

PeopleSoft®

EnterpriseOne 8.9
CNCインプリメンテーション
PeopleBook

2003年9月

PeopleSoft EnterpriseOne 8.9
CNCインプリメンテーション PeopleBook
SKU AC89JCI0309

Copyright 2003 PeopleSoft, Inc. All rights reserved.

本書に含まれるすべての内容は、PeopleSoft, Inc. (以下、「ピープルソフト」) が財産権を有する機密情報です。すべての内容は著作権法により保護されており、該当するピープルソフトとの機密保持契約の対象となります。本書のいかなる部分も、ピープルソフトの書面による事前の許可なく複製、コピー、転載することを禁じます。これには電子媒体、画像、複写物、その他あらゆる記録手段を含みます。

本書の内容は予告なく変更される場合があります。ピープルソフトは本書の内容の正確性について責任を負いません。本書で見つかった誤りは書面にてピープルソフトまでお知らせください。

本書に記載されているソフトウェアは著作権によって保護されており、このソフトウェアの使用許諾契約書に基づいてのみ使用が許諾されます。この使用許諾契約書には、開示情報を含むソフトウェアと本書の使用条件が記載されていますのでよくお読みください。

PeopleSoft、PeopleTools、PS/nVision、PeopleCode、PeopleBooks、PeopleTalk、Vantiveはピープルソフトの登録商標です。Pure Internet Architecture、Intelligent Context Manager、The Real-Time Enterpriseはピープルソフトの商標です。その他すべての会社名および製品名は、それぞれの所有者の商標である場合があります。ここに含まれている内容は予告なく変更されることがあります。

オープンソースの開示

この製品には、Apache Software Foundation (<http://www.apache.org/>) が開発したソフトウェアが含まれています。Copyright (c) 1999-2000 The Apache Software Foundation. All rights reserved. このソフトウェアは「現状のまま」提供されるものとし、特定の目的に対する商品性および適格性の黙示保証を含む、いかなる明示または黙示の保証も行いません。Apache Software Foundationおよびその供給業者は、損害の発生原因を問わず、責任の根拠が契約、厳格責任、不法行為(過失および故意を含む)のいずれであっても、また損害の可能性が事前に知らされていたとしても、このソフトウェアの使用によって生じたいかなる直接的損害、間接的損害、付随的損害、特別損害、懲罰的損害、結果的損害に関しても一切責任を負いません。これらの損害には、商品またはサービスの代用調達、使用機会の喪失、データまたは利益の損失、事業の中断が含まれますがこれらに限らないものとします。

ピープルソフトは、いかなるオープンソースまたはシェアウェアのソフトウェアおよび文書の使用または頒布に関しても一切責任を負わず、これらのソフトウェアや文書の使用によって生じたいかなる損害についても保証しません。

目次

CNC インプリメンテーション	1
インプリメンテーション・チームの役割.....	1
テクノロジー・チームの役割.....	2
開発チームの役割.....	2
ファンクショナル・チームの役割.....	2
システム・インテグレーション・チームの役割.....	3
パス・コードの設定	4
パス・コードの理解.....	4
インストール時のパス・コードの理解.....	4
ランタイムにおけるパス・コードの理解.....	4
開発時のパス・コードの理解.....	5
パス・コードの処理.....	5
新規パス・コードの追加.....	5
パス・コード定義の作成.....	7
新しいパス・コードに対するテーブルの修正.....	7
オブジェクト構成マネージャ(OCM)	13
オブジェクト構成マネージャ(OCM)の理解.....	13
例:アプリケーションのロジック要求.....	14
オブジェクト構成マネージャ (OCM)の処理.....	15
オブジェクトのマッピング.....	16
汎用テキストのマッピング.....	23
オブジェクト構成マネージャ・レポートの実行.....	25
OCM レポートの処理オプションの設定.....	25
〈オブジェクト構成マッピングの比較〉レポートの実行.....	29
〈オブジェクト構成の一括更新〉レポートの実行.....	30
〈オブジェクト構成の削除〉レポートの実行.....	31
〈オブジェクト構成の削除〉レポートの実行.....	31
〈OCM 検証〉レポートの実行.....	32
〈オブジェクト構成システム・テーブルの更新〉レポートの実行.....	34
〈ビジネス関数用 OCM レコードの作成〉レポートの実行.....	36
〈旧日付別ジョブ・マスター削除〉レポートの実行.....	36
〈OCM カテゴリの更新/削除〉レポートの実行.....	37
サーバー上のアプリケーション・ロジックの分割.....	39

2 階層ネットワークの一般的なトラフィック	39
3 階層ネットワークの分割トラフィック	40
マスター・ビジネス関数(MBF)の処理	40
分散アプリケーション・ロジックのサーバー処理	41
ロジックを分散しない場合	43
ロジックの分散に関する問題	44
ビジネス関数処理の問題	44
サーバーへの接続の失敗	45
ビジネス関数のロード失敗	46
ビジネス関数実行時の失敗	48
サーバー上のキャッシュのリセット	48
環境のセットアップ	49
環境の理解	49
環境定義の保管場所	49
環境の処理	50
環境の追加	51
環境のコピー	51
新しい環境の新規パス・コードへのコピー	52
環境の削除	55
環境ディレクタ	56
異なる環境からの環境ディレクタの実行	61
共有データ・ソースと環境データ・ソース	61
データベース構成と権限	61
データ・ソースのカスタマイズ	62
データ・クラス	62
データ・クラスの作成	62
環境別データ・ソースのテンプレート	62
テーブル・データ・クラス	63
データのロード	63
OCM の作成方法	63
新規作成またはコピー	63
新しい環境を作成した場合の TBLE と GT	64
環境をコピーした場合の TBLE と GT	64
新規作成または既存の環境をコピーする場合	64
データ・ソース	66
データ・ソースの理解	66
データベース・データ・ソースの理解	66
ロジック・データ・ソースの理解	67
必須データ・ソース・タイプ	67
データ・ソースの計画	72
データ・ソースの処理	73

データ・ソース命名規則	73
データ・ソースの追加または修正	74
データ・ソース・レポートの実行	82
データ・ソース・レポートの処理オプションの設定	82
データ・ソース・マスターの比較レポートの実行	84
データ・ソース・マスター・レポートの実行	85
ミドルウェア	87
JDENet 通信ミドルウェアの理解	87
JDENet 通信ミドルウェア	88
JDEBase データベース・ミドルウェアの理解	90
処理モード	91
ダイレクト接続処理	91
オフライン処理	91
バッチオブワン処理	92
ゼロクライアント処理	93
ダイレクト接続処理について	93
データとロジックの分散に関する推奨事項	93
ダイレクト接続処理の設定	94
ダイレクト接続環境のオブジェクト・マッピング	94
オフライン処理について	95
オフライン処理の設定	95
オフライン処理の管理	97
オフライン処理に必要なテクニカル・データ・テーブル	103
オフライン処理に必要なビジネス・データ・テーブル	104
標準的なカスタマイズ構成	110
標準的な構成	110
環境の理解	110
標準データ・ソースの理解	119
システム構成一覧表	122

CNC インプリメンテーション

『コンフィギュレーション・プランニングおよびセットアップ』スイートは、テクニカル・コンサルタント、システム管理者、ネットワーク管理者、サーバー管理者を対象に設計されています。これらの各ガイドでは、このソフトウェアの初期インストールと、標準データ・ソース、パス・コード、および環境の定義を完了していることを前提としています。各ガイドでは、初期インストール後に各種設定を変更、追加する方法を掲載しています。

『コンフィギュレーション・プランニングおよびセットアップ』スイートには次のガイドが含まれています。

- 『CNC インプリメンテーション』ガイド
主にテクニカル・コンサルタントを対象としています。
- 『システム・アドミニストレーション』ガイド
主にシステム管理者を対象としています。
- 『パッケージ管理』ガイド
主にシステム管理者およびカスタマイズ環境を管理する方を対象としています。
- 『サーバー&ワークステーション・アドミニストレーション』ガイド
主にネットワーク管理者を対象としています。

『CNC インプリメンテーション』ガイドは、主にソフトウェアのインストール後に必要なクライアント独自のセットアップについて説明しています。主にテクニカル・コンサルタントを対象していますが、他の担当者が参照しても役立つ重要な情報が含まれています。

インプリメンテーション・チームの役割

インプリメンテーション・メソドロジーは、ソリューションの設計、インストール、コンフィギュレーションに関わる特定の役割を定義しています。通常、これらの役割を担う担当者は次の4つのインプリメンテーション・チームに配置されます。

- テクノロジー – インストールとアップグレード、システム管理、セキュリティ、変更管理
- 開発 – データ・コンバージョン、インターフェイス、カスタム修正
- ファンクショナル – ビジネス・プロセス、アプリケーション構成、インテグレーションおよびテスト、エンドユーザー・トレーニング
- システム・インテグレーション – データ・センター設計、ハードウェア・サポート、ネットワーク構成、サードパーティ・ソフトウェア

これらのインプリメンテーション・チームは、コンサルタントとカスタマーの両方で構成されます。インプリメンテーションが完了すると、コンサルタントの役割は減りますが、カスタマーの役割は残り、しばしば責任レベルにおいて役割が大きくなります。このため、各役割を担うカスタマーの担当者は、必要なトレーニングを受け、十分な知識を持っている必要があります。

テクノロジー・チームの役割

通常、テクノロジー・プロジェクト・チームはテクニカル・コンサルタント 1 人、カスタマーのシステム管理者と変更管理責任者の 2 人がリードします。テクニカル・コンサルタントとシステム管理者は、ソフトウェアのインストールおよび、環境、ユーザー、セキュリティの設定、分散処理、データ・レプリケーション、その他のシステム管理および運用サポートを担当します。テクニカル・コンサルタントと変更管理責任者は、バージョン管理、ソフトウェア・アップデートとサービス・パックの導入、環境全体にわたるコードとデータの検証と更新、およびサーバーとワークステーションへの配布などの作業を担当します。

開発チームの役割

開発プロジェクト・チームは通常、1 人のカスタム・ソリューション・コンサルタントと、1 人またはそれ以上のアプリケーション開発者がリードします。カスタム・ソリューション・コンサルタントは、アプリケーション開発を行うことによって業務上の問題を解決します。主な役割には、データ移行計画の作成、旧システムおよびサードパーティ・システムとのインターフェイス設計、アップグレードを考慮に入れたカスタマイズ設計があります。アプリケーション開発者は、旧システムからのデータ移行、インターフェイスのコーディングとテスト、プログラムのカスタマイズとテスト、全変更箇所のシステムへの統合を行います。

変更管理担当者はソフトウェア変更の検証、プロモーション、配布をコントロールするバージョン管理を行い、カスタム・ソリューション・コンサルタントは、アプリケーション開発サイクルを明確にするために内部手順の作成をサポートする必要があります。さらに、開発チームのメンバーは、テクノロジーがインターフェイスとカスタマイズの設計と運用に与える影響だけでなく、変更管理ツールと手順を把握している必要があります。

ファンクショナル・チームの役割

ファンクショナル・プロジェクト・チームは、コンサルティングのプロジェクト責任者とカスタマーのプロジェクト責任者がリーダーとなり、アプリケーション・コンサルタントとカスタマーの各業務における実務責任者で構成されます。これらのプロジェクト・メンバーは、アプリケーションの設計、構成、配布とともに、この製品のアプリケーションで実現するすべての業務プロセスのモデル化を行います。アプリケーション・コンサルタントは、製品のインストール、構成、ロールアウト後も製品のエキスパートとして引き続き関与します。アプリケーション・コンサルタントはテクニカル・レベルのソリューションは提供しませんが、アプリケーションの問題は技術面の設計および構成に影響を与えるため、このソフトウェアの分散処理、データ・レプリケーション、環境などの処理方法を理解しておく必要があります。さらに、アプリケーション・コンサルタントと実務責任者は問題発生時のトラブルシューティングおよび、テクニカルな問題とアプリケーションの問題の切分けのエキスパートになる必要があります。

システム・インテグレーション・チームの役割

システム・インテグレーション・チームは、PeopleSoft が行う作業以外の多くのタスクを受け持ちます。サードパーティ・コンサルタントは、この中のいくつかのサービスを提供し、必要に応じて CNC、ネットワーク設計、カスタマイズなどのコンサルティングも行います。これに加えて、カスタマーはハードウェア構成およびネットワーク構成のサポートを提供します。

ソフトウェアのインプリメンテーションでは、PeopleSoft のソフトウェアおよびサービス適用範囲外のタスクが多く発生します。システム・インテグレーション(サードパーティ)コンサルタントは、補足的なサードパーティ製品を組み込んで全体的なビジネス・ソリューションを拡張するだけでなく、アプリケーションとランタイム・サービスを最適にサポートするインフラを構築するためにこれらのサービスを提供します。これらのコンサルタントはデータ・センター設計、IT プロセスの改善、ネットワーク・インフラの整備などのサービス/アドバイスを提供することができます。また、インストール、システム・セットアップ、PeopleSoft のソフトウェア・ソリューションを強化、拡張するサードパーティのハードウェアおよびソフトウェア製品の統合をサポートすることもできます。これらのプロジェクト・メンバーは EnterpriseOne のシステム構成と動作内容とともに、さまざまな技術要素が各種オペレーティング・システムやデータベース管理システム、サードパーティ製ミドルウェア、ネットワークとどのように対応するかを理解している必要があります。

パス・コードの設定

パス・コードは、オブジェクトとそのロケーションを指定します。すべてのオブジェクトに対して、オブジェクト・パス・マスター(F00942)でパス・コードを定義する必要があります。

パス・コードの理解

オブジェクトの保管場所はパス・コードによって定義されています。すべてのオブジェクトに対し、オブジェクト・パス・マスター(F00942)でパス・コードを定義する必要があります。たとえば、次のオブジェクトに対してそれぞれパス・コードを定義することをお勧めします。

- プリステイン・オブジェクト
- 本稼動用オブジェクト
- 開発用オブジェクト
- カンファレンス・ルーム・パイロット(CRP)オブジェクト

〈パス・コード・マスター〉アプリケーション(P980042)はオブジェクト・パス・マスター(F00942)を使用します。

インストール時のパス・コードの理解

すべてのセントラル・オブジェクトに対してオブジェクト・パス・マスターにパス・コードを定義する必要があります。このソフトウェアのオブジェクトは、セントラル・オブジェクト・データ・ソースとオブジェクトのディレクトリから構成され、これにはビジネス関数ソース、インクルード・ファイル、オブジェクト・ファイル、ダイナミック・リンク・ライブラリ(DLL)が含まれます。パス・コード定義には、セントラル・オブジェクト・スペックのデータ・ソース名、およびオブジェクトのディレクトリ・パスが含まれます。

ワークステーション用のパッケージをビルドする場合、パス・コードを指定する必要があります。パッケージのソースとして使用するセントラル・オブジェクトとパッケージの配布先ディレクトリは、このパス・コードによって決定されます。

ワークステーションにジャストインタイム・インストール(JITI)が設定されている場合、そのワークステーション上にないアプリケーションを実行すると、デプロイメント・サーバーのデータ・ソースからワークステーションに必要なオブジェクトがインストールされます。

ランタイムにおけるパス・コードの理解

ランタイムでは、パス・コードは次のように使用されます。

- 有効な環境を検証する
サインオン時には毎回、使用している環境で定義されたパス・コードが、ワークステーションに実際にインストールされているパス・コードのディレクトリと照合されます。環境で定義したパス・コードがワークステーションにない場合、ログインしてもその環境は表示されません。
- 要求されたオブジェクトのディレクトリ・ロケーションを確定する

開発時のパス・コードの理解

開発のためにオブジェクトをチェックアウトする場合、オブジェクト管理ワークベンチを使ってパス・コードを指定します。セントラル・オブジェクトのロケーションはパス・コードで指定されており、そのロケーション(データベースとファイル・サーバーの両方)からオブジェクトをチェックアウトします。

オブジェクトをチェックインする場合も、オブジェクト管理ワークベンチを使ってパス・コードを指定します。パス・コードによって、オブジェクトを配置するセントラル・オブジェクトのロケーションが決まります。

パス・コードの処理

インストール時に作成したパス・コード以外に、新しくパス・コードを作成することができます。ここでは、パス・コードの追加とパス・コード定義の作成について説明します。パス・コードを追加する場合、いくつかのオブジェクト・ライブラリアン・テーブルを修正する必要があります。

新規パス・コードの追加

インストール時にパス・コード(PD7334)が作成されています。このパス・コードの情報を使って、別のパス・コード(DV7334 など)を作成できます。パス・コードを追加する場合、本稼動環境のパス・コードをコピーするなど、必ず既存のオブジェクトからコピーして追加してください。既存のパス・コードであれば、どのパス・コードでもコピーできます。オブジェクトが正しく設定されていない場合はパス・コードを追加できません。

▶ 新しいパス・コードを追加するには

注:

ここでは、DV9 が新しいパス・コードです。

1. デプロイメント・サーバー上に新しいパス・コードを格納するディレクトリを作成します。たとえば、DV9 というディレクトリを作成します。
各セントラル・オブジェクト・データ・ソースには、デプロイメント・サーバー上に関連ディレクトリ・パスが必要です。デプロイメント・サーバーの PD9 ディレクトリには、すべてのオブジェクト(ビジネス関数ソース、インクルード・ファイル、オブジェクト・ファイル、DLL など)と標準パッケージがあります。新しいセントラル・オブジェクト・ディレクトリを作成するテンプレートとして、このディレクトリを使用します。
2. 新しく作成したディレクトリ(DV9)に PD9 パス・コードのサブディレクトリをコピーします。
3. 新しいパス・コードのセントラル・オブジェクト・スペックを格納するために、新しい Oracle または SQL Server データベースを作成(あるいは既存の Oracle または SQL Server データベースの容量を拡張)します。
4. 新しいセントラル・オブジェクト・スペック(DV9)のテーブル・オーナーを作成します。
5. 新しい DV9 パス・コードについて、オブジェクト・ライブラリアン - 状況詳細テーブル(F9861)とバージョン・リスト・テーブル(F983051)を修正します。

6. そのデータベースにセントラル・オブジェクト・スペックの本稼動環境パス・コード(PD9)をコピーします。
7. 新たにコピーしたセントラル・オブジェクト・データベース(Central Objects - DV9)に対して新しいデータ・ソースが作成されていることを確認します。
作成されていない場合はマニュアルで作成する必要があります。
8. 〈オブジェクト・パス・マスター〉を使って、DV9 のパス・コード定義を修正します。[デプロイメント・データ・ソース]フィールドで、新しくコピーしたセントラル・オブジェクト・データ・ソース(DV9)に対して新しいデータ・ソースが指定されていることを確認します。[ロケーション]と[サーバー共有パス]フィールドで、新しい DV9 パス・コードのロケーションが定義されていることを確認します。
9. インストール時に作成されたパッケージ定義(DV9_A と DV9_B)があることを確認します。
10. PD9 パス・コードから既存のパッケージを新しい DV9 パス・コードにコピーします。パス・コードのディレクトリを新しいパス・コードのディレクトリに変更して、新しいパス・コードのパッケージ INF ファイルを修正します。
11. インストール時に DV9 環境が作成されていることを確認します。
インストール時に環境が作成されていない場合は、既存の環境をコピーして作成してください。パス・コードは新しいパス・コードに変更してください。

注:

ユーザーがログインする際に選択できるよう、新しい環境をユーザー・プロファイルに追加してください。

12. 新しい環境の OCM マッピングを修正します。

以前のセントラル・オブジェクト・データ・ソースにマップされたテーブルに対して、新たに作成したセントラル・オブジェクト・データ・ソースを参照するレコードを追加します。バージョンリスト・テーブル(F983051)と処理オプション・テキスト・テーブル(F98306)が Central Objects - DV9 データ・ソースを参照するようマッピングします。デフォルトのデータ・ソースとして Business Data - Test を使用すると便利です。

注:

OCM では、F987*シリーズのテーブルのロケーションは決定されません。セントラル・オブジェクトのスペック・テーブルを含むこれらのテーブルのロケーションは、デプロイメント・データ・ソースによって決定されます。

参照

- デプロイメント・サーバーでの新しいディレクトリの作成については、『インストール』ガイドの「パス・コード・オブジェクトのコピー」
- オブジェクト・ライブラリアンの修正については、『CNC インプリメンテーション』ガイドの「新しいパス・コードに対するテーブルの修正」

- 新しいデータ・ソースの確認については、「データベース・データ・ソースの追加」
- パッケージ定義の確認については、『パッケージ管理』ガイドの「パッケージの作成およびデプロイメントの概要」
- 既存のパッケージのコピーについては、『パッケージ管理』ガイドの「パッケージ・ビルド」
- 新しい環境の確認については、『CNC インプリメンテーション』ガイドの「環境の追加」
- スペック・テーブルのロケーション決定については、『CNC インプリメンテーション』ガイドの「オブジェクト構成マネージャの処理」

はじめる前に

- 新しいパス・コードを追加する場合、次の命名規則に従ってください。
 - 10 文字以内
 - 大文字のみ

パス・コード定義の作成

オブジェクト・パス・マスター (F00942) には、コンフィギュレーションのすべてのパス・コード定義が含まれています。このテーブルは、システム・データ・ソースにあります。

▶ パス・コード定義を作成するには

〈環境〉メニュー (GH9053) から〈パス・コード・マスター〉 (P980042) を選択します。

1. 〈パス・コードの処理〉で、[追加] をクリックします。
2. 〈パス・コードの改訂〉で、次のフィールドに値を入力して [OK] をクリックします。
 - パス・コード
 - 記述
 - ロケーション
 - リリース No.
 - デプロイメント・データ・ソース
 - UNC フラグ

新しいパス・コードに対するテーブルの修正

オブジェクト・ライブラリアン - 状況詳細テーブル (F9861) とバージョン・リスト・テーブル (F983051) を修正して、新しいパス・コードを反映させる必要があります。この 2 つのテーブルを修正することにより、開発者が正しいセントラル・オブジェクト情報を使ってチェックイン/チェックアウト処理を行うことができます。

▶ オブジェクト・ライブラリアン - 状況詳細テーブル(F9861)を修正するには

1. 使用しているプラットフォームによって、次のようにログオンします。
 - Oracle の場合
SQL Plus ユーティリティにログオンします。
 - Query Analyzer の場合
ユーザーJDE で SQL Server にログオンします。
 - DB2/400 の場合
STRSQL と入力して SQL を起動します。
 - DB2/UDB の場合
Windows:

[スタート]-[プログラム]-[IBM DB2]-[コマンド・ウィンドウ]を選択して、OWSH9 に接続します。

UNIX:

インスタンス・オーナーとしてログオンして OWSH9 に接続します。
2. マスター・テーブルF9861 のバックアップ・コピーを次のように保存します。
 - Oracle
CREATE TABLE F9861SAV AS SELECT * FROM OBJ9.F9861
 - Query Analyzer
SELECT * INTO F9861SAV FROM OBJ9.F9861
 - DB2/400
[F13]を押します。

オプション 1 を選択します。

Select output を 3(ファイル)に変更します。

[Enter]を押します。

出力ファイルを OL9/F9861SAV に変更します。

[Enter]を押します。

[Enter]を押します。

Select * FROM OL9/F9861

[Enter]を押します。

[F13]を押します。

オプション 1 を選択します。

Select output を 1(表示)に変更します。

[Enter]を押します。

[Enter]を押します。

- DB2/UDB

DB2 “CREATE TABLE OBJ9.F9861SAV LIKE OBJ9.F9861”

DB2 “INSERT INTO OBJ9.F9861SAV SELECT * FROM OBJ9.F9861”

3. 次のようにマスター・テーブル(F9861)のレコード・カウントを取得します。

- Oracle と Query Analyzer

SELECT COUNT(*) FROM OBJ9.F9861

- DB2/400

SELECT COUNT(*) FROM OL9/F9861

- DB2/UDB

DB2 “SELECT COUNT (*) FROM OBJ9.F9861”

4. 次のようにマスターF9861 レコードのサブセットを含む一時テーブルを作成します。

- Oracle

CREATE TABLE TEMPF9861 AS SELECT * FROM OBJ9.F9861 WHERE SIMKEY = `デプロイメント・サーバー名` AND SIPATHCD = `PD9`; COMMIT

SELECT COUNT(*) FROM TEMPF9861

- Query Analyzer

SELECT * INTO TEMPF9861 FROM OBJ9.F9861 WHERE SIMKEY = `デプロイメント・サーバー名` AND SIPATHCD = `PD9`

- DB2®/400

[F13]を押します。

オプション 1 を選択します。

Select output を 3(ファイル)に変更します。

[Enter]を押します。

出力ファイルを OL9/TEMP9861 に変更します。

[Enter]を押します。

[Enter]を押します。

```
SELECT * FROM OL9/F9861 WHERE SIMKEY = 'デプロイメント・サーバー名' AND  
SIPATHCD = `PD9`
```

[Enter]を押します。

- DB2/UDB

```
DB2 "CREATE TABLE OBJ9.TEMPF9861 LIKE OBJ9.F9861"  
DB2 "INSERT INTO OBJ9.TEMPF9861 SELECT * FROM OBJ9.F9861 WHERE  
SIMKEY = 'DEPLOYMT' AND SIPATHCD = 'PD9' "
```

5. 次のように F9861 の一時レコードを新しいパス・コードに更新します。

- Oracle

```
UPDATE TEMPF9861  
  
SET SIPATHCD = `DV9`  
  
COMMIT
```

- Query Analyzer

```
UPDATE TEMPF9861  
  
SET SIPATHCD = `DV9`
```

- DB2/400

```
UPDATE OL9/TEMPF9861  
  
SET SIPATHCD = `DV9`
```

[Enter]を押します。

- DB2/UDB

```
DB2 "UPDATE OBJ9.TEMPF9861 SET SIPATHCD = 'DV9' "
```

6. 次のようにマスター・テーブル F9861 に一時レコードを追加します。

- Oracle

```
INSERT INTO OBJ9.F9861  
  
AS SELECT * FROM TEMPF9861;  
  
COMMIT;
```

- Query Analyzer

```
INSERT INTO OBJ9.F9861  
  
SELECT * FROM TEMPF9861
```

- DB2/400
INSERT INTO OL9/F9861;

SELECT * FROM OL9/TEMPF9861
 - DB2/UDB
DB2 “INSERT INTO OBJ9.F9861 SELECT * FROM OBJ9.TEMP9861”
7. 次のようにマスター・テーブル F9861 の新しいレコード件数が正しいことを確認します。
- Oracle と Query Analyzer
SELECT COUNT(*) FROM OBJ9.F9861;
 - DB2/400
[F13]を押します。

オプション 1 を選択します。

Select output を 1(表示)に変更します。

[Enter]を押します。

[Enter]を押します。

SELECT COUNT(*) FROM OL9/F9861
 - DB2/UDB
DB2 “SELECT COUNT (*) FROM OBJ9.F9861”
8. 次のように F9861 の一時テーブルをドロップします。
- Oracle と Query Analyzer
DROP TABLE TEMPF9861;
 - DB2/400
DROP TABLE OL9/TEMPF9861
 - DB2/UDB
DB2 “DROP TABLE OBJ9.TEMP9861”
9. 次のようにバックアップの F9861SAV テーブルをドロップします。
- Oracle と Query Analyzer
DROP TABLE F9861SAV;
 - DB2/400
DROP TABLE OL9/F9861SAV

- DB2/UDB
DB2 “DROP TABLE OBJ9.F9861SAV”

▶ **バージョン・リスト・テーブル(F983051)を修正するには**

Oracle の場合は SQL Plus ユーティリティ、SQL Server の場合は ISQL/w にユーザー JDE でログインしてください。

次の SQL コマンドを実行して、テーブル OBJ9.F9861 のバックアップを作成します。

- Oracle の場合：
UPDATE DV9.F983051 SET VRMKEY=`デプロイメント・サーバー名`,VRENHV=`DV9`;

COMMIT;
- ISQL/w の場合：
UPDATE DV9.F983051

SET VRMKEY=`デプロイメント・サーバー名`,VRENHV=`DV9`

オブジェクト構成マネージャ(OCM)

ワークステーションの能力とサーバー上のデータの整合性が最適となるようにデータやロジックを分散することができます。次のような機能を備えています。

- パーティション区切りを柔軟に決定。要件変更に応じて、すばやくシステムのパーティションを再分割することができます。
- 必要なデータやロジックのオブジェクトを独自に使用。たとえば、販売担当者は、外出先で見積りや在庫の確認に必要なデータやロジックだけをダウンロードできます。帰社してからは、サーバーに接続してバッチ処理でオーダーをかけることができます。
- システムの拡張。事実上無限の組合せのシステム上でオブジェクトを移動できるため、必要に応じてシステムを拡張することができます。

データの格納場所とロジックの処理が行われるロケーションをトラッキングするために、オブジェクト構成マネージャ(OCM)というツールを使用します。このツールを使ってデータやロジック処理のロケーションを指定できます。

オブジェクト構成マネージャ(OCM)の理解

〈オブジェクト構成マネージャ〉ツール(P986110)は、プログラミングなしで分散処理やデータを動的に構成します。環境やユーザーに応じて、オブジェクト構成マネージャ(OCM)は次の項目の正確なロケーションを示します。

