特に複雑な環境の場合は、異なるユーザーを別々の TCF Servers に割り当てる方が妥当な場合があります。このためには、これらのプロファイルを設定して、「アプリケーション」レベルで値を識別します。TCF Server の構成オプションの詳細は、インストール後のインストラクションを参照してください。

このプロファイル・オプションは、4つのどのレベルでも参照できますが、更新できるのはサイト・レベルとアプリケーション・レベルのみです。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	NO
ユーザー	YES	NO

このプロファイル・オプションの内部名は TCF:HOST です。

参照：「TCF Server の管理」

TCF: PORT

「TCF:HOST」プロファイルとこのプロファイルを組み合わせて設定すると、TCF Server のネットワーク位置が明確になります。TCF Server では、いくつかの関連サーバー・ロジックを実行してデータベースへアクセスできるようにすると、Oracle Applications の UI の様々な部分がサポートされます。

ほとんどの構成では、これらのプロファイルは、TCF Server が起動すると同時に、TCF Server の管理ユーティリティである 'ServerControl' によって設定されます。ServerControl を使用すると、これらの 2 つのプロファイル（TCF:HOST、TCF:PORT）はサイト・レベルで設定されます。

特に複雑な環境の場合は、異なるユーザーを別々の TCF Servers に割り当てる方が妥当な場合があります。このためには、これらのプロファイルを設定して、「アプリケーション」レベルで値を識別します。TCF Server の構成オプションの詳細は、「Oracle Applications のインストール」を参照してください。

ユーザーは、このプロファイル・オプションを参照できますが、更新することはできません。

このプロファイル・オプションは、4 つのどのレベルでも参照できますが、更新できるのはサイト・レベルとアプリケーション・レベルのみです。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	NO
ユーザー	YES	NO

このプロファイル・オプションの内部名は TCF:PORT です。

参照：「TCF Server の管理」

2 タスク

データベースに対する TWO_TASK。このプロファイルを「ゲートウェイ・ユーザー ID」プロファイルと一緒に使用すると、Web Server 用の動的 URL の作成に使用する接続文字列を設定できます。この文字列はデータベースの SQL*NET 別名に設定してください。

注意：TWO_TASK は、WebServer が稼動しているノード上で有効である必要があります。

ユーザーは、このプロファイル・オプションを参照できますが、更新することはできません。

このプロファイル・オプションは、どのレベルでも参照できますが、更新できるのはサイト・レベルのみです。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	NO
職責	YES	NO
ユーザー	YES	NO

このプロファイル・オプションの内部名は、TWO_TASK です。

ユーティリティ：診断

「ユーティリティ：診断」では、ユーザーが診断機能を自動的に使用できるかどうかを設定されます。「ユーティリティ：診断」を YES に設定すると、ユーザーはこれらの機能を自動的に使用できます。「ユーティリティ：診断」を NO に設定すると、ユーザーは診断機能を使用する場合に APPS スキーマのパスワードを入力する必要があります。

ユーザーは、このプロファイル・オプションを参照することも更新することもできません。

このプロファイル・オプションは、すべてのレベルで参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は、DIAGNOSTICS です。

ユーティリティ：SQL トレース

SQL トレースファイルをそれぞれのコンカレント・プログラムについて生成できるようになりました。プロファイル「ユーティリティ：SQL トレース」を YES に設定することにより、ユーザー・レベルで追跡を実行できます。このプロファイルをユーザーに対して有効化できるのは、システム管理者のみです。したがって、このプロファイルが誤って有効化されることがなく、しかもディスク使用状況をモニターできます。

SQL トレースの詳細は、『Oracle 8i SQL リファレンス』マニュアルを参照してください。

ユーザーは、このプロファイル・オプションを参照することも更新することもできません。

このプロファイル・オプションは、全レベルで参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

ビューア : HTML、PCL、PDF、ポストスクリプトおよびテキスト・アプリケーション用

これらのプロファイル・オプションを設定すると、ユーザーが指定の出力フォーマットでレポートを表示するのに使用するアプリケーションが決まります。たとえば、テキスト・レポートを Microsoft Word で表示するには、「ビューア : テキスト・アプリケーション用」を 'application/word' に設定できます。

有効な値は、システム管理者にが「ビューア・オプション」フォームで定義します。

ユーザーは、これらのプロファイル・オプションを参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

これらのプロファイル・オプションの内部名は、FS_MIME_HTML、FS_MIME_PCL、FS_MIME_PDF、FS_MIME_PS および FS_MIME_TEXT です。

ビューア : デフォルト・フォント・サイズ

この新規プロファイル・オプションを使用すると、Report Viewer にレポート出力を表示するときのデフォルトのフォント・サイズを設定できます。

このオプションに有効な値は 6、8、10、12、14 です。

ユーザーは、このプロファイル・オプションを参照および更新できます。

このプロファイル・オプションは、4 つのどのレベルでも参照および更新できます。

レベル	可視	更新可能
-----	----	------

サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は FNDCPVWR_FONT_SIZE です。

ビューア : テキスト

「ビューア : テキスト」プロファイル・オプションを設定すると、デフォルトの Report Viewer を使用することなく、レポート出力を直接ブラウザ・ウィンドウに送ることができます。このプロファイル・オプションに「ブラウザ」と入力すると、この機能を利用できます。

ユーザーは、この「ビューア : テキスト」プロファイル・オプションを参照および更新できます。

このプロファイル・オプションは、4 つのどのレベルでも参照および更新できます。

レベル	可視	更新可能
サイト	YES	YES
アプリケーション	YES	YES
職責	YES	YES
ユーザー	YES	YES

このプロファイル・オプションの内部名は、EDITOR_CHAR です。

登録済みアラートの使用

Oracle Alert の概要

Oracle Alert は、例外管理の方法として完成されたものです。

Oracle Alert は、データベースにおける重大なアクティビティをすぐに表示します。そのため、重要または異常な業務イベントが発生したとき、事態をすぐに把握できます。Oracle Alert を使用すると、部門およびその構成員のパフォーマンスをリアルタイムで計測できるため、障害が発生しそうな点に注意を集中することができます。定常的なトランザクションは Oracle Alert で自動化できるため、貴重な時間を重要度の高い作業に振り分けることができます。しかも、Oracle Alert ではこれがすべてオンラインで処理されるため、山積みの書類と格闘する必要がありません。

Oracle Alert は望ましい方法で業務情報をモニターするために必要なフレキシビリティを提供します。

基本的な業務上のニーズ

Oracle Alert は、次のように業務上の基本的ニーズを満たす機能を提供します。

- 例外条件が発生すると通知します。
- 監視する例外条件および通知頻度を指定できます。
- 共通のアプリケーション、つまり電子メールを通じてアラート・メッセージを送信して例外条件を通知します。
- アラート・メッセージに対する応答に基づき、指定の処理を実行します。
- 定義されたスケジュールに従って、データベースの定常的作業を自動的に実行します。
- 使用中の電子メール・システムと完全に統合できます。

Oracle Alert のランタイム機能

完全な Oracle Alert 製品のライセンス・コピーを持っていない場合でも、Oracle Applications 製品パッケージに組み込まれている登録済みアラートを使用することにより、主要な Oracle Alert 機能から得られる利点を利用することができます。

すべての Oracle Applications 製品のパッケージには、Oracle Alert のランタイム・バージョンが含まれています。このランタイム・バージョンでは Oracle Alert ウィンドウがすべて使用できますが、各ウィンドウ内の全機能が使用できるわけではありません。Oracle Alert のランタイム・バージョンでは、製品のパッケージにあらかじめ登録されているアラートのみを実行できます。新しいアラートの作成はできません。

アラート定義

アラート

データベース内で特定の例外条件をチェックするメカニズム。アラートは、それに含まれている SQL SELECT 文に特徴があります。SQL SELECT 文は、どのデータベース例外を識別するか、およびその例外についてどのような出力を生成するかをアプリケーションに伝えます。

たとえば、100 万円を超える発注にフラグを立てるアラートを定義し、そのアラートで、発注を要求した個人の名前、およびその上司であるマネージャの名前が出力されるように定義できます。登録済みアラートはすべて、Oracle Alert の「アラート」ウィンドウに一覧表示されます。

イベント・アラート

イベント・アラートは、データベース内での特定の例外または変更の発生をモニターします。Oracle Applications のウィンドウを使用して情報を追加または更新すると、データベースで例外が発生します。イベント・アラートは、指定の SQL SELECT 文に基づいて、データベースに発生する例外をモニターします。

定期アラート

定期アラートは、定義された計画に従って、重要な情報を定期的に通知します。定期アラートは、イベント・アラートのようにデータベース内の即時例外イベントを通知するのではなく、SQL SELECT 文に指定された特定のデータベース情報を計画された間隔で走査します。

アラート処理

アラート処理とは、ユーザーのアラートで実行させる処理です。アラート処理は、アラートからの出力に依存するように設定できます。アラート処理は、次の 3 つのカテゴリのいずれかに入ります。

- 詳細処理 -- データベース内で検出された 1 つの例外を表す処理

- 要約処理 -- データベース内で検出された複数の例外を表す処理
- 例外なし処理 -- データベース内で例外がまったく検出されなかったことを表す処理

処理には、メール ID への電子メール・メッセージの送信、Oracle Applications プログラムの実行、オペレーティング・システムのプログラムやスクリプトの実行またはデータベース内の情報を修正するための SQL スクリプトの実行を組み込むことができます。

1 つのアラートには複数の処理を指定することが可能で、処理にはアラートの出力を取り込むことができます。たとえば、ある例外が発生したときに、特定のアラートでマネージャにメッセージを送信し、同時に Oracle Applications プログラムを実行するように設定できます。

処理セット

処理セットは、特定のアラートについて使用できる一連のアラート処理です。処理セットに組み込む各処理には、順序番号を割り当てて、処理の実行順序を指定できます。登録済みアラートの中には、複数の処理セットがあるものもあります。各処理セットにも順序番号を割り当てることができ、それを使用して各処理セットの実行順序を指定できます。

関連項目

- 「Oracle Alert の DBA アラート」(B-8)
- 「Oracle Alert のメール削除アラート」(B-13)
- 「Oracle Alert の削除アラート」(B-12)
- 「Oracle Alert の事前コード化アラート」(B-7)
- 「事前定義済アラート」(B-3)
- 「登録済みアラートのカスタマイズ」(B-5)

事前定義済アラート

次の 2 種類の登録済みアラートがあります。

- イベント・アラート -- たとえば、Oracle Purchasing 用の「受入通知」アラートは、品目が受け入れられ「受入れ」ウィンドウに入力されると、メール・メッセージによりリクエストに通知されます。
- 定期アラート -- たとえば、Oracle Material Planning 用の「超過消費予測」アラートは、消費予測の超過を毎日チェックし、「予測入力」ウィンドウに表示されている現行予測量がマイナスになると、メール・メッセージを送信します。

提案：ご使用の Oracle Applications 製品のパッケージに含まれている登録済みアラートのリストについては、その製品のリファレンス・ガイドを参照してください。

登録済みアラートの使用

登録済みアラートは、初期状態では使用禁止になっています。使用するには、アラートを使用可能にする必要があります。登録済みアラートを表示または使用するには、Oracle Applications の起動時に「Oracle アラート・マネージャ」職責を選択します。「アラート・マネージャ」職責では、Oracle Alert メニューにアクセスすることができます。

登録済みアラートを使用可能にするか編集するには、「アラート」ウィンドウにナビゲートします。Oracle Applications 製品用の登録済みアラートを表示するには、ご使用の Oracle Applications の製品名を「アプリケーション」フィールドに指定して、問合せを実行します。

「名前」フィールドに、登録済みアラートの名前が表示されます。「タイプ」フィールドには、アラートがイベント・アラートと定期アラートのどちらであるかが表示されます。

「使用可」チェックボックスをチェックすることによって、アラートを実行可能にすることができます。「終了日」を入力して、このアラートを実行する最終日を指定することもできます。

「アラート詳細」ウィンドウをオープンするには、「アラート詳細」ボタンを選択します。使用可能なインストレーションを表示するには、「アラート・インストレーション」タブ付リージョンを選択します。

アラートの実行対象となるアプリケーション・インストレーションの Oracle ID を入力します。選択できるのは、アラートを所有しているアプリケーションに関連付けられた Oracle ID のみです。「使用可」チェックボックスをオフにすることによって、Oracle ID を一時的に使用禁止にすることができます。

「処理」ウィンドウをオープンするには、「処理」ボタンを選択します。Oracle Alert は、アラートについて定義された処理を自動的に表示します。

「処理タイプ」が「詳細」になっている場合は、「処理」ウィンドウで「処理詳細」ボタンを選択すると、その処理についての詳細が表示されます。

アラート処理は、「メッセージ詳細」ゾーンの「宛先」フィールドに表示されているメール ID 宛てにアラート処理メッセージを送信します。メール ID の書式が (Name はアラートにより定義されている出力) であれば、このフィールドを修正する必要はありません。しかし、「宛先」フィールドのメール ID が上記の形式でない場合、またはこのフィールドに値が表示されていない場合は、アラート処理メッセージを受け取るユーザーのメール ID を入力する必要があります。このウィンドウの内容を修正したら、作業結果を保存してください。

「Oracle Alert オプション」ウィンドウにナビゲートします。登録済みアラートと統合する電子メールを指定するには、このウィンドウを使用します。

「アラート」ウィンドウで「処理セット」ボタンを選択して、「処理セット」ウィンドウにナビゲートします。Oracle Alert は、アラートについて定義された処理のセットを自動的に表示します。

使用する各処理セットについて、「使用可」チェックボックスをオンにします。「終了日」フィールドに日付を入力して、このアラート処理セットを使用可能にしておく最終日を指定することもできます。

さらに、「処理セット・メンバー」ブロックで、その処理セット内で使用する各処理セット・メンバーごとに「使用可」チェックボックスをチェックします。

「終了日」を入力して、このアラート処理セット・メンバーを使用可能にしておく最終日を指定することもできます。終了するときは、作業結果を保存してください。

これで、登録済みアラートはいつでも使用できる状態になりました。

登録済みアラートのカスタマイズ

次の方法で、業務要件に合わせて登録済みアラートをカスタマイズできます。

電子メールの統合

Oracle Alert は、Oracle Office と完全に統合でき、Oracle Office を使用してユーザーに電子メールを送信することができます。Oracle Office は他の電子メール・システムへのゲートウェイを備えているので、他の電子メール・システムのユーザーへも同様に Oracle Alert からメッセージを送信できます。Oracle Alert は、UNIX メール・システム、VMS メール・システムまたはカスタム・メール・システムを使用してユーザーに電子メールを送信できます。