- データ
- バッチ処理
- ビジネス関数

オブジェクト構成マネージャ(OCM)は、データの格納場所や処理を実行するロケーションを特定するための情報をテーブルに保管します。ランタイムでは、オブジェクト構成マネージャ(OCM)によってデータと処理のロケーションが決定されます。

ビジネス・データ・オブジェクト(テーブル)は、データベース・データ・ソースにマップされます。バッチ処理とビジネス関数は、ロジック・データ・ソースにマップされます。

常に少なくとも次の 2 つの OCM テーブルが必要です。

- すべてのワークステーションで共有するテーブル。このテーブルをシステム・データ・ソースに格納します。通常、システム・データ・ソースはプライマリのデータ・サーバーに格納されます。プライマリのサーバーが使用できない場合、ワークステーションの jde.ini ファイルを参照してセカンダリのデータ・サーバーに接続します。
- 各ロジック・サーバー用のテーブル。サーバーとワークステーションでは処理が異なります。たとえば、各ロジック・サーバーのサーバー・マップ・データ・ソースには、サーバー処理用の OCM テーブルが別に格納されています。

例:アプリケーションのロジック要求

次の図は、〈オブジェクト構成マネージャ(OCM)〉による一般的なロジック要求の処理方法です。

ステップ 1: アプリケーションのロジック要求は、データ要求と似ています。オブジェクト構成マネージャ(OCM)は、すべてのビジネス関数とバッチ処理が処理される場所をコントロールします。

たとえば購買オーダーを追加する場合、ビジネス関数が呼び出されるたびに、オブジェクト構成マネージャ(OCM)を参照してそのビジネス関数を処理する場所が決定されます。[OK]をクリックして購買オーダーを完了すると、すべての情報を検証してレコードを記録するためにマスター・ビジネス関数が呼び出されます。これらのビジネス関数はローカルとサーバーのどちらでも処理できます。

オブジェクト構成マネージャ(OCM)の重複しない主キーとなるインデックスは次のとおりです。

- 環境 (PD9、DV9 など)
- ユーザー (特定のユーザーID/ロールまたは*PUBLIC)
- オブジェクト名 (F0101、B401002、R09801 など)
- データベース・パス

次の表は、OCM がロジック要求に対する正しいデータ・ソースを検索する順序を示しています。例として次のような条件を使用します。

- 環境は PD9(本稼動用)
- 状況は AV(アクティブ)
- オブジェクト・タイプはバッチ処理(UBE)とビジネス関数(BSFN)のどちらも可

検索順序	オブジェクト名	ユーザー/ グループ	検索条件
1	B0900049	SI5745669 (ユーザー ID)	指定された環境で状況アクティブ、タイプ UBE または BSFN の指定されたオブジェクトとユーザーのレコード
2	B0900049	OWTOOL (ロール)	指定された環境で状況アクティブ、タイプ UBE または BSFN の指定されたオブジェクトとグループのレコード
3	B0900049	*PUBLIC	指定された環境で状況アクティブ、タイプ UBE または BSFN の指定されたオブジェクトで、*PUBLIC のレコード
4	DEFAULT	SI5745669 (ユーザー ID)	指定された環境で状況アクティブ、タイプ UBE または BSFN の不特定オブジェクト(デフォルト)で、指定されたユーザーのレコード
5	DEFAULT	OWTOOL (ロール)	指定された環境で状況アクティブ、タイプ UBE または BSFN の不特定オブジェクト(デフォルト)で、指定されたグループのレコード
6	DEFAULT	*PUBLIC	指定された環境で状況アクティブ、タイプ UBE または BSFN の不特定オブジェクト(デフォルト)で、*PUBLIC のレコード
7			このオブジェクト・タイプのレコードがない場合、ワークステーション上で処理を実行します。

ステップ 2: データ・ソースが確定すると、そのデータ・ソース定義が JDENet に渡されます。

ステップ 3: JDENet は、サーバーにメッセージを送信して、ロジックの処理を開始します。サーバーの JDENet がメッセージを受信すると、サーバーの ERP 9.0 プロセスは、要求されたロジック・オブジェクトを処理することによってメッセージに応答します。

オブジェクト構成マネージャ (OCM) の処理

オブジェクト構成マネージャ(OCM)を使って柔軟にデータやバッチ・アプリケーション、ビジネス関数をデータ・ソースにマップして、ERP 9.0 の分散処理を管理することができます。たとえば、テーブル・オブジェクトをデータベース・データ・ソースに、ロジック・オブジェクトをロジック・データ・ソースにマップすることができます。

下に説明されているように、OCM テーブルとデータ・ソース・マスターは少なくとも 2 つずつ必要です。

全ワーク ステーション用	ワークステーション処理に使用する OCM テーブルとデータ・ソース・マスターは、通常エンタープライズ・サーバー上にある中央のシステム・データ・ソースに保管されます。システム・データ・ソースが使用できない場合、ワークステーションの jde.ini ファイルを参照してセカンダリのロケーションが検索されます。
各ロジック・ サーバー用	ロジック・サーバーが使用する OCM テーブルとデータ・ソース・マスターは、そのサーバー上のサーバー・マップ・データ・ソースに保管されます。ロジック・サーバー・タイプごとに、独自のサーバー・マップ・データ・ソースが必要です。

ビジネス・オブジェクトは、ランタイムで分散処理と分散データを構成するのに使用されます。この構成はオブジェクト構成マスター(F986101)で定義されています。サーバー・オブジェクト・マップを処理して、このテーブルのレコードを修正できます。

ワークステーション・マッピングは、中央のシステム・データ・ソースに保管されます。エンタープライズ・サーバーが使用するテーブル F986101 は、そのサーバー上のサーバー・マップ・データ・ソースに保管されます。エンタープライズ・サーバーごとに、サーバー・マップ・データ・ソースが必要です。

ロジック・オブジェクトを処理するエンタープライズ・サーバーでは、データを取り込むロケーションについて、ワークステーションとは異なるアプローチがとられます。ユーザー定義コードを要求するワークステーションでは、テーブル F986101 (システム・データ・ソースにあります) はローカルのデータベースを参照します。

エンタープライズ・サーバーがユーザー定義コードを要求する際、ワークステーションでこの情報を検索するのは無意味です。したがって、エンタープライズ・サーバーにはユーザー定義コードに対して固有のマッピングを設定する必要があります。このマッピングは、サーバー・マップ・データ・ソース内のオブジェクト構成マスター(F986101)で設定します。

ワークステーションのテーブル F986101 を変更した場合、サーバー・マップでもこのテーブルを変更する必要があるか確認してください。たとえば、ワークステーションに対して固有のマッピングを持つ新しい環境がある場合、エンタープライズ・サーバー上で対応するマッピングも変更すべきかどうか確認する必要があります。

OCM は、オブジェクト・マッピングの管理に役立つバッチ処理も提供しています。これらのバッチ・処理では、OCM レコードの比較、更新、コピー、削除などが実行されます。

オブジェクト構成マネージャ(OCM)アプリケーション(P986110)は、オブジェクト構成マスター(F986101)を更新します。

オブジェクトのマッピング

オブジェクトのマッピングは環境別に行います。作成されている環境を選択して、その環境のオブジェクトを、オブジェクトが使用するデータ・ソースにマップします。あるオブジェクト・タイプのすべてのインスタンスに対して1つのデータ・ソースへのデフォルト・マッピングを設定したり、データ・ソースに個々のオブジェクトをマップすることができます。

オブジェクト・タイプのマッピング: デフォルト・マップ オブジェクト・タイプに対してデフォルト・マップを作成するには、DEFAULT というオブジェクト名のマッピングを作成します。次に、オブジェクト・タイプ(TBLE など)とデータ・ソースを入力します。オブジェクト・タイプ TBLE に対するデフォルト・マップを作成すると、個別にマップされていないテーブル・オブジェクトはデフォルトのデータ・ソースを参照するようになります。

個々のオブジェクトのマッピング 1つの環境内で各オブジェクトをそれぞれマップすることができます。たとえば、セキュリティ・ワークベンチ・テーブル(F00950)などの特定のテーブルをシステム・データ・ソースのようなデフォルト以外のデータ・ソースにマップできます。

OCM で個別にオブジェクトがマップされていない場合は、そのオブジェクト・タイプのデフォルト・マップが使用されます。

注意:

テーブル・オブジェクトにはデフォルトのロケーションがないため、各環境では*PUBLIC のユーザー・プロファイルに対して TBLE(テーブル)オブジェクトのデフォルト・マップを作成する必要があります。テーブル・オブジェクトにデフォルト・マップがなく、個別にマップされていない場合は、選択/失敗エラー・メッセージが表示されます。

そのオブジェクト・タイプに対するデフォルト・マップがない場合、バッチ・アプリケーションとビジネス関数はローカルで実行されます。

▶ オブジェクトをマッピングするには

〈システム・アドミニストレーション・ツール〉メニュー(GH9011)から〈オブジェクト構成マネージャ(OCM)〉プログラム(P986110)を選択します。

1. 〈マシンの検索/選択〉で、処理する OCM テーブルが格納されたデータ・ソースをハイライトして[選択]をクリックします。

このフォームには、〈データ・ソースの改訂〉プログラム(P986115)の〈上級設定〉フォームで [OCM データ・ソース] オプションがオンになっているデータ・ソースが表示されます。グリッドのデータ・ソースの横のマシン名は、データ・ソースが存在するマシンのことです。ワークステーションは、マッピングにシステム・データ・ソースを使用します。各エンタープライズ・サーバーには、それぞれのサーバー・マップ・データ・ソースがあります。

2. 〈オブジェクト・マッピングの処理〉で、オブジェクト・マッピングを変更するデータ・ソースを検索したり、データ・ソースに新しいオブジェクト・マッピングを追加するフォームにアクセスするには[追加]をクリックします。
3. 〈オブジェクト・マッピングの改訂〉で、次のフィールドに入力してオブジェクトをマップするデータ・ソースを指定します。

- 環境名
 - オブジェクト名
データ・ソースにあるすべてのオブジェクトに対するパラメータを定義するには、このフィールドに DEFAULT と入力します。
 - 1 次データ・ソース
マップするオブジェクトのタイプに対して有効なデータ・ソースを選択してください。たとえば、データベース・データ・ソースにマップできるのは TBLE と GT のみ、ロジック・データ・ソースにマップできるのは BSFN のみです。
 - ユーザー
 - データ・ソース・モード
ビジネス関数 (BSFN) データ・ソースに対しては、セカンダリ・データ・ソースのみ使用してください。
 - QBE 使用
4. [OK]をクリックしてオブジェクト・マッピングを保存します。
- マッピングを行うたびに[OK]をクリックして、オブジェクトのマッピングを継続できます。[キャンセル]をクリックして〈オブジェクト・マッピングの処理〉フォームに戻る前に、必ず[OK]をクリックして最新のマッピングを保存してください。

注:

〈オブジェクト・マッピングの処理〉フォームの[オブジェクト状況]フィールドに NA と表示されている場合、無効な状況でマッピングが作成されたことを示します。

5. マッピングを有効にするには、〈オブジェクト・マッピングの処理〉フォームでマップをハイライトして[ロー]メニューから[状況の変更]を選択します。[オブジェクト状況]フィールドには AV(有効)と表示されます。

参照

- TBLE オブジェクト・タイプの検証については、『CNC インプリメンテーション』ガイドの「オブジェクト構成管理の処理オプション」

フィールド記述

記述	用語解説
環境名	インストール・アプリケーションの場合、環境名はプラン名とも呼ばれ、インストール/再インストールのための環境を識別します。環境またはバージョン・アプリケーションの場合、アプリケーションまたはバージョンのスペック・データの場所を識別するパス・コードです。

オブジェクト名

システム・オブジェクトを識別する名前。J.D. Edwards ERP アーキテクチャはオブジェクト指向です。ソフトウェアの個々のオブジェクトはすべてのアプリケーションのビルディング・ブロックとなっており、複数のアプリケーションでオブジェクトを再使用できます。各オブジェクトは、オブジェクト・ライブラリアンで管理されます。オブジェクトの例は次のとおりです。

- バッチ・アプリケーション(レポートなど)
- 対話型アプリケーション
- ビジネス・ビュー
- ビジネス関数
- ビジネス関数データ構造体
- イベント・ルール
- メディア・オブジェクト・データ構造体

--- フォーム固有 ---

このフォームでは、マッピングする ONEWORLD オブジェクトです。全オブジェクトタイプに対してデフォルトマップを作成するには、このフィールドにデフォルトのリテラル値を入力します。その後、オブジェクトタイプフィールドにオブジェクトタイプを入力します。

ユーザー

セキュリティを目的とし、ユーザーをグループに分類するために使用されるプロファイル。グループ・プロファイルは、グループ・メンバーに特定のプログラムへのアクセス権を与えるのに使用します。

ユーザー・クラス・グループの作成ルールは次のとおりです。

クラス・グループ・プロファイルは、他のシステム・プロファイルとの衝突を避けるため、*で始めます。

ユーザー・クラス・グループ・フィールドは、新しいグループ・プロファイルの入力時にはブランクにします。

--- フォーム固有 ---

このフォームには、各ユーザー、グループ名、またはリテラル値の"*PUBLIC"を入力できます。

オブジェクト・タイプ

処理中のオブジェクトのタイプ。たとえば、テーブルを処理している場合、オブジェクトタイプは TBLE です。ビジネス関数の場合は BSFN となります。

--- フォーム固有 ---

データ(オブジェクトタイプは TBLE)、バッチアプリケーション(オブジェクトタイプは UBE)、ビジネス関数(オブジェクトタイプは BSFN)、汎用テキスト(オブジェクトタイプは GT)をマップすることができます。

データ・ソース・モード	基本データ・ソースと第2データ・ソースのどちらを使用するかを示します。
QBE 使用	データ・ソースに対するロー・レベルでのレコード・ロックを ON/OFF する際に、このフラグを使用します。データベースの整合性を保つには、このフラグを ON にします。JDEBASE ミドルウェアは、このフラグを使用して、ロー・レベルでのレコード・ロックを使用するかどうかを確定します。

▶ **オブジェクト・ライブラリアン・テーブルのマッピングを変更するには**

〈システム・アドミニストレーション・ツール〉メニュー(GH9011)から〈オブジェクト構成マネージャ(OCM)〉プログラム(P986110)を選択します。

1. 〈マシンの検索/選択〉で、処理する OCM テーブルのマシンとデータ・ソースをハイライトして[選択]をクリックします。

データ・ソースはマシン上にあります。ワークステーションでは、システム・データ・ソースがマッピングに使用されます。エンタープライズ・サーバーごとに、独自のサーバー・マップ・データ・ソースがあります。

2. 〈オブジェクト・マッピングの処理〉で、データ・ソースに対してオブジェクト・マッピングを追加、変更するには[フォーム]メニューから[オブジェクト・ライブラリアン・データ・ソースの改訂]を選択します。

注:

オブジェクト・ライブラリアンのテーブルをマップする場合、同じパス・コードに基づくすべての環境で、オブジェクト・ライブラリアンのテーブルが同じデータ・ソースにマップされているかどうか検証されます。オブジェクト・ライブラリアンのテーブルを異なるデータ・ソースにマップした場合、エラー・メッセージが表示されます。

3. 〈OL データ・ソースの改訂〉で、次のフィールドに入力してオブジェクト・ライブラリアンのテーブルに対する新しいマッピングを定義します。

- パス・コード
- OL データ・ソース

4. データ・ソースへのマッピングの変更を保存するには[OK]をクリックします。

[OK]をクリックすると、[パス・コード]フィールドに入力したパス・コードを含む各環境のオブジェクト・ライブラリアンのデータ・ソース・マッピングが自動的に更新されます。

フィールド記述

記述	用語解説
パス・コード	ONEWORLD オブジェクトのポインタとなり、OENWORLD 内でのオブジェクトとそのロケーションをトラッキングするのに使用します。
OL データ・ソース	データ・ソースの名前。

Oracle パラメータ・テーブルの更新

Oracle を使用していて J.D. Edwards が推奨する命名規則に従わない場合や、新しい Oracle データ・ソースを追加する場合、Oracle パラメータ・テーブルを更新する必要があります。

▶ Oracle パラメータ・テーブルを更新するには

〈システム・アドミニストレーション・ツール〉メニュー(GH9011)から〈オブジェクト構成マネージャ(OCM)〉プログラム(P986110)を選択します。

1. 〈マシンの検索/選択〉で、処理する OCM テーブルのマシンとデータ・ソースをハイライトして[選択]をクリックします。

データ・ソースはマシン上にあります。ワークステーションでは、システム・データ・ソースがマッピングに使用されます。エンタープライズ・サーバーごとに、独自のサーバー・マップ・データ・ソースがあります。

2. 〈オブジェクト・マッピングの処理〉で、[検索]をクリックしてオブジェクト・マッピングを表示します。
3. オブジェクトをハイライトして[ロー]メニューから[Oracle パラメータ]を選択します。
4. 〈Oracle データベース・オブジェクトのサイズ変更〉で、[検索]をクリックします。

このフォームでは、リリース・レベル、データ・ソース、オブジェクト名で検索を行うことができます。新しいパラメータを追加する場合は、[検索]をクリックする必要はありません。

このアプリケーションは、〈データ・ソース〉プログラム(P986115)からもアクセスできます。

5. 〈Oracle データベース・オブジェクトのサイズ変更〉で、次のいずれかの操作を行います。
 - レコードをハイライトして[選択]をクリックします。
 - [追加]をクリックします。

〈Oracle データベース・オブジェクトのサイズ変更〉フォームが表示されます。このフォームで、データ・ソースまたはテーブルに対して表領域や索引情報の修正、追加を行うことができます。

また、〈データ・ソースの処理〉フォームの[ロー]メニューから[デフォルト・データベース]を選択して〈データベース・オブジェクトのサイズ変更〉フォームを表示することもできます。この場合、[オブジェクト名]フィールドには自動的に DEFAULT という値が入力され、デフォルト値が使用されます。オブジェクト名を DEFAULT 以外の値に変更すると、もともとデフォルト値が入力されていたフィールドが表示されます。

6. 新しいパラメータを追加する場合は、次のフィールドに値を入力します。
 - データ・ソース
 - オブジェクト名
データ・ソースにあるすべてのオブジェクトに対するパラメータを定義するには、このフィールドに DEFAULT と入力します。
 - リリース No.
7. DEFAULT マップまたは特定のオブジェクトのどちらに対してデータ・ソースを選択するかによって、〈Oracle データベース・オブジェクトのサイズ変更〉フォームに表示されるフィールドが異なります。DEFAULT マップに対しては、表領域または索引が常駐する Oracle 領域の名前しか定義できません。ただし、特定のオブジェクトに対しては、Oracle テーブルが使用する領域サイズなどのパラメータも定義できます。

DEFAULT マップに対しては、次のフィールドに値を入力します。

 - テーブルスペース名
 - インデックススペース名

特定のオブジェクトに対して、次のフィールドに値を入力します。

 - インデックス初期ストレージ
 - インデックスの次のストレージ
 - インデックスパーセント増加ストレージ
 - インデックス最小エクステント
 - インデックス最大エクステント

この値は、テーブル情報、インデックス情報の両方で有効です。
8. 一時変更情報を入力するには、[フォーム]メニューから[一時変更]を選択します。

このオプションは、〈データベース・オブジェクトのサイズ変更〉フォームの[ロー]メニューにもあります。
9. 〈テーブルおよびデータ・ソースの一時変更の改訂〉フォームで、次のフィールドに値を入力するか既存の値を変更します。
 - データのコピー(Y/N)
 - テーブルの作成 (1/0)
10. [OK]をクリックして、1 つ前のフォームに戻ります。〈システム・アドミニストレーション・ツール〉メニューに戻るまで、すべてのフォームで[OK]をクリックします。

フィールド記述

記述	用語解説
データ・ソース	データ・ソースの名前。
オブジェクト名	<p>システム・オブジェクトを識別する名前。J.D. Edwards ERP アーキテクチャはオブジェクト指向です。ソフトウェアの個々のオブジェクトはすべてのアプリケーションのビルディング・ブロックとなっており、複数のアプリケーションでオブジェクトを再使用できます。各オブジェクトは、オブジェクト・ライブラリアンで管理されます。オブジェクトの例は次のとおりです。</p> <ul style="list-style-type: none">○ バッチ・アプリケーション(レポートなど)○ 対話型アプリケーション○ ビジネス・ビュー○ ビジネス関数○ ビジネス関数データ構造体○ イベント・ルール○ メディア・オブジェクト・データ構造体
リリース No.	WORLD では、ソフトウェアバージョンのリポジトリに定義されたリリース番号。ONEWORLD では、リリースマスターに定義されたリリース番号をいいます。
テーブルスペース名	テーブルを保管するために作成された ORACLE スペース名
インデックススペース名	インデックスを保管するために作成されたオラクルのスペース領域名
インデックス初期ストレージ	初期のセグメント・スペース、またはエクステント内で、あるインデックスを保管するために要するオラクル・スペース容量(バイト)
インデックスの次のストレージ	オラクル・スペースの使用済セグメントが無くなった場合に、インデックスに割り当てられるオラクルスペースのバイト数
インデックスパーセント増加ストレージ	あるインデックスに割り当てられていた、これまでのエクステント・スペースを増やすために、新たなエクステント・スペースの増分パーセンテージ値を示すオラクル・パラメータ
インデックス最小エクステント	あるインデックスに割り当てるスペース・セグメントまたはエクステントの最小値を示すオラクル・パラメータ
インデックス最大エクステント	あるインデックスに割り当てられるスペース・セグメントあるいはエクステントの最大数値を示すオラクル・パラメータ。
テーブル初期ストレージ	初期のセグメントスペースまたはエクステント内で、あるテーブルを保管するために要する ORACLE スペースのバイト数

テーブル最大エクステント	あるテーブルに割り当てられるスペースセグメントまたはエクステントの最大数値を示す ORACLE パラメータ
テーブル最小エクステント	あるテーブルに割り当てられるスペース・セグメントあるいはエクステントの最小数値を示すオラクル・パラメータ。
テーブル次のストレージ	あるテーブルに対し、以前に使用されていたセグメントスペースがいっぱいになった時に割り当てられる ORACLE スペースのバイト数
テーブルパーセント増加ストレージ	あるテーブルに割り当てられていた、これまでのエクステント・スペースを増やすために、新たなエクステント・スペースの増分パーセンテージ値を示すオラクル・パラメータ。
データのコピー(Y/N)	ファイルとそのデータを本稼働環境にコピーするかどうかを示します。値が N の場合は データなしでファイルのみ移動します。JDFDATA から本稼働データ・ライブラリを作成 する場合、コピー時にデータを含めるかどうかをこのフィールドで指定します。
テーブルの作成 (1/0)	テーブルが、このデータソースに対して自動的に作成されることを示します。 UDC H96/CRを参照してください。

処理オプション:オブジェクト構成マネージャ(P986110)

〈オブジェクト構成マネージャ〉アプリケーション(P986110)には、エラー処理を制御する処理オプションがあります。

処理タブ

1. 存在しないテーブル・エラー

マップしようとする TBLE オブジェクトがマッピング先のデータソースで見つからない場合にエラーと警告のどちらを表示するか指定します。有効な値は次のとおりです。

1 エラー

ブランク 警告

汎用テキストのマッピング

汎用テキスト言語状況テーブル(F001651)は、格納されているデータをさまざまな用途で使用できるため、他のオブジェクトよりも柔軟にマッピングを行うことができます。汎用テキストに分類されるのは次のようなデータです。

- すべての環境で使用する共通データ。
たとえば、データ辞書の用語解説とビジネス関数の注記は、すべての環境で共通です。
- 特定の環境だけの本稼働用データ。
たとえば、本稼働環境、テスト環境、デモ・データ環境など、環境によって在庫品目の注記や住所録補足データの内容が異なる場合があります。

▶ 汎用テキストをマップするには

〈システム・アドミニストレーション・ツール〉メニュー(GH9011)から〈オブジェクト構成マネージャ (OCM)〉プログラム(P986110)を選択します。

〈オブジェクト構成マネージャ〉では、オブジェクト・タイプGTを使って汎用テキスト・オブジェクトをマップします。

次の表は、標準的な設定でERP 9.0 をインストールした場合のオブジェクト構成マネージャ・テーブル(F986101)の一例です。この例では、PD9(本稼動)とTS9(テスト)環境を示しています。

環境	オブジェクト名	記述	データ・ソース	オブジェクト・タイプ
PD9	F00165	汎用テキスト・テーブル	Business Data – Prod	TBLE
PD9	GT92002	データ辞書用語解説	Data Dictionary	GT
PD9	GT9860A	オブジェクト・ライブラリアン	Object Librarian	GT
PD9	GT9862A	ビジネス関数の注記	Object Librarian	GT
PD9	GT98DSA	データ構造体の注記 – 構造	Object Librarian	GT
PD9	GT98DSB	データ構造体の注記 – 構造、項目	Object Librarian	GT
PD9	GT98TMPL	メディア・オブジェクト・テンプレート	Object Librarian	GT
TS9	F00165	汎用テキスト・テーブル	Business Data – Test	TBLE
TS9	GT92002	データ辞書用語解説	Data Dictionary	GT
TS9	GT9860A	オブジェクト・ライブラリアン	Object Librarian	GT
TS9	GT9862A	ビジネス関数の注記	Object Librarian	GT
TS9	GT98DSA	データ構造体の注記 – 構造	Object Librarian	GT
TS9	GT98DSB	データ構造体の注記 – 構造、項目	Object Librarian	GT
TS9	GT98TMPL	メディア・オブジェクト・テンプレート	Object Librarian	GT

オブジェクト構成マネージャ・レポートの実行

オブジェクト構成マネージャには、マッピングをトラッキングするためのレポートを印刷するバッチ処理が備わっています。

いくつかの OCM レポートはメニューから選択して実行することができます。その他の OCM レポートは、レポート・ドライバ・ジョブから呼び出されます。結果は同じですが、レポート・ドライバから呼び出されたバッチ・プロセスによって処理オプションとデータ選択の設定は異なります。

ドライバからレポートを呼び出す場合、そのドライバから起動されるすべてのレポートで処理オプションの設定が同じになります。たとえば、R988611A ドライバは、〈マッピングの比較〉、〈一括更新〉、〈OCM の削除〉、〈OCM のコピー〉レポートを呼び出します。

レポートのデータ選択は、ドライバからではなく、その特定のレポートに対して入力する必要があります。処理オプションとデータ選択を設定した後、ドライバからレポートを実行することができます。

OCM レポートの処理オプションの設定

次のタスクは、レポート・ドライバを使ってデータ・ソース・レポートに対して処理オプションを設定する方法について説明しています。次の OCM レポートは、レポート・ドライバから呼び出されます。

- オブジェクト構成マッピングの比較
- オブジェクト構成の一括更新
- オブジェクト構成の削除
- オブジェクト構成のコピー
- 旧日付別ジョブ・マスターの削除

〈旧日付別ジョブ・マスターの削除〉を除くすべてのレポートは、R98611A レポート・ドライバから呼び出されます。〈旧日付別ジョブ・マスターの削除〉は R9861101 レポート・ドライバから呼び出されます。特に注記がない限り、これらのレポートの処理オプションの設定プロセスは選択するレポートに関わらず同じです。

▶ OCM レポートの処理オプションを設定するには

〈上級操作〉メニュー(GH9012)から、レポート・ドライバから呼び出すオブジェクト構成マネージャ(P986110)のレポートを1つ選択します。

1. 〈バッチ・バージョンの処理 - 使用可能なバージョン〉で、バージョンをハイライトして[ロー]メニューから[処理オプション]を選択します。
2. 〈旧日付別ジョブ・マスターの削除〉を選択した場合、フィールドに入力する処理オプションのレポートのタブをクリックします。
3. 処理オプションを保存して〈バッチ・バージョンの処理 - 使用可能なバージョン〉フォームに戻るには、[OK]をクリックします。

処理オプション:オブジェクト構成バッチ・アプリケーション

次のバッチ・アプリケーションには、同じ処理オプション・フォームを使用して R98611A レポートから入る処理オプションがあります。

- OCM マッピング比較(R986101)
- OCM 一括更新(R986110)
- オブジェクト構成の削除(R986120)
- オブジェクト構成のコピー(R986121)

OCM バッチ・アプリケーションの処理オプションでは、データの比較、更新、削除、コピーを行う際に使うデータ・ソースを指定できます。これらの処理オプションでは、特定のバッチ・アプリケーションで使用されるその他のパラメータも設定できます。

現在実行中のレポートの処理オプションのみが使用されます。他のレポートに関する処理オプションのフィールドは、そのレポートを実行するまで無視されます。

1. オブジェクト構成マネージャのレポート名

実行するレポート名を指定します。有効な値は次のとおりです。

R986101 (OCM マッピング比較)

R986110 (OCM 一括更新)

R986120 (オブジェクト構成削除)

R986121 (オブジェクト構成コピー)