登録済みアラートとともに使用する電子メール・アプリケーションを指定するには、「Oracle Alert オプション」ウィンドウをオープンし、「メール・システム」タブ付リージョンを使用します。「名前」に電子メール・アプリケーションの名前を入力し、「コマンド」にメール・アプリケーションを起動するために使用するオペレーティング・システム・コマンドを入力し、メール・プログラムに渡すパラメータがある場合はそれを「パラメータ」に入力します。

Oracle Office を使用する場合は、オペレーティング・システム・コマンドを指定する必要はありません。メール・アプリケーションについての情報を入力したら、「使用中」チェックボックスをオンにし、その後で作業結果を保存してください。一時点で使用可能にできるメール・アプリケーションは1つのみです。

標準アラート・メッセージ・テキスト

すべてのアラート・メッセージ処理に表示されるメッセージのヘッダーおよびフッターのテキストをカスタマイズできます。「Oracle Alert オプション」ウィンドウの「メッセージ要素」タブ付リージョンにナビゲートすると、4つのメッセージ要素が自動的に表示されます。各要素は、すべてのアラート・メール・メッセージに表示される特定タイプのメッセージ・テキストを表します。

Oracle Alert のランタイム・バージョンの場合は、編集する必要があるのは「メッセージ処理ヘッダー」要素および「メッセージ処理フッター」要素のみです。表示されるテキストをカスタマイズするだけで、各アラート・メッセージの始めと終わりのテキストを変更できま

す。テキストをブランクにして、アラート・メッセージに標準テキストが何も表示されないようにすることもできます。このウィンドウで変更を行った場合は、必ず作業結果を保存してください。

アラート頻度

登録済みの各定期アラートを実行する頻度を計画することができます。一部のアラートは毎日チェックし、一部は月に一度のみ、その他は明示的に要求したときにのみチェックできるようにすることができます。重大な例外条件を毎日、または必要なら 1 日に 2 回以上監視するように設定できるフレキシビリティがあります。また、重要度の低い例外条件はスケジュール頻度を低くして、たとえば月に一度に設定することもできます。

登録済みアラートの頻度を変更するには、「アラート」ウィンドウにナビゲートします。問合せを実行して修正する登録済みの定期アラートを表示し、その定期アラートの「頻度」を変更します。

アラート履歴

Oracle Alert は、特定のアラートについて例外および処理の履歴を記録することができます。アラートについて記録する履歴の日数を変更するには、「アラート」ウィンドウを使用します。そして、「記録日数」フィールドの値を、記録する履歴の日数に変更してください。

重複の抑制

同じアラート例外について Oracle Alert が同じメッセージを繰り返し送信しないようにするには、重複メッセージの抑止を選択できます。Oracle Alert は、同じアラートについて重複する例外条件を検出すると、そのアラートの処理セット・メンバーの再実行を中止します。

このオプションを指定するには、「アラート」ウィンドウの「処理セット」ブロックの「重複の抑止」チェックボックスを使用します。「重複の抑止」チェックボックスのデフォルトは「チェックなし」です。「重複の抑止」チェックボックスをオンにする場合は、アラート履歴を記録する日数が、アラートのチェックの間隔日数より最低 1 日は長くなるように注意してください。Oracle Alert は、例外が重複しているかどうかを判別するのに履歴情報を使用します。

メッセージ処理

登録済みアラートにメッセージ処理が含まれている場合は、そのメッセージ処理の特定の局面をカスタマイズすることができます。「アラート」ウィンドウで「処理」ボタンを選択することにより、「処理」ブロックにナビゲートしてください。このブロックで、カスタマイズするメッセージ処理が表示されている行にカーソルを移動し、「処理詳細」ボタンを押すと、そのメッセージ処理についての「処理詳細」ウィンドウがオープンされます。メッセージ処理の次の機能を修正することができます。

- 受信人リスト – 「リスト」、「宛先」、「Cc」、「Bcc」または「ユーザー印刷」の各フィールドのメール ID を追加または削除することができます。**Name** の形式で表示されてい

るメール ID は修正しないでください。これは、アラート出力により定義されるメール ID を表しています。

- プリント --Oracle Alert でメッセージを送信する宛先プリンタの名前を修正することができます。
- テキスト --アラート・メッセージで送信するボイラープレート・テキストを修正することができます。テキストのボディ中に使用されているアラート出力 (**Name** 形式のもの) は編集しないでください。要約メッセージの場合は、その要約メッセージ内の開始テキストおよび終了テキストのみを編集します。修正を完了したら、作業結果を保存してください。

要約しきい値

登録済みアラートは、詳細処理、要約処理、例外なし処理という 3 つの処理タイプのうち 1 つを使用します。例外なし処理は、アラートについて例外がまったく検出されない場合に Oracle Alert が定義済み処理を実行するという点で、単純明快です。

しかし、Oracle Alert は、詳細処理または要約処理を実行する時点をどのようにして認識するのでしょうか。Oracle Alert は、検出したすべての例外について、その例外の数に関係なく詳細処理を実行したり、あるいは特定の一連の例外について要約処理を実行することができます。たとえば、アラートにより検出される各例外について個別のメール・メッセージを受け取るようにもできるし、同じアラートにより検出されたすべての例外を要約して示す単一のメッセージを受け取るようにもできます。

「アラート」ウィンドウの「処理セット」ブロックの「メンバー」タブ付リージョンで「要約しきい値」を設定することにより、Oracle Alert が例外をいくつ検出した時点で詳細処理から要約処理に切り替えるかを指定できます。

関連項目

- [「Oracle Alert の DBA アラート」](#) (B-8)
- [「Oracle Alert のメール削除アラート」](#) (B-13)
- [「Oracle Alert のランタイム機能」](#) (B-2)
- [「Oracle Alert の概要」](#) (B-1)
- [「Oracle Alert の削除アラート」](#) (B-12)
- [「Oracle Alert の事前コード化アラート」](#) (B-7)

Oracle Alert の事前コード化アラート

Oracle Alert インストレーションには、データベースおよび Oracle Alert の使用中に生成したデータの管理に役立つように設計されたカスタム・アラートが含まれています。Oracle

Alert には、システムの使用可能な表領域およびディスク領域、割当に関する問題を系統立ててモニターするための 8 つのアラートがあるため、データベース管理者の作業効率が上がりデータベースのパフォーマンスも向上します。

時には、古いコンカレント要求、アラート・チェックおよび処理セット・チェックをデータベースから削除することが必要になる場合があります。Oracle Alert は、古いファイルを定期的に削除して貴重な表領域を解放し、データベースのパフォーマンスを向上させるための 2 つのアラートを提供します。また、Oracle Alert には、古いメッセージが入った電子メール・フォルダを消去し、送信メール・アカウントおよび応答メール・アカウントを管理しやすいサイズに維持するためのアラートもあります。

ここでは、この 11 のアラートの概要を説明し、各アラートを使用してシステム・パフォーマンスを向上させるためのヒントを示します。

用語

事前コード化アラートについての説明を読む前に、用語集に記載されている次の用語の意味を理解しておいてください。

- 定期アラート
- 例外
- 処理
- 詳細処理
- 要約処理
- 例外なし処理
- 入力

Oracle Alert の DBA アラート

Oracle Alert の DBA アラートは、データベースの管理に役立つアラートで、次の事項を定期的に通知します。

- 次のエクステントが割当不可能な表および索引
- 表領域の割当制限に達しかけているユーザー
- 十分な空き領域がない表領域
- サイズを超過したかまたは断片化した表および索引
- 最大エクステント数に達しかけている表および索引

カスタマイズ可能なアラート頻度

Oracle Alert の DBA アラートは定期アラートです。したがって、データベースをチェックする頻度を指定できます。データベースのニーズに応じて、日に 1 回または週に 1 回、月に 1 回の実行を設定してください。

要約メッセージおよび例外なしメッセージ

Oracle Alert は、DBA アラートに指定されている例外を検出すると、検出された例外すべてを要約したメッセージをデータベース管理者に送信します。例外をまったく検出しなかった場合は、例外が検出されなかったことを示すメッセージを送信します。つまり、データベースのステータスが変わらない場合でも、Oracle Alert からステータスの通知が送られてくることになります。

カスタマイズ可能なアラート入力

入力により、DBA アラートをカスタマイズできます。アラートのターゲットにする Oracle ユーザー名または表、索引を指定することができ、Oracle Alert の検索対象とするエクステンツ数のしきい値、最大エクステンツ数およびブロック数を指定することができます。また、入力値は、処理セット・レベルで定義できるため、異なるユーザー名、表および索引をターゲットとする複数の処理セットを作成できます。処理セットは必要に応じていくつでも作成できます。

複数のデータベース・インスタンスのサポート

Oracle Applications の DBA アプリケーションは Oracle Alert の DBA アラートを所有します。したがって、Oracle Alert は、Oracle Alert のデータベースの外部にあるインスタンスも含めて、ユーザー作成の各データベース・インスタンスについて DBA アラートを実行します。

Applications DBA アラートの説明

各 DBA アラートのカスタマイズ可能な頻度および入力について、次に説明します。

次のエクステンツが割当不可能な表

このアラートは、次のエクステンツが最大空きエクステンツより大きくなる表を検索します。

頻度	N 日ごと
入力	表名、ORACLE ユーザー名

次のエクステンツが割当不可能な索引

このアラートは、次のエクステンツが最大空きエクステンツより大きくなる索引を検索します。

頻度	N 日ごと
入力	索引名、ORACLE ユーザー名

表領域の割当制限に近いユーザー

このアラートは、表領域の割当制限にまもなく達してしまうユーザーを検出します。

頻度	N 日ごと
入力	ORACLE ユーザー名
表領域名	

残りの最小空き領域（%）のチェック
最大使用可能領域（%）のチェック
残りの最小空き領域合計（バイト数）
最大使用可能領域（%）のチェック

十分な空き領域がない表領域

このアラートは、指定された最小量の空き領域がない表領域を検索します。

頻度	N 日ごと
入力	表領域名

残りの合計空き領域のチェック
使用可能エクステントの最大サイズのチェック
使用可能エクステントの最大サイズ（バイト数）
残りの最小空き領域合計（バイト数）

サイズ超過または断片化した索引

このアラートは、指定のブロック数またはエクステント数を超えた索引を検出します。

頻度	N 日ごと
入力	索引名
入力	ORACLE ユーザー名

最大ブロック数のチェック
最大エクステント数のチェック

最大ブロック数
最大エクステント数

サイズ超過または断片化した表

このアラートは、指定のブロック数またはエクステント数を越えた表を検出します。

頻度 N 日ごと
入力 表名
 ORACLE ユーザー名
 最大ブロック数のチェック
 最大エクステント数のチェック
 最大ブロック数
 最大エクステント数

最大エクステント数に近い表

このアラートは、最大エクステント数に指定されたエクステント数以内にある表および索引を検索します。

頻度 N 日ごと
入力 表名
 ORACLE ユーザー名
 残りの最小エクステント数

最大エクステント数に近い索引

このアラートは、最大エクステント数に指定されたエクステント数以内にある表および索引を検索します。

頻度 N 日ごと
入力 索引名
 ORACLE ユーザー名
 残りの最小エクステント数

Oracle Alert の削除アラート

Oracle Alert 事前コード化アラートの中の 2 つは、Oracle Alert の使用時に生成するデータの管理に役立つように設計されています。Oracle Alert の使用時に、次のことができます。

- 指定日数以上を経過した古いコンカレント要求を自動的に削除します。
- 指定日数を経過した古いアラート・チェックおよび処理セット・チェックを自動的に消去します。

カスタマイズ可能なアラート頻度

削除アラートの実行計画はユーザーが決定できます。Oracle Alert は、定義された計画に基づいてコンカレント・マネージャに削除アラートを発行し、古いコンカレント要求をすべて削除します。

カスタマイズ可能なアラート入力

入力によりアラートをカスタマイズできます。削除アラートのターゲットにするアプリケーションおよびコンカレント・プログラムを指定し、データが不要になる、つまり「古い」状態になる時期を決定してください。入力値は処理セット・レベルで定義するので、異なるアプリケーションおよび異なるコンカレント・プログラムをターゲットにして複数の処理セットを作成することができます。処理セットは必要に応じていくつでも作成できるので、システムへの不要なファイルの蓄積が解消されます。

Oracle Alert の削除アラートの説明

各削除アラートのカスタマイズ可能な頻度および入力について、次に説明します。

アラート・チェックおよび処理セット・チェックの削除

このアラートは、指定日数を経過した古いアラート・チェックおよび処理セット・チェックを検索し、それを削除する SQL 文スクリプトを実行します。

アラート・タイプ	定期
周期性	N 日ごと
入力	アプリケーション名、アラート・チェック以降の日数

注意： オープン中の応答のある応答処理アラートについては、Oracle Alert はアラート・チェックも処理セット・チェックも削除しません。

コンカレント要求の削除

このアラートは、指定日数を経過した古いコンカレント要求およびそのログ・ファイルと出力ファイルを検索し、それを削除するコンカレント・プログラムを実行します。コンカレント・プログラム名を入力する場合は、プログラム名（FND_CONCURRENT_REQUESTS 表の USER_CONCURRENT_PROGRAM_NAME 列にある名前）を使用する必要があります。「要求」ウィンドウに表示されるコンカレント・プログラム名に付随したオプション記述は使用しないでください。

アラート・タイプ	定期
周期性	N 日ごと
入力	アプリケーション名 コンカレント・プログラム名 コンカレント要求をコンカレント・マネージャに発行した後の経過日数
オペレーティング・システム・プログラム	ログ・ファイル、出力ファイルおよび各コンカレント要求の対応レコードを削除します。
引数	コンカレント要求 ID

Oracle Alert のメール削除アラート

Oracle Alert の事前コード化アラートの 1 つは、Oracle Office フォルダを管理可能なサイズに保つために役立つように設計されています。特に、応答処理を使用している場合、応答アカウントに古いメッセージが蓄積するのは望ましい状態ではありません。Oracle Alert の使用時に、次のことができます。

- 定義した Oracle Alert の Oracle Office アカウントから古くなったメール・メッセージを自動的に削除します。
- 古いメッセージを消去するための Oracle Office アカウントおよび Oracle Office フォルダを指定します。
- 削除するメッセージを判別します。

カスタマイズ可能なアラート頻度

アラートの実行計画はユーザーが決定できます。Oracle Alert は、定義された計画に基づいてコンカレント・マネージャにメール削除警告を発行します。

カスタマイズ可能なアラート入力

入力を使用して、削除する Oracle Office アカウント、メール・フォルダおよびメッセージを Oracle Alert に指示します。入力値は処理セット・レベルで定義するので、異なるメール・

アカウントおよび異なるメール・フォルダをターゲットにして複数の処理セットを作成できます。処理セットは、メール・アカウントを最新の状態に保つために必要に応じていくつでも作成できます。