2. Version バージョン

実行するレポートのバージョンを指定します。

R986101 (マッピング比較) タブ

この処理オプションでマッピングの比較に使うデータ・ソースを指定できます。比較方法やレポートの例外の扱い方も指定できます。

処理オプションの詳細内容を見るには、フィールドを右クリックして[ポップ・ヒント]を選択します。または、処理オプションのフィールドをクリックして[F1]キーを押してください。

1. 比較元データ・ソースおよび 2. 比較先データ・ソース

比較するデータ・ソース名を指定します。

3. 比較方法

使用する比較方法を指定します。この比較方法によって、データ・ソースの検索方法が決定されます。一方向の比較では、データ・ソース 1 だけにあつてデータ・ソース 2 にはないレコードが検索されます。双方向の比較では、データ・ソース 1 にあつてデータ・ソース 2 にはないレコードと、データ・ソース 1 になくてデータ・ソース 2 にあるレコードの両方が検索されます。

1 一方向のみで比較する。第 2 データ・ソースに存在せず、第 1 データ・ソースにあるレコードのみを印刷します。

ブランク 双方向で比較する。第 2 データ・ソースに存在せず第 1 データ・ソースに存在するレコード、および第 1 データ・ソースに存在せず第 2 データ・ソースに存在するレコードを印刷します。

4. 例外

レポートの例外のみ印刷するか、データ選択で選択されたすべてのレコードをデータ・ソースの違いとともに印刷するかを指定します。有効な値は次のとおりです。

1 例外のみ印刷する

ブランク 全てのレコードを印刷

一括更新タブ (P986110)

更新に使われるデータ・ソースを指定できます。既存のレコードから新しい OCM レコードを作成するか、新しい値を入力するかも指定できます。既存のレコードから新しい OCM レコードを作成するよう指定した場合、既存の値はすべて[処理]タブの処理オプションに入力した値で上書きされます。

処理オプションの詳細内容を見るには、フィールドを右クリックして[ポップ・ヒント]を選択します。または、処理オプションのフィールドをクリックして[F1]キーを押してください。

1. 処理モード

レポートがデータを処理するモードを指定します。有効な値は次のとおりです。

1 テスト・モード。削除するレコードを表示するレポートを作成します。レコードは削除されません。

ブランク 最終モード。削除するレコードのレポートを作成します。レコードは削除されます。

2. データ・ソース名

OCM テーブルのデータ・ソース名を指定します。このレポートを実行して更新されるすべての変更点は、指定したデータ・ソース内の OCM テーブルに反映されます。

3. 処理制御

新しい OCM レコードを作成するか、既存の OCM を変更するかを指定します。有効な値は次のとおりです。

1 既存の OCM から新しい OCM レコードを作成する。作成されるレコードは、選択したレコードと似ています。処理オプションに入力された新しい OCM の値は必要に応じて置き換えられます。

ブランク 選択したレコードを新しい OCM の値で変更する。データ選択を使用して、処理オプションに入力した新しい OCM の値を反映するレコードを指定します。

4. 新しい OCM 値

新しい OCM 値を入力します。既存のレコードから新しい OCM レコードを作成する場合、この処理オプションで入力したすべての値は、既存のものを上書きします。有効な値は次のとおりです。

環境名

オブジェクト名

1 次データ・ソース

ユーザー

オブジェクト・タイプ

データ・ソース・モード

第 2 データ・ソース

R986120(オブジェクト構成の削除)タブ

OCM レコードを削除するデータ・ソースを指定します。また、実行モードを選択し、〈オブジェクト構成のコピー〉バッチ・アプリケーション(R986121)を実行するかどうかを指定できます。これにより、レコードを削除する前にレコードのコピーを作成できます。

処理オプションの詳細内容を見るには、フィールドを右クリックして[ポップ・ヒント]を選択します。または、処理オプションのフィールドをクリックして[F1]キーを押してください。

1. 処理モード

レポートがデータを処理するモードを指定します。有効な値は次のとおりです。

1 テスト・モード削除するレコードのレポートを作成します。レコードは削除されません。

ブランク 最終モード削除するレコードのレポートを作成します。レコードは削除されます。

2. データ・ソース名

OCM テーブルのデータ・ソース名を指定します。このレポートを実行して更新されるすべての変更点は、指定したデータ・ソース内の OCM テーブルに反映されます。

3. 処理モード

レポートがデータを処理するモードを指定します。有効な値は次のとおりです。

1 テスト・モード削除するレコードのレポートを作成します。レコードは削除されません。

ブランク 最終モード削除するレコードのレポートを作成します。レコードは削除されます。

4. オブジェクトのコピー

〈オブジェクト構成のコピー〉プログラム(R986121)を〈オブジェクト構成の削除〉プログラムとともに実行するかどうかを指定します。有効な値は次のとおりです。

1 オブジェクト構成のコピー(R986121)を実行する。これにより、レコードを削除する前にコピーを作成します。〈オブジェクト構成のコピー〉を実行する際、処理オプションの[R986121]タブとデータ選択を設定してください。データ選択は、R986120(オブジェクト構成の削除)で指定した R986121 と同じバージョンで定義する必要があります。

ブランク オブジェクトのコピー(R986121)を実行しない。これによりコピーを作成せずにレコードが削除されます。

R986121(オブジェクト構成のコピー)タブ

OCM レコードのコピーに使用するコピー元とコピー先のデータ・ソースを指定します。

処理オプションの詳細内容を見るには、フィールドを右クリックして[ポップ・ヒント]を選択します。または、処理オプションのフィールドをクリックして[F1]キーを押してください。

1. 処理モード

レポートがデータを処理するモードを指定します。有効な値は次のとおりです。

- 1 テスト・モード。削除するレコードのレポートを作成します。レコードは削除されません。
- ブランク 最終モード。削除するレコードのレポートを作成します。レコードは削除されます。

2. コピー元データ・ソース

レコードをコピーする OCM テーブルのデータ・ソース名を指定します。

3. コピー先データ・ソース

コピー先の OCM テーブルのデータ・ソース名を指定します。

〈オブジェクト構成マッピングの比較〉レポートの実行

このレポートを実行すると、2つの異なるデータ・ソースのオブジェクト構成マスター(F986101)を比較してその違いを見ることができます。たとえば、システム・データ・ソースの F986101 テーブルと任意のサーバー・マップ・データ・ソースの F986101 テーブルを比較できます。

はじめる前に

- レポート用の処理オプションは、実際のレポートではなく、レポート・ドライバから設定してください。

▶ 〈オブジェクト構成マッピングの比較〉レポートのデータ選択の設定

〈システム・アドミニストレーション・ツール〉メニュー(GH9011)から〈バッチ・バージョン〉プログラム(P98305)を選択します。

1. 〈バッチ・バージョンの処理 - 使用可能なバージョン〉で、[バッチ・アプリケーション]フィールドに”R986101”と入力して[検索]をクリックします。
2. 処理オプションの[ドライバ]タブで指定したバージョンをダブルクリックします。
3. 〈バージョン・プロンプト〉で、[データ選択]オプションをオンにして[投入]をクリックします。
4. 〈データ選択〉で、適切なカラムから選択して比較するレコードを指定します。
5. [OK]をクリックします。

▶ 〈オブジェクト構成マッピングの比較〉レポートを実行するには

〈上級操作〉メニュー(GH9012)から〈バッチ・バージョン〉(P98305)を選択します。

1. 〈バッチ・バージョンの処理 - 使用可能なバージョン〉で、実行するレポートと一致するバージョンを選択します。
2. [選択]をクリックします。
3. 〈バージョン・プロンプト〉で、[投入]をクリックします。

〈オブジェクト構成の一括更新〉レポートの実行

この処理は、同じオブジェクト構成マスター(F986101)テーブル内でのオブジェクト・マッピングの一括更新および一括コピーを実行します。このバッチ・アプリケーションは、対話型アプリケーション(P986110)の代わりに使用できます。このアプリケーションは、複数のレコードを更新、コピーするのに便利です。

はじめる前に

- レポート用の処理オプションは、実際のレポートではなく、レポート・ドライバから設定してください。

▶ 〈オブジェクト構成の一括更新〉レポートのデータ選択の設定

〈システム・アドミニストレーション・ツール〉メニュー(GH9011)から〈バッチ・バージョン〉(P98305)を選択します。

1. 〈バッチ・バージョンの処理 - 使用可能なバージョン〉で、[バッチ・アプリケーション]フィールドに”R986110”と入力して[検索]をクリックします。
2. 処理オプションの[ドライバ]タブで指定したバージョンをダブルクリックします。
3. 〈バージョン・プロンプト〉で、[データ選択]オプションをオンにして[投入]をクリックします。
4. 〈データ選択〉で、適切なカラムから選択して更新またはコピーするレコードを指定します。
5. [OK]をクリックします。

▶ 〈オブジェクト構成の一括更新〉レポートを実行するには

〈上級操作〉メニュー(GH9012)から〈OCM カテゴリの更新/削除〉(R986101B)を選択します。

1. 〈バッチ・バージョンの処理 - 使用可能なバージョン〉で、実行するレポートと一致するバージョンを選択します。
2. [選択]をクリックします。
3. 〈バージョン・プロンプト〉で、[投入]をクリックします。
4. 処理オプションの設定を確認します。

注:

このレポートを実行すると、次のメッセージが表示される場合があります。

- Duplicate Key - Update/Copy not done
既存のレコードのキーを使って、レコードを更新したり新しいレコードを作成することはできません。
 - Active DEFAULT *Public records cannot be updated
*PUBLIC グループの DEFAULT レコードは更新できません。
-

〈オブジェクト構成の削除〉レポートの実行

このバッチ処理は、1つのデータ・ソースから特定の OCM テーブルのレコードを削除します。

はじめる前に

- レポート用の処理オプションは、実際のレポートではなく、レポート・ドライバから設定してください。

▶ 〈オブジェクト構成の削除〉レポートのデータ選択を設定するには

〈システム・アドミニストレーション・ツール〉メニュー(GH9011)から〈バッチ・バージョン〉プログラム(P98305)を選択します。

1. 〈バッチ・バージョンの処理 - 使用可能なバージョン〉で、[バッチ・アプリケーション]フィールドに”R986120”と入力して[検索]をクリックします。
2. 処理オプションの[ドライバ]タブで指定したバージョンをダブルクリックします。
3. 〈バージョン・プロンプト〉で、[データ選択]オプションをオンにして[投入]をクリックします。
4. 〈データ選択〉で、適切なカラムから選択して削除するレコードを指定します。
5. [OK]をクリックします。

▶ 〈オブジェクト構成の削除〉レポートを実行するには

〈上級操作〉メニュー(GH9012)から〈OCM カテゴリの更新/削除〉(R98610B)を選択します。

1. 〈バッチ・バージョンの処理 - 使用可能なバージョン〉で、実行するレポートと一致するバージョンを選択します。
2. [選択]をクリックします。
3. 〈バージョン・プロンプト〉で、[投入]をクリックします。

〈オブジェクト構成の削除〉レポートの実行

この処理では、データ・ソースの OCM レコードを他のデータ・ソースにコピーして、重複レコードを削除します。たとえば、既存の環境をコピーして新しい環境を作成すると、システム・データ・ソースに新しい環境の OCM レコードが作成されます。

〈オブジェクト構成のコピー〉バッチ処理を使用して、これらのレコードを該当するサーバー・マップ・データ・ソースにコピーします。レコードをコピーした後、〈オブジェクト構成一括更新〉プログラム(R986110)を使って、少なくとも該当するサーバーのロケーションに対して LOCAL のマッピングを変更します。

はじめる前に

- 処理オプションは、実際のレポートではなく、レポート・ドライバから入力してください。

〈OCM 検証〉レポートの実行

〈OCM 検証〉レポート(R9861130)は、次のことを検証するのに便利なトラブルシューティング・ツールです。

- すべてのデータ・ソース・マスター定義に OCM マッピングがある。
- OCM マッピングの整合性がとれており、データ・ソース定義があり、重複していない。
- 適切なスペック・ファイルが特定のサーバー上にある。
- ビジネス関数 DLL が特定のサーバー上にある。

このレポートを実行し、ワークステーションやサーバーで上記の項目を検証します。このレポートを実行するマシンによって、検証される OCM テーブルとデータ・ソース・マスター・テーブルが決まります。したがって、ローカル・マシンと ERP 9.0 を実行する各サーバー上でこのレポートを実行する必要があります。デプロイメント・サーバー上でも、このレポートを実行することができます。

▶ 〈OCM 検証〉レポートを実行するには

〈上級操作〉メニュー(GH9012)から〈OCM の確認〉(R98611A)を選択します。

1. 〈バッチ・バージョンの処理 - 使用可能なバージョン〉で、バージョンをハイライトして[選択]をクリックします。
2. 〈バージョン・プロンプト〉で、[データ選択]をオンにして[投入]をクリックします。
3. 処理オプションで、次のフィールドに値を入力します。

処理オプションは、カテゴリに分かれており、必要な検証タイプを選択できます。最初の 2 つのオプションは無関係なデータ・ソースについての検証で、その他のオプションは OCM マッピングの検証に関連しています。

検証を行わない場合は、その処理オプションを空白にしてください。

処理オプションで次のように設定することができます。

- データ・ソース・マスター定義を検証する。1 を入力すると、すべてのデータ・ソース・マスター(F98611)定義に OCM マッピングがあるかどうかを検証されます。
- 検証する環境名を入力する。*ALL を入力するとすべての環境が検証されます。この環境は、この後のすべての検証で使用されます。
- オブジェクト・マッピングを検証するユーザーまたはグループ(*PUBLIC を含む)を入力する。すべてのユーザーを検証するには、*ALL と入力するか空白のままにします。ユーザーID を入力した場合、*PUBLIC レコードもレポートに印刷されます。
- アクティブまたはアクティブでないマッピングを検証するかを指定する。1 を入力すると、入力したユーザーや環境で有効なすべての OCM マッピングについて、データ・ソース・マスター(F98611)にデータ・ソース定義があるかどうかを検証されます。2 を入力すると、アクティブ、非アクティブ両方のマッピングが検証されます。
- 重複レコードを検証する。1 を入力すると、入力したユーザーや環境に重複する OCM マッピングがないかどうかを検証されます。ただし、アクティブなマッピングに対してのみ重複チェックが行われます。

- 連結 DLL 名を確認する。1 を入力すると、特定のサーバー上に各ビジネス関数の連結 DLL 名があり、その DLL 名が有効かどうかを検証されます。
- スペック・ファイルを検証する。1 を入力すると、ERP 9.0 の実行に必要なすべてのスペック・ファイルがこのパス・コードにあるかどうかを検証されます。このオプションは、サーバーに対して実行する場合のみ有効です。

処理オプション: OCM の検証 (R9861130)

処理オプションの詳しい内容を見るには、フィールドを右クリックして[ポップ・ヒント]を選択します。または、処理オプションのフィールドをクリックして[F1]キーを押してください。

1. データ・ソース・マッピング

データ・ソース・マスター定義のデータ・ソース検証を実行するか指定します。有効な値は次のとおりです。

1 各データ・ソースに定義された OCM マッピングが最低 1 つあるかを検証する

ブランク データ・ソース定義を検証しません。

2. 環境の確認

検証を行う環境を指定します。有効な値は次のとおりです。

*All すべての環境の検証を実行します。

特定の環境名 指定した環境のみ検証します。

親 DLL タブ

各ビジネス関数に有効な親 DLL が割り当てられているか検証します。

処理オプションの詳しい内容を見るには、フィールドを右クリックして[ポップ・ヒント]を選択します。または、処理オプションのフィールドをクリックして[F1]キーを押してください。

1. ビジネス関数の親 DLL

各ビジネス関数に親 DLL があるかどうかを検証します。有効な値は次のとおりです。

1 各ビジネス関数に親 DLL が割り当てられているかどうかを検証する。また、DLL 名が正しいかどうかを検証します。

ブランク 検証しない

スペック・タブ

ERP 9.0 を実行するために必要なすべてのスペックがサーバー上にあるかどうかを検証します。

処理オプションの詳しい内容を見るには、フィールドを右クリックして[ポップ・ヒント]を選択します。または、処理オプションのフィールドをクリックして[F1]キーを押してください。

1. サーバー・スペックの確認

ERP 9.0 を実行するために必要なすべてのスペックがパス・コードにあるかどうかを検証します。この処理オプションは、サーバーに対して実行する場合のみ有効です。有効な値は次のとおりです。

1 必要なスペックがあるかどうかを検証する

blank スペックを検証しない

OCM タブ

検証するユーザーID および OCM マッピングを指定したり、OCM マッピングの重複チェックを行うかどうかを指定します。

処理オプションの説明を参照するには、処理オプションのフィールドを右クリックして、[表示]-[ポップヒント]を選択します。または、処理オプションのフィールドをクリックして[F1]キーを押してください。

1. 確認するユーザーID

OCM 検証を実行するユーザーID を指定します。有効な値は次のとおりです。

blank すべてのユーザーを検証する

*ALL すべてのユーザーを検証する

*PUBLIC *PUBLIC のみ検証する

特定のユーザーID 指定したユーザーのみ検証します。

2. OCM マッピング

OCM マッピングを検証するかどうかを指定します。有効な値は次のとおりです。

blank OCM マッピングを検証しない。

1 すべてのアクティブな OCM マッピングを検証する

2 アクティブおよび非アクティブ両方の OCM マッピングを検証する

3. 重複する OCM マッピング

OCM レコードの重複チェックを行うかどうかを指定します。有効な値は次のとおりです。

blank 重複チェックを行わない

1 アクティブな OCM マッピングのレコードが重複していないかチェックする

〈オブジェクト構成システム・テーブルの更新〉レポートの実行

この処理では、環境詳細テーブル(F00941)にあるすべての環境に対して、指定したテーブル、ユーザーID、データ・ソースの有効な OCM レコードが追加されます。データ選択を使って、OCM レコードを追加する環境を指定することができます。

▶ 〈オブジェクト構成システム・テーブルの更新〉レポートを実行するには

〈上級操作〉メニュー(GH9012)から〈オブジェクト構成システム・テーブルの更新〉(R986101A)を選択します。

1. 〈バッチ・バージョンの処理 - 使用可能なバージョン〉で、バージョン XJDE0001 を選択します。
2. 〈バージョン・プロンプト〉で、[データ選択]オプションをオンにして[投入]をクリックします。
3. 〈データ選択〉で、更新するレコードを指定して[OK]をクリックします。
4. 処理オプションで、次のフィールドに値を入力します。
 - マッピングを作成するテーブル名
 - テーブルのマッピング先のデータ・ソース
 - レコードのマッピング先のユーザーID

*PUBLIC を入力すると、すべてのユーザーのレコードがマップされます。グループ別にマップすることもできます。

 - テスト・モードまたは最終モード

ブランクの場合、テスト・モードで実行されます。まずテスト・モードで実行して、正しい結果になるかどうかを確認してください。最終モードで実行する場合は 1 を入力します。
5. [OK]をクリックします。

処理オプション: システム・テーブル・マッピングの作成(R986101A)

処理タブ

OCM マッピングを作成するのに使用するテーブル名、データ・ソース、ユーザーIDを指定します。テスト・モード、最終モードの指定もできます。

処理オプションの詳しい内容を見るには、フィールドを右クリックして[ポップ・ヒント]を選択します。または、処理オプションのフィールドをクリックして[F1]キーを押してください。

1. テーブル名

各環境にマッピングするテーブル名を指定します。このテーブル名とデータ・ソースおよびユーザーIDの処理オプションで指定した値を使って、OCM マッピングが作成されます。

2. データ・ソース

各環境にマッピングするデータ・ソースを指定します。このデータ・ソースとテーブル名およびユーザーIDの処理オプションで指定した値を使って、OCM マッピングが作成されます。

3. ユーザーID

各環境にマッピングするユーザーIDを指定します。このユーザーIDとテーブル名およびデータ・ソースの処理オプションで指定した値を使って、OCM マッピングが作成されます。

4. テスト/最終モード

バッチ・アプリケーションをテスト・モードと最終モードのどちらで実行するかを指定します。テスト・モードで実行すると、レコードは変更されません。最終モードで実行した場合は、レコードの更新と追加が行われます。有効な値は次のとおりです。

ブランク テスト・モード

1 最終モード

〈ビジネス関数用 OCM レコードの作成〉レポートの実行

このバッチ処理では、オブジェクト・ライブラリアンからサーバーのビジネス関数が読み込まれ、処理オプションで指定したターゲット・データ・ソースにそれらのビジネス関数に対する OCM レコードが作成されます。また、OCM レコードを作成する元となるデータ・ソースや環境を指定も処理オプションで指定できます。

▶ 〈サーバー・ビジネス関数用 OCM レコードの作成〉を実行するには

〈上級操作〉メニュー(GH9012)から〈ビジネス関数用 OCM レコードの作成〉(R986140)を選択します。

1. 〈バッチ・バージョンの処理 - 使用可能なバージョン〉、XJDE0001 をハイライトして[選択]をクリックします。
2. 〈バージョン・プロンプト〉で、[データ選択]オプションをオンにして[投入]をクリックします。
3. 〈データ選択〉で、作成するレコードを指定して[OK]をクリックします。
4. 処理オプションで、次のフィールドに値を入力します。
 - テスト・モードまたは最終モードを指定します。
1 を入力すると最終モードで実行され、レポートの印刷と更新が行われます。0 を入力するかフィールドをブランクにするとテスト・モードで実行され、レポートは印刷されませんが更新は行われません。
 - ビジネス関数を実行するサーバーに対応したマシン・データ・ソース。
 - リアルタイム・イベントを使用する環境。
 - クライアントからサーバーのマシンが使用するシステム・データ・ソース。

〈旧日付別ジョブ・マスター削除〉レポートの実行

このバッチ・アプリケーションを実行すると、サーバーに投入済みの古い印刷ジョブの一覧を印刷することができます。レポートのみ作成するか、レポートを作成してF986110 テーブルから古いレコードを削除するかを選択できます。

このバッチ・アプリケーションは、レポート・ドライバによって起動されます。すべてのデータ選択は、バッチ・アプリケーションから入力しますが、処理オプションは実際のアプリケーション(R9861102)からではなく、レポート・ドライバ(R9861101)から入力します。

▶ ジョブ・マスター削除のデータ選択を設定するには

〈システム・アドミニストレーション・ツール〉メニュー(GH9011)から〈バッチ・バージョン〉プログラム(P98305)を選択します。

1. 〈バッチ・バージョンの処理 - 使用可能なバージョン〉で、[バッチ・アプリケーション]フィールドに”R9861102”と入力して[検索]をクリックします。
2. 次のバージョンの1つをダブルクリックします。
 - XJDE0001 = Jobs With All Status(すべてのジョブ)
 - XJDE0002 = Jobs With Done Status(完了ジョブ)
 - XJDE0003 = Jobs With Error Status(エラーのあったジョブ)
 - XJDE0004 = Jobs With Wait Status(待機中ジョブ)
3. 〈バージョン・プロンプト〉で、[データ選択]オプションをチェックして[投入]をクリックします。
4. 〈データ選択〉で、適切なカラムから選択して比較するレコードを指定します。
5. [OK]をクリックします。

▶ 〈旧日付別ジョブ・マスターの削除〉プログラム(R9861101)を実行するには

〈上級操作〉メニュー(GH9012)から〈旧日付別ジョブ・マスターの削除〉(R9861101)を選択します。

1. 〈バッチ・バージョンの処理 - 使用可能なバージョン〉で、実行するレポートに対応するバージョンを選択します。
2. [選択]をクリックします。
3. 〈バージョン・プロンプト〉で[投入]をクリックします。
4. 処理オプションで、次の情報を入力して[OK]をクリックします。
 - データ・ソース(デフォルト・タブ)
 - 古い日付(デフォルト・タブ)
 - 制御モード(デフォルト・タブ)
 - ジョブ状況 UBE(バージョン・タブ)

〈OCM カテゴリの更新/削除〉レポートの実行

このバッチ・アプリケーションを実行すると、指定した OCM テーブル(F986101)にメンバー・オブジェクトを追加したりテーブルから削除することができます。オブジェクトを個別に追加または削除する代わりに、同じカテゴリのすべてのメンバーの OCM マッピングを追加、削除できます。このバッチ・アプリケーションの処理オプションでは、マッピングを追加または削除するかどうかを指定したり、使用するパス・コード、環境、データ・ソースを指定できます。

▶ <OCM カテゴリの更新/削除>プログラムを実行するには

<上級操作>メニュー(GH9012)から<OCM カテゴリの更新/削除>(R986101B)を選択します。

1. <バッチ・バージョンの処理 - 使用可能なバージョン>で、バージョン XJDE0001 をハイライトして[選択]をクリックします。
2. <バージョン・プロンプト>で、[データ選択]をクリックして[投入]をクリックします。
3. <データ選択>で、追加または削除するレコードを指定して[OK]をクリックします。
4. 処理オプションで[OCM モード]タブをクリックして、次の情報を入力します。

- 処理するモード(テストまたは最終)

テスト・モードでレポートを実行するには 1 を入力します。レポートは印刷されますが、OCM テーブルは更新されません。最終モードでレポートを実行するには 2 を入力します。レポートが印刷され、OCM テーブルが更新されます。

- 使用するモード(追加または削除)

カテゴリにオブジェクトの OCM マッピングを追加するには 1、削除するには 2 を入力します。

- マッピングの一時変更

- 環境とユーザーに対して既にマッピングされているオブジェクトの OCM マッピングを削除するには 1 を入力します。競合する OCM マッピングのレコードを削除しないで無効にするには、2 を入力するかブランクのままにします。

5. [OCM 設定]タブをクリックして、次の情報を入力します。

- OCM マッピングの追加または削除時に使用するパス・コード

パス・コードを入力すると、入力したパス・コードのすべての環境でマッピングが追加または削除されます。処理オプションで環境名を入力した場合、パス・コードを入力する必要はありません。このフィールドをブランクにすると、OCM マッピングはパス・コードにかかわらず追加、削除されます。

- OCM マッピングの追加または削除時に使用する環境

このフィールドをブランクにすると、OCM マッピングは環境にかかわらず追加、削除されます。処理オプションでパス・コードを入力した場合、そのパス・コードが使用されます。環境とパス・コード両方の処理オプションがブランクの場合、OCM マッピングは環境詳細(F00941)テーブルにあるすべての環境で追加または削除されます。

- ユーザー・クラス/グループ

OCM マッピングを削除する場合にこのフィールドをブランクにすると、ユーザーやグループにかかわらずすべての OCM レコードが削除されます。OCM マッピングを追加する場合、このフィールドは入力必須です。

- データ・ソース

OCM マッピングを削除する場合にこのフィールドをブランクにすると、データ・ソースにかかわらずすべての OCM レコードが削除されます。OCM マッピングを追加する場合、このフィールドは入力必須です。

サーバー上のアプリケーション・ロジックの分割

アプリケーションのロジックは、ビジネス関数のコンポーネントを個別にマップするかまとめてマップすることにより、分割してリモートで実行することが可能で、ワークステーションの代わりにアプリケーション・サーバーまたはエンタープライズ・サーバー上で実行することができます。

マスター・ビジネス関数やビジネス関数を含む、特定のビジネス関数コンポーネントを再展開すると、ワークステーションでは分散処理のパフォーマンスが向上し、同時にネットワーク・トラフィックが減少します。この再展開には、オブジェクト構成マネージャ(OCM)の標準メソッドロジーを使ったオブジェクトの再マッピングが含まれます。

このような構成の例としては、Windows シン・クライアント/ヘビー・サーバーや Java シン・クライアント/ヘビー・サーバーのモデルが挙げられます。どちらのモデルでも、アプリケーション・スペックはクライアント、ビジネス関数コンポーネントはサーバー上にあります。両者の主な違いは、Windows では JDENet コミュニケーション・ミドルウェアを使用するのに対し、Java では JDENet/CORBA ミドルウェアを使用する点です。

すべてのビジネス関数コンポーネントが分割できるよう設計されていますが、最大の利点はマスター・ビジネス関数(MBF)を分割できることです。

トランザクションの検証やデータベースの更新といった処理を行う MBF の概念に基づいて、トランザクション指向のアプリケーションが開発されています。トランザクション指向アプリケーションの大部分の I/O サービスは MBF によって実行されます。トランザクションのビジネス・ロジックの大部分を分散化し、アプリケーション・サーバー上で実行するために MBF を分割することによって、ネットワーク・トラフィックを最低限に抑えて分散環境や WAN 環境でのアプリケーションのパフォーマンスを大幅に向上させることができます。

MBF がクライアントで処理される 2 階層設定では、WAN 上のクライアントとサーバー間で多くのやり取りが行われます。クライアントとデータ・サーバー、アプリケーション・サーバーまたはエンタープライズ・サーバーで構成される 3 階層設定では、LAN 上の 2 つのサーバー間でトランザクション処理が発生します。このため、WAN 上におけるクライアントとサーバー間のやり取りは、クライアントでの入力とサーバーからの結果の送信だけで済みます。この 3 階層構成によって、WAN 上のトラフィックを大幅に減少させることができます。