Oracle Alert のメール削除アラートの説明

メール削除アラートのカスタマイズ可能な頻度および入力について、次に説明します。

Oracle Office メッセージの削除

頻度	週次
入力	有効日数
フォルダ	
	Oracle Office アカウント

関連項目

- [「Oracle Alert のランタイム機能」](#) (B-2)
- [「Oracle Alert の概要」](#) (B-1)
- [「事前定義済アラート」](#) (B-3)
- [「登録済みアラートのカスタマイズ」](#) (B-5)

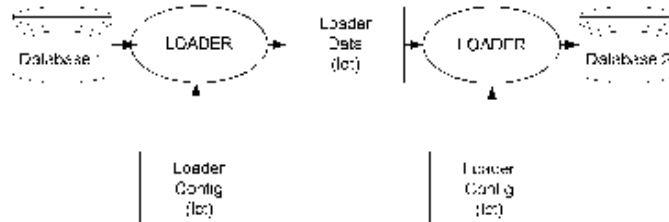
一般ローダー

一般ローダー（FNDLOAD）は、Oracle Applications データをデータベースとテキスト・ファイル形式間で移動できるコンカレント・プログラムです。このローダーでは、構成ファイルが読み込まれ、アクセスするデータが決定されます。特定の構成ファイルの情報については、製品グループの『オープン・インタフェース・ガイド』を参照してください。

警告： Oracle Applications により提供されるローダー・ファイルのみを使用してください。Oracle Applications により提供されたファイル以外のものを使用したり、提供されたスクリプトを変更すると、データベースが破損する危険があります。Oracle では、カスタム・ローダー・ファイルまたは変更された Oracle Applications ロード・ファイルはサポートしません。

概要

一般ローダーでは、アプリケーション・エンティティからデータを編集可能かつ移植性のあるテキスト・ファイルにダウンロードできます。このファイルは、他のデータベースへアップロードし、データをコピーできます。データベース・ストアおよびファイル書式間の変換は、ローダーによって読み込まれる構成ファイルにより指定されます。情報のフローは次のとおりです。



このローダーは、ダウンロードまたはアップロードの2つのモードのいずれかで作動します。ダウンロード・モードの場合、データは、データベースからテキスト・ファイルにダウンロードされます。アップロード・モードの場合、データはテキスト・ファイルからデータベースにアップロードされます。

このローダーによってサポートされるデータ構造は、マスター / ディテールの関係および外部キー参照の関係が含まれます。

ダウンロードおよびアップロード時、関連するデータの構造は、構成ファイルによって説明されます。構成ファイルでは、データの構造とデータをデータベースへ、またはデータベースからコピーする際に使用されるアクセス方式が説明されます。同じ構成ファイルがアップロードおよびダウンロードの両方で使用される場合があります。

ダウンロードの際、一般ローダーでは、データ・ファイルという補助ファイルが作成されます。このファイルには、ダウンロード用に選択された構造化データが含まれます。データ・ファイルには、ダウンロードされるデータを表す標準構文があります。アップロードの際、一般ローダーでは、データ・ファイルが読み込まれ、アップロードするデータが取得されます。多くの場合、データ・ファイルは、事前のダウンロードにより生成されます。ただし、別のソースから生成される場合もあります。データ・ファイルは、対応する使用可能な構成ファイルがない場合には、解析できません。

テキスト・ファイルへのデータベース情報のダウンロード

テキスト・ファイルは人が判読可能で移植性があり、任意のエディタを使用して検査および修正ができます。テキスト・ファイルに書き出されたレコードを識別するには、一般に開発者キーが使用されます。たとえば、「プロファイル」構成ファイルにあるレコードの識別には `PROFILE_OPTION_ID` ではなく `PROFILE_OPTION_NAME` が使用されます。

テキスト・ファイル内情報のデータベースへのアップロード（マージ）

データベース内の行がすでに正しい場合には、その行は変更されません。行が存在し、異なる属性を持つ場合、その行は更新されます。行が存在していない場合は、新しい行が挿入されます。アップロードされたファイルに該当する行が存在しない場合でも、既存の行が削除されることはありません。

このダウンロード機能とアップロード機能により、1つのデータベースで定義されたプロファイル値情報を容易に他のデータベースに伝播できます。この機能は Oracle Applications のシード・データを顧客に配布したり、顧客のプロファイル定義を1次サイトから別のサイトへコピーしたりする際にも使用できます。

プロファイル値データのテキスト・ファイル・バージョンは、大量の編集操作にも使用できます。この作業の場合は、フォームではなくテキスト・エディタを使用した方が効率的です。

FNDLOAD 実行ファイル

一般ローダーは、FNDLOAD という名前のコンカレント・プログラムです。コンカレント実行ファイルに指定できるパラメータは、次のとおりです。

```
FNDLOAD apps/pwd 0 Y mode configfile datafile entity [ param ... ]
```

各パラメータの説明を次に示します。

- <apps/pwd>** フォーム username/password[@connect_string] での APPS スキーマおよびパスワード。connect_string を省略すると、このパラメータは、名前 TWO_TASK を使用している環境から、プラットフォーム固有の方式で指定されます。
- <0 Y>** コンカレント・プログラムのフラグ
- mode** UPLOAD または DOWNLOAD。UPLOAD を指定すると、datafile はデータベースにアップロードされます。DOWNLOAD を指定すると、ローダーによって行が取り込まれ、それが datafile に書き込まれます。
- <configfile>** 使用する構成ファイル（通常拡張子 .lct が付きますが、ローダーによって強制または指定はされません）。
- <datafile>** 書込み対象のデータ・ファイル（通常拡張子 .ldt が付きますが、ローダーによって強制または指定はされません）。データ・ファイルがすでに存在する場合、そのファイルは上書きされません。
- <entity>** ダウンロードまたはアップロードを開始するエンティティ・タイプ。datafile 内のすべてのエンティティ・タイプをアップロードする場合は、エンティティ・タイプとしてダッシュ (-) を指定します。
- <[param]>** アクセス SQL にバインド値（UPLOAD および DOWNLOAD の両方）を指定する場合、0 個以上のパラメータを使用します。各パラメータはフォーム NAME=VALUE に指定されます。NAME には、ロードされるエンティティの属性名と競合しない値を指定してください。

ファイル指定

構成ファイルおよびデータ・ファイルのパラメータは、次の 2 つの方法のうち、いずれかの方法で指定します。

```
@<application_short_name>:[<dir>/.../]file.ext
```

例を次に示します。

```
@fnd:patch/115/loader/fndapp.lct
@po:install/data/poreq.ldt
```

もうひとつの方法として、パラメータは次のように指定できます。

<native_path>

例を次に示します。

```
mydata.ldt
c:¥loader¥config¥cfg102.lct
```

例

ダウンロードの例は次のとおりです。

```
FNDLOAD apps/apps@devdb 0 Y
      DOWNLOAD testcfg.lct out.ldt FND_APPLICATION_TL APPSNAME=FND
```

このコマンドによる実行内容は次のとおりです。

- apps/apps@devd に接続されます。
- 構成ファイル testcfg.lct によってデータがダウンロードされます。
- データ・ファイル out.ldt にデータが書き込まれます。
- APPSNAME パラメータの値が 'FND' と定義された状態で FND_APPLICATION_TL エンティティがダウンロードされます。

アップロードの例は次のとおりです。

```
FNDLOAD apps/apps@custdb 0 Y
      UPLOAD @FND:admin/115/import/fndapp.lct
      @FND:patch/115/import/US/fnd1234.ldt -
```

このコマンドによる実行内容は次のとおりです。

- apps/apps@custdb に接続されます。
- \$FND_TOP/patch/107/loader/fndapp.lct の構成ファイルを使用して、\$FND_TOP/patch/107/loader/fnd1234.ldt にあるデータ・ファイルからデータがアップロードされます。

データ・ファイル全体の内容がアップロードされます。

構成ファイル

一般ローダーの操作は、指定された構成ファイルにより管理されます。構成ファイルには、次のものが含まれます。

- DEFINE ブロック

- DOWNLOAD ブロック
- UPLOAD ブロック

構成ファイルの内容では、データ構造およびデータ・ファイルとデータベース間でデータを移動する際に使用するアクセス方法が指定されます。

DEFINE ブロック

DEFINE ブロックでは、データファイル・レコードの構造が指定されます。定義ブロック書式は、既存の AOL ロードーによってすでに生成されたものと同じです。このセクションの体系は次のとおりです。

```
DEFINE <エンティティ>
    KEY <key_attribute_name> <データ型>
    ...
    (BASE|TRANS|CTX) <attribute_name> <データ型>
    ...
    [DEFINE <child_entity> ...]
END <エンティティ>
```

例

```
DEFINE FND_LOOKUP_TYPE
    KEY          VIEW_APPSNAME/VARCHAR2 (50)
    KEY          LOOKUP_TYPE/VARCHAR2 (30)
    BASE         OWNER/VARCHAR2 (6)
    TRANS        MEANING/VARCHAR2 (80)
    TRANS        DESCRIPTION/VARCHAR2 (240)
    DEFINE FND_LOOKUP_VALUE
        KEY      LOOKUP_CODE/VARCHAR2 (30)
        BASE     END_DATE_ACTIVE/VARCHAR2 (10)
        BASE     OWNER/VARCHAR2 (6)
        TRANS    MEANING/VARCHAR2 (80)
        TRANS    DESCRIPTION/VARCHAR2 (240)
        CTX      TAG/VARCHAR2 (30)
        END FND_LOOKUP_VALUE
    END FND_LOOKUP_TYPE
VIEW_APPSNAME
```

1 つ以上の KEY 属性では、各エンティティの主キーが定義されます。BASE および CTX 属性では、変換は必要ありません。TRANS 属性では、変換は必要です。BASE および CTX 属性は、同様に扱われるので注意してください。つまり、CTX は、BASE のシノニムです。CTX 属性タイプは、ユーザーによる BASE 属性間の区別を可能にする（オプション）ためにより提供されます。たとえば、BASE 属性を取り除いて .ldt ファイルを単純化するという変

換を行う場合、属性の一部を BASE、別の属性を CTX として示すことにより、どの属性を削除するか管理できます。

データ型は、現在サポートされている VARCHAR2 を除いて、標準の Oracle スカラー型です。属性は、構成ファイルにある他のエンティティへの外部キー参照として定義される場合もあります。

外部キー・エンティティは、最上位エンティティであり、そのダウンロード文には、各キー属性の WHERE 句にフィルタ・パラメータが含まれている必要があります。また、パラメータ名は、キー属性名と一致している必要があります。

エンティティ定義は、マスター / デティール関連にネストできます。ネストされたエンティティ定義では、その親エンティティのキー属性が受け継がれ、それは再定義できません。

DOWNLOAD 文

DOWNLOAD 文は、ダウンロードする行を選択する SQL 文です。この文は、可能な場合、他の表へ結合して、生成された ID 番号を開発者キーへと解決します。DOWNLOAD 文には、コマンド行のパラメータ値が代入されるフォーム 'NAME' のバインド値が含まれます。DOWNLOAD 文のフォームは次のとおりです。

```
DOWNLOAD <エンティティ>
    "... から <属性式> を選択 "
```

例

```
DOWNLOAD FND_LOOKUP_TYPE
"select VA.APPLICATION_SHORT_NAME VIEW_APPSNAME,
  LT.LOOKUP_TYPE,
  OA.APPLICATION_SHORT NAME,
  LT.CUSTOMIZATION_LEVEL,
  decode(LT.LAST_UPDATED_BY, 1, 'SEED', 'CUSTOM') OWNER,
  LT.MEANING,
  LT.DESCRPTION
from   FND_LOOKUP_TYPES_VL LT,
       FND_APPLICATION VA,
       FND_APPLICATION OA,
       FND_SECURITY_GROUPS SG
where  VA.APPLICATION_ID = LT.VIEW_APPLICATION_ID
and    OA.APPLICATION_ID = LT.APPLICATION_ID
and    (:VIEW_APPSNAME is null or
        (:VIEW_APPSNAME is not null
        and VA.APPLICATION_SHORT_NAME like :VIEW_APPSNAME))
and    (:LOOKUP_TYPE is null or
        (LOOKUP_TYPE is not null and LT.LOOKUP_TYPE like :LOOKUP_TYPE))
and    SG.SECURITY_GROUP_ID = LT.SECURITY_GROUP_ID
and    ((:SECURITY_GROUP is null and SG.SECURITY_GROUP_KEY = 'STANDARD') or
```

```
(:SECURITY_GROUP is not null
and SG.SECURITY_GROUP_KEY = :SECURITY_GROUP))
order by 1, 2 "
```

親エンティティのキー属性または任意のコマンド行パラメータを参照する可能性のある子エンティティの文をダウンロードします。

UPLOAD 文

UPLOAD 文は、ファイル・データを受け入れ、それをデータベースに適用する SQL 文または PL/SQL 無名ブロックです。文は、データ・ファイルから読み込まれる各レコードごとに 1 度実行されます。文のバインド値は、ファイル・データまたはコマンド行パラメータからの属性により満たされます。

UPLOAD 文のフォームは次のとおりです。

```
UPLOAD <エンティティ>
    "..."
```

例

```
UPLOAD FND_LOOKUP_TYPE
BEGIN
    " begin
        if (:UPLOAD_MODE = 'NLS') then
            fnd_lookup_types_pkg.TRANSLATE_ROW(
                x_lookup_type => :LOOKUP_TYPE,
                x_security_group => :SECURITY_GROUP,
                x_view_application => :VIEW_APPSNAME,
                x_owner => :OWNER,
                x_meaning => :MEANING,
                x_description => :DESCRIPTION);
        else
            fnd_lookup_types_pkg.LOAD_ROW(
                x_lookup_type => :LOOKUP_TYPE,
                x_security_group => :SECURITY_GROUP,
                x_view_application => :VIEW_APPSNAME,
                x_owner => :OWNER,
                x_meaning => :MEANING,
                x_description => :DESCRIPTION);
        end if;
    end; "
```

DOWNLOAD のように、子エンティティの UPLOAD 文では、親レコードの任意の属性が参照される場合があります。

例

```
UPLOAD FND_LOOKUP_VALUE
" begin
    if (:UPLOAD_MODE = 'NLS') then
        fnd_lookup_values_pkg.TRANSLATE_ROW(
            x_lookup_type => :LOOKUP_TYPE,
            x_lookup_code => :LOOKUP_CODE,
            x_security_group => :SECURITY_GROUP,
            x_view_application => :VIEW_APPSNAME,
            x_owner => :OWNER,
            x_meaning => :MEANING,
            x_description => :DESCRIPTION);
    else
        fnd_lookup_values_pkg.LOAD_ROW(
            x_lookup_type => :LOOKUP_TYPE,
            x_lookup_code => :LOOKUP_CODE,
            x_security_group => :SECURITY_GROUP,
            x_view_application => :VIEW_APPSNAME,
            x_owner => :OWNER,
            x_meaning => :MEANING,
            x_description => :DESCRIPTION,
            x_tag => :TAG);
    end if;
end;"
```

データ・ファイル

データ・ファイルは、移植可能なテキスト・ファイルです。上記の構成ファイルを使用したダウンロードにより作成されるデータ・ファイルには、次のようなファイルがあります。

-- エンティティ定義開始 --

```
DEFINE FND_LOOKUP_TYPE
KEY   VIEW_APPSNAME VARCHAR2(50)
KEY   LOOKUP_TYPE VARCHAR2(30)
BASE  OWNER VARCHAR2(6)
TRANS MEANING VARCHAR2(80)
TRANS DESCRIPTION VARCHAR2(240)
DEFINE FND_LOOKUP_VALUE
KEY   LOOKUP_CODE VARCHAR2(30)
BASE  END_DATE_ACTIVE VARCHAR2(10)
BASE  OWNER VARCHAR2(6)
TRANS MEANING VARCHAR2(80)
TRANS DESCRIPTION VARCHAR2(240)
BASE  TAG VARCHAR2(30)
END FND_LOOKUP_VALUE
```

```
END FND_LOOKUP_TYPE

# -- エンティティ定義終了 --

BEGIN FND_LOOKUP_TYPE "FND" "YES_NO"
  OWNER = "SEED"
  MEANING = "Yes or No"

  BEGIN FND_LOOKUP_VALUE Y
    OWNER = "SEED"
    MEANING = "Yes"
  END FND_LOOKUP_VALUE

  BEGIN FND_LOOKUP_VALUE N
    OWNER = "SEED"
    MEANING = "No"
  END FND_LOOKUP_VALUE
END FND_LOOKUP_TYPE
```

Application Object Library 構成ファイル

Oracle Application Object Library では、一般ローダー用の構成ファイルがいくつか提供されています。設定データと共にこの構成ファイルを使用することができます。

これらの構成ファイルは、次のデータを操作します。

- コンカレント・プログラム定義
- 要求グループ
- 機能セキュリティ・データ
- 参照タイプおよび参照値
- プロファイル・オプションおよびプロファイル・オプション値
- フレックスフィールド設定データ
- 添付定義
- メッセージ

コンカレント・プログラム構成ファイル

コンカレント・プログラム構成ファイル `afcpprog.lct` は、コンカレント・プログラム定義をダウンロードしたり、アップロードします。このファイルは、プログラム名とアプリケーション名をパラメータとして使用します。

エンティティ定義は次のとおりです。

```

DEFINE PROGRAM
KEY   CONCURRENT_PROGRAM_NAME    VARCHAR2 (30)
KEY   APPLICATION                  VARCHAR2 (50)
CTX   OWNER                        VARCHAR2 (7)
TRANS USER_CONCURRENT_PROGRAM_NAME VARCHAR2 (240)
BASE  EXEC_APPLICATION             VARCHAR2 (50)
BASE  EXECUTABLE_NAME             VARCHAR2 (30)
BASE  EXECUTION_METHOD_CODE       VARCHAR2 (1)
BASE  ARGUMENT_METHOD_CODE        VARCHAR2 (1)
BASE  QUEUE_CONTROL_FLAG          VARCHAR2 (1)
BASE  QUEUE_METHOD_CODE           VARCHAR2 (1)
BASE  REQUEST_SET_FLAG            VARCHAR2 (1)
BASE  ENABLED_FLAG                VARCHAR2 (1)
BASE  PRINT_FLAG                  VARCHAR2 (1)
BASE  RUN_ALONE_FLAG              VARCHAR2 (1)
BASE  SRS_FLAG                    VARCHAR2 (1)
TRANS DESCRIPTION                 VARCHAR2 (240)
BASE  CLASS_APPLICATION           VARCHAR2 (50)
BASE  CONCURRENT_CLASS_NAME       VARCHAR2 (30)
BASE  EXECUTION_OPTIONS           VARCHAR2 (250)
BASE  SAVE_OUTPUT_FLAG            VARCHAR2 (1)
BASE  REQUIRED_STYLE               VARCHAR2 (1)
BASE  OUTPUT_PRINT_STYLE          VARCHAR2 (30)
BASE  PRINTER_NAME                VARCHAR2 (30)
BASE  MINIMUM_WIDTH               VARCHAR2 (50)
BASE  MINIMUM_LENGTH              VARCHAR2 (50)
BASE  REQUEST_PRIORITY            VARCHAR2 (50)
BASE  ATTRIBUTE_CATEGORY          VARCHAR2 (30)
BASE  ATTRIBUTE1                  VARCHAR2 (150)
BASE  ATTRIBUTE2                  VARCHAR2 (150)
BASE  ATTRIBUTE3                  VARCHAR2 (150)
BASE  ATTRIBUTE4                  VARCHAR2 (150)
BASE  ATTRIBUTE5                  VARCHAR2 (150)
BASE  ATTRIBUTE6                  VARCHAR2 (150)
BASE  ATTRIBUTE7                  VARCHAR2 (150)
BASE  ATTRIBUTE8                  VARCHAR2 (150)
BASE  ATTRIBUTE9                  VARCHAR2 (150)
BASE  ATTRIBUTE10                 VARCHAR2 (150)
BASE  ATTRIBUTE11                 VARCHAR2 (150)
BASE  ATTRIBUTE12                 VARCHAR2 (150)
BASE  ATTRIBUTE13                 VARCHAR2 (150)
BASE  ATTRIBUTE14                 VARCHAR2 (150)
BASE  ATTRIBUTE15                 VARCHAR2 (150)
BASE  OUTPUT_FILE_TYPE            VARCHAR2 (4)
BASE  RESTART                     VARCHAR2 (1)
BASE  NLS_COMPLIANT               VARCHAR2 (1)
BASE  CD_PARAMETER                VARCHAR2 (240)

```

```

BASE    INCREMENT_PROC                VARCHAR2 (61)
BASE    MLS_EXECUTABLE_APPLICATION    VARCHAR2 (50)
BASE    MLS_EXECUTABLE_NAME           VARCHAR2 (50)
BASE    ENABLE_TIME_STATISTICS        VARCHAR2 (1)
BASE    SECURITY_GROUP_NAME           NUMBER
BASE    RESOURCE_CONSUMER_GROUP       VARCHAR2 (30)
BASE    ROLLBACK_SEGMENT              VARCHAR2 (30)
BASE    OPTIMIZER_MODE                VARCHAR2 (30)
END PROGRAM

```

要求グループ構成ファイル

ファイル afcpreqg.lct を使用して、要求グループ・データをロードします。

この構成ファイルは、要求グループ名とアプリケーションをパラメータとして使用します。

エンティティ定義は次のとおりです。

```

DEFINE REQUEST_GROUP
KEY    REQUEST_GROUP_NAME    VARCHAR2 (30)
KEY    APPLICATION_SHORT_NAME VARCHAR2 (50)
CTX    OWNER                 VARCHAR2 (7)
TRANS  DESCRIPTION           VARCHAR2 (800)
BASE    REQUEST_GROUP_CODE   VARCHAR2 (30)
END REQUEST_GROUP

```

機能セキュリティ構成ファイル

このファイルは、変換データのアップロードだけに使用します。

非変換データのダウンロードおよびアップロードには、FNDSLOAD 実行ファイルを使用します。参照：「[機能セキュリティ・ローダー](#)」(C-19) 機能セキュリティ・ローダー。

エンティティ定義は次のとおりです。

```

DEFINE FORM
KEY    APPLICATION_SHORT_NAME VARCHAR2 (50)
KEY    FORM_NAME              VARCHAR2 (30)
TRANS  USER_FORM_NAME        VARCHAR2 (80)
TRANS  DESCRIPTION           VARCHAR2 (240)
CTX    OWNER                 VARCHAR2 (7)
END FORM

DEFINE FUNCTION
KEY    FUNCTION_NAME          VARCHAR2 (30)
BASE    FORM REFERENCES FORM
BASE    TYPE                  VARCHAR2 (30)

```

```

BASE  PARAMETERS                VARCHAR2(2000)
BASE  WEB_HOST_NAME              VARCHAR2(80)
BASE  WEB_AGENT_NAME             VARCHAR2(80)
BASE  WEB_HTML_CALL              VARCHAR2(240)
BASE  WEB_ENCRYPT_PARAMETERS     VARCHAR2(1)
BASE  WEB_SECURED                VARCHAR2(1)
BASE  WEB_ICON                   VARCHAR2(30)
TRANS USER_FUNCTION_NAME        VARCHAR2(80)
TRANS DESCRIPTION                VARCHAR2(240)
CTX   OWNER                      VARCHAR2(7)
END FUNCTION

DEFINE MENU
KEY   MENU_NAME                  VARCHAR2(30)
TRANS USER_MENU_NAME            VARCHAR2(80)
TRANS DESCRIPTION                VARCHAR2(240)
CTX   OWNER                      VARCHAR2(7)

DEFINE ENTRY
TRANS PROMPT                     VARCHAR2(60)
TRANS DESCRIPTION                VARCHAR2(240)
CTX   SUBMENU REFERENCES MENU
CTX   FUNCTION REFERENCES FUNCTION
CTX   OWNER                      VARCHAR2(7)
END ENTRY
END MENU

```

参照構成ファイル

ファイル afcpregg.lct を使用して、参照タイプおよび参照値をロードします。
エンティティ定義は次のとおりです。

```

DEFINE FND_LOOKUP_TYPE
KEY   VIEW_APPSNAME              VARCHAR2(50)
KEY   LOOKUP_TYPE                VARCHAR2(30)
CTX   APPLICATION_SHORT_NAME     VARCHAR2(50)
BASE  CUSTOMIZATION_LEVEL        VARCHAR2(1)
CTX   OWNER                      VARCHAR2(6)
TRANS MEANING                    VARCHAR2(80)
TRANS DESCRIPTION                VARCHAR2(240)

DEFINE FND_LOOKUP_VALUE
KEY   LOOKUP_CODE                VARCHAR2(30)
BASE  ENABLED_FLAG               VARCHAR2(1)
BASE  START_DATE_ACTIVE          VARCHAR2(10)
BASE  END_DATE_ACTIVE            VARCHAR2(10)

```

```

BASE TERRITORY_CODE          VARCHAR2 (2)
BASE TAG                     VARCHAR2 (30)
BASE ATTRIBUTE_CATEGORY     VARCHAR2 (30)
BASE ATTRIBUTE1             VARCHAR2 (150)
BASE ATTRIBUTE2             VARCHAR2 (150)
BASE ATTRIBUTE3             VARCHAR2 (150)
BASE ATTRIBUTE4             VARCHAR2 (150)
BASE ATTRIBUTE5             VARCHAR2 (150)
BASE ATTRIBUTE6             VARCHAR2 (150)
BASE ATTRIBUTE7             VARCHAR2 (150)
BASE ATTRIBUTE8             VARCHAR2 (150)
BASE ATTRIBUTE9             VARCHAR2 (150)
BASE ATTRIBUTE10            VARCHAR2 (150)
BASE ATTRIBUTE11            VARCHAR2 (150)
BASE ATTRIBUTE12            VARCHAR2 (150)
BASE ATTRIBUTE13            VARCHAR2 (150)
BASE ATTRIBUTE14            VARCHAR2 (150)
BASE ATTRIBUTE15            VARCHAR2 (150)
CTX OWNER                   VARCHAR2 (6)
TRANS MEANING               VARCHAR2 (80)
TRANS DESCRIPTION          VARCHAR2 (240)
END FND_LOOKUP_VALUE
END FND_LOOKUP_TYPE

```

プロファイル・オプションおよびプロファイル値構成ファイル

ファイル afscprof.lct を使用して、プロファイル・オプションおよびプロファイル値をロードします。

エンティティ定義は次のとおりです。

```

DEFINE PROFILE
KEY PROFILE_NAME            VARCHAR2 (80)
CTX OWNER                   VARCHAR2 (7)
CTX APPLICATION_SHORT_NAME  VARCHAR2 (50)
TRANS USER_PROFILE_OPTION_NAME VARCHAR2 (240)
TRANS DESCRIPTION          VARCHAR2 (240)
BASE USER_CHANGEABLE_FLAG  VARCHAR2 (1)
BASE USER_VISIBLE_FLAG     VARCHAR2 (1)
BASE READ_ALLOWED_FLAG     VARCHAR2 (1)
BASE WRITE_ALLOWED_FLAG    VARCHAR2 (1)
BASE SITE_ENABLED_FLAG     VARCHAR2 (1)
BASE SITE_UPDATE_ALLOWED_FLAG VARCHAR2 (1)
BASE APP_ENABLED_FLAG      VARCHAR2 (1)
BASE APP_UPDATE_ALLOWED_FLAG VARCHAR2 (1)
BASE RESP_ENABLED_FLAG     VARCHAR2 (1)
BASE RESP_UPDATE_ALLOWED_FLAG VARCHAR2 (1)

```

```

BASE    USER_ENABLED_FLAG                VARCHAR2 (1)
BASE    USER_UPDATE_ALLOWED_FLAG         VARCHAR2 (1)
BASE    START_DATE_ACTIVE                 VARCHAR2 (11)
BASE    END_DATE_ACTIVE                   VARCHAR2 (11)
BASE    SQL_VALIDATION                     VARCHAR2 (2000)

DEFINE FND_PROFILE_OPTION_VALUES
KEY      LEVEL                            VARCHAR2 (50)
KEY      LEVEL_VALUE                       VARCHAR2 (100)
KEY      LEVEL_VALUE_APP                   VARCHAR2 (50)
CTX      OWNER                             VARCHAR2 (7)
BASE     PROFILE_OPTION_VALUE              VARCHAR2 (240)

END FND_PROFILE_OPTION_VALUES
END PROFILE

```

フレックスフィールド設定データ構成ファイル

ファイル `afcpreqg.lct` を使用して、フレックスフィールド・データをロードします。

警告：フレックスフィールド構成ファイルを使用してダウンロードするデータ・ファイルは、変更しないでください。フレックスフィールド・データが破壊される危険があります。Oracle Applications は、これらのデータ・ファイルへの変更を一切サポートしていません。