次の図では、MBF をクライアントで処理する 2 階層ネットワークと、MBF をサーバーで処理する 3 階層ネットワークのトラフィックを比較しています。

2 階層ネットワークの一般的なトラフィック

2 階層構成では、GUI、イベント・ルール、MBF は通常クライアントで処理され、データはサーバーに保管されます。WAN 上のクライアントとサーバー間では通常次のような処理が発生します。

- レコードのフェッチ(クライアントからサーバーへ)
- レコードの送信(サーバーからクライアントへ)
- データ・フォーマットの検証(クライアントからサーバーへ)
- フォーマット OK(サーバーからクライアントへ)
- レコード詳細の送信(クライアントからサーバーへ)
- 詳細 OK(サーバーからクライアントへ)
- トランザクションの終了(クライアントからサーバーへ)

3 階層ネットワークの分割トラフィック

3 階層構成では、GUI やイベント・ルールはクライアントで処理されますが、MBF はアプリケーション・サーバーまたはエンタープライズ・サーバー上で処理されます。データはデータベース・サーバーで保管されます。WAN 上のクライアントとアプリケーション・サーバーまたはエンタープライズ・サーバー間では、次の処理が発生します。

- データ入力または処理要求(クライアントからサーバーへ)
- 処理結果の送信(サーバーからクライアントへ)

LAN 上のアプリケーション・サーバーまたはエンタープライズ・サーバーとデータベース・サーバー間では、ローカルで次の処理が発生します。

- レコードのフェッチ
- レコードの送信
- データ・フォーマットの検証
- フォーマット OK
- レコード詳細の送信
- 詳細 OK
- トランザクション終了

マスター・ビジネス関数(MBF)の処理

次の処理(イベント)は、一般的なアプリケーションでマスター・ビジネス関数(MBF)がどのように使用されるかを示しています。この例では(受注オーダー入力)アプリケーションを使用します。

- 受注オーダー最終行
受注オーダーの最終行に達すると最初のイベントが発生し、クライアントのアプリケーションから jdeCallObject API が呼び出されます。このコマンドによって MBF にメッセージが送信されます。メッセージには、行のデータ(データ構造体として)が含まれます。アプリケーションは、関連するデータとともに非同期的にメッセージを送信します。つまり、アプリケーションではメッセージを送信すると同時に次の行に処理を進めます。
- MBF による行メッセージの取得
MBF が行のデータを含む JDENet メッセージを受け取るとこのイベントが発生します。行データは、サーバーの共有メモリにキャッシュとして保存されます。
- MBF による行の拡張および編集
MBF が受注オーダー行を取得して検証する際にこのイベントが発生します。行の取得、検証に必要なデータは、通常 LAN 上のローカルでアクセスされます。このデータ要求はデータベース独自の SQL コールによって行われ、適切なオープン・データベース・コネクティビティ(ODBC)または Oracle コール・レベル・インターフェイス(OCI)によって送信されます。

- MBF からクライアント・アプリケーションへのメッセージの返信

この 4 番目のイベントは、MBF が受注オーダー行を取得、検証して、エラーがある場合はエラー・コードとともに、取得した行をクライアントに返す際に発生します。この返信メッセージは JDENet を使って送信されます。受注オーダーに関連するすべての行に対して、1~4 のイベントが非同期的に繰り返されます。

- 受注オーダーの終了 (OK ボタン)

このイベントは、ユーザーがすべての受注オーダー行の入力を完了したことを示します。検証済みのすべての行がクライアントに返された後、ユーザーが [OK] をクリックすることによってこのイベントが発生します。ユーザーが [OK] をクリックすると、MBF にトランザクション終了メッセージが送信されます。クライアントではすぐに次のトランザクションを入力することができます。

- MBF による全トランザクションの処理

この処理は、すべてのトランザクション行が保管されている共有メモリのキャッシュを MBF で非同期的に読み込み、データベースを更新する際に行われます。

- データベースの更新と MBF クリーンアップ

ODBC および OCI 経由で通常ローカルでデータベースを更新 (すべてのトランザクションを保存) し、保存したトランザクションのキャッシュを共有メモリからクリアします。

MBF をサーバーで実行するようマップすると、1 台のサーバー (エンタープライズ・サーバー) または 2 台の LAN 接続マシン (アプリケーション・サーバーとデータ・サーバー) 間で大量のデータ送受信が行われます。つまり、トランザクションは最低限のネットワーク・トラフィックで処理されることとなります。このタイプのアプリケーション・トランザクションは、分散環境や WAN 環境で理想的なパフォーマンスを実現します。

分散アプリケーション・ロジックのサーバー処理

一般的なヘビー・クライアントに比べ、マスター・ビジネス関数やその他ビジネス関数などのコンポーネントをサーバーで実行するようにアプリケーション・ロジックを分割すると、サーバーで実行されるプロセスが増加し、多くのユーザー・セッションが必要になります。このように負荷が増大した状態でサーバーの処理を制御するために、jde.ini で設定すべき重要なパラメータがあります。

参照

- jde.ini パラメータの設定については、『システム・アドミニストレーション』ガイドの「ミドルウェア」

例: 処理規則 1 - クライアント上の JDENet

クライアントでは、JDENet 関数は jdenet.dll と呼ばれるダイナミック・リンク・ライブラリ (DLL) にあります。これらの関数はエクスプローラ・プログラム (Oexplore.exe) から呼び出されます。つまり、これらの関数は別々のプロセスやサービスとして実行されるのではなく、Oexplore.exe プロセスの中で実行されます。

クライアントが (OCM で定義された) サーバーと通信セッションを開始すると、クライアントはサーバー上の特定の JDENet プロセスと通信するよう割り当てられます。この割当ては、セッションが終了する持続されます。つまり、ユーザーがサインオンしている限り同じ論理接続が維持されます。

例:処理規則 2 - サーバー上の JDENet

サーバー上で複数の JDENet プロセスを実行するよう設定できます。サーバーにある jde.ini ファイルのパラメータによって、そのサーバーで開始可能な JDENet プロセス数やネットワーク接続総数が指定されています。

複数の JDENet プロセスを指定した場合、JDENet プロセスが最大数になるまで、セッション要求に応じて 1 対 1 ベースでプロセスが開始されます。後続のセッションは順次 JDENet プロセスに割り当てられます。各 JDENet プロセスでは、メッセージは一度に 1 つずつ待ち行列に入って処理されます。たとえば次のようになります。

Session 1:セッション 1: JDENET_1 JDENET_1
Session 2:セッション 2: JDENET_2 JDENET_2
Session 3:セッション 3: JDENET_3 JDENET_3
Session 4:セッション 4: JDENET_1 JDENET_1
Session 5:セッション 5: JDENET_2 JDENET_2
Session 6:セッション 6: JDENET_3 JDENET_3

上記の例は、下の表に示す特徴に基づいています。

特徴	値	jde.ini パラメータ
JDENet プロセス数	3	[JDENet] maxNetProcesses=3
サーバーごとの接続数	nnn	稼働環境によって異なります。接続数はできるだけ余裕を持って設定してください。 例: [JDENet] maxNetConnections=800
セッション数	6	なし

例:処理規則 3 - サーバー上のカーネル

さまざまな機能を取り扱うために、サーバー上でさまざまなタイプのカーネル処理を実行することができます。jdeCallObject API を使って分散オブジェクトを処理するのは、タイプ 2 のカーネルです。サーバーにある jde.ini ファイルのパラメータで、特定のタイプのカーネルを各自いくつ開始できるかが指定されています。

カーネルに接続できる有効なセッションの総数を直接コントロールすることはできません。これは、使用可能なカーネル・タイプにその都度セッションが動的に割り当てられるためです。ただし、クライアントからサーバー、サーバーからクライアントへの接続数に基づいて十分なカーネル数を指定することで、カーネルごとのユーザー数を間接的にコントロールすることは可能です。たとえば、十分なカーネルを指定して、各カーネルに 1 ユーザーまたは 2 ユーザーを設定することができます。

必要なカーネル数を決定する際、各カーネル・プロセスではサーバーのメモリが使用されることに注意してください。使用されるメモリの正確な容量は、パフォーマンス面ほど重要ではありません。パフォーマンスが極端に低くならないようにするには、1つのカーネルを同時に使用できるユーザー（またはMBF）の数を考慮してください。

カーネル・プロセスは、JDENet ネットワーク通信セッションと同様の方法で開始されます。各カーネル・タイプについて、開始可能な最大数のカーネルが開始されるまで、新しいセッションごとに新しいカーネルが開始されます。最大数に達すると、特定のカーネル・プロセスに順次セッションが割り当てられます。

タイプ 2 の各カーネルは、jdeCallObject API の要求を一度に 1 つずつ待ち行列に入れて処理します。1つのカーネルに複数のセッションが割り当てられている場合、jdeCallObject ルーチンが完了すると、そのカーネル・タイプの待ち行列から次の要求が取り出されます。

サーバーの jde.ini ファイルを設定する際は注意が必要です。このカーネル調整方法ではと、本来のジョブを実行中のジョブを分散することはできません。次の例に示すように、カーネル 2 に負荷の大きい受注オーダー入力処理がロードされる一方で、カーネル 1 は負荷の小さい軽い処理で余裕があります。ハードウェアの階層を決定する際、この負荷調整を考慮すると効果的です。例：

Session 1:セッション 1: JDENET_1, Kernel_1 JDENET_1、カーネル 1
Session 2:セッション 2: JDENET_2, Kernel_2 JDENET_2、カーネル 2
Session 3:セッション 3: JDENET_3, Kernel_1 JDENET_3、カーネル 1
Session 4:セッション 4: JDENET_1, Kernel_2 JDENET_1、カーネル 2
Session 5:セッション 5: JDENET_2, Kernel_1 JDENET_2、カーネル 1
Session 6:セッション 6: JDENET_3, Kernel_2 JDENET_3、カーネル 2

上記の例は、下の表に示す特徴に基づいています。

特徴	値	jde.ini パラメータ
JDENet プロセス数	3	[JDENet] maxNetProcesses=3
セッション数	6	なし
タイプ 2 のカーネル数	2	[JDENet_KERNEL_DEF2] maxNumberOfProcesses=2
タイプ 2 のカーネルを要求するセッション数	6	なし

参照

- カーネルについて詳しくは『システム・アドミニストレーション』ガイドの「ミドルウェア」

ロジックを分散しない場合

次の場合はロジックを分散しないでください。

- サーバーにマップされているビジネス関数によって、そのビジネス関数を呼び出すアプリケーションから起動する別のビジネス関数に必要なキャッシュが作成される場合、それらのビジネス関数も同じロジック・サーバーにマップする必要があります。

- クライアント専用のビジネス関数に、Windows NT が搭載されたワークステーションでのみ有効な GUI または Windows 機能が含まれている場合、ロジック・サーバーに分散されたビジネス関数からクライアントでのみ実行可能なビジネス関数を呼び出すことができない場合があります。

注:

マスター・ビジネス関数がロジック・サーバーへマッピングされている場合、サーバーは自身のサーバー・マッピング・データ・ソースを参照して、処理する OCM テーブルを確定します。たとえば、マスター・ビジネス関数がロジック・サーバーへマッピングされている場合、そのマスター・ビジネス関数によって呼び出されるロジックは、OCM 内のサーバー・マッピング・データ・ソースに基づいて呼び出されます。

ロジックの分散に関する問題

分散されたロジックには次のような問題があります。

- バッチ制御機能では、分散されたロジックに問題が発生する場合があります。アプリケーションを終了する際、バッチ制御ビジネス関数により、入力結果の確認のためのフォームが表示されます。アプリケーションがバッチ制御ビジネス関数を直接呼び出す場合は問題ありません。ただし、マッピングされたマスター・ビジネス関数は、サーバーから GUI 表示を実行できないため、バッチ制御ビジネス関数を呼び出せません。たとえば、ロジックがサーバーにマップされている場合、購買、在庫アプリケーションではバッチ制御を使用できません。
- ローカライズ、カスタム・ロジックをインプリメンテーションする場合、GUI または Windows 機能を含むロジックをマップしないでください。

ビジネス関数処理の問題

コンフィギュラブル・ネットワーク・コンピューティング(CNC)ソリューションを使用して、開発者や管理者はロジック処理のためにビジネス関数をアプリケーション・サーバー(複数可)にマッピングできます。サーバーで問題が発生すると、ビジネス関数を実行するためにアプリケーション・サーバーへの再接続が試みられます。サーバーに再接続してビジネス関数が実行できると、処理は中断なく続行されます。

ただし、次のような状況ではビジネス関数の処理が複雑になる場合があります。

- サーバーのプロセスが停止しているため、クライアント・ワークステーションはアプリケーション・サーバーに再接続できない。
- ビジネス関数の処理で、プロセスが停止しているアプリケーション・サーバー上にキャッシュまたはエラー・ログ情報を作成する。
- ビジネス関数が原因でサーバー上のプロセスが停止する。
- サーバーがシャットダウンされ、そのサーバーがサーバー・クラスタの一部であるため、クライアント・ワークステーションからアプリケーション・サーバーに再接続できない。

クライアント・ワークステーションがサーバーと通信できない場合、ビジネス関数処理はセカンダリ・サーバーに割り当てられます。CallObject コードには、基本サーバーとセカンダリ・サーバーの名前が指定されています。

注:

インストール時のデフォルト設定では、セカンダリ・サーバーは定義されていません。サーバーを定義する場合、OCM マッピングを変更する必要があります。セカンダリ・サーバーが定義されていない場合、フェイルオーバーが発生すると、ビジネス関数処理は問題が発生したサーバーからクライアント・ワークステーションにマップされます。

ビジネス関数処理で、プロセスが停止しているアプリケーション・サーバー上にキャッシュが作成された場合、クライアント・ワークステーションはアプリケーション・サーバーに再接続しますが、ユーザーはアプリケーションを終了して再度実行する必要があります。

ビジネス関数実行時にサーバー上でプロセスが停止した場合、クライアント・ワークステーションはサーバーに再接続します。ビジネス関数が原因でjdenet_k プロセスが停止したため、ビジネス関数の呼出しは失敗します。

クライアント・ワークステーションがサーバー・クラスタ内のサーバーと通信できない場合、そのサーバーをクラスタの一部と見なして再接続の試行が続けられます。クラスタのサーバーから別のクラスタのサーバーに処理を移すには数分かかる場合があります。

CNC ソリューションでは、サーバーのダウンやカーネル・プロセスの停止によってアプリケーション・サーバーのロジック処理が終了した場合でも、ビジネス関数の失敗に対応して処理を続行できる方法が提供されています。さらに、フェイルオーバーが発生した場合はjde.log にメッセージが書き込まれ、トラブルシューティングをすることができます。

サーバーへの接続の失敗

ビジネス関数がサーバーへ接続できない原因のメカニズムは、ネットワークでのサーバーの構成によります。このセクションでは、次のような設定エラーについて説明します。

- 非クラスタ・サーバー構成のサーバーへの接続の失敗
- クラスタ構成のサーバーへの接続の失敗

非クラスタ・サーバー構成のサーバーへの接続の失敗

非クラスタ・サーバー構成では、プライマリ・サーバーに接続できない場合、ビジネス関数処理の割当てが再度行われます。次のステップは、アプリケーション・サーバー上で実行するためにビジネス関数を呼び出す初期段階で行われる処理です。

1. ユーザーがサーバー上でビジネス関数を呼び出します。
2. サーバーがプライマリ・サーバーからセカンダリ・サーバーまたはクライアント・ワークステーションへフェイルオーバーしているかチェックします。
3. 処理が他のサーバーに割り当てられている場合、ビジネス関数は再度マップされ、ビジネス関数を実行するセカンダリ・サーバーまたはクライアント・ワークステーションに CallObject メッセージが送信されます。
4. サーバーがフェイルオーバーしていない場合は、ビジネス関数を実行する元のサーバーに CallObject メッセージが送信されます。

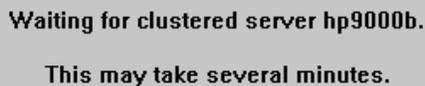
ビジネス関数処理の第 2 段階では、アプリケーション・サーバーまたはクライアント・ワークステーション上でロジックの実行が試みられます。次のステップは、この第 2 段階で行われる一連の処理です。

5. 元のサーバーまたは代替りのサーバーでビジネス関数がエラーなく実行された場合、要求は処理されています。
6. クライアント・ワークステーションの要求がサーバーで正しく処理されない場合、再接続回数をカウントしながら再接続が試みられます。
7. 再接続カウンタの値が 1 より大きい場合、ビジネス関数は失敗します。再接続カウンタの値が 1 より小さい場合は、サーバーに再接続してビジネス関数を実行しようとします。
8. サーバーに再接続できない場合、セカンダリ・サーバーが定義されている場合はセカンダリ・サーバーに、定義されていない場合はクライアント・ワークステーションに要求が転送されます。

サーバー上にキャッシュが作成されている場合、アプリケーションを終了して再度実行する必要があります。

クラスタ構成のサーバーへの接続の失敗

クラスタ構成のサーバーがダウンしてビジネス関数が失敗した場合、セカンダリ・サーバーまたはクライアント・ワークステーションにフェイルオーバーする代わりに、そのクラスタで新しいマシンが使用可能になるまで待機して、ビジネス関数の要求を再度送信します。再接続を試行している間は、次の一時ウィンドウが表示されます。



Waiting for clustered server hp9000b.
This may take several minutes.

このウィンドウは 1 分ごとにリフレッシュされ、クライアントがクラスタ・サーバーに再接続されるまで表示されます。

サーバーがダウンする前にそのサーバーにビジネス関数のキャッシュが作成された場合、サーバー・クラスタにビジネス関数要求は送信されません。この場合、アプリケーションを終了してビジネス関数を再度実行する必要があります。

ビジネス関数のロード失敗

サーバーがクライアント・ワークステーションからビジネス関数を実行するよう要求された場合、サーバーではビジネス関数をロードする必要があります。この処理は次の 2 つの理由で失敗する場合があります。

- ビジネス関数が保管されているライブラリをロードできない。
- ビジネス関数のアドレスを取得できない。

ビジネス関数が保管されているライブラリをロードできない

サーバーがビジネス関数ライブラリをロードできない場合、クライアント・ワークステーション上に次のメッセージが表示されます。このメッセージはそのマシンの jde.log ファイルに書き込まれます。



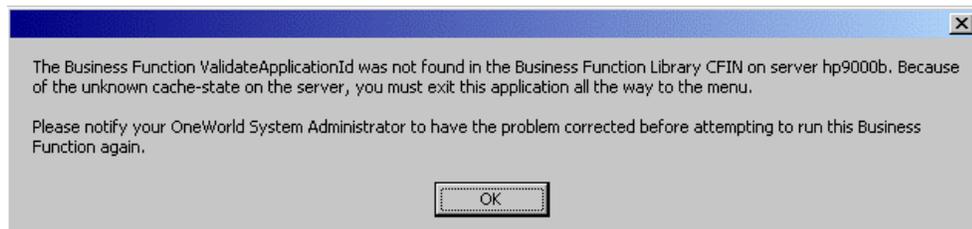
ライブラリがロードできない原因として次のことが考えられます。

- パッケージ・ビルドでビジネス関数ライブラリのビルドに失敗した。
- ライブラリが誤って削除されたか名前を変更された。
- 権限に関して問題がある。

ライブラリのロードが失敗した場合、メニューに戻るまでアプリケーションを終了して、システム管理者に連絡してください。ビジネス関数の再実行を試みる前に、問題が解決されていることを確認してください。

ビジネス関数のアドレスを取得できない

サーバーがライブラリ内にあるビジネス関数のアドレスを取得できない場合、クライアント・ワークステーション上に次のメッセージが表示されます。このメッセージはそのマシンの jde.log ファイルに書き込まれます。



サーバーがビジネス関数のアドレスを取得できない原因として次のことが考えられます。

- パッケージ・ビルドでビジネス関数を含むモジュールが作成できなかったため、そのモジュールがビジネス関数ライブラリに含まれなかった。
- サーバーより新しいパッケージがクライアント上にあり、ビジネス関数がサーバーにはなくてクライアントにある。

このエラーが発生した場合、メニューに戻るまでアプリケーションを終了して、システム管理者に連絡してください。ビジネス関数の再実行を試みる前に、問題が解決されていることを確認してください。

ビジネス関数実行時の失敗

ビジネス関数そのものが原因でサーバー上のプロセスが停止する場合があります。この場合、ビジネス関数が問題の原因であることを示すダイアログ・ボックスが表示されます。

このダイアログ・ボックスが表示された場合、OCM マッピングを変更するか、ビジネス関数のバグを修正する必要があります。

サーバー上のキャッシュのリセット

最初にビジネス関数が実行されなかった場合、サーバー上にキャッシュが作成されているかどうか確認されます。キャッシュが作成されておらず、プライマリ・サーバーへの再接続の試みが失敗した場合、セカンダリ・サーバーかクライアント・ワークステーション上でビジネス関数の実行を試みます。

サーバー上にキャッシュが作成されている場合、アプリケーションを終了して再度実行するようメッセージが表示されます。このメッセージは、クライアントの `jde.log` ファイルに書き込まれます。

サーバー上にキャッシュを作成するのは、ビジネス関数の重要な処理です。1 つのビジネス関数の実行時に作成されるキャッシュのデータは、後続の関数でも使用されます。たとえば、最初のビジネス関数がキャッシュを作成、初期化し、2 つ目の関数がデータを追加し、3 つ目の関数がデータにアクセスしデータベースにそのデータを挿入します。

最初のビジネス関数がキャッシュを作成した後でサーバーのプロセスが停止し、クライアント・ワークステーションがそのキャッシュを含むサーバー上のプロセスと通信できない場合、次のビジネス関数はそのキャッシュにアクセスできません。したがって、このような場合には強制的にアプリケーションが終了して再起動されます。

注:

UBE およびテーブル変換では、サーバー上でキャッシュが作成された場合でも、失敗後にビジネス関数の処理が続行されます。

環境のセットアップ

環境とは、データおよびオブジェクトのロケーションを示すポイントの集合です。環境定義には、パス・コードとオブジェクト構成マネージャ(OCM)マッピングが含まれます。

環境の理解

環境は、オブジェクト構成マネージャ(OCM)マッピングとパス・コードから構成されます。OCM マッピングによって次のことがわかります。

- ユーザーのロジック・オブジェクトはどこで処理されるのか
- ユーザーのデータ・オブジェクトはどこにあるのか

環境に関連付けられているパス・コードによって次のことがわかります。

- オブジェクトはどのディレクトリにあるのか

環境には次のような設定項目があります。

- 分散データと分散ロジックの環境
データの保管場所とアプリケーションを処理する場所を決定します。
- 処理環境モード
オフライン処理、ダイレクト接続、バッチオブワンの 3 つのトランザクション処理をサポートします。
- オブジェクト環境グループ
本稼動環境、開発環境、プリステイン環境など、どのオブジェクトを使用するかを管理します。

環境定義の保管場所

環境定義は次の 5 つのテーブルに保管されます。

- ライブラリ・リスト管理(F0093)
- ライブラリ・リスト・マスター(F0094)
- 環境詳細 - OneWorld (F00941)
- OCM (F986101)
- オブジェクト・パス・マスター(F00942)

ライブラリ・リスト管理テーブル(F0093)

ライブラリ・リスト管理テーブル(F0093)には、各ユーザーが使用可能な環境が含まれています。ユーザーがログオンできるよう、各ユーザーに少なくとも 1 つの有効な環境に割り当てる必要があります。ライブラリ・リスト管理テーブルで各ユーザーに割り当てられた環境は、ERP 9.0 起動時にワークステーション上の ERP 9.0 のディレクトリに対して検証されます。ログオン時には、ユーザーがログオンを許可されていて、使用しているマシンにインストールされている環境のみが表示されます。

〈ユーザー・プロファイル〉アプリケーションの〈環境の改訂〉フォームで、特定の環境にサインオンできるユーザーを指定することができます。

ライブラリ・リスト・マスター(F0094)

ライブラリ・リスト・マスター(F0094)には、環境名とその記述が含まれています。

環境詳細 - OneWorld テーブル(F00941)

このテーブルには、環境名とパス・コード・マスター(F00942)の関連するパス・コードが含まれています。

各環境にはパス・コードを関連付ける必要がありますが、複数の環境で同じパス・コードを共有することができます。たとえば、本稼動用オブジェクトを使用する2つの環境で、メイン・サーバーと部門サーバーにあるデータを処理するユーザーのグループに異なるデータのロケーションをマップすることができます。

OCM テーブル(F986101)

オブジェクト構成マネージャ(OCM)テーブルには、各環境のデータおよびロジック・オブジェクトのマッピングが含まれています。マッピングが同じ環境であっても、各環境のレコードをこのテーブルに設定する必要があります。2つの環境で、異なるパス・コードを持ちながら、データとロジックに対して同じマッピングを使用することもできます。たとえば、開発者とテスト担当者が異なるオブジェクトを使用しながら、同じデータベースのデータを使ってローカルでロジックを実行することができます。

オブジェクト・パス・マスター(F00942)

このテーブルには、パス・コードのセントラル・オブジェクト、パス・コードと関連付けられたリリース、その他の詳細に関するロケーションが保管されています。

環境の処理

新しい環境を追加するか既存の環境をコピーして、新しい環境を作成することができます。〈インストール・プランナ〉および〈環境ディレクタ〉アプリケーションのカスタム環境を作成する機能を使用すると、このプロセスが簡単になります。

オブジェクト構成マネージャ(OCM)マッピングは環境の重要な要素であるため、新しい環境を作成するか既存の環境をコピーするか決定する際には、これらのマッピングを決定することが最初のステップです。

環境をコピーすると、その環境の OCM マッピングも同時にコピーされます。作成する環境の OCM マッピングが既存の環境のマッピングと非常に似ている場合は、既存の環境をコピーして新しい環境を作成すると便利です。たとえば、開発環境とテスト環境のマッピングが似ている場合、環境をコピーして異なるマッピングのみ変更する方が簡単です。

既存の環境のオブジェクト・マッピングを使用しない場合は、それらの環境をコピーするのではなく新しい環境を追加してください。たとえば、オフライン処理を設定する場合、オフライン処理ユーザーの OCM マッピングはダイレクト接続ユーザーのものとは異なるため、新しい環境を追加することをお勧めします。

注:

既存の環境をコピーしないで環境を追加する場合、OCM マッピングを手作業で作成する必要があることに注意してください。このため、通常は既存の環境をコピーして OCM マッピングを修正する方がすべてを手作業で作成するより簡単です。

環境の追加

既存の環境の OCM マッピングを使用しない場合は、既存の環境をコピーするのではなく、新しい環境を追加してください。

参照

- OCM マッピングの追加については、『CNC インプリメンテーション』ガイドの「オブジェクト構成マネージャ(OCM)の処理」

環境のコピー

既存の環境をコピーして新しい環境を作成する場合、コピーする環境に関連付けられている OCM マッピングも同時にコピーされます。また、環境のコピー時にその他の OCM マッピングもコピーするよう処理オプションを設定することができます。

既存の環境から環境をコピーする以外に、テンプレートとして使用できるサンプルの本稼動環境を提供する Installation Planner (インストール・プランナ)を使用することもできます。これらのテンプレートには、すべてのテーブルの推奨マッピングが含まれています。

▶ 環境をコピーするには

〈環境〉メニュー(GH9053)から〈環境設定マスター〉プログラム(P0094)を選択します。

1. 〈環境の処理〉フォームで、[検索]をクリックします。

[J.D. Edwards 環境のみを表示]オプションがオンの場合、このフォームには環境詳細テーブル(F00941)にある環境のみ表示されます。

2. グリッドのローを選択して[ロー]メニューから[環境のコピー]をクリックします。
3. 〈環境のコピー〉で、[新しい環境]フィールドに環境名を入力します。

コピーする環境の*PUBLIC の OCM マッピングのみコピーするには、[*PUBLIC レコードのみコピー]オプションをオンにします。*PUBLIC だけでなく、環境および個々のオブジェクトのマッピングをコピーするには、このオプションをオフにします。

4. [OK]をクリックします。

ライブラリ・リスト・マスター(F0094)に新しい環境が追加されます。環境詳細テーブル(F00941)に新しい環境のレコードが作成されると同時に、システム・データ・ソースの OCM テーブル(F986101)にも一連のマッピングが作成されます。処理オプションの設定によって、サーバー・マップ・データ・ソースにもマッピングが作成される場合があります。

5. 新しい環境のパス・コードを変更するには、〈環境の処理〉フォームで環境を選択してパス・コードを変更します。
6. 必要に応じてその他の OCM マッピングを変更します。

参照

- OCM マッピングの追加について詳しくは、『CNC インプリメンテーション』ガイドの「オブジェクトのマッピング」

フィールド記述

記述	用語解説
新しい環境	<p>特定のライブラリ・リストと関連した名前。J98INITIA 初期プログラムは、これらのライブラリ・リスト名を使用してユーザーがサインオンできる環境を制御します。ライブラリ・リストのこれらの構成は、ライブラリ・リスト・マスター(F0094)で管理されています。</p> <p>J.D. Edwards の場合、J.D. Edwards の実行に使用する有効な環境がこのフィールドに示されます。環境は、パスコード(オブジェクト)とデータ・ソース(データ)を含んでいます。その 2 つが共に機能してシステム内での作業が可能となります。</p>