この構成ファイルには、次のエンティティが含まれます。

- 値セット
- 付加フレックスフィールド
- キー・フレックスフィールド定義
- セキュリティ・ルール
- 積上グループ
- 値セット値

添付設定データ構成ファイル

ファイル `afattach.lct` を使用して、添付設定データをロードします。

エンティティ定義は次のとおりです。

```

DEFINE FND_ATTACHMENT_FUNCTIONS
    KEY    FUNCTION_NAME                VARCHAR2 (30)
    KEY    FUNCTION_TYPE                VARCHAR2 (1)
    KEY    APP_SHORT_NAME              VARCHAR2 (50)
    CTX    OWNER                      VARCHAR2 (7)
    BASE    SESSION_CONTEXT_FIELD      VARCHAR2 (61)
    BASE    ENABLED_FLAG                VARCHAR2 (1)

DEFINE FND_ATTACHMENT_BLOCKS
    KEY    BLOCK_NAME                  VARCHAR2 (30)
    CTX    OWNER                      VARCHAR2 (7)
    BASE    QUERY_FLAG                 VARCHAR2 (1)
    BASE    SECURITY_TYPE              VARCHAR2 (50)
    BASE    ORG_CONTEXT_FIELD          VARCHAR2 (61)
    BASE    SET_OF_BOOKS_CONTEXT_FIELD VARCHAR2 (61)
    BASE    BUSINESS_UNIT_CONTEXT_FIELD VARCHAR2 (61)
    BASE    CONTEXT1_FIELD             VARCHAR2 (61)
    BASE    CONTEXT2_FIELD            VARCHAR2 (61)
    BASE    CONTEXT3_FIELD            VARCHAR2 (61)

DEFINE FND_ATTACHMENT_BLK_ENTITIES
    KEY    BLK_ENTITY                  REFERENCES FND_DOCUMENT_ENTITIES
    BASE    DISPLAY_METHOD             VARCHAR2 (1)
    BASE    INCLUDE_IN_INDICATOR_FLAG VARCHAR2 (1)
    CTX    OWNER                      VARCHAR2 (7)
    BASE    PK1_FIELD                 VARCHAR2 (61)
    BASE    PK2_FIELD                 VARCHAR2 (61)
    BASE    PK3_FIELD                 VARCHAR2 (61)
    BASE    PK4_FIELD                 VARCHAR2 (61)
    BASE    PK5_FIELD                 VARCHAR2 (61)
    BASE    SQL_STATEMENT              VARCHAR2 (2000)
    BASE    INDICATOR_IN_VIEW_FLAG     VARCHAR2 (1)
    BASE    QUERY_PERMISSION_TYPE      VARCHAR2 (1)
    BASE    INSERT_PERMISSION_TYPE     VARCHAR2 (1)
    BASE    UPDATE_PERMISSION_TYPE     VARCHAR2 (1)
    BASE    DELETE_PERMISSION_TYPE     VARCHAR2 (1)
    BASE    CONDITION_FIELD            VARCHAR2 (61)
    BASE    CONDITION_OPERATOR          VARCHAR2 (50)
    BASE    CONDITION_VALUE1           VARCHAR2 (100)
    BASE    CONDITION_VALUE2           VARCHAR2 (100)
END FND_ATTACHMENT_BLK_ENTITIES
END FND_ATTACHMENT_BLOCKS

DEFINE FND_DOC_CATEGORY_USAGES
    KEY CATEGORY_USAGE REFERENCES
                                FND_DOCUMENT_CATEGORIES
    BASE ENABLED_FLAG            VARCHAR2 (1)

```

```

        CTX    OWNER                                VARCHAR2 (7)
    END FND_DOC_CATEGORY_USAGES
END FND_ATTACHMENT_FUNCTIONS

DEFINE FND_DOCUMENT_ENTITIES
    KEY    DATA_OBJECT_CODE                        VARCHAR2 (30)
    BASE   APP_SHORT_NAME                          VARCHAR2 (50)
    BASE   TABLE_NAME                            VARCHAR2 (30)
    BASE   ENTITY_NAME                            VARCHAR2 (40)
    CTX    OWNER                                VARCHAR2 (7)
    BASE   PK1_COLUMN                            VARCHAR2 (30)
    BASE   PK2_COLUMN                            VARCHAR2 (30)
    BASE   PK3_COLUMN                            VARCHAR2 (30)
    BASE   PK4_COLUMN                            VARCHAR2 (30)
    BASE   PK5_COLUMN                            VARCHAR2 (30)
    TRANS  USER_ENTITY_NAME                      VARCHAR2 (240)
    TRANS  USER_ENTITY_PROMPT                    VARCHAR2 (40)
END FND_DOCUMENT_ENTITIES

DEFINE FND_DOCUMENT_CATEGORIES
    KEY    CATEGORY_NAME                          VARCHAR2 (30)
    BASE   APP_SHORT_NAME                          VARCHAR2 (50)
    CTX    OWNER                                VARCHAR2 (7)
    BASE   START_DATE_ACTIVE                      VARCHAR2 (11)
    BASE   END_DATE_ACTIVE                       VARCHAR2 (11)
    BASE   ATTRIBUTE_CATEGORY                     VARCHAR2 (30)
    BASE   ATTRIBUTE1                            VARCHAR2 (150)
    BASE   ATTRIBUTE2                            VARCHAR2 (150)
    BASE   ATTRIBUTE3                            VARCHAR2 (150)
    BASE   ATTRIBUTE4                            VARCHAR2 (150)
    BASE   ATTRIBUTE5                            VARCHAR2 (150)
    BASE   ATTRIBUTE6                            VARCHAR2 (150)
    BASE   ATTRIBUTE7                            VARCHAR2 (150)
    BASE   ATTRIBUTE8                            VARCHAR2 (150)
    BASE   ATTRIBUTE9                            VARCHAR2 (150)
    BASE   ATTRIBUTE10                           VARCHAR2 (150)
    BASE   ATTRIBUTE11                           VARCHAR2 (150)
    BASE   ATTRIBUTE12                           VARCHAR2 (150)
    BASE   ATTRIBUTE13                           VARCHAR2 (150)
    BASE   ATTRIBUTE14                           VARCHAR2 (150)
    BASE   ATTRIBUTE15                           VARCHAR2 (150)
    BASE   DEFAULT_DATATYPE_ID                    VARCHAR2 (50)
    BASE   APP_SOURCE_VERSION                     VARCHAR2 (255)
    TRANS  USER_NAME                              VARCHAR2 (255)
END FND_DOCUMENT_CATEGORIES

DEFINE FND_DOCUMENT_DATATYPES

```

```

KEY    DATATYPE_ID          VARCHAR2 (50)
KEY    NAME                  VARCHAR2 (30)
CTX    OWNER                 VARCHAR2 (7)
BASE   START_DATE_ACTIVE    VARCHAR2 (11)
BASE   END_DATE_ACTIVE      VARCHAR2 (11)
TRANS  USER_NAME            VARCHAR2 (30)
END FND_DOCUMENT_DATATYPES

```

メッセージ構成ファイル

ファイル afmdmsg.lct を使用して、データベース内のメッセージをアップロードしたり、ダウンロードします。

一般ローダーと afmdmsg.lct は、データベース間でのメッセージ転送でのみ使用してください。バイナリ・ランタイム・ファイルおよび可読テキスト・ファイルへのメッセージの移動には、メッセージ・ディクショナリ生成プログラムを使用します。参照：[「メッセージ・ディクショナリ生成プログラム」](#) (C-17)。

エンティティ定義は次のとおりです。

注意：ダウンロードしている言語を変更するには、ローダーを実行する前に環境変数 NLS_LANG を設定します。

```

DEFINE FND_NEW_MESSAGES
KEY    APPLICATION_SHORT_NAME  VARCHAR2 (50)
KEY    MESSAGE_NAME            VARCHAR2 (30)
CTX    OWNER                   VARCHAR2 (7)
CTX    MESSAGE_NUMBER          VARCHAR2 (50)
TRANS  MESSAGE_TEXT            VARCHAR2 (2000)
CTX    DESCRIPTION              VARCHAR2 (240)
CTX    TYPE                     VARCHAR2 (30)
CTX    MAX_LENGTH               NUMBER
END FND_NEW_MESSAGES

```

メッセージ・ディクショナリ生成プログラム

メッセージ・ディクショナリ生成プログラム (FNDMDGEN) は、Oracle Applications メッセージ・ディクショナリ・メッセージ用のデータベースからバイナリ・ランタイム・ファイルを生成するコンカレント・プログラムです。次の項では、メッセージ・ディクショナリ生成プログラムの操作を説明します。

メッセージ・ディクショナリの使用とメッセージ作成の詳細は、『Oracle Applications 開発者ガイド』を参照してください。

注意: メッセージ・テキスト・ファイルのアップロードやデータベースへのダウンロードには、一般ローダーおよび対応する構成ファイルを使用します。

メッセージ・リポジトリ

メッセージ情報は2つの異なるリポジトリに格納されます。これらのリポジトリには、それぞれ独自の形式があり、特定のニーズを満たす機能を備えています。各メッセージ・リポジトリについて、格納するメッセージ属性も含めて次に説明します。

データベース

データベース内の FND_NEW_MESSAGES 表は、すべての言語に対して Oracle Applications メッセージをすべて格納します。データベース・メッセージを直接使用するのは、ストアード・プロシージャのメッセージ・ディクショナリ API だけです。データベース・メッセージ・データは、「メッセージ」フォームを使用して編集できます。

データベース属性: APPLICATION, LANGUAGE, NAME, NUMBER, TEXT, DESCRIPTION

ランタイム

1つのランタイム・バイナリ・ファイルは、1つのアプリケーションおよび1つの言語についてのメッセージを格納します。ファイルは、個々のメッセージをその名前 (NAME) で高速に検索できるように最適化されています。

ランタイム・ファイルは次の場所にあります。

<APPL_TOP>/<APPLMSG/<LANGUAGE>.msb

ここで、<APPL_TOP> はアプリケーションのベースパス、APPLMSG は環境変数でその通常値は「mesg」、<LANGUAGE> は NLS 言語コード (「US」、「F」など) です。代表的なメッセージ・ファイルは \$FND_TOP/mesg/US.msb のようになります。

ランタイム属性: NAME, NUMBER, TEXT

使用方法

プログラム引数を指定しないで (つまり FNDMDGEN dbuser/dbpassword 0 Y) 「メッセージ・ディクショナリ生成プログラム」を起動したときに表示されるヘルプは、次のとおりです。

FNDMDGEN <Oracle ID/ パスワード> 0 Y <言語コード名>
[アプリケーション短縮名] [モード] [ファイル名]

モードは次のとおりです。

DB_TO_RUNTIME データベースからランタイム・ファイルへ (.msb)

注意: リリース 11i では、旧リリースと異なり、モード DB_TO_RUNTIME だけがサポートされます。

ワイルド・カード

値「ALL」を渡すことで、<言語コード名>または[アプリケーション短縮名]のどちらかにワイルドカードを使用できます。ワイルドカードの使用方法を次に示します。

データベースから	メッセージは FND_NEW_MESSAGES 表から検索されます。ワイルド・カードを指定すると、データベース内のすべてのメッセージが対象となります。
ランタイム・ファイルへ	ワイルド・カードの場合は、言語およびアプリケーションの個々の組合せについて別個のランタイム・ファイルが作成されます。

機能セキュリティ・ローダー

「機能セキュリティ・ローダー (FNDSLOAD)」は、データベースとテキスト・ファイル表現との間で Oracle Applications 機能セキュリティ情報を移動できるコンカレント・プログラムです。次の各節では、「機能セキュリティ・ローダー」の操作について説明します。

「機能セキュリティ・ローダー」を使用して、アップグレード前のファイルのダウンロードとアップグレード後のファイルのアップロードを行うことにより、アップグレード時にカスタム職責とメニュー情報を保存します。

サポートされている操作

「機能セキュリティ・ローダー」を使用すると、データベース（ランタイム操作用に使用）とテキスト・ファイル表現（データを配布用に使用可能）との間で、機能セキュリティ・データを移動できます。特に、次のような操作が可能となります。

データベース情報をテキスト・ファイルにダウンロードする

テキスト・ファイルは人が判読可能で移植性があり、任意のエディタを使用して検査および修正ができます。テキスト・ファイルに書き出されたレコードを識別するには、一般に「開発者キー」が使用されます。つまり、レコードの識別には、FUNCTION_ID でなく、FUNCTION_NAME が使用されます。

テキスト・ファイル内の情報をデータベースにアップロード（マージ）する

データベースにすでに正しい行が存在している場合には、その行は変更されません。行が存在するがその属性が異なる場合には、その行は更新されます。行が存在していない場合は、新しい行が挿入されます。アップロードされたファイルに該当する行がない場合でも、既存の行が削除されることはありません。

これらのダウンロード機能およびアップロード機能により、あるデータベース内に定義されている機能セキュリティ情報を別のデータベースに簡単に反映させることができます。これは、Oracle Applications シード・データを顧客に配布したり、顧客の機能セキュリティ定義を主要サイトから他のサイトにコピーしたりする場合に便利です。

機能セキュリティ・データのテキスト・ファイル・バージョンは、大量の編集操作の場合にも便利です。大量編集の場合は、フォームを使用するよりテキスト・エディタを使用する方が効率よく操作できます。

使用方法

「機能セキュリティ・ローダー」は次の引数を指定します。

```
FNDSLOAD <ユーザー名 / パスワード> 0 Y
  [{ LOCAL <filepath> } |
   { <appsname> <subdir>/.../<subdir>/<fname>.ext<} ]
  [UPLOAD | {DOWNLOAD <menuname> ... <menuname>}]
```

この場合

<username/ password> は、APPLSYS アカウントです。

<appsname> は、アプリケーション短縮名または「LOCAL」です。

<fname> は、読み込みまたは書き込みが行われるファイルです。

<menuname> は、ダウンロードするメニュー名のリストです。

ファイルの位置は、引数 **<appsname>** および **<fname>** により決定します。**< appsname>** は、ファイルが特定のアプリケーションのインストール / インポート・ディレクトリにあるかどうか、またはローカル・ディレクトリにあるかどうかを示します。

アプリケーション名	ファイル	場所
<APP>	< ファイル >	\$(APP_TOP)/install/import/< ファイル >.slt
LOCAL	< ファイル >	< ファイル >

表 1

例

FNDSLOAD applsys/fnd 0 Y FND install/import/sysadmin.slt UPLOAD

ファイル "\$FND_TOP/install/import/sysadmin.slt" を参照します。

FNDSLOAD applsys/fnd 0 Y LOCAL sysadmin.slt UPLOAD

ファイル「sysadmin.slt」を参照します。

UPLOAD および DOWNLOAD

ローダーは、アップロード・モードまたはダウンロード・モードで動作します。UPLOAD モードでは、ファイル全体がデータベースに読み込まれ、マージされます（メニューの既存の項目との間に矛盾が発生すると、ファイル内の項目リストが優先され、既存項目は削除されます）。

DOWNLOAD モードでは、指定された各 <menuname> をルートとするメニュー・ツリーがデータベースから読み込まれ、各メニューが参照するすべての機能およびフォームとともに、ファイルに書き込まれます。<menuname> は実メニュー名で、「ユーザー・メニュー名」でないことに注意してください。

FNDSLOAD applsys/fnd 0 Y FND sysadmin.slt DOWNLOAD FND_NAVIGATE4.0

「FND_NAVIGATE4.0」の下すべてのメニューをファイル sysadmin.slt にダウンロードします。

FNDSLOAD applsys/fnd 0 Y FND sysadmin.slt UPLOAD

ファイル sysadmin.slt 内のすべてをデータベースにアップロードします。

機能セキュリティ・ファイル形式

機能セキュリティ情報は、各フォームおよび機能、メニュー用の一連の「レコード」を書き出すことにより、テキスト・ファイル形式で格納されます。ファイルは、どのテキスト・エディタでも表示および編集可能なプレーン・テキストです。このファイル形式を意識する必要があるのは、ファイルを編集する場合のみです。

このファイルは、テキスト行の集合から構成され、それぞれのファイルの長さは 1024 バイト未満でなければなりません（書き込まれた行の長さはいずれも 80 バイト以下です）。

行は、コメント行、つまりファイル内のデータに影響を与えない行にすることができます。コメント行は、「#」で始まり、改行で終わります。

これはコメント行です。

ファイル内の情報は、一連のトークンで構成され、トークンとトークンの間はスペースで区切られています。トークン自体にスペースが含まれていることがありますが、その場合にはスペースを二重引用符で囲みます。トークンには、「エスケープ」文字（円記号）が先頭に

付いていない限り、印刷不能文字を入れることはできません。あらかじめ定義されているエスケープ文字には次のものがあります。

文字列	内容
¥e	エスケープ
¥n	改行
¥r	改行
¥b	バックスペース
¥v	垂直タブ
¥f	用紙送り
¥"	二重引用符
¥¥	円記号
表 2	

その他のエスケープ文字は、「¥007」のように、8 進数の前に円記号が付いた形になります。行が長すぎて 80 文字以内に収まらないと、必要数の「継続」行に分割されます。1 つの円記号で終了する行は、次の行に継続するものとみなされます。この場合、円記号に続く改行は無視され、スペース区切り記号とはみなされません。このような行の後に続く行は「分割」行の継続部分とみなされます。この行自体が円記号で終了し、さらに継続することを示す場合もあります。連続行の場合は、先頭の「#」はコメントを示す記号とは見なされず、連続するデータ項目の一部と見なされます。

ファイルは、次に示す 3 行で始まります。

```
LANGUAGE = "AMERICAN"
CODESET = "WE8ISO8859P1"
TRANSLATED = "Y"
```

この 3 行は、ファイルの NLS_LANG 言語とコードセット、および変換された列の値の現在の変換ステータスを示します。

次に、一連の形式のオブジェクト定義があります。

```
BEGIN <オブジェクト・タイプ> <オブジェクト識別子>
  <属性> = <値>
  ...
BEGIN <サブオブジェクト・タイプ> <オブジェクト識別子>
  ...
END <サブオブジェクト・タイプ>
...
```

END <オブジェクト・タイプ>

3つの主要オブジェクト・タイプ、および1つのサブオブジェクト・タイプがあります。メイン・オブジェクトはFORMおよびFUNCTION、MENUで、サブオブジェクトはENTRYです。各オブジェクト・タイプにはなんらかの識別子、およびそれに続く一連の属性値があります。属性値が指定されていないければ、NULLが属性値と見なされます。

FORM レコード

フォーム情報は次の形式で書き出されます。

```
BEGIN FORM <app_short_name> <form_name>
  USER_FORM_NAME = "<user_form_name: 80 文字翻訳済み>"
  DESCRIPTION = "< 摘要 : 240 文字翻訳済み>"
END FORM
```

フォームはapp_short_name および form_name により識別されます。FORM レコードは、ダウンロード中のメニューにより参照されている各フォームについて1つずつ書き出されます。

例：

```
BEGIN FORM SQLGL GLXJEENT
  USER_FORM_NAME = " 仕訳 / 引当データの入力 "
  DESCRIPTION = " 仕訳 / 引当データの入力 "
END FORM
```

FUNCTION レコード

機能情報は次の形式で書き出されます。

```
BEGIN FUNCTION <function_name>
  FORM = <app_short_name> <form_name>
  TYPE = <タイプ :30 文字>
  PARAMETERS = "<パラメータ : 240 文字>"
  USER_FUNCTION_NAME = "<user_function_name: 80 char    translated>"
  DESCRIPTION = "< 摘要 : 240 文字翻訳済み>"
END FUNCTION
```

機能は開発者 function_name で識別されます。FUNCTION レコードは、ダウンロード中のメニューにより参照されている各機能について1つずつ書き出されます。

例：

```
BEGIN FUNCTION SQLGL_GLXJEENT_A
  FORM = SQLGL GLXJEENT
  TYPE = FORM
  PARAMETERS = "actual_flag=¥"A¥"                                HELP_TARGET=¥"GLXJEENT_A¥"
```

```
USER_FUNCTION_NAME = " 仕訳データの入力 "  
DESCRIPTION = " 仕訳データの入力 "  
END FUNCTION
```

MENU レコード

メニュー情報は、実際は、複数の ENTRY レコードが入っている（囲み込まれている）1 個の MENU レコードで構成された複合レコードです。メニュー情報の形式は次のとおりです。

```
MENU <menu_name>  
  USER_MENU_NAME = "<user_menu_name: 80 文字翻訳済み >"  
  DESCRIPTION = "< 摘要 : 240 文字翻訳済み >"  
  BEGIN ENTRY  
    PROMPT = "< プロンプト : 30 文字翻訳済み >"  
    DESCRIPTION = "< 摘要 : 240 文字翻訳済み >"  
    SUBMENU = <sub_menu_name>  
    FUNCTION = <function_name>  
  END ENTRY  
  ... さらにレコードの入力 ...  
END MENU
```

例：

```
MENU GL_SUPERVISOR_GUI  
  USER_MENU_NAME = GL_SUPERVISOR_GUI  
  BEGIN ENTRY  
    PROMPT = 仕訳  
    DESCRIPTION = " 実際および引当仕訳の入力と転記 "  
    SUBMENU = GL_SU_JOURNAL_GUI  
  END ENTRY  
  BEGIN ENTRY  
    PROMPT = 予算  
    DESCRIPTION = " 予算の定義と入力 "  
    SUBMENU = GL_SU_BUDGET_GUI  
  END ENTRY  
  BEGIN ENTRY  
    PROMPT = " レポートの実行 "  
    DESCRIPTION = " レポートの実行 "  
    FUNCTION = SQLGL_FNDRSRUN_GLMODE  
  END ENTRY  
END MENU
```

メニューは、開発者メニュー名により識別されます。メニュー項目自体には識別子はありませんが、定義順に並んでいると仮定されるので、順序番号が暗黙の識別子となります。指定された各メニューは、それが参照するすべてのサブメニューとともに（再帰的に）ダウンロードされます。

機能セキュリティ・ファイルの例

```
#
# $Header: SYS00027170.htm 115.0 1999/11/12 23:31:27 nlo ship $
#
LANGUAGE = "AMERICAN"
CODESET = "WE8ISO8859P1"
TRANSLATED = "Y"
BEGIN FORM FND_FNDCPQCR
    USER_FORM_NAME = " 要求の表示 "
    DESCRIPTION = " 個別および明示的なコンカレント要求を表示するためのユーザー・フォーム "
    END FORM
BEGIN FORM FND_FNDPOMSV
    USER_FORM_NAME = " 個別プロファイル・オプション値の更新 "
    DESCRIPTION = " 個別プロファイル・オプションを設定するためのユーザー・フォーム "
    END FORM
BEGIN FORM FND_FNDRSRUN
    USER_FORM_NAME = " レポートの実行 "
    END FORM
BEGIN FORM FND_FNDCPDIA
    USER_FORM_NAME = " ヘルプの要求 "
    DESCRIPTION = " 要求診断フォーム (関連: レポートの表示) "
    END FORM
BEGIN FORM FND_FNDRSSET
    USER_FORM_NAME = " レポート・セットの管理 "
    DESCRIPTION = " 全レポート・セットを管理するためのシステム管理者フォーム "
    END FORM
BEGIN FUNCTION FND_FNDCPQCR
    FORM = FND_FNDCPQCR
    TYPE = FORM
    USER_FUNCTION_NAME = " コンカレント要求: 全表示 "
    DESCRIPTION = " コンカレント要求: 表示フォーム "
    END FUNCTION
BEGIN FUNCTION FND_FNDPOMSV
    FORM = FND_FNDPOMSV
    TYPE = FORM
    USER_FUNCTION_NAME = " プロファイル・ユーザー値 "
    DESCRIPTION = " プロファイル・ユーザー値フォーム "
    END FUNCTION
BEGIN FUNCTION FND_FNDRSRUN
    FORM = FND_FNDRSRUN
    TYPE = FORM
    USER_FUNCTION_NAME = " 要求: 発行 "
    DESCRIPTION = " 要求: フォームの実行 "
    END FUNCTION
BEGIN FUNCTION FND_FNDCPDIA_VIEW
    FORM = FND_FNDCPDIA
```

```

TYPE = FORM
PARAMETERS = "MODE=¥"VIEW¥"
USER_FUNCTION_NAME = " コンカレント要求：完了済の表示 "
DESCRIPTION = " コンカレント要求診断フォーム：要求の表示モード "
END FUNCTION
BEGIN FUNCTION FND_FNDRSSET_USER
  FORM = FND_FNDRSSET
  TYPE = FORM
  PARAMETERS = "MODE=¥"USER¥"
  USER_FUNCTION_NAME = " 要求セット（ユーザー・モード） "
  DESCRIPTION = " 要求セット：ユーザー・モード・フォーム "
END FUNCTION
BEGIN MENU GL_SU_MANAGER_GUI
  USER_MENU_NAME = GL_SU_MANAGER_GUI
  DESCRIPTION = "Called by GL_SUPERUSER4.0, GL_USER4.0, GL_SUPERVISOR4.0, GL_BY
UDGETUSER4.0, GL_BUDGETSUPER4.0"
  BEGIN ENTRY
    PROMPT = Requests
    DESCRIPTION = " コンカレント・マネージャ要求の検討 "
    FUNCTION = FND_FNDCPQCR
  END ENTRY
  BEGIN ENTRY
    PROMPT = Profile
    DESCRIPTION = " 個別プロファイル・オプションの検討 "
    FUNCTION = FND_FNDPOMSV
  END ENTRY
  BEGIN ENTRY
    PROMPT = レポート
    DESCRIPTION = " グループと標準レポートの要求 "
    SUBMENU = FND_REPORT4.0
  END ENTRY
END MENU
BEGIN MENU FND_REPORT4.0
  USER_MENU_NAME = "FND_REPORT 4.0"
  DESCRIPTION = " フォーム 4.0 の標準レポート発行およびレポートの表示 "
  BEGIN ENTRY
    PROMPT = 実行
    DESCRIPTION = " 要求の発行 "
    FUNCTION = FND_FNDRSRUN
  END ENTRY
  BEGIN ENTRY
    PROMPT = 表示
    DESCRIPTION = " 完了済要求の表示 "
    FUNCTION = FND_FNDCPDIA_VIEW
  END ENTRY
  BEGIN ENTRY
    PROMPT = セット

```

```
DESCRIPTION = " 標準要求セットの定義 "  
FUNCTION = FND_FNDRSSET_USER  
END ENTRY  
END MENU
```


OracleApplications システム管理者の 文字モード・フォームおよび対応する GUI ウィンドウ

次の表に、システム管理の文字モード・フォームと、GUI 製品で同じ機能を持つウィンドウまたはプロセスを示します。

システム管理者職責を使用すれば、ほとんどのウィンドウにアクセスできます。次に示すナビゲーション・パスはすべて、システム管理者職責があることを前提としています。

特に注記のない限り、GUI ウィンドウまたはプロセスの詳細は、『Oracle Applications システム管理者ガイド』（リリース 11i）を参照してください。

キャラクタ・モード・フォームおよび メニュー・パス	GUI ウィンドウまたはプロセスおよびナビゲータ・ パス
コンカレント・マネージャの管理 ¥N: 業務 C: コンカレント M: マネー ジャ A: 管理	「コンカレント・マネージャの管理」ウィンドウ 参照: コンカレント・マネージャの管理 ナビゲータ: 「コンカレント」> 「マネージャ」> 「管理」
要求セットの管理 ¥N: 業務 C: コンカレント S: セット ¥N: 業務 R: レポート S: セット	「要求セット」ウィンドウ 参照: 要求セット (『Oracle Applications ユーザー ズ・ガイド』) ナビゲータ: 「コンカレント」> 「セット」または ナビゲータ: 「レポート」> 「セット」
付加フレックスフィールド・セキュリ ティ・ルールの割当 ¥N: 業務 S: セキュリティ R: 職責 F: フ レックスフィールド D: 付加 A: 割当	「セキュリティ・ルールの割当」ウィンドウ 参照: セキュリティ・ルールの割当 (『Oracle Applications フレックスフィールド・ガイド』) ナビゲータ: 「セキュリティ」> 「職責」> 「値セッ ト」> 「割当」 「検索」ウィンドウでは「付加フレックスフィール ド」を選択します。