新しい環境の新規パス・コードへのコピー

新しい環境を新たなパス・コードにコピーするには、新たなパス・コードにデモ・データをロードして新しいテスト環境を作成する方法を把握する必要があります。

▶ 環境を追加するには

1. 〈環境の処理〉で、[追加]をクリックします。

[J.D. Edwards 環境のみを表示]オプションがオンの場合、このフォームには環境詳細 - OneWorld テーブル(F00941)にある環境のみ表示されます。
2. 〈環境の改訂〉フォームで、次のフィールドに値を入力して[OK]をクリックします。
 - 環境名
 - 記述
 - パス・コード
 - リリース No.
 - ジャスト・イン・タイム・インストール
 - 開発者 (Y/N)

ライブラリ・リスト・マスター(F0094)に新しい環境が追加されます。環境詳細テーブル(F00941)には、この環境のレコードが新たに作成されます。

3. 次のオプションをクリックします。

- WAN 設定済み環境
- デタッチ・モード

〈環境の改訂〉フォームで入力したパス・コードが環境マスター(F0094)にない場合、〈データ・ソースの選択〉フォームが表示されます。

4. 〈データ・ソースの選択〉で、[OL データ・ソース]フィールドに値を入力します。

システムでは、このフォームの入力データを使用して、新しい環境のオブジェクト・ライブラリアン OCM マッピングが作成されます。OCM テーブル(F986101)に必要なレコードが書き込まれます。

〈環境の改訂〉フォームに入力したパス・コードが既に他の環境レコードにある場合、その環境のオブジェクト・ライブラリアン OCM レコードに関連付けられたデータ・ソースが確定されます。この値を使って OCM テーブル(F986101)に新しいレコードが作成されます。

▶ 新しい環境を新規パス・コードにコピーするには

〈環境〉メニュー(GH9053)から〈環境設定マスター〉プログラム(P0094)を選択します。

1. 〈環境の処理〉で、[検索]をクリックします。
[*Public レコードのみ]オプションが正しく設定されていることを確認してください。
2. 必要な環境に類似した環境を選択して[ロー]メニューから[環境のコピー]を選択します。
3. 〈環境のコピー〉で、[新しい環境]フィールドに入力し、[*Public レコードのみコピー]オプションをオン(またはオフ)にします。
4. [OK]をクリックして〈環境の処理〉フォームに戻ります。
5. 〈環境の処理〉で、[検索]をクリックします。
6. 新しい環境を選択して、ステップ 1 で作成した新しいパス・コードに変更します。
7. 終了したら[OK]をクリックします。
8. 〈環境〉メニュー(GH9053)から環境設定マスター(P0094)を再度選択して、パス・コードが変更されていることを確認します。

▶ テスト・バッチ・ファイルを作成するには

1. LOADALL と JDESET ファイルのすべての行をコメント行にして、本稼働環境用セットと同じ形式でテスト環境用セットを作成します。
2. LOADPD をコピーして LOADTEST.BAT ファイルを作成し、必要なパラメータを JDESET.BAT で指定したパラメータに変更します。
3. LOADALL.BAT を実行します。
4. 現行のプランナ環境に必要なデータ・ソースを追加します。

ロジック・データ・ソースとして、Business Data Test、Central Objects Test、OneWorld Local Test、Control Tables Test が必要です。

デプロイメント・サーバー・ソースが Cental Objects データ・ソースおよび LOADALL のスペックと一致していることを確認してください。

▶ **AS/400 サーバー・マップに新しい環境を反映するには**

〈システム・インストール・ツール〉メニュー(GH961)から〈カスタム・インストール・プラン〉(P98240)を選択します。

1. 〈インストール計画の処理〉で、プランを選択してツリーを開きます。
2. ホストをクリックして[選択]をクリックします。
3. [フォーム]メニューから[環境]を選択し、新環境をグリッドに追加します。追加した環境を保存します。
4. 〈ホスト・プランナ〉では、[ロー]メニューから[ホストの改訂]を選択します。
5. [フォーム]メニューから[サーバー・マップの生成]を選択します。

▶ **AS/400 に新規パス・コードを追加するには**

1. B9/hosts/as400/(マシン名)の LIBRARY ファイルで、最初の行を B9SYS に変更します。
2. PD9 や PY9 など、必要なパス・コードの名前を追加します。
3. AS/400 で、ENDNET および CLRIPC コマンドを使用して B9 サービスを終了します。
4. AS/400 に QSECOFR でログオンします。
5. CLRLIB JDEOW コマンドを使って AS/400 の JDEOW ライブラリをクリアします。
6. CHGCURLIB JDEOW コマンドを使って AS/400 の現行のライブラリを JDEOW に変更します。
7. “FTP デプロイメント・サーバー名”コマンドを使って、デプロイメント・サーバーからファイルを転送します。
8. JDE でサインオンして次のコマンドを入力します。
 - cd x:¥jdedwardsoneworld¥B9¥hosts¥as400 ¥(エンタープライズ・サーバー名)
 - get ftpinput
 - cd..
 - bin
 - get jdesav1
 - quit
9. AS/400 の画面で次のコマンドを入力します。
 - RSTOBJ OBJ(J98OW10 J98OW20) SAVLIB (JDEOW) DEV(*SAVF) SAVF(JDEOW/JDESAV1) MBROPT(*ALL) ALWOBJDIF(*ALL) FRCOBJCVN(*YES *ALL) RSTLIB(JDEOW)
 - SBMJOB CMD(CALL PGM(J98OW10) PARM('deploymentservername' 'JDEOW'))
10. STRNET コマンドを入力して ERP 9.0 サービスを起動します。
11. 各環境で PORTTEST を実行し、正常に動作していることを確認します。

環境の削除

環境を削除する場合、環境の定義が除去され、その環境に関連付けられた OCM レコードも削除されます。処理オプションを設定して、環境を削除する際に OCM マッピングも削除することができます。

▶ 環境を削除するには

〈環境〉メニュー(GH9053)から〈環境設定マスター〉プログラム(P0094)を選択します。

〈環境の処理〉フォームで、環境を選択して[削除]をクリックします。

処理オプション:環境マスター(P0094)

処理タブ

これらの処理オプションを使用すると、この環境の OCM のマッピングを他のデータ・ソースにコピーまたは削除できるようにするかどうかを指定できます。これにより OCM マッピングのコピーおよび削除を柔軟に行うことができます。

使用している環境の OCM マッピングは、これらの処理オプションの設定に関わらず自動的にコピーまたは削除されます。処理オプションに左右されるのは、他のデータ・ソースにマッピングをコピー/削除できるかどうかです。

処理オプションの情報については、処理オプションのフィールドを右クリックしてメニューから[ポップ・ヒント]を選択してください。または、処理オプションのフィールドをクリックして[F1]キーを押します。

1. OCM マッピングの削除

他のデータ・ソースでこの環境の OCM マッピングを削除できるようにするかどうかを指定します。有効な値は次のとおりです。

ブランク OCM マッピングを削除できません。

1 OCM マッピングを削除できます。

2. OCM マッピングのコピー

他のサーバー・マップ・データ・ソースでこの環境の OCM マッピングをコピーできるようにするかどうかを指定します。有効な値は次のとおりです。

ブランク OCM マッピングをコピーできます。

1 OCM マッピングをコピーできません。

環境ディレクタ

〈環境ディレクタ〉プログラム(P989400)は、システム環境を簡単に作成できるように設計された新しいアプリケーションです。

環境ディレクタでは、環境を作成するために必要なすべてのステップが提供されています。新しい環境の作成または既存の環境のコピー、既存のパス・コードの共有またはコピー、データ・ソースの設定、クライアントおよびサーバー・マップの OCM マッピングの作成、既存環境からのデータのコピーをすべてここで行うことができます。

環境ディレクタはカスタム環境を作成するのにも便利なツールですが、カスタム環境を作成する場合は〈インストール・プランナ〉プログラム(P9840)と〈インストール・ワークベンチ〉プログラム(P9841)を使用することをお勧めします。これらのプログラムは、システムを設定するための標準のアプリケーションです。環境ディレクタとインストール・プランナおよびインストール・ワークベンチは、カスタム環境を設定する大部分の機能を共有しています。

▶ 新しい環境を作成するには

既存の環境から設定をコピーしない場合にこの方法を使用してください。新しい環境の設定を柔軟に行うために、高速モードではなくディレクタ・モードを使用することをお勧めします。

〈上級操作〉メニュー(GH9012)から〈環境ディレクタ〉(P989400)を選択します。

1. 〈環境ディレクタ〉で、[新しい環境]オプションをクリックして[次へ]をクリックします。
2. 〈環境〉で、[ディレクタ]オプションをクリックして[次へ]をクリックします。
3. 〈環境プロパティ〉で、次のフィールドに値を入力します。

- 環境名

環境は、パス・コード(オブジェクト)とデータ・ソース(データ)の両方を含みます。この2つを指定することによって、ユーザーの環境が正しく指定されます。

- 記述

ユーザーが定義した名前または説明。

- コード

ビジネス・データとコントロール・テーブルのデータ・ソースを作成するのに使用されます。たとえば、画面の環境のデータ・ソースは Business Data - TN と Control Tables - TN になります。デフォルトは環境名の最初の2文字になります。

- リリース

リリース・マスターで定義されているリリース番号です。デフォルトは現在のリリースです。

- ジャスト・イン・タイム・インストール
この環境にサインオンしたすべてのユーザーに対してジャスト・イン・タイム・インストールを行うかどうかを指定します。修正したアプリケーションを本稼働用パス・コードに転送する前は、このオプションをオフにすることを考慮してください。アプリケーションを完全にテストして本番環境のユーザーに変更を適用できる段階になったら、このオプションをオンに戻すことができます。
 - JAS/WTS 環境
Java または Windows ターミナル・サーバーの環境の場合はこのオプションをオンにしてください。このオプションは、この環境でのビジネス関数マッピングの作成方法およびログオン時に表示される環境に影響します。
4. [次へ]をクリックします。
 5. <パス・コード>で、[既存のパス・コードの共有]または[既存のパス・コードのコピー]を選択します。共有またはコピーするパス・コード名を入力して[次へ]をクリックします。
パス・コードのリリース・レベルは新しい環境のリリース・レベルと一致している必要があります。[既存のパス・コードのコピー]を選択した場合は<パス・コードのプロパティ>、そうでない場合は<マシン>フォームが表示されます。
 6. <パス・コードのプロパティ>で、次のフィールドに値を入力します。
 - パス・コード
パス・コードはシステム・オブジェクトに対するポインタとなり、オブジェクトとそのロケーションをトラッキングするために使用されます。
 - 記述
ユーザーが定義した名前または説明。
 - リリース
リリース・マスターで定義されているリリース番号です。環境のリリースと一致している必要があります。
 - 累積記述
パス・コードのリリース・レベルをさらに詳しく指定します。SAR インテグレーション用に OMW によって使用されるのみです。
 - ロケーション
ロケーションあるいはマシンキーは、ネットワーク(サーバーまたはワークステーション)上のマシン名を示します。
 - サーバー共有パス
このパス・コードの共有ディレクトリを指定します。ファイル・サーバーに保管されているオブジェクトはこのパスにあります。

- UNC フラグ

サーバー・パスの作成方法を決定します。有効な値は次のとおりです。

オン

相対パスを使ってパスを作成します。2つのバックスラッシュ(¥¥)を入力してください。

オフ

実際のドライブの文字を使ってパスを作成します。

デプロイメント・データ・ソース

パス・コードに対応する Central Object Specifications データ・ソースのロケーション(データ・ソース)を指定します。たとえば、PDxxxx(xxxx は最新のリリース・レベル)というパス・コードの環境の場合、このパス・コードに対する有効なデータ・ソースは Central Objects - PDxxxx になります。

状況コード

開発サイクルにおけるソフトウェアの状況を決定します。

マージ・オプション

標準のオブジェクトにカスタマー独自のオブジェクトをマージするかどうかを示します。パス・コード・レベルでこのオプションを設定すると、そのパスにチェックインされたすべてのオブジェクトに対してこのオプションが適用されます。

7. [次へ]をクリックします。
8. <マシン>で、新しい環境を実行するエンタープライズ・サーバーを選択します。<マシンの選択>で、左のツリーからマシンをハイライトして右矢印ボタンをクリックします。選択したマシンが右のツリーに移動します。
9. 左のツリーに表示するマシンを絞り込むには、[マシンキー]と[ポート番号]フィールド(どちらかまたは両方)に入力して[検索]をクリックします。既存の環境から新しい環境をコピーする場合、リンクをクリックすると既存の環境のマシンが自動的に選択されます。
10. 続行するには[次へ]をクリックします。
11. <データ・ソースの改訂>で、新しい環境の各データ・ソースが表示されます。これらは<環境別データ・ソースのテンプレート>メニュー(GH9012、P98503、F98511)で定義されています。共有データ・ソースは表示されません。
12. <データ・ソースの改訂>で、[OK]をクリックします。環境の各データ・ソースに対して同じフォームが表示されます。すべてのデータ・ソースを定義したら、<データ・ロード>フォームが表示されます。
13. <データ・ロード>で、次のデータを選択または選択解除します。
 - Load Business Data (R98403 XJDE0021)
選択すると、ソース環境を入力する必要があります。
 - Load Control Tables (R98403 XJDE0022)
選択すると、ソース環境を入力する必要があります。

- Central Objects And Versions (R98403 XJDE0019)
- Deployment Server Directory (R9800942 XJDE0001)

この UBE は次の処理を行います。

ローカルのファイル・システムまたはデプロイメント・サーバーのパス・コードのコピー。デプロイメント・サーバーから実行する場合、これらは同じになります。

デプロイメント・サーバーのパッケージ inf ファイルのコピー。

ソフトウェア・パッケージ詳細(F9631)、ソフトウェア・パッケージ・ビルド見出し(F96021)、ソフトウェア・パッケージ・ビルド見出し - 履歴(F96215)の各テーブルのコピー。

レコードのチェックアウト(R989861 XJDE0001)

ESU 履歴(R9672 XJDE0001)

14. [次へ]をクリックします。
15. <環境ディレクトリの改訂>で、ノードのプロパティを表示するにはノードをハイライトして[選択]をクリックします。
16. 環境を作成して OCM を検討するには[OCM]をクリックします。
17. 環境ディレクトリで選択したレコードを検討するには[戻る]、環境ディレクトリを終了するには[キャンセル]をクリックします([キャンセル]をクリックするとすべての変更が失われます)。
18. 環境の作成を完了するには[終了]をクリックします。

次の処理が行われます。

- OCM がない場合は OCM の生成
- F98403 に新しい環境のレコードを作成する(これによってインストール・プランナに表示されるようになります)
- ビジネス・データのロード
- コントロール・テーブルのロード
- セントラル・オブジェクトとバージョンのロード
- パス・コード・ディレクトリとパッケージのコピー
- チェックアウト・レコードのコピー
- ESU 履歴のコピー
- エンタープライズ・サーバーの構成
- サーバー・マップ OCM の作成
- サーバー・マップ F98611 レコードの作成
- F9651 にエンタープライズ・サーバーのレコードを書き込む
- BSFN および UBE マッピングの構成

▶ 高速オプションを使って新しい環境を作成するには

既存の環境から設定をコピーしない場合にこの方法を使用してください。新しい環境の設定を柔軟に行うために、高速モードではなくディレクタ・モードを使用することをお勧めします。

〈上級操作〉メニュー(GH9012)から〈環境ディレクタ〉(P989400)を選択します。

1. [新しい環境]オプションをオンにして[次へ]をクリックします。
2. 〈環境〉で、[高速]オプションをオンにして次のフィールドに値を入力します。
 - 環境名
環境は、パス・コード(オブジェクト)とデータ・ソース(データ)の両方を含みます。
 - 記述
ユーザーが定義した名前または説明。
 - パス・コード名
既存のパス・コードの名前。
3. [エンタープライズ・サーバー]の下のリンクをクリックします。
4. 〈マシン〉で、新しい環境を実行するエンタープライズ・サーバーを選択します。〈マシンの選択〉で、左のツリーからマシンをハイライトして右矢印ボタンをクリックします。選択したマシンが右のツリーに移動します。
5. 左のツリーに表示するマシンを絞り込むには、[マシンキー]と[ポート番号]フィールド(どちらかまたは両方)に入力して[検索]をクリックします(既存の環境から新しい環境をコピーする場合、既存の環境のマシンが自動的に選択されます)。
6. 続行するには[OK]をクリックします。
〈環境〉フォームに戻ります。
7. 環境のデフォルトを作成するには[次へ]をクリックします。
8. 〈環境ディレクタの改訂〉で、ノードのプロパティを表示するにはノードをハイライトして[選択]をクリックします。
9. 環境を作成して OCM を検討するには[OCM]、環境ディレクタを終了するには[キャンセル]をクリックします([キャンセル]をクリックするとすべての変更が失われます)。
10. 環境の作成を完了するには[終了]をクリックします。
次の処理が行われます。
 - OCM がない場合は OCM の生成
 - F98403 に新しい環境のレコードを作成する(これによってインストール・プランナに表示されるようになります)
 - ビジネス・データのロード
 - コントロール・テーブルのロード
 - エンタープライズ・サーバーの構成
 - サーバー・マップ OCM の作成

- サーバー・マップ F98611 レコードの作成
- F9651 にエンタープライズ・サーバーのレコードを書き込む
- BSFN および UBE マッピングの構成

異なる環境からの環境ディレクタの実行

インストール・プランナを使って JDEPLAN 環境からカスタム環境を作成することをお勧めします。これによって、プランナ・データベースに環境が作成され、システム・データ・ソースに配布されます。環境にソフトウェア・アップデートを適用するには、その環境がプランナ・データベースに存在している必要があります。

インストール・プランナの〈環境の選択〉フォームには、カスタム環境をプランに追加できるように[追加]ボタンと[コピー]ボタンがあります。この 2 つのボタンを使って環境ディレクタのいくつかのフォームで環境を定義し、インストール・プランナとインストール・ワークベンチで環境のその他の項目(データ・ソースやデータ・ロードなど)を設定します。

環境ディレクタは、デプロイメント環境から実行することはできません。デプロイメント環境はデプロイメント・サーバーにのみ存在します。デプロイメント環境では B9 システム・プランナにある自らのテーブル(F98611 および F986101)が使用され、それ以外のシステム・テーブルはシステム・データ・ソースにあります。これらのテーブルが別の場所にあるため、プランナとシステム定義の間で不整合が生じます。

環境ディレクタを単独で実行する場合は、Windows クライアントから実行してください。環境はシステム・データ・ソースに作成されます。ソフトウェア・アップデートを適用する前に、プランナ環境にその環境をコピーする必要があります。システムからプランナに環境定義を自動的にコピーする機能(プログラム R9698611)があります。

共有データ・ソースと環境データ・ソース

環境ディレクタでは共有データ・ソースは表示されません。これらのデータ・ソースはすべての環境で共有され、あらかじめ構成されているため再構成の必要はありません。共有データ・ソースのデフォルトは、システム、オブジェクト・ライブラリアン、データ辞書のデータ・ソースです。環境データ・ソースのみ構成します。環境データ・ソースのデフォルトは、ビジネス・データ、コントロール・テーブル、セントラル・オブジェクト、バージョンのデータ・ソースです。

データベース構成と権限

環境ディレクタで新しい環境にデータをロードする前に、新しいデータベースとテーブル・スペース、およびライブラリを作成する必要があります。

通常、システム、オブジェクト・ライブラリアン、データ辞書のデータ・ソースは既存の環境と共有しているため、新しいデータベースは必要ありません。ビジネス・データとコントロール・テーブルのデータ・ソース、および独自のデータ・ソースは、他の環境と共有していない場合には新しいデータベースを作成する必要があります。セントラル・オブジェクトとバージョンのデータ・ソースは、新しい環境で新しいパス・コードを使用する場合にデータベースを作成する必要があります。

参照

- データベースの設定方法については『インストール』ガイドおよび『アップグレード』ガイドの「システム・データベースのインストール」

データ・ソースのカスタマイズ

ここでは、環境別のさまざまなデータ・クラスとデータ・ソースのテンプレートについて説明します。

データ・クラス

データ・クラスは OCM の自動生成を行う中心となるところにあります。データ・クラスによって、データ・ソースとテーブルに保管されているデータの種類を分類します。たとえば、ビジネス・データのデータ・クラスは B、システムのデータ・クラスは S です。ビジネス・データを複数のデータ・ソースに分割する場合、新しいデータ・クラスを作成して、データ・ソースのテンプレート(環境ディレクトリまたはプランナでデータ・ソースが表示されます)を修正し、そのデータ・クラスにテーブルと GT オブジェクトを割り当てる必要があります。これらの情報に基づいて OCM が作成されます。

データ・クラスの作成

新しいデータ・クラスを作成するには、ユーザー定義コード H96/DU にデータ・ソースの値、H96/CL にテーブルと GT オブジェクトの値を追加してください。

環境別データ・ソースのテンプレート

データ・ソースのテンプレートでは、新しい環境で使用するデータ・ソース、およびデータ・ソースのプロパティのデフォルト値を定義されています。これらのテンプレートは環境別データ・ソース・テーブル(F98511)に保管されています。

テンプレートからデータ・ソースを作成すると、F98511 にはその環境のデータ・ソースも保存されます。

インストール・プランナと環境ディレクトリでは、データ・ソースを表示したり OCM を作成するのにこのテーブルが使用されます。データ・ソースを表示する際、データ・ソース・マスター(F98611)にそのデータ・ソースが定義されているかどうかをまずチェックされます。定義されている場合、F98511 のデフォルトは無視され、既存のデータ・ソース定義に基づいてデータ・ソースが表示されます。

このテーブルの情報は、〈環境別データ・ソース・テンプレート〉メニュー(GH9611)を使って変更することができます。このアプリケーションを使用すると、環境の作成が簡単に行えるだけでなく、標準命名規則に従ってカスタム環境を作成することができます。

新しい環境を作成すると、その環境のデータ・ソースを作成するためにテンプレート・レコードがコピーされ、プロパティのトークンがその環境の値に更新されます。環境の値が*SHARED のレコードは、共有データ・ソースのテンプレートとして使用されます。これは、共有データ・ソースを設定するために一度だけ使用され、環境が DEFAULT の新しいレコードが作成されます。環境ディレクトリでは共有データ・ソースは表示されません。共有データ・ソースはインストール・プランナで設定してください。共有データ・ソースのデフォルトは、システム、オブジェクト・ライブラリアン、データ辞書のデータ・ソースです。

環境の値が*ENV のレコードは、新しい環境のデータ・ソースに使用されます。新しい環境を作成する場合は常に*ENV のレコードがコピーされます。環境名は*ENV に置き換えられ、トークンは実際のデータ・ソース名とデフォルトのプロパティに置き換えられます。環境データ・ソースのデフォルトは、ビジネス・データ、コントロール・テーブル、セントラル・オブジェクト、バージョンのデータ・ソースです（セントラル・オブジェクトのデータ・ソースもパス・コード・マスターで指定します）。

環境の値が*LOGIC のレコードは、ロジックおよびサーバー・マップのデータ・ソースに使用されます。これらのレコードは、エンタープライズ・サーバーを選択する際に設定します。

テーブル・データ・クラス

テーブルは、オブジェクト・ライブラリアン・テーブル(F9860)のデータ・クラス・フィールド(SICLDF)を使ってデータ・クラスと関連付けられています。テーブル・データ・クラスは次の 2 つのプログラムで修正することができます。

- オブジェクト管理ワークベンチ(GH902) - OMW の左のツリーでテーブルを選択して[設計]アイコンをクリックします。[インストール/マージ・コード]タブをクリックして[データ・クラス]フィールドを変更します。[OK]をクリックします。データ・クラスを変更しても既存の OCM マッピングには影響しません。変更したデータ・クラスは新しい環境のマッピングを作成するためにのみ使用されます。
- テーブル・データ・クラス - このアプリケーションは、複数のテーブルのデータ・クラスを一度に修正する場合に使用します。システム管理者だけがこのアプリケーションを使用するようにしてください。〈上級操作〉メニュー(GH9611)から〈テーブル・データ・クラス〉を選択します。グリッドの QBE を使って検索するテーブルを絞り込みます。変更するデータ・クラスを持つテーブル(複数可)をハイライトします。[選択]をクリックします。新しいデータ・クラスの値を入力して[OK]をクリックします。

データのロード

環境ディレクタでは、UBE を使って環境にデータをロードする処理を自動的に行うことができます。ただし、DTS、BCP、インポート/エクスポート、CPYLIB などのサードパーティのツールを使ってデータをロードする場合があります。その場合は、環境ディレクタでデフォルトのデータをロードしないようにすることができます。

OCM の作成方法

OCM マッピング(TBLE、GT、BSFN、UBE)は、新しい環境またはコピーした環境に対して作成することができます。

新規作成またはコピー

新しい環境の OCM は 2 つのステップを使って作成します。最初にテーブルと GT のマッピングを設定します。次に BSFN と UBE のマッピングを構成します。

新しい環境を作成した場合の TBLE と GT

各テーブルまたは GT オブジェクトのデータ・クラスは、オブジェクト・ライブラリアン(F9860)に保管されます。オブジェクトの環境とデータ・クラスに基づいて、環境別データ・ソース・テーブル(F98511)からデータ・ソース名が取り込まれます。レコードが見つからない場合、環境名が DEFAULT (共有データ・ソース)のデータ・クラスが検索されます。そのデータ・ソースがデフォルトのマッピングの場合、オブジェクト名が DEFAULT のマッピングが作成されます。デフォルト・マッピングでない場合は、実際のオブジェクト名が使用されます。プランナおよび内部データ・クラスのテーブルに対しては、マッピングは作成されません。

環境をコピーした場合の TBLE と GT

コピーしようとする環境の TBLE および GT オブジェクトの*PUBLIC の各 OCM レコードが検索されます。次に、オブジェクト・ライブラリアンからオブジェクトのデータ・クラスが取り込まれます。そして、そのデータ・クラスの新しい環境の F98511 でデータ・ソースが検索されます。データ・ソースが OCM のデータ・ソースと異なる場合、新しいデータ・ソースにマッピングが変更されます。

新規作成または既存の環境をコピーする場合

UBE と BSFN に対して OCM を作成するアルゴリズムは同じですが、デフォルトのマッピングが LOCAL かエンタープライズ・サーバーかによって結果が異なります。

デフォルトのマッピングが LOCAL の場合

LOCAL に対するデフォルト・マッピングが作成されます。サーバー・マッピングは、OMW のロケーション(BSFN は Business Function Location、UBE は Process Location と表示)がサーバー専用(F9861 の SIBFLOCN が 3)またはクライアント/サーバー(F9861 の SIBFLOCN が 2)で、OCM カテゴリがマッピング先サーバー(F9860 の SIOLCD02 が 002)であれば、どんなオブジェクトに対しても作成することができます。

デフォルトのマッピングがサーバーの場合

エンタープライズ・サーバーに対するデフォルト・マッピングが作成されます。LOCAL マッピングは、OMW のロケーション(BSFN は Business Function Location、UBE は Process Location と表示)がクライアント専用(F9861 の SIBFLOCN が 1)またはクライアント/サーバー(F9861 の SIBFLOCN が 2)で、OCM カテゴリがローカルのみマップ(F9860 の SIOLCD02 が 001)であれば、どんなオブジェクトに対しても作成することができます。

デフォルトのマッピングは次のようになります。

	基本環境	JAS/WTS 環境
	(DV9、JD9、PD9、PY9)	(JDV9、JJD9、JPD9、JPY9)
UBE	エンタープライズ・サーバー	エンタープライズ・サーバー
BSFN	ローカル	エンタープライズ・サーバー

オブジェクト名	データ・クラス
F0094	S
F9860	O
F9200	D
F0101	B
F01012	B
F0004	T
F98710	C
F983051	V
GT92002	D
GT9860A	O
GT3711	B
GT4801	B

環境	オブジェクト名	ユーザー/グループ	データ・ソース
ENV1	DEFAULT	*PUBLIC	Business Data - ENV1
ENV1	F0094	*PUBLIC	System - B9
ENV1	F9860	*PUBLIC	Object Librarian - B9
ENV1	F9200	*PUBLIC	Data Dictionary - B9
ENV1	F0004	*PUBLIC	Control Tables - ENV1
ENV1	F983051	*PUBLIC	Versions - PY9
ENV1	GT92002	*PUBLIC	Data Dictionary - B9
ENV1	GT9860A	*PUBLIC	Object Librarian - B9
ENV2	DEFAULT	*PUBLIC	Business Data - ENV2
ENV2	F0094	*PUBLIC	System - B9
ENV2	F9860	*PUBLIC	Object Librarian - B9
ENV2	F9200	*PUBLIC	Data Dictionary - B9
ENV2	F0004	*PUBLIC	Control Tables - ENV2
ENV2	F983051	*PUBLIC	Versions - DV9
ENV2	GT92002	*PUBLIC	Data Dictionary - B9
ENV2	GT9860A	*PUBLIC	Object Librarian - B9