キャラクタ・モード・フォームおよびメニュー・パス	GUI ウィンドウまたはプロセスおよびナビゲータ・パス
機能パラメータの割当 ¥N: 業務 A: アプリ F: フレックス フィールド F: フレックスビルダー A: 割当	フレックスビルダーの機能は、Oracle Workflow を使用した勘定科目ジェネレータ機能で代用できます。 参照: 勘定科目ジェネレータ (『Oracle Applications フレックスフィールド・ガイド』)
キー・フレックスフィールド・セキュリティ・ルールの割当 ¥N: 業務 S: セキュリティ R: 職責 F: フレックスフィールド K: キー A: 割当	「セキュリティ・ルールの割当」ウィンドウ 参照: セキュリティ・ルールの割当 (『Oracle Applications フレックスフィールド・ガイド』) ナビゲータ: 「セキュリティ」 > 「職責」 > 「値セット」 > 「割当」 「検索」ウィンドウでは「キー・フレックスフィールド」を選択します。
パラメータ・セキュリティ・ルールの割当 ¥N: 業務 S: セキュリティ R: 職責 R: レポート R: ルール A: 割当	「セキュリティ・ルールの割当」ウィンドウ 参照: セキュリティ・ルールの割当 (『Oracle Applications フレックスフィールド・ガイド』) ナビゲータ: 「セキュリティ」 > 「職責」 > 「値セット」 > 「割当」 「検索」ウィンドウでは「コンカレント・プログラム」を選択します。
プリンタ・ドライバの割当 ¥N: 業務 I: 導入 P: プリンタ D: ドライバ A: 割当	「プリンタ・ドライバ」ウィンドウ 参照: プリンタ・ドライバ ナビゲータ: 「導入」 > 「プリンタ」 > 「ドライバ」
セキュリティ・ルールの割当 ¥N: 業務 S: セキュリティ R: 職責 V: 値セット A: 割当	「セキュリティ・ルールの割当」ウィンドウ 参照: セキュリティ・ルールの割当 (『Oracle Applications フレックスフィールド・ガイド』) ナビゲータ: 「セキュリティ」 > 「職責」 > 「値セット」 > 「割当」
端末セキュリティの割当 ¥N: 業務 S: セキュリティ R: 職責 T: 端末 A: 割当	GUI では廃止
アプリケーション・ユーザーの定義 N: 業務 S: セキュリティ U: ユーザー D: 定義	「ユーザー」ウィンドウ 参照: ユーザー ナビゲータ: 「セキュリティ」 > 「ユーザー」 > 「定義」
組合せ特殊化ルールの定義 ¥N: 業務 C: コンカレント M: マネージャ R: ルール	「組合せ特殊化ルール」ウィンドウ 参照: 組合せ特殊化ルール ナビゲータ: 「コンカレント」 > 「マネージャ」 > 「ルール」

キャラクタ・モード・フォームおよびメニュー・パス	GUI ウィンドウまたはプロセスおよびナビゲータ・パス
コンカレント・マネージャの定義 ¥N: 業務: C: コンカレント M: マネージャ D: 定義	「コンカレント・マネージャ」ウィンドウ 参照: コンカレント・マネージャ ナビゲータ: 「コンカレント」 > 「マネージャ」 > 「定義」
コンカレント・プログラムの定義 ¥N: 業務: C: コンカレント P: プログラム D: 定義	「コンカレント・プログラム」ウィンドウ 参照: コンカレント・プログラム ナビゲータ: 「コンカレント」 > 「プログラム」 > 「定義」
コンカレント・プログラム実行ファイルの定義 ¥N: 業務: C: コンカレント P: プログラム E: 実行ファイル	「コンカレント・プログラム実行ファイル」ウィンドウ 参照: コンカレント・プログラム実行ファイル ナビゲータ: 「コンカレント」 > 「プログラム」 > 「実行ファイル」
コンカレント要求タイプの定義 ¥N: 業務: C: コンカレント P: プログラム T: タイプ	「コンカレント要求タイプ」ウィンドウ 参照: コンカレント要求タイプ ナビゲータ: 「コンカレント」 > 「プログラム」 > 「タイプ」
相互検証ルールの定義 ¥N: 業務: A: アプリケーション F: フレックスフィールド K: キー C: 相互検証	「相互検証」ウィンドウ 参照: 相互検証ルール (『Oracle Applications フレックスフィールド・ガイド』) ナビゲータ: 「アプリケーション」 > 「フレックスフィールド」 > 「キー」 > 「相互検証」
通貨の定義 ¥N: 業務: A: アプリケーション C: 通貨	「通貨」ウィンドウ 参照: 通貨の定義 (『Oracle Applications General Ledger ユーザーズ・ガイド』) ナビゲータ: 「アプリケーション」 > 「通貨」
データ・グループの定義 N: 業務 S: セキュリティ O: Oracle D: データ G: グループ	「データ・グループ」ウィンドウ 参照: データ・グループ ナビゲータ: 「セキュリティ」 > 「Oracle」 > 「データ・グループ」
付加フレックスフィールド・セキュリティ・ルールの定義 ¥N: 業務 S: セキュリティ R: 職責 F: フレックスフィールド D: 付加 D: 定義	「セキュリティ・ルールの定義」ウィンドウ 参照: セキュリティ・ルールの割当 (『Oracle Applications フレックスフィールド・ガイド』) ナビゲータ: 「セキュリティ」 > 「職責」 > 「値セット」 > 「定義」 「検索」ウィンドウでは「付加フレックスフィールド」を選択します。

キャラクタ・モード・フォームおよびメニュー・パス	GUI ウィンドウまたはプロセスおよびナビゲータ・パス
付加フレックスフィールド・セグメントの定義 ¥N: 業務: A: アプリケーション F: フレックスフィールド D: 付加 S: セグメント	「付加フレックスフィールド・セグメント」ウィンドウ 参照: 付加フレックスフィールド・セグメント (『Oracle Applications フレックスフィールド・ガイド』) ナビゲータ: 「アプリケーション」 > 「フレックスフィールド」 > 「付加」 > 「セグメント」
文書カテゴリの定義 ¥N: 業務: A: アプリケーション D: 文書 C: カテゴリ	「文書カテゴリ」ウィンドウ 参照: 文書カテゴリ ナビゲータ: 「アプリケーション」 > 「文書」 > 「カテゴリ」
文書連番の定義 ¥N: 業務: A: アプリケーション D: 文書 D: 定義	「文書連番」ウィンドウ 参照: 文書連番 ナビゲータ: 「アプリケーション」 > 「文書」 > 「定義」
フレックスフィールド・パラメータの定義 ¥N: 業務: A: アプリケーション F: フレックスフィールド F: フレックスビルダー D: 定義	フレックスビルダーの機能は、Oracle Workflow を使用した勘定科目ジェネレータ機能で代用できます。 参照: 勘定科目ジェネレータ (『Oracle Applications フレックスフィールド・ガイド』)
ヘルプ・テキストの定義 ¥N: 業務: A: アプリケーション T: テキスト	GUI では廃止
キー・フレックスフィールド・セキュリティ・ルールの定義 ¥N: 業務 S: セキュリティ R: 職責 F: フレックスフィールド K: キー D: 定義	「セキュリティ・ルールの定義」ウィンドウ 参照: セキュリティ・ルールの割当 (『Oracle Applications フレックスフィールド・ガイド』) ナビゲータ: 「セキュリティ」 > 「職責」 > 「値セット」 > 「定義」 「検索」ウィンドウでは「キー・フレックスフィールド」を選択します。
キー・フレックスフィールド・セグメントの定義 ¥N: 業務: A: アプリケーション F: フレックスフィールド K: キー S: セグメント	キー・フレックスフィールド・セグメント 参照: キー・フレックスフィールド・セグメント (『Oracle Applications フレックスフィールド・ガイド』) ナビゲータ: 「アプリケーション」 > 「フレックスフィールド」 > 「キー」 > 「セグメント」
キー・セグメント値の定義 ¥N: 業務: A: アプリケーション F: フレックスフィールド K: キー V: 値	「セグメント値」ウィンドウ 参照: セグメント値 (『Oracle Applications フレックスフィールド・ガイド』) ナビゲータ: 「アプリケーション」 > 「フレックスフィールド」 > 「キー」 > 「値」

キャラクタ・モード・フォームおよびメニュー・パス	GUI ウィンドウまたはプロセスおよびナビゲータ・パス
論理データベースの定義 ¥N: 業務 C: コンカレント D: データベース	リリース 11 では廃止 論理データベースはコンカレント衝突ドメインにかわりました。 参照: コンカレント衝突ドメイン ナビゲータ: 「コンカレント」 > 「衝突ドメイン」
メニューの定義 ¥N: 業務 : A: アプリケーション M: メニュー	「メニュー」ウィンドウ 参照: メニュー ナビゲータ: 「アプリケーション」 > 「メニュー」
パラメータ・セキュリティ・ルールの定義 ¥N: 業務 S: セキュリティ R: 職責 R: レポート R: ルール D: 定義	「セキュリティ・ルールの定義」ウィンドウ 参照: セキュリティ・ルールの割当 (『Oracle Applications フレックスフィールド・ガイド』) ナビゲータ: 「セキュリティ」 > 「職責」 > 「値セット」 > 「定義」 「検索」ウィンドウでは「コンカレント・プログラム」を選択します。
パラメータ値の定義 ¥N: 業務 A: アプリケーション V: 検証 R: レポート	値セット値の定義 参照: セグメント値 (『Oracle Applications フレックスフィールド・ガイド』) ナビゲータ: 「アプリケーション」 > 「検証」 > 「値」 「検索」ウィンドウでは「値セット」を選択します。
印刷形式の定義 ¥N: 業務 I: 導入 P: プリンタ S: 形式	「印刷形式」ウィンドウ 参照: 印刷形式 ナビゲータ: 「導入」 > 「プリンタ」 > 「形式」
プリンタ・ドライバの定義 ¥N: 業務 I: 導入 P: プリンタ D: ドライバ	「プリンタ・ドライバ」ウィンドウ 参照: プリンタ・ドライバ ナビゲータ: 「導入」 > 「プリンタ」 > 「ドライバ」
プリンタ・タイプの定義 ¥N: 業務 I: 導入 P: プリンタ T: タイプ	「プリンタ・タイプ」ウィンドウ 参照: プリンタ・タイプ ナビゲータ: 「導入」 > 「プリンタ」 > 「タイプ」
レポート・グループの定義 N: 業務 : S: セキュリティ R: 職責 R: レポート G: グループ	「要求グループ」ウィンドウ 参照: 要求グループ ナビゲータ: 「セキュリティ」 > 「職責」 > 「要求」
職責の定義 N: 業務 : S: セキュリティ R: 職責 D: 定義	「職責」ウィンドウ 参照: 職責 ナビゲータ: 「セキュリティ」 > 「職責」 > 「定義」

キャラクタ・モード・フォームおよびメニュー・パス	GUI ウィンドウまたはプロセスおよびナビゲータ・パス
ロールアップ・グループの定義 ¥N: 業務: A: アプリケーション F: フレックスフィールド K: キー G: グループ	「ロールアップ・グループ」ウィンドウ 参照: ロールアップ・グループ (『Oracle Applications フレックスフィールド・ガイド』) ナビゲータ: 「アプリケーション」 > 「フレックスフィールド」 > 「キー」 > 「グループ」
セグメント値の定義 ¥N: 業務: A: アプリケーション V: 検証 V: 値	「セグメント値」ウィンドウ 参照: セグメント値 (『Oracle Applications フレックスフィールド・ガイド』) ナビゲータ: 「アプリケーション」 > 「検証」 > 「値」 または ナビゲータ: 「アプリケーション」 > 「フレックスフィールド」 > 「キー」 > 「値」
短縮別名の定義 ¥N: 業務: A: アプリケーション F: フレックスフィールド K: キー A: 別名	「短縮別名」ウィンドウ 参照: 短縮別名 (『Oracle Applications フレックスフィールド・ガイド』) ナビゲータ: 「アプリケーション」 > 「フレックスフィールド」 > 「キー」 > 「別名」
端末グループの定義 ¥N: 業務 S: セキュリティ R: 職責 T: 端末 G: グループ	GUI では廃止
値セットの定義 ¥N: 業務: A: アプリケーション V: 検証 S: セット	「値セット」ウィンドウ 参照: 値セット (『Oracle Applications フレックスフィールド・ガイド』) ナビゲータ: 「アプリケーション」 > 「検証」 > 「セット」
稼動シフトの定義 ¥N: 業務: C: コンカレント M: マネージャ W: 稼動シフト	「稼動シフト」ウィンドウ 参照: 稼動シフト ナビゲータ: 「コンカレント」 > 「マネージャ」 > 「稼動シフト」
ズームの定義 ¥N: 業務: A: アプリケーション Z: ズーム	GUI では廃止
フレックスビルダー・テスト画面 ¥N: 業務 A: アプリケーション F: フレックスビルダー T: テスト	フレックスビルダーの機能は、Oracle Workflow を使用した勘定科目ジェネレータ機能で代用できます。 参照: 勘定科目ジェネレータ (『Oracle Applications フレックスフィールド・ガイド』)

キャラクタ・モード・フォームおよびメニュー・パス	GUI ウィンドウまたはプロセスおよびナビゲータ・パス
アプリケーション・ユーザーのモニター ¥N: 業務:S: セキュリティ U: ユーザー M: モニター	「ユーザーのモニター」ウィンドウ 参照: ユーザーのモニター ナビゲータ: 「セキュリティ」> 「ユーザー」> 「モニター」
アプリケーションの登録 ¥N: 業務:A: アプリケーション R: 登録	「アプリケーション」ウィンドウ 参照: アプリケーション ナビゲータ: 「アプリケーション」> 「登録」
ノードの登録 ¥N: 業務:I: 導入 N: ノード	「ノード」ウィンドウ 参照: ノード ナビゲータ: 「導入」> 「ノード」
Oracle ID の登録 ¥N: 業務:S: セキュリティ O:Oracle R: 登録	「Oracle ユーザー」ウィンドウ 参照: Oracle ユーザー ナビゲータ: 「セキュリティ」> 「Oracle」> 「登録」
プリンタの登録 ¥N: 業務:I: 導入 P: プリンタ R: 登録	プリンタ 参照: プリンタ ナビゲータ: 「導入」> 「プリンタ」> 「登録」
端末の登録 ¥N: 業務:I: 導入 T: 端末	GUI では廃止
監査データの検索 ¥N: 業務:S: セキュリティ A: 監査証跡	GUI では廃止
レポートの実行 ¥N: 業務 R: レポート R: 実行	「要求の発行」ウィンドウ 参照: 要求の発行 (『Oracle Applications ユーザーズ・ガイド』) ナビゲータ: 「レポート」> 「実行」
導入情報の更新 ¥N: 業務:I: 導入 I: 情報	GUI では廃止
個別プロファイル・オプションの更新 ¥N: 業務:P: プロファイル P: 個別	「個別プロファイル値」ウィンドウ 参照: パーソナル・プロファイル値 (『Oracle Applications フレックスフィールド・ガイド』) ナビゲータ: 「プロファイル」> 「個別」
システム・プロファイル・オプションの更新 ¥N: 業務:P: プロファイル S: システム	「システム・プロファイル値」ウィンドウ 参照: システム・プロファイル値 ナビゲータ: 「プロファイル」> 「システム」

キャラクタ・モード・フォームおよびメニュー・パス	GUI ウィンドウまたはプロセスおよびナビゲータ・パス
<p>コンカレント要求ログの表示 ¥N: 業務 C: コンカレント R: 要求</p>	<p>「要求」ウィンドウ 参照: 要求の表示 (『Oracle Applications ユーザーズ・ガイド』) ナビゲータ: 「コンカレント」 > 「要求」 次を選択してください。 [出力の表示] ボタン - 「要求出力」の表示用 [ログの表示] ボタン - 「要求ログ」の表示用 「メニュー」を使用して、次の順に選択してください。 「ツール」 > 「マネージャ・ログ」 - 「マネージャ・ログ」の表示用</p>
<p>レポートの表示 ¥N: 業務 R: レポート V: 表示</p>	<p>「要求」ウィンドウ 参照: 要求の表示 (『Oracle Applications ユーザーズ・ガイド』) ナビゲータ: 「コンカレント」 > 「要求」 次を選択してください。 [出力の表示] ボタン - 「要求出力」の表示用 [ログの表示] ボタン - 「要求ログ」の表示用</p>
表 3	