データ・ソース

データ・ソースは、この製品のシステム・セットアップを行う上で必要不可欠なものです。データ・ソースには、このシステムに必要なすべてのデータベースとロジック・サーバーを定義します。

データ・ソースの理解

データ・ソースには、データベース・テーブルの場所、およびロジック・オブジェクトが実行される場所を定義します。データ・ソースは次のような項目を指定することができます。

- 特定のロケーションのデータ・ソース(¥b7¥dataにある JDELocal などの MSDE データベースや、PRODDATA などの AS/400 データ・ライブラリ)
- ロジックを処理するサーバー

データ・ソース定義は、データ・ソース・マスター(F98611)に保管されます。各ワークステーションは、通常エンタープライズ・サーバーのシステム・データ・ソースに置かれる共有の F98611 テーブルを使用します。ロジック処理とデータ要求を行うサーバーには、データ・ソースの固有な定義が必要です。このため、各サーバーのサーバー・マップ・データ・ソースに F98611 があります。

少なくとも 2 つの F98611 があります。通常エンタープライズ・サーバーにあって各ワークステーションがアクセスするシステム・データ・ソースと、各ロジック・サーバーが必要とするサーバー・マップ・データ・ソースの 2 か所にあります。

データベース・データ・ソースの理解

データベースとは、データベース管理システムにおけるテーブルのグループです。データベースにアクセスするアプリケーションに対して、使用するデータベースを設定する必要があります。ネットワーク上でデータベースを分散し、さまざまなサーバーおよびデータベース管理システムをネットワーク構成に組み込むことができます。データベース・データ・ソースには、データベースに接続するために必要なデータベース情報を設定します。

データ・ソース名

データ・ソースを区別するためにデータ・ソースの名前を定義します。データ・ソース名は、データ・ソースの内容が反映されるようにします。たとえば、本稼動環境のビジネス・データを保管していることを示すには Business Data - Prod などのデータ・ソース名にします。

インストール時に提供されるデフォルトのデータ・ソース名を使用することができます。

参照

- データ・ソース名の定義について詳しくは、『EnterpriseOne 8.9 インストール・ガイド』および『EnterpriseOne 8.9 インストール・リファレンス・ガイド』

データベース名

データ・ソース定義には、データベース、およびそのデータベースがあるサーバーの情報を指定する必要があります。データベースの種類によってデータベースの指定方法が異なります。たとえば、Oracle データベースは Oracle SQL*Net V.2 接続文字列によって指定します。ODBC を介してアクセスするデータベースに対しては、ODBC データ・ソース名を指定する必要があります。

ネットワーク・マシン名(サーバー名)

データベースはサーバー上にあります。ネットワークにこのサーバーを設定して、他のコンピュータがそのサーバーにアクセスできるようにする必要があります。データ・ソース定義に、データベースが置かれているサーバー名を指定してください。

JDEBase(DLL 名)

テクノロジーの変化に対応できるビジネス・ソリューションが必要不可欠です。JDEBase ミドルウェア製品では、各データベース管理システムで異なる SQL を解決する API が提供されています。JDEBase は自動的に適切なプログラムを決定します。これらのプログラムは、一般的なデータ操作言語(DML)の要求を適切な SQL 文に変換し、サードパーティ製通信ソフトウェアを使ってネットワーク経由で要求を伝達します。

ロジック・データ・ソースの理解

ロジック・マシンとは、バッチ・アプリケーションおよびビジネス関数が実行されるサーバーまたはクライアント・マシンのことです。データ・ソース定義でこれらのロジック・マシンを指定する必要があります。このデータ・ソース定義には、HP9000 といったサーバー名など、マシンに関するネットワーク情報を指定してください。

分散処理のためにロジック・オブジェクトをマッピングすると、ロジック・オブジェクトを処理するターゲット・ロケーションとして、マシン・データ・ソース(分散処理データ・ソース)が使用されます。

必須データ・ソース・タイプ

EnterpriseOne 8.9 を実行するには、必要最低限のデータ・ソースを設定する必要があります。必須データ・ソースのうち 2 つには、ロジックを処理するマシンを定義します。それ以外のデータ・ソースには、使用するさまざまなデータベースを定義します。

このソフトウェアには、システム構成に必須のデータ・ソースのサンプルが含まれています。

分散処理データ・ソース

このデータ・ソース定義には、ネットワーク上のロジック・マシンを指定した情報が含まれています。データ・ソースとして、各ロジック・マシンを定義する必要があります。

ローカル・データ・ソース

このデータ・ソースにはワークステーションを定義します。このデータ・ソースを使用して、サーバー上で実行するためにオブジェクト構成マネージャ(OCM)でマップしたバッチ・アプリケーションの処理ロケーションを一時変更します。

ビジネス・データ・データ・ソース

このデータ・ソースは、ビジネス・データを複数の所有者またはライブラリに分割する場合に使用されます。これは同じエンタープライズ・サーバー上に置くことも異なるエンタープライズ・サーバー上に置くこともできます。各データのグループにはそれぞれデータ・ソースが必要です。

ビジネス・データには、次のような補足データ例があります。

- 本稼動用データ(会計や製造データなど、テクニカル関連以外のデータ)
- テスト・データ
- デモ・データ(デモまたはトレーニング用)
- カンファレンス・ルーム・パイロット(CRP)データ

この製品をインストールすると、対応するホスト・データベースにコピーできるデモ用データもインストールされます。データ・ソース名は Business Data - JDE です。

コントロール・テーブル・データ・ソース

このデータ・ソースは、ユーザー定義コード、メニュー、自動採番レコードで構成されます。

リリース別データ辞書データ・ソース

このデータ・ソースによってデータ辞書のマスター・テーブルを 1 か所に保管できるため、変更管理が簡単になります。これらのマスター・テーブルをグループ化してデータ辞書データベースを形成します。本稼動用パス・コード(PD9 など)と開発用パス・コード(DV9 など)で 1 つのデータ辞書を共有する必要があります。パス・コードごとにデータ辞書を設定することもできますが、複数のデータ辞書の使用はお勧めしません。また、それに対するサポートも行っていません。データ辞書データ・ソースの名前には基準リリース番号が付けられています。たとえば、Data Dictionary - B7334、Data Dictionary - B9、Data Dictionary - B732 などです。

システム・データ・ソース

このデータ・ソースは、この製品に含まれるすべてのアプリケーションを実行するのに使用するテクニカル・テーブルで構成されています。リリースごとにシステム・データ・ソースを 1 つ設定する必要があります。アプリケーションの実行時には、システム・テーブルによって次の情報が提供されます。

- オブジェクト・マッピング(テーブル、バッチ・プロセス、ビジネス関数のロケーション)
- データ・ソース定義
- セキュリティ
- 自動採番 ID(開発用のみ)

すべてのワークステーションは、通常デプロイメント・サーバーではなくエンタープライズ・サーバーに保管されているシステム・テーブルを使用します。各ロジック・サーバーには専用のシステム・テーブルが必要です。これらのサーバーのシステム・テーブルは、サーバー・マップ・データ・ソースに保管されます。

参照

- サーバー・システム・テーブルについて詳しくは『CNC インプリメンテーション』ガイドの「サーバー・マップ・データ・ソース」

システム・テーブルのキャッシュ作成

初めてこのソフトウェアにログオンすると、指定したユーザーID と環境を使ってシステム・テーブルからそのユーザーと環境の情報が取り込まれます。この情報は、ワークステーションのメモリにキャッシュとして保存されます。サーバーのシステム・テーブルに変更があるたびに、セッションが確立されているワークステーションでは、システム情報のキャッシュが動的に作成されます。

システム・データ・ソースへの接続方法

ワークステーションでこのソフトウェアを起動すると、そのワークステーションの jde.ini ファイルで指定されたプライマリのデータ・ソースへの接続が試みられます。このデータ・ソースに接続できない場合は、システム情報を得るためにセカンダリのデータ・ソースに接続しようとします。このため、セカンダリ・データ・ソースのロケーションにも最新の情報が含まれることが重要です。セカンダリ・データ・ソースの情報は、テーブル変換またはデータ複製によって管理できます。

jde.ini ファイルでプライマリ・データ・ソース接続に関する設定は次のようになります。

```
[DB SYSTEM SETTINGS]
```

```
.
```

```
.
```

```
Default Env=DEMOB7A
```

```
Default PathCode=DEMO
```

```
Base Datasource=System B733
```

```
Database=System B733
```

```
.
```

```
.
```

```
.
```

```
セカンダリ・システム・データ・ソースへの接続
```

```
[DB SYSTEM SETTINGS - SECONDARY]
```

```
Base Datasource=Access32
```

Object Owner=
Server=
Database=Access32
Load Library=JDBODBC.DLL

インストール時に、〈リリース・マスター〉アプリケーションによってシステム・データ・ソースとリリースが関連付けられます。リリースを設定すると、ワークステーションのインストール時に jde.ini ファイルを作成するのに使用される setup.ini ファイルが更新されます。

参照

- システム・データ・ソースへの接続について詳しくは『CNC インプリメンテーション』ガイドの「オブジェクト・ライブラリアンとセントラル・オブジェクトの保管」
- 〈リリース・マスター〉アプリケーションについて詳しくは『インストール・リファレンス』ガイドの「主要な ERP 9.0 テクニカル・テーブル」

オブジェクト・ライブラリアン・データ・ソース

このデータ・ソースには、カスタム開発で使用するオブジェクト・ライブラリアン・テーブルを指定します。設定するパス・コード(セントラル・オブジェクト)の数にかかわらず、オブジェクト・ライブラリアン・テーブルはソフトウェアの各リリースごとに 1 セットのみ設定してください。このデータ・ソースは、J.D. Edwards がサポートするすべてのプラットフォームに置くことができます。オブジェクト・ライブラリアン・データ・ソースの名前には、Object Librarian - B9 などのように基準リリース番号が付けられています。

セントラル・オブジェクト・データ・ソース

このデータ・ソースには、ソース・オブジェクト(セントラル・オブジェクト・スペック)およびユーザー時変更テーブル(F98950)を指定します。セントラル・オブジェクト・データ・ソースはデータベースです。

複数のパス・コードがある場合、パス・コードごとにセントラル・オブジェクト・データ・ソースが必要です。開発者がオブジェクトを変更する場合、セントラル・オブジェクト・データ・ソースからオブジェクトをチェックアウトします。開発者がこのオブジェクトをチェックインすると、オブジェクトは開発者のワークステーションからセントラル・オブジェクト・データ・ソース上のリレーショナル・データベース・テーブルにコピーされます。構成に必要なパス・コードごとに、それぞれセントラル・オブジェクト・データ・ソースを設定する必要があります(Central Objects - PD9, Central Objects - DV9 など)。

次のようなオブジェクトにセントラル・オブジェクト・データ・ソースが必要です。

- プリステイン・オブジェクト
- 本稼動用オブジェクト
- 開発用オブジェクト

構成に応じて作成した環境で使用するパス・コードを各セントラル・オブジェクト・データ・ソースに接続します。セントラル・オブジェクト・データ・ソースは SQL Server または Oracle データベースに置く必要があります。これらのデータベースが実行可能なすべての J.D. Edwards 対応プラットフォームに置くことができます。パフォーマンスを最大限にしてテーブル・スペースを有効に利用するには、セントラル・オブジェクトを UNIX 上に置いてください。UNIX エンタープライズ・サーバーがない場合は、デプロイメント・サーバーでは Microsoft SQL Server または Oracle を使用してください。

オブジェクト・ライブラリアンとセントラル・オブジェクトの保管

オブジェクト・ライブラリアンとセントラル・オブジェクト・データ・ソースは、Oracle または Microsoft SQL Server に置くことができます。また、オブジェクト・ライブラリアンは DB2/400 上に置くこともでき、Oracle または Microsoft SQL Server にあるセントラル・オブジェクト・データ・ソースとは別のサーバーに置くことができます。

次の設定をお勧めします。

- Oracle/UNIX プラットフォームの場合、オブジェクト・ライブラリアンとセントラル・オブジェクト・データ・ソースは Oracle に保管してください。それ以外の場合は、デプロイメント・サーバーの Oracle または Microsoft SQL Server データベースに保管してください。
- セントラル・オブジェクトのリレーショナル・データベース(RDB)のコンポーネントを保管するデータベースに関係なく、C 言語のコンポーネントは、デプロイメント時にすべてのワークステーションにアクセス可能なデプロイメント・サーバーに保管する必要があります。このため、C コンポーネントはデプロイメント・サーバーに保管することをお勧めします。

サーバー・マップ・データ・ソース

このデータ・ソースを使用して、各ロジック・サーバーに独自のシステム・テーブルを作成することができます。このシステム・テーブルはサーバー・マップ・テーブルと呼ばれます。サーバー・マップ・テーブルは各ロジック・サーバーに必要です。サーバー・マップ・テーブルは、ワークステーションのシステム・テーブルとの整合性が確実に保たれるように管理する必要があります。

サーバー・マップ・データ・ソースを使用して、ロジック・サーバーに固有のオブジェクト・マッピングを作成してください。サーバー上で実行中のバッチ・ジョブとビジネス関数がデータを要求する際、サーバー・マップ・データ・ソースの OCM テーブルとデータ・ソース・マスターを参照します。これは、この 2 つのテーブルのマッピングが異なるために必要です。

たとえば、静的ローカル・データはワークステーションにマップされ、動的トランザクション・データ、ビジネス関数およびバッチ・プロセスがサーバーにマップされている環境にログオンするとします。受注オーダーを入力して[OK]をクリックすると、サーバー上で受注オーダー入力のマスター・ビジネス関数が実行されます。マスター・ビジネス関数がワークステーションに戻ってユーザー定義コードや税情報を取り込むのは意味がないため、サーバー・マップ OCM テーブルではすべてのデータが適切なサーバー・データ・ソースにマップされます。

サーバー・マップ・データベースにある次のテーブルは、サーバー独自の処理に使用されます。

オブジェクト構成 マスター(F986101)

サーバーが要求するデータに対して処理されるロジック・オブジェクト、およびその他のロジック・オブジェクトを提供します。これらの要求がサーバーに送信された場合、OCM テーブルでそのデータとロジック・オブジェクトの正確なマッピングを検索する必要があります。サーバーとワークステーションのマッピング要件は異なる場合があります。

たとえば対話型処理では、パフォーマンス向上のためにワークステーションに対してすべてのユーザー定義コードをローカルにマップする必要があります。サーバー処理では、それらのファイルをサーバーのデータベースに対してローカルにマップすると、サーバー処理のパフォーマンスが向上します。

ジョブ・コントロール 状況マスター(F986110)

サーバーで実行されるバッチ・ジョブの情報を記録します。

ジョブ番号マスター (F986111)

サーバーで実行されるバッチ・ジョブのジョブ番号(自動採番)を記録します。

参照

- ロジック・オブジェクトの処理については、『CNC インプリメンテーション』ガイドの「オブジェクト構成マネージャ(OCM)の処理」
- テーブルの複製については、『システム・アドミニストレーション』ガイドの「データの複製」

バージョン・データ・ソース

このデータ・ソースは、Versions - PD9 などのようにパス・コードに対応しています。このデータ・ソースには、バージョンおよび処理オプションの情報が保管されます。このデータ・ソースには次のテーブルが含まれます。

- バージョン・リスト (F983051)
- 処理オプション・テキスト(F98306)

データ・ソースの計画

データ・ソースは、システム構成の基盤となるため、ビジネス・ニーズを満たすすべてのデータ・ソースを正しく設定することが非常に重要です。

次のタイプのデータ・ソースには、各リリースごとにデータ・ソース定義が1つ必要です。

- オブジェクト・ライブラリアン
- システム
- データ辞書
- ローカル

上記以外の次のタイプのデータ・ソースについては、業務要件に応じてタイプごとに複数のデータ・ソース定義が必要な場合があります。

- ビジネス・データ(本稼動用、テスト、CRP、JDE デモ・データ)
- 分散処理(各ロジック・サーバーに1つ)
- サーバー・マップ(各ロジック・サーバーに1つ)
- セントラル・オブジェクト(各パス・コードに1つ)
- コントロール・テーブル

データ・ソースの計画では、次のことを行います。

9. システム・データのロケーションを評価します。
10. ロジックを置くすべてのエンタープライズ・サーバーをリストアップして、次のように設定します。
 - 各エンタープライズ・ロジック・サーバーに分散処理データ・ソースを1つ設定する
 - 各分散処理データ・ソースにサーバー・マップ・データ・ソースを1つ設定する
11. 本稼動環境、プリスティン(JDE オリジナル)環境、開発環境などの環境をサポートするために必要なオブジェクトの数を確認します。各オブジェクトのグループに対してセントラル・オブジェクト・データ・ソースを1つ設定します。
12. ビジネス・データのロケーションをリストアップして、次のように設定します。
 - アプリケーションが使用するビジネス・データのテーブルが含まれているデータベースおよびライブラリごとにビジネス・データ・ソースを1つ設定する
 - 本稼動用およびテスト用に、それぞれコントロール・テーブル・データ・ソースを設定する

データ・ソースの処理

データベース・データ・ソースおよびロジック・データ・ソースの追加/変更を行うには、それぞれ<データベース・データ・ソース>および<ロジック・データ・ソース>アプリケーションを使用します。新しいデータ・ソースを追加したら、<リリース/データ・ソース・マップ>アプリケーションを使用してリリース/データ・ソース・マップ・テーブル(F00948)を更新する必要があります。

データ・ソース命名規則

新しいデータ・ソースを追加する場合、次の命名規則に従ってください。

- 30文字以内
- 大文字と小文字を区別
- スペースの有無を区別

Client Access データ・ソースについては、次の例外があります。

データ・ソース名

- 32 文字以内
- 1 文字目には英字を使用
- 次の文字は使用不可 {}[]()?*=!@;

注:

Client Access ODBC ドライバを使って AS/400 のデータにアクセスするには、データ・ソース名を入力する必要があります。

データ・ソース記述

- 80 文字以内

テーブル所有者(オブジェクト所有者 ID)

Oracle および SQL Server データベース管理システムでは、各テーブルにテーブル所有者 ID があります。たとえば、システム・テーブルを含むデータベースには、B733SYS のような所有者 ID が付けられます。Oracle および SQL Server のテーブル所有者 ID を指定してください。

データ・ソースの追加または修正

ここでは、システム・データ・ソース内のデータ・ソース・マスター(F98611)にデータベースまたはロジック・データ・ソースを追加したり、既存のデータ・ソースを修正する方法について説明します。初期インストール後に、このアプリケーションを使ってデータ・ソース定義を修正、追加します。

追加または修正を行う際、サイズを含むテーブルとデータ・ソースの情報も変更することができます。

▶ データ・ソースを追加または修正するには

〈システム・アドミニストレーション・ツール〉メニュー(GH9011)から〈データベース・データ・ソース〉または〈ロジック・データ・ソース〉プログラム(P986115)を選択します。

これらのプログラムでは、〈データ・ソースの改訂〉プログラム(P986115)の〈上級設定〉フォームで[OCM データ・ソース]オプションが選択されたデータ・ソースが表示されます。

1. 〈マシンの検索/選択〉で、処理するデータ・ソースをハイライトして[選択]をクリックします。

このフォームには、〈データ・ソースの改訂〉プログラム(P986115)の〈上級設定〉フォームで [OCM データ・ソース]オプションがオンになっているデータ・ソースが表示されます。参照するデータ・ソースがあるマシンを選択します。ワークステーションは、マッピングにシステム・データ・ソースを使用します。各エンタープライズ・サーバーには独自のサーバー・マップ・データ・ソースがあります。

2. データ・ソースを追加するには、〈データ・ソースの処理〉で[追加]をクリックします。

3. 既存のデータ・ソースを修正する場合は、〈データ・ソースの処理〉で[検索]をクリックして、データ・ソースを表示します。
4. 修正するデータ・ソースをハイライトして[選択]をクリックします。

注意

データ・ソースを追加または変更しないで照会のみ行う場合は、[OK]ではなく[キャンセル]をクリックして〈データ・ソースの改訂〉フォームを終了してください。[OK]をクリックすると、ユーザーがデータ・ソースを追加または変更したと見なされ、既存の ODBC ドライバが正常に動作しなくなる可能性があります。その場合は、コントロール・パネルの管理ツールにある[ODBC データ・ソース]を使用して、ドライバを修正してください。

5. 〈データ・ソースの改訂〉フォームで、必要なフィールドに値を入力して[OK]をクリックします。設定するデータ・ソースに必要なフィールドが表示されます。次のようなフィールドがあります。
 - データ・ソース名
 - データ・ソース・タイプ
 - データ・クラス
 - プラットフォーム
 - JDBNET データ・ソース
 - データベース・サーバー名
 - JDBNet サーバー名
 - ロジック・サーバー名
 - JDBNet サーバー・マップ・データ・ソース名
 - サーバーで使用するデータ・ソース
 - オブジェクト所有者 ID
 - スキーマ
 - ライブラリ名
 - データベース名
 - データベース名(TNS)
 - データベースのエイリアス名
 - データベース・インスタンス
 - ODBC データ・ソース名
 - サーバー・マップ・データ・ソース名

6. さらに詳しいオプションを設定するには、[フォーム]メニューから[詳細]を選択します。
設定するデータ・ソースに関連するオプションが表示されます。次のようなオプションがあります。
- クラスタ化
 - Unicode(ユニコード)
 - 小数点シフトを使用
 - ユリウス日付を使用
 - 更新サポート
 - OGM データ・ソース
 - テーブル所有者を使用
 - LOB データ・タイプ・サポート
 - AS/400 BLOB サポート
7. データベース・サーバーのポート情報を指定するには、[フォーム]メニューから[ポート]を選択します。
- データベース・サーバーのポート番号
 - Oracle SID 名
8. [OK]をクリックします。
- DB2/400、UDB、SQL Server データベースに対して ODBC ドライバの入力/変更ができるよう ODBC データ・ソース管理者アプレットが自動的に開きます。ODBC ドライバを追加、更新して、追加または変更したデータ・ソースが反映されるようにします。

参照

- DLL 名の一覧については『CNC インプリメンテーション』ガイドの「データベース・ミドルウェア・ドライバ」

フィールド記述

記述	用語解説
データ・ソース名	データ・ソースの名前。
使用するデータ・ソース	データ・ソースの構成方法、サーバー(SVR)の UBE とビジネス関数の実行方法、またはデータベース(DB)のテーブルデータへのアクセス方法を示します。
データ・ソース・タイプ	データベースのタイプ
データ・クラス	将来使用。
プラットフォーム	データベースが常駐している、実際のハードウェアのタイプ

JDBNET データ・ソース	データ・ソースが JDEBNET を使用するかどうかを示すフラグ。Access を除くすべてのデータ・ソースに必須です。
JDBNET サーバー・マップ・データ・ソース名	データ・ソース・マスター(F98611)で将来使用として予約されている 30 文字の文字列。
サーバーで使用するデータ・ソース名	HPDEVORAP または HP9000 など、インストール中にデータベースに割り当てられる名称。
オブジェクト所有者 ID	データベース・テーブルの接頭辞または所有者
スキーマ	データベース・テーブルの接頭辞または所有者
ライブラリ名	SQL サーバー・データベースの名前。SQL Server ODBC および SQL Server OLEDB データ・ソースに必須です。
データベース名	HPDEVORAP または HP9000 など、インストール中にデータベースに割り当てられる名称。 --- フォーム固有 --- データソースタイプによって、このフィールドの使用目的は異なります。データソースタイプが ORACLE の場合は、ORACLE コネクト文字列が含まれます。MICROSOFT ACCESS、SQL*SERVER、または CLIENT ACCESS の場合、WINDOWS ODBC データソース名が含まれます。SQL*SERVER への接続数を最小にするには、ODBC データソース名を SQL*SERVER データベースが常駐するマシン名にします。また、カタログ名も個々のデータベース名に設定します。たとえば、データベースが(A と B)2 つある場合、マシン(INTELNT)では、このフィールドには"INTELNT"が含まれ、1 つのデータソースに対してカタログ名がデータベース A に設定されます。もう 1 つのデータソースにはデータベース B が設定されます。
データベース名(TNS)	HPDEVORAP または HP9000 など、インストール中にデータベースに割り当てられる名称。
データベース・エイリアス名	SQL サーバー・データベースの名前。SQL Server ODBC および SQL Server OLEDB データ・ソースに必須です。
データベース・インスタンス	サーバー・インスタンスの名前。SQL Server ODBC および SQL Server OLEDB データ・ソースに必須です。
ODBC データ・ソース名	HPDEVORAP または HP9000 など、インストール中にデータベースに割り当てられる名称。
サーバー・マップ・データ・ソース名	HPDEVORAP または HP9000 など、インストール中にデータベースに割り当てられる名称。
クラスタ	使用されていません。

Unicode フラグ	データベースが Unicode をサポートするかどうかを示すオプション。このオプションはすべてのデータ・ソースで必須です。
テーブル所有者を使用	これを使って、所有者 ID フィールドを使用可能にします。
ユリウス日付を使用	ユリウス形式で日付を保管する場合にチェックします。チェックしない場合は、WINDOWS のコントロールパネルに定義されているとおりに日付が保管されます。ONEWORLD は、ONEWORLD 以外のテーブルに対してはこのフラグを自動的にオフにします。
小数点シフトを使用	データの検索または更新時に、データ辞書のスペックに基づいて小数点位置を自動的にシフトするにはこれをチェックします。このフィールドは、ONEWORLD テーブル専用です。ONEWORLD 以外のテーブルは、小数点位置のシフト機能なしで別々のデータソースにある必要があります。ONEWORLD テーブルに ONEWORLD 以外のテーブルを持ち込むと、ONEWORLD は小数点位置をシフトされた、そのテーブルのフィールドを認識しません。これは、テーブル変換ではチェックされません。
更新サポート	データ・ソースに対するロー・レベルでのレコード・ロックを ON/OFF する際に、このフラグを使用します。データベースの整合性を保つには、このフラグを ON にします。JDEBASE ミドルウェアは、このフラグを使用して、ロー・レベルでのレコード・ロックを使用するかどうかを確定します。
OCM データ・ソース	このフラグを使用して、そのデータ・ソースが OCM テーブル(F986101)をもつかどうかを指定します。このフラグを有効にすると、ONEWORLD の〈データ・ソースの 処理〉フォームの OCM データ・フィールドに 1 が表示されます。このフラグは、システム・データ・ソースとサーバー・マップ・データ・ソースに対してのみ有効にします。システム・データ・ソースやサーバー・マップ・データ・ソースではない データベース・データ・ソースやログ・データ・ソースに対しては、この設定を 有効にしないでください。
LOB データ・タイプ・サポート	データベースが LOB データ・タイプをサポートするかどうかを示します。すべての AS400 と Oracle データベースに必須です。
AS/400 BLOB サポート	AS400 バイナリ・オブジェクト(BLOB)データ・ソースに対する設定は、あるデータ・ソースが ONEWORLD BLOB テーブルを含められるかどうかを示します。
データベース・サーバー・ポート No.	JAS サーバーのポート番号。SQL Server ODBC、SQL Server OLEDB、および Oracle データ・ソースに必須です。
Oracle SID 名	サーバー・インスタンスの名前。SQL Server ODBC および SQL Server OLEDB データ・ソースに必須です。

リリース/データ・ソース・マップ・テーブルの修正

〈リリース/データ・ソース・マップ〉プログラム(P00948)を使って、各タイプのデータ・ソースのリリース・レベルを定義します。新しいデータ・ソースを追加する場合は、リリース/データ・ソース・マップ・テーブル(F00948)を更新してください。このテーブルには、データ辞書、オブジェクト・ライブラリアン、バージョン・リスト、ワークフローのレコードが含まれ、リリース別に保管されています。たとえば、B73.1、B73.2、B73.3 などのリリースに対してそれぞれデータ辞書レコードが必要です。

データ・ソース間でオブジェクトをコピーまたは転送すると、ソースとターゲットのデータ・ソースのリリースに互換性があるかどうかチェックされます。Unicode コンバージョンが必要な場合 (Xe から EnterpriseOne へのアップグレード時など)、EnterpriseOne 以降のリリースでは自動的にデータが変換されます。

互換性のないリリースのデータ・ソース間では、オブジェクトの移動やコピーを行うことはできません。データ・ソースの追加時に、入力したデータ構造とデータ・タイプがリリース/データ・ソース・マップ・テーブル(F00948)にある場合はエラーが表示されます。データ・ソースは各データ・タイプに対して1つのリリース・レベルしか設定できません。

▶ テーブルおよびデータ・ソースのサイズ変更または情報の一時変更をするには

〈システム・アドミニストレーション・ツール〉メニュー(GH9011)から〈データベース・データ・ソース〉または〈ロジック・データ・ソース〉プログラム(P986115)を選択します。

1. 〈マシンの検索/選択〉で、マシンをハイライトして[選択]をクリックします。〈データ・ソースの処理〉で、処理するデータ・ソースをハイライトして[選択]をクリックします。
2. 〈データ・ソースの改訂〉で、[フォーム]メニューから[データベースのサイズ変更]または[デフォルトのサイズ変更]を選択します。

注:

このアプリケーションは、〈オブジェクト構成マネージャ〉アプリケーションからもアクセスできます。

3. テーブルおよびデータ・ソースのサイズ変更情報を追加するには、データベースのサイズ変更フォームで[追加]をクリックします。

データベースによって〈Oracle データベース・オブジェクトのサイズ変更〉または〈DB2 UDB データベース・オブジェクトのサイズ変更〉フォームが表示されます。このフォームで、テーブルを変更、追加したり、データ・ソースまたはテーブルのインデックス情報を変更、追加することができます。

4. 既存のテーブルおよびデータ・ソースのサイズを変更するには、該当するデータベース・オブジェクトのサイズ変更フォームで、変更するデータ・ソースを選択します。

〈Oracle データベース・オブジェクトのサイズ変更〉または〈DB2 UDB データベース・オブジェクトのサイズ変更〉フォームには入力済みの情報が表示されます。

注:

〈データ・ソースの処理〉フォームで[ロー]メニューから[デフォルト・データベース]を選択してもこのフォームが表示されます。この場合、[オブジェクト名]フィールドには自動的に DEFAULT という値が入力され、デフォルト値が使用されます。オブジェクト名を DEFAULT 以外の値に変更すると、もともとデフォルト値が入力されていたフィールドが表示されます。

5. 新しいパラメータを追加する場合は、次のフィールドに値を入力します。

Oracle データ・ソースの場合:

- データ・ソース
- オブジェクト名

UDB データ・ソースの場合:

- データ・ソース
 - テーブル名
-

注:

[オブジェクト/テーブル名]フィールドに DEFAULT と入力して、データ・ソースのすべてのオブジェクトに対してパラメータを定義することができます。

6. データベース・オブジェクトのサイズ変更フォームで、次のフィールドに値を入力するか値を変更します。

Oracle データ・ソースの場合:

- インデックス初期ストレージ
- インデックスの次のストレージ
- テーブルパーセント増加ストレージ
- テーブル最小エクステント
- テーブル最大エクステント

この値は、テーブル情報、インデックス情報の両方で有効です。

UDB データ・ソースの場合:

- ターゲット・オペレーティングシステム
- テーブル - テーブル・スペース・マネージャ
- データベース・エイリアス名
- 物理データベース名
- テーブル - テーブル・スペース

- LOB デフォルト・サイズ (バイト)
- インデックス・ページあたりの空き容量%
- テーブル - テーブル・スペース 32K
- インデックス・テーブル・スペース
- LOB - テーブル・スペース
- テーブル・ページあたりの空き容量%
- 空き容量の最小パーセント
- LOB - ログの変更
- LOB - コンパクト・データ
- 初期ログ
- 一時テーブル
- 追加モード

注:

Oracle データ・ソースの場合は、オブジェクト名 DEFAULT に対して〈データベース・オブジェクトのサイズ変更〉フォームで変更できるフィールドは、テーブル・スペース名およびインデックス・スペース名のみです。

7. Oracle データ・ソースの情報を変更するには、[ロー]メニューから[一時変更]を選択します。
このオプションは、〈Oracle データベース・オブジェクトのサイズ変更〉または〈DB2 UDB データベース・オブジェクトのサイズ変更処理〉フォームの[ロー]メニューにもあります。
8. 〈テーブル/データ・ソースの一時変更の改訂〉で、次のフィールドに値を入力するか値を変更します。
 - データのコピー (Y/N)
 - テーブルの作成 (1/0)
9. [OK]をクリックして、1 つ前のフォームに戻ります。〈システム・アドミニストレーション・ツール〉メニューに戻るまで、すべてのフォームで[OK]をクリックします。

参照

- データ・ソースについて詳しくは、『CNC インプリメンテーション』ガイドの「データ・ソースの追加または修正」

処理オプション:データベース・データ・ソースおよびロジック・データ・ソース (P986115)

デフォルト・タブ

この処理オプションを設定して次のことができます。

- ローカル(データベース)データ・ソースまたはサーバー・データ・ソースを指定する。
- 〈ワークステーション・インストール〉プログラムの実行時に、ODBC データ・ソースを作成するかどうかを指定する。

処理オプションの情報については、処理オプションのフィールドを右クリックしてメニューから[ポップ・ヒント]を選択してください。または、処理オプションのフィールドをクリックして[F1]キーを押します。

1. データ・ソース・タイプ

UBE とビジネス関数を実行するサーバーまたはテーブルのデータにアクセスするデータベース(DB)といった、データ・ソースの構成方法を指定します。

2. ODBC データ・ソース

ワークステーションのインストール時に ODBC データ・ソースを作成するかどうかを指定します。有効な値は次のとおりです。

ブランク ODBC データ・ソースを作成しない

- 1 Client Access、SQL Server、DB/2 データ・ソースを作成する

データ・ソース・レポートの実行

データ・ソース・マスターのバッチ・プロセスでは、データ・ソースのレポート、または 2 つのデータ・ソースの比較レポートを作成できます。

データ・ソース・マスターのバッチ・プロセスはレポート・ドライバ(R98611B)から呼び出されますが、その他のレポートはレポート・ドライバを必要としません。R98611B レポート・ドライバを使用して、各レポートの処理オプションを設定します。レポートのデータ選択は、ドライバからではなく、その特定のレポートに対して入力する必要があります。処理オプションとデータ選択を設定した後、ドライバからレポートを実行できます。

データ・ソース・レポートの処理オプションの設定

ここでは、レポート・ドライバを使用してデータ・ソース・レポートの処理オプションを設定する方法について説明しています。

▶ データ・ソース・レポートの処理オプションを設定するには

〈上級操作〉メニュー(GH9012)から〈データ・ソース・マスター・レポート〉または〈データ・ソース・マスターの比較〉を選択します。

1. 〈バッチ・バージョンの処理 - 使用可能なバージョン〉で、レポートを検索して選択します。
2. [ロー]メニューから[処理オプション]を選択します。
3. 処理オプションが表示されます。
4. [UBE ドライバ]タブで、レポート・ドライバの情報を入力します。
5. 処理オプションを入力するレポートのタブをクリックします。各レポートのタブでデータ・ソースの情報を入力します。
6. [OK]をクリックして保存します。

参照

- 『CNC インプリメンテーション』ガイドの「処理オプション: データ・ソース・マスター(R98611)およびデータ・ソース・マスターの比較(R986112)」

処理オプション: データ・ソース・マスター(R98611)およびデータ・ソース・マスターの比較(R986112)

レポート・ドライバから処理オプションを設定すると、〈データ・ソース・マスター・レポート〉プログラム(R98611)および〈データ・ソース・マスターの比較〉プログラム(R986112)にパラメータを入力できます。

実行するレポートの処理オプションを設定する必要があります。レポートはバージョンが XJDE0001 が実行されるため、データ選択が必要な場合は、バージョン XJDE0001 のデータ選択を指定してください。

UBE ドライバ・タブ

この処理オプションで実行するデータ・ソース・マスター・レポートを指定できます。表示されているレポート番号が実行するレポートに対応していることを確認してください。レポートに使用するバージョンを指定することも可能です。

処理オプションの詳しい内容を見るには、フィールドを右クリックして[ポップ・ヒント]を選択します。または、処理オプションのフィールドをクリックして[F1]キーを押します。

1. データ・ソース・マスター・レポート名

実行するレポート名を指定します。有効な値は次のとおりです。

R98611 データ・ソース・マスター・レポート

R986112 データ・ソース・マスターの比較

2. バージョン

実行するレポートのバージョンを指定します。

R986112 タブ

比較する2つのデータ・ソース、比較方法、例外印刷の処理方法を指定します。

処理オプションの詳しい内容を見るには、フィールドを右クリックして[ポップ・ヒント]を選択します。または、処理オプションのフィールドをクリックして[F1]キーを押します。

1. データ・ソース 1

2. 比較データ・ソース 2

比較するデータ・ソース名を指定します。

3. 比較方法

使用する比較方法を指定します。この比較方法によって、データ・ソースの検索方法が決定されます。一方向の比較では、データ・ソース 1 だけにあるデータ・ソース 2 にはないレコードが検索されます。双方向の比較では、データ・ソース 1 にあるデータ・ソース 2 にはないレコード、およびデータ・ソース 1 になくてデータ・ソース 2 にあるレコードが検索されます。

有効な値は次のとおりです。

1 一方向のみで比較する。データ・ソース 1 にあるデータ・ソース 2 にはないレコードのみを印刷します。

Blank 双方向で比較する。データ・ソース 1 にはあるレコード、およびデータ・ソース 1 にはなくデータ・ソース 2 には存在するレコードを印刷します。

4. 例外のみ

比較レポートに例外のみを印刷するか、すべてのレコードを印刷するかを指定します。有効な値は次のとおりです。

1 2つのデータ・ソース間で異なるレコードのみ印刷する

Blank 両方のデータ・ソースにあるすべてのレコードを印刷する

R98611 タブ

レポートに使用するデータ・ソースを指定します。

処理オプションの詳しい内容を見るには、フィールドを右クリックして[ポップ・ヒント]を選択します。または、処理オプションのフィールドをクリックして[F1]キーを押します。

1. データ・ソース名

レポート作成に使用するデータ・ソースを指定します。

データ・ソース・マスターの比較レポートの実行

〈データ・ソース・マスターの比較〉プログラム(R986112)を実行して、2つのデータ・ソース間で、データ・ソース・マスター(R98611)テーブルを比較します。たとえば、システム・データ・ソースの F98611 テーブルと任意のサーバー・マップ・データ・ソースの F98611 テーブルを比較できます。ただし、データ・ソースは各マシンで異なる定義が必要です。

はじめる前に

- レポート・ドライバを使用して、レポートの処理オプションを入力します。

▶ データ・ソース・マスターの比較のデータ選択を設定するには

〈システム・アドミニストレーション・ツール〉メニュー(GH9011)から〈バッチ・バージョン〉プログラム(P98305)を選択します。

1. 〈バッチ・バージョンの処理 - 使用可能なバージョン〉で、[バッチ・アプリケーション]フィールドに R986112 と入力して[検索]をクリックします。
2. バージョン XJDE0001 をダブルクリックします。
3. 〈バージョン・プロンプト〉で、[データ選択]オプションをチェックして[投入]をクリックします。
4. 〈データ選択〉で、適切なカラムから比較するレコードを選択します。
5. [OK]をクリックします。

▶ データ・ソース・マスターの比較を実行するには

〈上級操作〉メニュー(GH9012)から〈データ・ソース・マスターの比較〉プログラム(R98611B)を選択します。

1. 〈バッチ・バージョンの処理 - 使用可能なバージョン〉で、実行するレポートに対応するバージョンを選択します。
2. [選択]をクリックします。
3. 〈バージョン・プロンプト〉で、[投入]をクリックします。

データ・ソース・マスター・レポートの実行

データ・ソース・マスター・レポート(R98611)を実行して、特定のデータ・ソースのデータ・ソース・マスター(F98611)のレポートを印刷します。

▶ データ・ソース・マスター・レポートのデータ選択を設定するには

〈システム・アドミニストレーション・ツール〉メニュー(GH9011)から〈バッチ・バージョン〉プログラム(P98305)を選択します。

1. 〈バッチ・バージョンの処理〉フォームで、[バッチ・アプリケーション]フィールドに“R98611”と入力して[検索]をクリックします。
2. バージョン XJDE0001 をハイライトして[ロー]メニューから[データ選択]を選択します。
3. 処理オプションを設定して[OK]をクリックします。
4. 〈レポート出力先〉で、画面またはプリンタを選択して[OK]をクリックします。
5. 〈データ選択〉で、適切なカラムから選択するレコードを指定します。
6. [OK]をクリックします。

▶ データ・ソース・マスター・レポートを実行するには

〈上級操作〉メニュー(GH9012)から〈データ・ソース・マスター・レポート〉プログラム(R98611B)を選択します。

1. 〈バッチ・バージョンの処理 - 使用可能なバージョン〉で、実行するレポートに対応するバージョンを選択します。
2. [選択]をクリックします。
3. 〈バージョン・プロンプト〉で、[投入]をクリックします。

ミドルウェア

クライアント/サーバー環境では、アプリケーションが異なるプラットフォーム間でデータを送受信する必要があります。プラットフォームによって、通信プロトコル、データベース管理システム、オペレーティング・システムが異なる場合があります。クライアントとサーバーの通信、サーバー同士の通信のために、複数のプロトコルやベンダーの問題を解決するメカニズムが必要です。この機能を果たすのが、ミドルウェアと呼ばれるソフトウェアウェア群です。ミドルウェアはオペレーティング・システムとビジネス・アプリケーションの中間に位置します。アプリケーション構造が一貫したミドルウェアに基づいていることが重要です。

次のようなミドルウェアを提供しています。

JDENet 通信ミドルウェア クライアントとサーバー、およびサーバー同士を接続し、要求に応答してメッセージを送信します。ピアツーピア、メッセージ基準、ソケット基準のマルチプロセス通信ミドルウェア・ソリューションです。

JDEBase データベース・ミドルウェア 複数のデータベース・アクセスに対応したプラットフォーム非依存型のアプリケーション・プログラム・インターフェイス(API)を提供します。これらの API は次の用途に使用されます。

- データ・ソース要求に基づき、プラットフォーム独自の構造化照会言語(SQL)を自動生成するアプリケーションによって。
- C 言語で上級ビジネス関数を書き込むオープン API として。これらの API を使ってプラットフォーム固有の SQL ステートメントが自動的に生成されます。

JDEBase は、クライアントとサーバー間およびサーバー同士のデータベース・アクセスも提供します。これを実現するために、IBM の Client Access や Microsoft の Open Database Connectivity (ODBC) のプログラミング・インターフェイスなど、さまざまなサードパーティのデータベース・ドライバに対応しています。

JDENet 通信ミドルウェアの理解

ネットワーク上で通信するためには、お互いのコンピュータが共通の通信プロトコルを使用する必要があります。通信プロトコルは、異なるデバイス間でデータ交換を行うための形式と関係を指定した正式な規則です。通信ミドルウェアは、次のようなプロトコル階層を処理します。

ネット・ワーク階層 ネットワーク階層は、情報の特定とルーティング、エラー・チェック、要求の再送信を処理します。

トランスポート階層 トランスポート階層は、ネットワーク全体で接続指向のデータ送信サービスを提供します。この階層は、安定した順序立ったデータ交換のためのセッションまたは相互接続を維持しながら、エンドツーエンドのデータ交換を提供します。

TCP/IP プロトコル・スイートがサポートされています。

アプリケーション階層 アプリケーション階層は、アプリケーション同士の交信およびデータ交換を提供します。JDENet は、アプリケーション階層の通信ミドルウェアです。

JDENet 通信ミドルウェア

JDENet は、J.D.Edwards が独自に開発した通信ミドルウェア・パッケージで、サーバー間およびクライアントとサーバー間の通信を提供します。

JDENet はピアツーピアのミドルウェアです。たとえば、ネットワーク上で通信を開始するのがクライアント、それに応答するのがサーバーであると考えます。この場合、クライアントは常に要求を発信することによって通信を開始し、別のマシンからの要求に応答する場合はサーバーとして機能します。ピアツーピアのミドルウェアでのクライアントとサーバーの区別は、どのマシンが通信を開始したかによって決まります。どのプラットフォームで稼動しているマシンかに関係なく、すべてのマシンはクライアントとサーバーのどちらとしても機能できます。

JDENet では、クライアントとサーバー間の通信はメッセージを通して行われます。JDENet のメッセージには、ビジネス関数、バッチ・ジョブ、データ複製、ログイン・セキュリティなどの処理要求があります。JDENet のメッセージはクライアントとサーバーのどちらからでも送信できます。JDENet は、複数サーバー間のサーバー・タイプが異なる場合に限りデータベースの要求を処理します。

アプリケーション要求(メッセージ)は、同期、非同期どちらでも送信できます。ビジネス関数を呼び出すなど同期メッセージの場合は、クライアントはサーバーで処理が完了するのを待つ必要があります。バッチ・プロセスなどの非同期メッセージの場合は、サーバーで要求が処理されている間、クライアントは他のタスクを続行することができます。場合によっては、ビジネス関数も非同期に呼び出すことができます。

ソケットベース通信

ソケットは、アプリケーションがネットワーク上でデータの packets を送受信する通信の終点になります。ソケットはシステムの多重通信チャネルを提供します。JDENet は、ストリーム・ソケットを使ってエンドツーエンドで通信を行います。ソケットはデータを損なうことなく確実に送信します。

メッセージベース通信

メッセージベースの通信では、アプリケーションがロジックまたはデータをメッセージ形式で要求します。送信されたメッセージは処理のため待ち行列に置かれます。メッセージの送信はミドルウェアによって行われるため、クライアントのアプリケーションでは他のタスクを処理することができます。このようなメッセージ処理サービスがない場合、アプリケーションでは要求が処理され、結果が戻ってくるまで待機する必要があります。

メッセージ処理はイベント指向のアプリケーションに最も適しています。これと正反対なのは、同期処理であるリモート・プロシージャ・コール(RPC)です。JDENet のメッセージ処理によって、メッセージを確実に送信します。

プロセスベース設計

クライアント・ワークステーションには複数の EnterpriseOne をインストールできますが、一度に実行できる Windows ベースの EnterpriseOne は 1 つだけです。このソフトウェアでは、JDENet と呼ばれる内部ネットワーク・プロセス(ネット・プロセスとも呼ばれます)を使ってサーバーに要求を送信します。

サーバーにも JDENet と呼ばれるネット・プロセスがあります。このプロセスでは、クライアント・ワークステーションと通信して、要求メッセージを適切なプロセスに割り当てます。次に専用プロセスでは、その処理が DLL、共有ライブラリ、ジョブ待ち行列といったプラットフォームごとのロジック・プロセスに割り当てられます。メイン・プロセスや専用プロセス、DLL、共有ライブラリ、ジョブ待ち行列は 1 つのサーバーで複数持つことができます。

このアーキテクチャの利点は、複数のワークステーションが同じサーバーに同時に要求を出せることです。必要な数のワークステーションからメインのサーバー・プロセスへのセッション接続を維持することができます。また、特定のタイプのワークステーション要求を処理する専用プロセスの総数（および各タイプの総数）も定義することができます。

ネットワーク・プロセス

ネットワーク・プロセス、専用プロセス、ロジック・プロセスは互いに関連しています。この関係を定義するのがエンタープライズ・サーバーの jde.ini ファイルです。各エンタープライズ・サーバーには、JDENET_n ジョブという最低 1 つのネットワーク・プロセスが必要です。このジョブは、EnterpriseOne のネットワーク接続およびトラフィックをコントロールします。

各サーバーの jde.ini ファイルで、複数の JDENET_n プロセスを設定することができます。存在する JDENET_n プロセスの数に関係なく、最初の JDENET_n プロセスは「マスター・リスナー」の機能を果たします。

複数の JDENET_n ジョブを指定した場合、ジョブを各要求に割り当て、要求通りにジョブが開始されます。最多数の JDENET_n プロセスが開始された場合、ネットワーク・プロセス上でロード・バランシング（負荷調整）を行いながら、接続数が最多になるまで現在実行中の JDENET_n ジョブが自動的に切り替えられます。サーバーで JDENET_n ジョブの接続が最大数に達した場合、既存のセッションを終了するまで、そのサーバーにはセッションを追加することができません。JDENET_n への接続は、セッションが確立されている間維持されるよう設計されています。

たとえば、サーバーの jde.ini ファイルで JDENET_n プロセス数を 4 と指定したとします。最初の JDENET_n の要求は、「マスター・リスナー」に送信されます。マスター・リスナーとは、サーバーの起動時に実行された最初の JDENET_n プロセスです。JDENET_n に 2 つ目の要求が送信されると、マスター・リスナーはその要求を 2 つ目の JDENET_n プロセスに割り当てます。これによって 2 つ目の JDENET_n が開始されます。要求を出したマシンとこのサーバーの間でセッションが継続している間、この割当てが維持されます。3 つ目と 4 つ目の JDENET_n の要求が行われた場合も同じ処理が発生します。5 つ目の要求が受信された場合、要求は最初の JDENET_n プロセスに割り当てられ、同様の処理が繰り返されます。

カーネル・プロセス

JDENET_n プロセスは、通信のネットワーク階層を処理する役割を果たします。JDENET_n ジョブが受信したメッセージをロジック処理要求だと判断した場合、適切な JDENET_k ジョブにその要求を送信します。メッセージ識別子に基づいて適切な JDENET_k のジョブが決定されます。JDENET_k のジョブは、JDENET_n のジョブとプラットフォームの該当するジョブ間のリンクを提供するプロセスです。JDENET_k プロセスはサーバーでのみ処理されます。

JDENET_k のジョブはさまざまなロジック・プロセス間で双方向のルーティングを処理し、JDENET_n のジョブは適切なマシンへの返信を処理します。専用カーネルには 11 のタイプがあり、それぞれが特定のタイプのプロセスを処理する役割を果たします。

ロジック・プロセスの例として、Windows NT のダイナミック・リンク・ライブラリ(.dll)、UNIX の共有ライブラリ(.sl)、AS/400 の JDENet プロセスがあります。

参照

- JDENet について詳しくは『システム・アドミニストレーション』ガイドの「ミドルウェア」

JDEBase データベース・ミドルウェアの理解

各データベース管理システム(DBMS)には、それぞれ独自の構造化照会言語(SQL)があります。たとえば、Microsoft SQL Server、Oracle、DB2/400 が同じ SQL ステートメントをどのように扱うかを次に示します。

Microsoft SQL Server `SELECT * FROM PRODDTA.F0101`

Oracle `SELECT * FROM PRODDTA.F0101;`

DB2/400 `SELECT * FROM PRODDTA/F0101`

データベースのミドルウェア階層は、異なる SQL を解釈するための共通のインターフェイスを提供します。JDEBase は PeopleSoft のデータベース・ミドルウェア製品です。これは共通のアプリケーション・プログラミング・インターフェイス(API)で、開発者はこの API を呼び出すことによって、データの要求とデータ操作ロジックを実行することができます。JDEBase は一般的な API を解釈し、システムがデータベースにアクセスするために SQL を適切なステートメントに変換します。

分散環境で複数のデータベースがある場合、データベースの整合性を確認するためのモニタリング・プログラムが必要です。このモニタリング・プログラムは、トランザクション・モニタと呼ばれます。JDEBase データベース・ミドルウェアには、トランザクション・モニタが備わっています。

JDEBase には次のような特徴があります。

- プラットフォームごとに異なる SQL コーディングを意識しないプログラミング
- ドライバのスピーディな開発
- サーバー間コミュニケーション
- トランザクション処理

JDEBase は開発者に API を提供し、このソフトウェアには変換プログラムを提供します。この変換プログラムはデータ・ソース定義に組み込まれています。

たとえば、住所録マスターにデータ要求が出されたとします。オブジェクト構成マネージャ(OCM)は、要求されたテーブルがどのデータ・ソースに含まれるかを判断します。データベース情報は、データ・ソース・マスター(F98611)にあります。

処理モード

対話型アプリケーションを実行するにはさまざまな方法があります。ユーザーのニーズとシステム構成に応じて適切な方法を選択することができます。どの方法を使用しても、ワークステーションの能力とサーバーのデータ整合性が最適となるようにデータやロジックを分散することができます。

対話型アプリケーションには 4 つの操作モードがあります。

ダイレクト接続処理

ダイレクト接続処理では、アプリケーションを実行すると主要なビジネス・データがあるサーバーに接続されます。ダイレクト接続モデルではデータおよびロジックを分散することができます。パフォーマンスを向上させるため、ネットワーク・トラフィックとサーバーでの不要な入出力を減らす方法でデータとロジックを分散することを考慮してください。

バッチ・アプリケーションをローカルで実行することは可能ですが、運用におけるその必要性はほとんどありません。このため、すべてのバッチ・アプリケーションはサーバーで処理(分散処理)するように設定してください。

参照

- ダイレクト接続モデルについて詳しくは『CNC インプリメンテーション』ガイドの「ダイレクト接続処理について」

オフライン処理

オフライン処理モデルは、サーバーに接続していないワークステーションでトランザクションを入力する必要があるユーザー用に設計されています。たとえば、外出先で営業担当者がラップトップ・コンピュータでオーダーを入力します。会社に戻ってから、ネットワークに接続してデータを転送することができます。

オフライン処理用に設計されたアプリケーションは、通常のアプリケーションとは別のアプリケーションです。これらのアプリケーションでは、静的データの検証だけでなく、要求を処理するために正確でなければならないその他の重要な情報の検証も行われます。基本的な検証が行われた後、ワークステーション上の Z ファイルと呼ばれるワーク・テーブルにトランザクションが保管されます。ネットワークに接続すると、Z ファイルがエンタープライズ・サーバーにアップロードされます。トランザクションはマスター・ビジネス関数により再び検証され、トランザクション・ファイルにレコードが更新されます。

ユーザーが開始するバッチ・プロセスであるアップロード処理では、ワークステーションからデータが転送され、次の処理が行われます。

- オーダー情報をワークステーションのワーク・テーブルからサーバーのワーク・テーブルにコピーする。
- サーバーのトランザクション制御テーブル(F0041Z1)で 1 件のトランザクションに対して 1 つのレコードを更新する。この際、更新済み、処理中、エラー終了、正常終了のいずれかの状況が与えられます。
- トランザクションがアップロード済みであることを示すようにワークステーションの F0041Z1 テーブルを更新する。これにより、同じトランザクションが再びアップロードされることを防ぎます。

- サーバー上でビジネス関数がすべてのローカル・データの再検証、およびオーダー入力時に重要と見なされなかったその他の情報の検証を行い、正式なトランザクション・テーブルにレコードを書き込む。
- ワークセンターに正常終了またはエラーを通知するメッセージを送信する。ユーザーはこのメッセージを確認して削除することができます。
- サーバーおよびワークステーションの F0041Z1 テーブルで、アップロードで処理された各トランザクションの状況を更新する。

ワークセンターでは、完了した各ジョブのメッセージ(エラーの有無が表示されます)が処理を実行したユーザーに送信されます。エラーを解消するには、詳細を参照してください。エラーを解決するには何らかのアクションが必要な場合、エラーが発生したトランザクションを見てください。

参照

- オフライン処理モデルについて詳しくは『CNC インプリメンテーション』ガイドの「オフライン処理について」

バッチオブワン処理

バッチオブワン処理は、オフライン処理の利点に、トランザクション・データのリアルタイム更新というダイレクト接続の利点に加えられた処理です。バッチオブワン処理では、PD7334 などの標準的な本稼動環境でオフライン処理アプリケーション(P4004Z)を使用します。つまり、ローカルのワークステーションは物理的にネットワークに接続されます。P4004Z アプリケーションは Z ファイルを使用しますが、これらのテーブルはエンタープライズ・サーバー上で検索されます。各受注オーダーは、エンタープライズ・サーバーに直接書き込まれます。

バッチオブワン処理では、P4004Z アプリケーションによってフィールドごとの検証がほとんど行われないため、ネットワーク・トラフィックが減少します。このため、広域ネットワーク(WAN)での受注オーダー入力には理想的な方法です。

バッチ・オブ・ワン処理を使用する場合、全受注オーダーの最新かつ完全なデータが保管されている F4201 および F4211 テーブルにはユーザーは直接アクセスできません。P4004Z アプリケーションはこれらのテーブルに直接アクセスしないため、Z ファイルにあるオーダー情報のみ参照、更新、削除することができます。オーダーが本稼動環境のテーブル(F4201 および F4211)に移動された後は、P4210 アプリケーションを使って受注オーダーを検討、更新、削除することができます。

サブシステムまたはオンライン・モード処理を設定するには、〈定期オーダー入力〉アプリケーション(P4004Z)の処理オプションを変更してください。

サブシステム

バッチオブワン処理のサブシステムには、R40211Z バッチ・アプリケーションがバックグラウンドで実行されるバージョンがあります。サブシステムのバッチ・アプリケーションは、定期的にエンタープライズ・サーバーの Z ファイルをチェック、処理可能なデータがある場合はバッチ検証を実行します。そして、エンタープライズ・サーバーの Z ファイルから本稼動環境の F4201 および F4211 テーブルにデータを転送します。

注:

バッチ・アプリケーション R40211Z のバージョン ZJDE0002 は 30 ミリ秒ごとに送信されます。つまり、このバッチ・アプリケーションは毎秒 33.3 回の頻度で実行され、エンタープライズ・サーバーの CPU を無駄に使うこととなります。バッチオブワンのサブシステム・モードを本稼動環境で使用する場合は、15,000 ミリ秒、つまり 15 秒ごとに実行するよう設定してください。

オンライン・モード

オンライン・モードでは、受注オーダーが完了するとプリンタ選択画面が表示されます。プリンタを選択した後、バッチ・アプリケーション R40211Z によってバッチの検証が行われ、エンタープライズ・サーバーの Z ファイルから本稼動環境の F4201 および F4211 テーブルにデータが転送されます。