Oracle Applications System Administrator の設定

この節には、Oracle Applications 製品を使用する前に完了する必要がある各作業の概要が記載されています。

Oracle Applications Implementation Wizard

複数の Oracle Applications 製品を実装する場合、Oracle Applications Implementation Wizard を使用することにより設定処理を調整できます。Implementation Wizard では、インストールしたアプリケーションの設定手順が示されます。この手順に従うことにより、製品間の導入上の依存関係を満たし、余分な設定手順を省くことができます。Implementation Wizard によって、（同時進行で作業する複数のグループごとに）個別に完了できる手順を識別できます。これにより、必要な導入プロセスを最も効率的に管理することができます。

Implementation Wizard をリソース・センターとして使用すると、設定手順をグラフィック表示したり、設定活動のオンライン・ヘルプを読んだり、該当の設定ウィンドウをオープンしたりできます。また、Implementation Wizard を使用して各手順のコメントを記録することにより、詳細の参照や確認のために実装内容を文書化できます。

詳細は、『Implementation Wizard Documentation』を参照してください。

設定チェックリスト

Oracle Applications を設定するには、Oracle System Administrator にログインしてから次の各ステップを最後まで実行します。

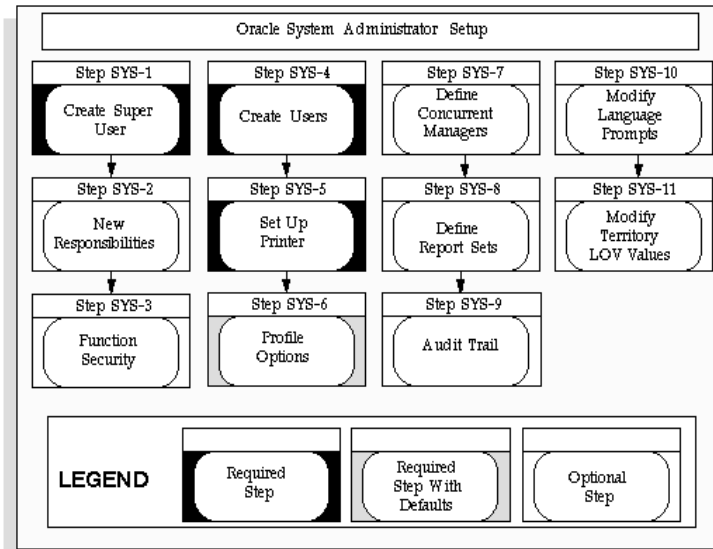
- ☐ 「ステップ 1 設定の完了段階での Oracle Applications ユーザーの作成」 (E-3)
- ☐ 「ステップ 2 新規職責の作成（オプション）」 (E-3)
- ☐ 「ステップ 3 機能セキュリティの導入（オプション）」 (E-4)

- ☐ 「ステップ 4 追加ユーザーの作成」 (E-4)
- ☐ 「ステップ 5 プリンタの設定」 (E-5)
- ☐ 「ステップ 6 サイト・レベルおよびアプリケーション・レベルのプロファイル・オプションの指定」 (E-5)
- ☐ 「ステップ 7 コンカレント・マネージャの定義 (オプション)」 (E-6)
- ☐ 「ステップ 8 要求セットの定義 (オプション)」 (E-7)
- ☐ 「ステップ 9 監査証跡の設定 (オプション)」 (E-7)
- ☐ 「ステップ 10 言語プロンプトの変更 (オプション)」 (E-7)
- ☐ 「ステップ 11 地域 LOV 値の変更 (オプション)」 (E-8)

設定フローチャート

Oracle System Administration にはさまざまな設定方法があり、オプション設定ステップについては、それに対応する機能を使用する準備ができるまで延期することもできます。しかし、次のフローチャートで示される順序で設定することをお勧めします。

Figure 1 - 25 Oracle System Administrator Setup



設定ステップ

ステップ 1 設定の完了段階での Oracle Applications ユーザーの作成

アプリケーションの設定を完了するには、Oracle Applications ユーザーを作成する必要があります。すべてのアプリケーションの設定に対して 1 人のユーザーを作成するか、または各製品または製品グループ別に定義するかを選択できます。

Oracle Applications にログオンするには、Oracle Applications アイコンの上でダブルクリックします。

サインオン・ウィンドウが表示されます。システム管理者職責にアクセスするには、ユーザー名 SYSADMIN とパスワード SYSADMIN を入力して「接続」を選択します。

注意: コマンド、ユーザー名またはパスワード、あるいはそのすべては、使用する組織に適合するようにアプリケーションのインストール時にコンサルタントが変更している場合があります。その場合は、コンサルタントから正しいログオン方法の指示を受けてください。

ユーザーを定義するには、ナビゲーション・リストから「セキュリティ」>「ユーザー」>「定義」を選択して「ユーザー」ウィンドウにナビゲートします。「ユーザー名」フィールドにユーザー名（たとえば、INSTALL）を入力して、そのユーザーのパスワードを選択します。この新規ユーザーが最初にサインオンするときに、Oracle Applications からパスワードの変更が求められます。このユーザーが Oracle Applications の設定のために登録したことを忘れないように、摘要を入力してください。

このステップで作成する各ユーザーについて、現在日付の終了日を設定します。この設定により、以後これらのユーザー名はアプリケーションにアクセスできなくなります。

新規ユーザーに職責を割り当てます。新規ユーザーには、インストールする各アプリケーションの全機能職責を割り当ててください。

関連項目

- 「「ユーザー」ウィンドウ」(1-9)

ステップ 2 新規職責の作成（オプション）

Oracle Applications の職責とは 1 つの権限レベルを指します。職責により、あるユーザーが使用できるアプリケーションの機能、およびユーザーが実行できる要求とコンカレント・プログラムが決定されます。また、要求とコンカレント・プログラムがアクセスできるアプリケーション・データも決定されます。Oracle Applications では、提供されている事前定義職責のセットを使用できます。また、提供されているセットが必要な職責と一致しない場合には、独自の職責を定義できます。

各職責は、データ・グループ、要求グループおよびメニューと関係づけられます。データ・グループでは、アプリケーションと ORACLE ユーザー名の対が定義されます。ORACLE ユーザー名は、職責がアクセスできるデータベース表と表特権を決めます。要求グループを使用すれば、この職責を持つユーザーは要求、要求セットまたはコンカレント・プログラムを「要求の発行」フォームから実行できます。事前定義メニューを選択します。メニューからは、機能および機能のメニューの階層的な配列を利用して、アプリケーション機能にアクセスすることができます。

新規職責を定義するには、「職責」ウィンドウを使用します。次に、新規職責をユーザーに割り当てるには、「ユーザー」ウィンドウを使用します。

関連項目

- [「ユーザー」ウィンドウ](#) (1-9)
- [「職責」ウィンドウ](#) (1-6)

ステップ 3 機能セキュリティの導入（オプション）

機能セキュリティとは、各アプリケーションの機能へのユーザー・アクセスを制御する機構です。

ある職責の機能を制限するためにメニューまたは機能を除外するには、「職責」フォームを使用します。

または

「メニュー」フォームを使用することにより、ある職責で使用する機能を含む新規メニューを作成します。

関連項目

- [「メニュー」ウィンドウ](#) (1-30)
- [「職責」ウィンドウ](#) (1-6)

ステップ 4 追加ユーザーの作成

追加アプリケーション・ユーザーを作成するには、ステップ 1 で説明した手順を使用してください。新規ユーザーを定義するには、1 つ以上の職責と 1 つのパスワードを割り当てます。このパスワードは、初回ログイン後にそのユーザーが変更します。「職責」フィールドで LOV を使用すると、指定する各アプリケーションの標準職責のリストを取得することができます。1 人のユーザーに複数の職責を割り当てられます。

関連項目

- [「ユーザー」ウィンドウ](#) (1-9)

ステップ 5 プリンタの設定

プリンタの設定方法は、「[プリンタの設定](#)」(5-10) のページを参照してください。Oracle Applications に添付されていないプリンタ・タイプをサイトで使用する場合は、各プリンタを定義してから、使用するオペレーティング・システムの規定に従ってその名称を登録してください。

定義する各カスタム・プリンタ・タイプまたは特殊印刷形式ごとに、「プリンタ・ドライバ」フォームを使用してプリンタ・タイプの各印刷形式で使用するプリンタ・ドライバを割り当てます。

関連項目

- 「[プリンタ・タイプの定義およびプリンタの登録](#)」(5-8)
- 「[プリンタ・ドライバ](#)」(5-9)
- 「[プリンタおよび印刷の概要](#)」(5-1)
- 「[印刷形式](#)」(5-9)

使用するプリンタ・オペレーティング・システム名の検索方法の詳細は『Oracle Applications インストレーション・マニュアル』の「印刷」の節を参照してください。

ステップ 6 サイト・レベルおよびアプリケーション・レベルのプロファイル・オプションの指定

「システム・プロファイル値」フォームにナビゲートします（「プロファイル」>「システム」）。「検索」ウィンドウで、「表示」レベルとして「サイト」と「アプリケーション」をチェックします。「アプリケーション」フィールドに「システム管理」を、「プロファイル」フィールドに「サイト名」を入力します。Oracle Applications は、サイト名として 'Not Specified' を表示します。**Not Specified** をユーザーのサイト名に変更します。

これ以外のオプションを指定するには、検索ウィンドウに戻って「プロファイル」フィールドを消去してから、「検索」を選択します。

さらに、AutoInstall により設定された他のプロファイル・オプションの値を検査して、どれを変更するかを決定する必要があります。サイト・レベル・プロファイル・オプションは、他のレベルでそれを上書きするまで、システムのデフォルトになります。

各「システム管理」プロファイル・オプションの摘要は「[Oracle Application Object Library のプロファイル・オプション](#)」(A-1) を参照してください。

重要なプロファイル・オプションには次のものが含まれます。

- アプリケーション Web エージェント
- ゲートウェイ・ユーザー ID

-
- TWO_TASK
 - サイト名
 - 添付ファイル・ディレクトリ
 - ヘルプ・システム・ベース URL
 - ヘルプ・システム・ルート
 - TCF ホスト
 - TCF ポート

注意: このリストは、「システム管理」プロファイル・オプションの完全なリストではありません。すべての「システム管理」プロファイル・オプションを検索するには、「検索」ウィンドウを使用します。

関連項目

- [「システム・プロファイル値ウィンドウ」 \(9-6\)](#)
- [「ユーザー・プロファイル・オプションの設定」 \(9-2\)](#)
- [「ユーザー・プロファイル設定の概要」 \(9-1\)](#)

ステップ7 コンカレント・マネージャの定義（オプション）

Oracle Applications のコンカレント・プロセスは、複数のタスクを同時に実行できる機能です。Oracle Applications コンカレント・プロセスを使用すると、ユーザーはオンライン操作と並行してデータ依存機能を長時間にわたって実行できます。コンカレント・マネージャはコンカレント・プロセス用のコンポーネントです。長時間を要するタスクをモニタリングしながら、コンピュータ・システムを停止させることなくタスクを実行します。

Oracle Applications は、すべての要求を実行できる単一の標準コンカレント・マネージャを自動的にインストールします。コンカレント・マネージャのフレキシブルな機能を利用して、システムのスループットを管理できます。

コンカレント・マネージャは必要に応じていくつでも定義できます。しかし、各コンカレント・マネージャは余分なメモリーを消費することに注意してください。

コンカレント・マネージャをそれぞれ特殊化すると、すべての要求、特定ユーザーの発行した要求、特定アプリケーションの発行した要求、その他の制約別に（またはそれらの制約の組合せ別に）実行させることが可能です。

並列コンカレント・プロセスをクラスタ環境、大規模並列環境または同種ネットワーク環境で使用する場合には、使用するノードを登録してから、コンカレント・マネージャを第1および第2ノードに割り当ててください。コンカレント・マネージャを使用可能なすべての

ノードに拡張することにより、ハードウェア資源を最大限に利用してコンカレント・プロセスを実行できます。

新規のコンカレント・マネージャを定義するには、「コンカレント・マネージャの定義」フォームを使用します。

関連項目

- 「コンカレント・プロセスの概要」 (4-1)
- 「マネージャおよびマネージャの稼働シフトの定義」 (4-22)

その他の関連項目

『Oracle Applications インストレーション・マニュアル』、メモリー所要量

『Oracle Applications インストレーション・マニュアル』、インストールまたはアップグレードの準備

ステップ 8 要求セットの定義（オプション）

要求セットとは、1 回の要求で発行するレポートまたはプログラムのグループです。要求セットを定義および保守するには、「要求セット」フォームを使用します。

ユーザーが独自のレポート・セットを定義することもできます。

ステップ 9 監査証跡の設定（オプション）

アプリケーション・ユーザーが行ったデータ変更の追跡が必要な場合には、該当する表の監査証跡を設定してください。

使用するサイトの監査証跡の定義には、監査グループの定義が含まれます。監査グループとは、変更の追跡対象となる表および列のグループです。次に、監査インストールを定義することにより、監査証跡の監査対象となる Oracle ID を指示します。最後に、「監査証跡更新レポート」を実行すると、監査証跡定義が有効になります。

関連項目

- 「監査グループ」ウィンドウ (2-32)
- 「導入の監査」ウィンドウ (2-31)
- 「ユーザー監査およびデータ監査の概要」 (2-1)

ステップ 10 言語プロンプトの変更（オプション）

「換算」ウィンドウに表示されるフィールド名を変更する場合は、「言語」ウィンドウで該当言語の「摘要」値を変更してください。

関連項目

- [「言語」ウィンドウ](#) (7-13)

ステップ 11 地域 LOV 値の変更（オプション）

LOV に表示される地域値を変更する場合は、「地域」ウィンドウで該当地域の「摘要」値を変更してください。

関連項目

- [「地域」ウィンドウ](#) (7-13)

TCF サーバーの管理

TCF (Thin Client Framework) サーバーは、Oracle Applications ユーザー・インタフェースの特定の Java コンポーネントと中間層およびデータベース層との通信を実現するための中間層プロセスです。このプロセスを使用する Oracle Applications の機能には、AK オブジェクト・ナビゲータ、機能セキュリティ・メニュー・ビューア、BOM Routings と MRPPegging、WIP Shop Floor Control Board などがあります。

TCF サーバー・プロセスは、本稼動インストールでは常に実行されている必要があります。