発生したエラーはすべてワーク・センターに送信されます。新しいメッセージが送信されるたびにプロンプトを表示するか、そうしない場合はすべてのメッセージをまとめて見ることができます。

ゼロクライアント処理

ゼロクライアント処理を使用すると、ワークステーションからインターネット・ブラウザだけを使ってアプリケーションおよびデータに接続することができます。この処理モードとダイレクト接続処理は、データとロジックが 1 か所に保管されるという点で類似しています。この方法は、ワークステーションのディスク・スペースを使わないため管理が簡単です。

参照

- ゼロクライアント処理について詳しくは『Web-Based Solutions (ウェブベース・ソリューション)』ガイド

ダイレクト接続処理について

ダイレクト接続処理では、ワークステーションはデータを保管してロジックを処理するサーバーに直接接続されます。パフォーマンスを向上させるため、ネットワーク・トラフィックとサーバーでの不要な入出力を減らす方法でデータとロジックを分散することを考慮してください。

データとロジックの分散に関する推奨事項

ダイレクト接続でパフォーマンスを最大にするために、データとロジックの処理には次のロケーションを使用することをお勧めします。

- トランザクション・データはデータ・サーバーにマップする。
- ユーザー定義コードとメニューはワークステーションにマップする。その他の静的ファイルもローカルにマップすることができます。静的ファイルをローカルにマップしてもメンテナンス費用に対して期待するパフォーマンスが得られない場合は、データ・サーバーにマップすることができます。
- すべてのバッチ・アプリケーションはエンタープライズ・ロジック・サーバーにマップする。

ダイレクト接続処理の設定

ダイレクト接続処理を設定する基本的なステップは次のとおりです。

- 本稼動環境を作成し、新しい環境が本稼動用パス・コードを使用していることを確認する
- サーバーにマップするマスター・ビジネス関数の名前を決定する(必要な場合)
- 新しい環境の OCM マッピングを修正する

参照

- 本稼動用パス・コードの確認については、『CNC インプリメンテーション』ガイドの「環境の追加」
- サーバーへのマッピングについて詳しくは『CNC インプリメンテーション』ガイドの「マスター・ビジネス関数の検索」
- ダイレクト接続処理について詳しくは『CNC インプリメンテーション』ガイドの「オブジェクト構成マネージャ(OCM)の処理」

ダイレクト接続環境のオブジェクト・マッピング

システム構成によって、複数のダイレクト接続環境が必要な場合があります。たとえば、多階層のシステム構成をサポートするために、すべてのアプリケーション処理を1つのサーバーにマップし、すべてのデータを本社サーバーにマップする環境が必要な場合などです。

マスター・データ管理環境

データ管理者は、パブリッシュされたテーブルを1つのロケーションで集中的に管理するのにマスター・データ管理環境を使用します。この環境は次の手順で設定します。

- すべてのテーブル・オブジェクトをサーバーのビジネス・データ・ソースにマップする
- ユーザー定義コードとメニューをサーバーのコントロール・テーブル・データ・ソースにマップする

コントロール・データには、パブリッシュ済みのユーザー定義コードが含まれています。

- システム・テーブルをシステム・データ・ソースにマップする
- オブジェクト・ライブラリアン・テーブルをオブジェクト・ライブラリアン・データ・ソースにマップする
- データ辞書テーブルをデータ辞書データ・ソースにマップする
- バッチ・アプリケーションをサーバーにマップする

マスター・ビジネス関数の検索

ダイレクト接続環境では、マスター・ビジネス関数を検索してサーバーにマップする必要があります。

オフライン処理について

オフライン処理では、サーバーに接続していないマシンでトランザクションを入力し、後でサーバーに接続してアップロードすることができます。たとえば、外出先の営業担当者がラップトップにトランザクションを入力し、会社に戻ってからサーバーにそのデータをアップロードすることが可能です。オフライン処理は社内でも利用できます。サーバーに接続しない状態でトランザクションを入力すると、アプリケーションではすべての検証をローカルで行うため処理効率がアップします。ローカルで入力したデータはネットワークのトラフィックが少ない時間にアップロードします。

オフライン処理では最小限のデータ検証しか行われなため、スピーディに入力することが可能です。ビジネス関数による完全な検証は、オフライン処理のトランザクションをエンタープライズ・サーバーにアップロードしたときに行われます。

オフライン処理では次の環境を用意する必要があります。

- 非接続時のトランザクション入力のためのオフライン処理環境
- エンタープライズ・サーバーにトランザクションをアップロードするためのダイレクト接続環境

重要

データの整合性を維持するために、オフライン処理を利用する各ユーザーに対して重複しない番号または伝票タイプが必要です。業務要件に応じて、次のようなソリューションをお勧めします。

- オフライン処理ユーザーが少数の場合、各ユーザーに異なる伝票タイプを割り当てる。たとえば、ユーザーAには伝票タイプ S1、ユーザーBには伝票タイプ S2 を割り当てます。ユーザーA がオーダー番号 1009 をアップロードすると、1009S1 という番号がオーダーに割り当てられます。ユーザーB がアップロードした場合は 1009S2 という番号になります。この方法では、各営業担当者が作成したデータが別個のビジネス・データとしてアップロードされるため、データの上書きや破損などの問題を防ぐことができます。

注:

伝票タイプはその他さまざまな処理で使用されるため、すべての伝票タイプをオフライン処理ユーザーに割り当てないでください。

伝票タイプは A0～Z9 まで使用できます。オフライン処理ユーザーにすべての伝票タイプを割り当てる必要がある場合、データの整合性を維持するために次の方法を検討してください。

- オフライン処理ユーザーが多数の場合、レコードに割り当てる自動採番の範囲が各ユーザーで重複しないようにします。たとえば、ユーザー1には 0001～9999、ユーザー2には 10000～19999 を割り当てます。同じ番号が複数のレコードに使用されないため、データの整合性を維持できます。この方法は管理が複雑になるため、他に適切な方法がない場合のみ使用してください。

オフライン処理の設定

オフライン処理の設定は、各ワークステーションのインストール後に1度だけ行います。

はじめる前に

- オフライン処理用の新しい環境を作成してください。この環境には本稼動用のパス・コードを使用する必要があります。この環境には SFPD9 またはこれに似た名前を付けることができます。

既存の環境をコピーするとその環境の OCM マッピングもコピーされてしまうため、コピーによって新しい環境を作成しないでください。また、オフライン環境ではジャストインタイム・インストールのフラグを N に設定してください。

注:

詳しくは『CNC インプリメンテーション』ガイドの「環境の追加」を参照してください。

▶ オフライン処理を設定するには

〈システム・アドミニストレーション・ツール〉メニュー(GH9011)から〈オブジェクト構成マネージャ(OCM)〉プログラム(P986110)を選択します。

1. ローカルの Microsoft Access データ・ソースを参照する OneWorld Local - PD9 というオフライン処理環境用のテーブル(TBLE)オブジェクトに対するデフォルト・マッピングを作成します。
2. オフライン処理環境を設定した後、オフライン処理ユーザーの環境リストに追加します。
3. デフォルトのオフライン処理データベース(StoreFwd.mdf)をシステム管理者のローカル・ワークステーションの b9¥pd9¥data ディレクトリにコピーします。

データベースは、デプロイメント・サーバーの¥¥デプロイメント・サーバー名¥b9¥planner¥data ディレクトリに置かれます。そのデータベースをシステム管理者のローカル・ワークステーションの b9¥pd9¥data ディレクトリにコピーします。これで、システム管理者のワークステーションの b9¥pd9¥data ディレクトリの下に 2 つのデータベースができます。

4. この b9¥pd0¥data ディレクトリから、JDELocal_Data.mdf データベースにある次のテーブルを StoreFwd.mdb データベースにエクスポートします。

F0004	ユーザー定義コード・タイプ
F0005	ユーザー定義コード
F0082	メニュー・マスター
F00821	メニュー選択
F0083	メニュー・テキスト一時変更
F0084	メニュー・パス

将来のリリースに最新のテーブルが含まれるように、システム管理者のワークステーションがメニューおよびユーザー定義コード・テーブルのサブスクリバとして設定されていることを確認してください。

5. システム管理者のワークステーションの b9¥pd9¥data ディレクトリから JDELocal_Data.mdf データベースを削除して、StoreFwd.mdf を JDELocal_Data.mdf に変更します。
6. Microsoft Access を使用して、システム管理者のワークステーション上の JDELocal_Data.mdf にすべてのテクニカル・データとビジネス・データのテーブルがあり、ユーザー定義コードとメニューを除くすべてのテーブルが空であることを確認します。

足りないテーブルがある場合、¥¥デプロイメント・サーバーのマシン名¥b9¥planner¥data にある JDELocal_Data.mdf からインポートしてください。構造のみインポートし、データはインポートしないでください。つまり、空のテーブルをインポートします(ユーザー定義コードとメニューを除く)。

7. すべてのデータ辞書テーブル(F92*)またはデータ辞書 TAM ファイルをオフライン処理のクライアント・マシンにコピーします。

オフライン処理の基本的な設定はこれで完了です。オフライン処理を有効にするには、この章で説明されているシステム管理者とエンドユーザーのタスクを最低 1 回は実行する必要があります。

参照

- オフライン処理用に新しい環境を作成する方法については、『CNC インプリメンテーション』ガイドの「環境の追加」
- テーブル・オブジェクトのデフォルト・マッピングについては、『CNC インプリメンテーション』ガイドの「オブジェクト構成マネージャ(OCM)の処理」
- テーブルの詳細情報と全テクニカル・データ・テーブルの一覧は、『CNC インプリメンテーション』ガイドの「オフライン処理に必要なテクニカル・データ・テーブル」
- テーブルの詳細情報と全ビジネス・データ・テーブルの一覧は、『CNC インプリメンテーション』ガイドの「オフライン処理に必要なビジネス・データ・テーブル」

オフライン処理の管理

システム管理者のワークステーションに JDELocal_Data.mdf データベースを作成してください。エンドユーザーがオフライン処理を実行するには JDELocal_Data.mdf データベースが必要です。このタスクはサーバーのオフライン処理データベースをリフレッシュします。

重要

上記のタスクは最低 1 回は実行してください。オフライン処理ユーザーに正確で最新のデータを提供するために、システム管理者は定期的にこのタスクを実行する必要があります。

テクニカル・データのダウンロード

オフライン処理を実行するには、本稼働用環境のテクニカル・データ・テーブルをシステム管理者のワークステーションにコピーする必要があります。〈環境データベース作成〉プログラム(R98403)を使ってこの処理を実行します。このプログラムでは Copy Table スクリプトが使用されます。この処理が完了すると、オフライン処理パッケージをインストールしたユーザーはテクニカル・データをローカルで実行できます。(オフライン・トランザクション処理中は、ユーザーはサーバーにアクセスすることができません。)

システム管理者のワークステーションには、最新の本番環境パッケージがインストールされている必要があります。テクニカル・データに変更があった場合にのみ、このタスクを実行してください。

▶ テクニカル・データをダウンロードするには

〈システム・アドミニストレーション・ツール〉メニュー(GH9011)から〈バッチ・バージョン〉プログラム(P98305)を選択します。

1. 〈バッチ・バージョンの処理〉フォームで、[バッチ・アプリケーション]フィールドに“R98403”と入力して[検索]をクリックします。
2. Populate System Tables for Store & Forward(システム・テーブルの入力追加)バージョンをハイライトして[選択]をクリックします。
3. 〈バージョン・プロンプト〉で、[データ選択]オプションをオンにして[投入]をクリックします。
4. 〈データ選択〉で、すべてのテクニカル・データ・テーブルが選択されていることを確認します。
5. [OK]をクリックします。
処理オプションが表示されます。
6. 処理オプションに次の値を入力して[OK]をクリックします。

環境タブ	
1. データベースを作成するターゲット環境名を入力してください(このレポートが別の処理から呼び出されている場合は、ターゲット環境が挿入されます)。	SFPD9 など、オフライン処理用に作成した環境の名前を入力します。
2. または、作成するデータベースのターゲット・データ・ソース名を入力してください。	上のオプションを入力しない場合は、オフライン処理のデータ・ソース名を入力してください。
3. 本稼働用データをロードするには“1”、デモ用データをロードするには“2”を入力してください。デフォルトでは本稼働用データがロードされます(このレポートが別の処理から呼び出されている場合は、このフラグが挿入されます)。	このバッチ・プロセスではオブジェクト・ライブラリアンのレコードの[データのコピー(Y/N)]フラグで、データをコピーするかどうか判断されるため、デモ用データの場合は常に“2”を入力してください。これらのテクニカル・テーブルには[データのコピー(Y/N)]フラグがNと指定されているものがあるため、この処理オプションに“2”を入力すると、このフラグを無視してすべての本番環境データがコピーされます。
4. データをロードするソース・データ・ソース名を入力してください(このレポートが別の処理から呼び出されている場合は、ソース・データ・ソースが挿入されます)。	多くのデータ・ソースからコピーする場合はこの処理オプションを空白にして、次の処理オプションを設定してください。この処理オプションを設定する場合、ERP 9.0 Local などのデータ・ソース名を入力します。

5. または、データベースのコピー元となるソース環境名を入力してください(このレポートが別の処理から呼び出されている場合は、ソース環境が挿入されます)。	データ・ソースではなくソース環境の名前を入力します。1つの環境の中にコピーするデータ・ソースが複数ある場合、上の処理オプションではなく、この処理オプションを使用してください。
更新タブ	
6. テスト・モードには"0"、環境データベースを作成するには"1"を入力してください。デフォルトはテスト・モードです。	レポートを生成してテクニカル・テーブルをローカル・マシンにコピーするには"1"を入力します。
7. テーブルの自動生成が可能なデータ・ソースに既存のテーブルを再生成するには、"A"を入力してください。デフォルトでは、テーブルは再生成されません。	ファイルをクリアしてデータを再生成するには"A"を入力します。
8. ソース・データ・ソースにあるテーブルのみをコピーするには、"1"を入力してください。デフォルトでは、すべてのテーブルがコピーされます。	ブランクのままにします。
9. ターゲット・テーブルを消去せずにレコードを追加するには、"Y"を入力してください。これは言語テーブルに使用されません。デフォルトでは、ターゲット・テーブルが消去されます。	ブランクのままにします。
印刷タブ	
10. 例外のみを印刷する場合は、"1"を入力してください。デフォルトでは、すべての行が印刷されます。	例外行のみを印刷する場合は、"1"を入力します。レポートを印刷するには、この処理オプションをブランクのままにします。 このレポートには、ダウンロード中に発生したエラーがすべて印刷されます。システム管理者のワークステーションの JDELocal_Data.mdf にないテクニカル・テーブルも印刷されます。
ライセンス・タブ	
11. すべてのテーブルを作成するには"Y"、ライセンスのあるテーブルのみを作成には"N"を入力してください。デフォルトでは、すべてのテーブルが作成されます。	すべてのテーブルを作成するには、"Y"を入力します。
12. すべてのテーブルをレポートに印刷するには"Y"、ライセンスのあるテーブルのみ印刷には"N"を入力してください。デフォルトでは、すべてのテーブルが印刷されます。	すべてのテーブルを印刷するには、"Y"を入力します。

処理オプションを設定した後テーブルのコピーが実行され、〈テーブルのコピー〉メッセージ・ボックスにダウンロード状況が表示されます。

ビジネス・データのダウンロード

ビジネス・データは複数のオフライン処理ユーザーで共有されるため、サーバーからすべてのオフライン処理アプリケーションを含むマスター・テーブルのレコードを JDELocal_Data.mdf データベースにダウンロードするこのバッチ・プロセスを実行してください。

はじめる前に

- データベース管理アプリケーションを使って、システム管理者の JDELocal_Data.mdf にすべてのビジネス・データ・テーブル定義があることを確認してください。

▶ ビジネス・データをダウンロードするには

〈システム・アドミニストレーション・ツール〉メニュー(GH9011)から〈バッチ・バージョン〉プログラム(P98305)を選択します。

1. 〈バッチ・バージョンの処理 - 使用可能なバージョン〉で、[バッチ・アプリケーション]フィールドに”P00140”と入力して[検索]をクリックします。

注:

レコードがないすべてのビジネス・データ・テーブルに対しては、〈環境データベース作成〉プログラム(R98403)を実行する必要があります。

製品出荷時に含まれているすべてのバージョンが表示されます。これらのバージョンは、ビジネス・データ・テーブルをダウンロードするバッチ・プロセスです。

システム管理者のローカル・マシンにコピーする必要があるビジネス・データ・テーブルの各バージョンを実行してください。

2. ビジネス・データ・テーブルのバージョンを選択して、[ロー]メニューから[バージョンの実行]を選択します。
3. 〈バージョン・プロンプト〉で、[投入]をクリックします。
4. 〈レポート出力先〉で、レポートの出力先を選択して[OK]をクリックします。
5. 〈環境一時変更〉で、次のフィールドにソース環境名を入力して[OK]をクリックします。
 - ソース環境一時変更
設定した全パブリッシャ・テーブルを参照する通常の本稼動用環境であることを確認してください。
 - ターゲット環境一時変更
すべての OCM マッピングが ERP 9.0 Local を参照するオフライン処理環境であることを確認してください。

注:

〈環境一時変更〉で[検索]ボタンをクリックすると、〈環境の検索〉フォームが表示されます。適切な環境をハイライトして[選択]をクリックしてください。

6. オフライン処理要件に応じて、〈オフライン処理ダウンロード〉プログラム(P00140)の適切なバージョンを実行します。

各バージョンは、JDELocal_Data.mdf に 1 つのマスター・テーブルをダウンロードします。バッチ・プロセスはサーバーではなくローカルで実行してください。

サーバーからワークステーションにデータベースをコピーした後、このバッチ・プロセスを実行することができます。つまり、システム管理者はこのステップを省略してビジネス・データ・テーブルを空にすることができます。

エンドユーザーのデータベース・アクセスの許可

テクニカルおよびビジネス・データのダウンロードによって作成された JDELocal_Data.mdf のデプロイメントを行う必要があります。このデータベースはオフライン処理ユーザーによってワークステーションにコピーされるため、オフライン処理ユーザーがこのデータベースにアクセスできるように権限を設定する必要があります。

▶ エンドユーザーのデータベース・アクセスを許可するには

1. システム管理者のワークステーションにある JDELocal_Data.mdf をデプロイメント・サーバー上の共有フォルダにコピーします。
2. ビジネス・データを同期化する必要がある場合は、このデータベースをリフレッシュします。

jde.ini ファイルのオフライン処理設定の確認

エンドユーザーのワークステーションでオフライン処理を実行するには、それらのユーザーの jde.ini の [DB SYSTEM SETTINGS] セクションを正しく設定する必要があります。

▶ jde.ini ファイルのオフライン処理設定を確認するには

1. オフライン処理エンドユーザーの jde.ini ファイルの [DB SYSTEM SETTINGS] セクションで、デフォルト環境が SFPD9 になっていることを確認します。
2. [DB SYSTEM SETTINGS - SECONDARY] セクションで、デフォルト値が ERP 9.0 Local - PD9 になっていることを確認します。このデフォルト環境は、プライマリ・データベースへの接続がタイムアウトすると、ローカルの JDELocal_Data.mdf にアクセスします。

また、setup.inf ファイルでデフォルト環境を本番環境の SFPD9 に変更すると、オフライン処理ユーザーはワークステーションをインストールするたびに jde.ini ファイルを更新する必要がありません。次のように設定してください。

```
[DB SYSTEM SETTINGS]
```

```
Default Env=SFPD9
```

```
Default Pathcode=PD9
```

```
[DB SYSTEM SETTINGS - SECONDARY]
```

```
Default Env=System Local - PD9
```

```
Default Pathcode=PD9
```

エンドユーザー・データベースの更新

ワークステーションでオフライン処理を実行できるようにするには、オフライン処理エンドユーザーはこのタスクを最低 1 回実行する必要があります。

注意:

正確かつ最新のデータにアクセスするために、ユーザーはこのタスクを定期的に行う必要があります。オフライン処理の実行頻度に基づいて、サーバーのデータベースからワークステーションの JDELocal_Data.mdf をリフレッシュする方針を決定してください。

▶ エンドユーザーのデータベースを更新するには

1. オフライン処理ユーザーのワークステーションに、最新の本番環境パッケージがインストールされていることを確認します。
2. ワークステーションがネットワークに接続された状態で、サーバーの共有フォルダにある JDELocal_Data.mdf をローカルの b9\data ディレクトリにコピーします。

オフライン処理トランザクションのアップロード

オフライン処理トランザクションをエンタープライズ・サーバーにアップロードする方法を説明します。ローカル・マシンのワークファイルからサーバーのワークファイルにトランザクションをアップロードするバッチ・アプリケーションを使用します。

▶ オフライン処理トランザクションをアップロードするには

オフラインでのトランザクション入力が完了したら、オフライン処理トランザクションをアップロードすることができます。いったん ERP 9.0 をログオフし、ワークステーションを再起動してネットワークに接続し、通常の本稼働環境にログオンします。

〈システム・アドミニストレーション・ツール〉メニュー(GH9011)からバッチ・バージョン(P98305)を選択します。

1. 〈バッチ・バージョンの処理 - 使用可能なバージョン〉で、[バッチ・アプリケーション]フィールドに次のいずれかの値を入力して[検索]をクリックします。
 - R0911Z1(オフライン仕訳のアップロード)
 - R0411Z1(オフライン処理レコードのアップロード)
 - R42101Z(オフライン受注オーダーのアップロード)

オフライン処理で入力したトランザクションのタイプごとに、必要なバージョンを実行する必要があります。

2. バージョンをハイライトして[選択]をクリックします。

注意:

データを正しく処理するために、このバッチ・アプリケーションはローカルで実行してください。

この処理によって、次のワークファイルがワークステーションからサーバーにコピーされます。

- トランザクション制御テーブル(F0041Z1)

- 仕訳アップロード・テーブル(F0911Z1)
- 伝票アップロード・テーブル(F0411Z1)
- バッチ入荷確認 - オーダー見出し(F4001Z)
- バッチ入荷確認 - オーダー明細(F4011Z)

各アップロード処理を実行すると、サーバーにアップロードされたすべてのトランザクションを表示するレポートが作成されます。

3. このレポートでトランザクションが正しくアップロードされたことを確認してください。
4. サーバーから ERP 9.0 を実行します。
5. 〈バッチ・バージョンの処理 - 使用可能なバージョン〉で、[バッチ・アプリケーション]フィールドに次のいずれかの値を入力して[検索]をクリックします。
 - R09110Z(バッチ仕訳処理)
 - R04110Z2(オフライン伝票バッチ処理)
 - R40211Z(定期オーダーの編集/作成)

このプログラムを実行すると、ローカル・マシンのワークファイルのデータを使ってサーバーのトランザクションにレコードが作成されます。

オフライン処理はこれで完了です。

オフライン処理に必要なテクニカル・データ・テーブル

オフライン処理では次のテクニカル・データ・テーブルが使用されるため、ユーザーのローカル・マシンに置く必要があります。

テーブル	説明
F0002	自動採番
F00021	会社/会計年度別自動採番
F0092	ライブラリ・リスト - ユーザー
F00921	ユーザー表示設定
F00922	ユーザー表示設定タグ・ファイル
F00924	ユーザー・インストール・パッケージ
F00925	ユーザー・アクセス定義
F0093	ライブラリ・リスト管理
F0094	ライブラリ・リスト・マスター
F00941	環境詳細 - OneWorld

F00942	オブジェクト・パス・マスター
F00945	リリース・マスター
F00950	セキュリティ・ワークベンチ・テーブル
F98101	イメージング固定情報
F983051	バージョン・リスト
F98306	処理オプション・テキスト
F9860	オブジェクト・ライブラリアン・マスター
F986101	オブジェクト構成マスター
F98611	データ・ソース・マスター
F98613	ビジネス・ビュー環境サーバー
F9865	フォーム情報ファイル
F98825	パッケージ・デプロイメント・スケジュール
F98950	ユーザー一時変更テーブル
F98980	言語別フォント一時変更

参照

- ローカル・マシンへのテーブルのコピーについて詳しくは『CNC インプリメンテーション』ガイドの「テクニカル・データのダウンロード」

オフライン処理に必要なビジネス・データ・テーブル

オフライン処理を行うユーザーのローカル・マシンには、〈伝票入力〉プログラム(P0411Z1)、〈仕訳入力〉プログラム(P0911Z1)、〈受注オーダー入力〉プログラム(P4004Z)を実行するために必要なビジネス・データ・テーブルが必要です。これらのテーブルをローカル・マシンにコピーするには、各テーブルをそれぞれコピーする必要があります。

はじめる前に

- 伝票入力、仕訳入力、受注オーダー入力などのアプリケーションを実行するためにユーザー定義コード・テーブル(F0004 および F0005)をダウンロードする前に、次の点を考慮してこれらのテーブルが必要かどうかを判断してください。
 - これらのテーブルでデータ選択を指定する必要がある。
 - システム管理者からエンドユーザーに提供された JDELocal_Data.mdf にはこれらのテーブルが含まれていない。

伝票入力(P0411Z1)

テーブル	説明
F0004	ユーザー定義コード・タイプ
F0005	ユーザー定義コード
F0006	ビジネスユニット・マスター
F0008	会計期間パターン
F0008B	52 会計期間パターン
F0009	一般会計固定情報
F0010	会社固定情報
F0012	AAI マスター
F0013	通貨コード
F0014	支払条件
F0022	税規則
F0041Z1	トランザクション制御
F0101	住所録マスター
F0111	人名録
F0150	住所組織構造マスター
F0116	日付別住所
F0401	仕入先マスター
F0411Z1	伝票アップロード・テーブル
F0901	勘定科目マスター
F0907	勘定科目表形式
F0911Z1	仕訳アップロード・テーブル
F4008	税域
F4801	作業オーダー・マスター

注:

これ以外にテーブルが必要な場合もあります。

仕訳入力(P0911Z1)

テーブル	説明
F0004	ユーザー定義コード・タイプ
F0005	ユーザー定義コード
F0006	ビジネスユニット・マスター
F0008	会計期間パターン
F0008B	52 会計期間パターン
F0009	一般会計固定情報
F0010	会社固定情報
F0012	AAI マスター
F0013	通貨コード
F0014	支払条件
F0022	税規則
F0041Z1	トランザクション制御
F0101	住所録マスター
F0111	人名録
F0150	住所組織構造マスター
F0116	日付別住所
F0301	顧客マスター
F0901	勘定科目マスター
F0907	勘定科目表形式
F0911Z1	仕訳アップロード・テーブル

F4008	税域
F4801	作業オーダー・マスター

注:

これ以外にテーブルが必要な場合もあります。

受注オーダー入力(P4004Z)

テーブル	説明
F0004	ユーザー定義コード・タイプ
F0005	ユーザー定義コード
F0006	ビジネスユニット・マスター
F0006D	ビジネスユニット代替記述マスター
F0008	会計期間パターン
F0008B	52 会計期間パターン
F0009	一般会計固定情報
F0010	会社固定情報
F0012	AAI マスター
F0013	通貨コード
F0014	支払条件
F0022	税規則
F0041Z1	トランザクション制御
F0101	住所録マスター
F0111	人名録
F0116	日付別住所
F0150	住所組織構造マスター

F0301	顧客マスター
F0401	仕入先マスター
F0901	勘定科目マスター
F0907	勘定科目表形式
F4001Z	バッチ入荷確認 - オーダー見出し
F40073	優先情報階層
F4008	税域
F4009	流通/製造固定情報
F40095	デフォルト保管場所/プリンタ
F4013	オーダー処理相互参照
F4011Z	バッチ入荷確認 - オーダー明細
F40205	行タイプ制御固定情報
F4070	価格調整スケジュール(上級価格設定システムのみ)
F4071	価格調整タイプ(上級価格設定システムのみ)
F4072	価格調整明細(上級価格設定システムのみ)
F4075	価格変動テーブル(上級価格設定システムのみ)
F4092	価格設定グループ定義
F4094	品目/顧客キーID マスター
F41001	在庫固定情報
F41002	計量単位換算係数
F41003	計量単位標準換算
F4101	品目マスター
F4102	事業所品目
F4106	品目基本価格
F4201	受注見出し
F4207	品目別価格

F4208 顧客別価格

F4801 作業オーダー・マスター

標準的なカスタマイズ構成

このソフトウェアのシステム構成はさまざまな方法で設定することができます。いくつかの例や推奨事項を次に示します。

やむを得ない理由で変更が必要でない限り、標準のセットアップと命名規則に従ってください。標準的なセットアップと命名規則に従っている場合、特殊な設定内容を知らないプロジェクト・チーム以外の担当者とのコミュニケーションも円滑に行うことができ、不要な混乱を防ぐことができます。

システム構成をカスタマイズする場合、標準設定の記述のみを変更します。名前は変更しないでください。標準の命名規則に従うと、アップグレードの際にも手作業の設定は最低限で済みます。

環境記述は、ログイン画面で表示され、ユーザーはここで環境を選択できるため、重要です。環境記述には次の内容を含める必要があります。

- パス・コード
- データ・タイプ(本番環境、テスト、プロトタイプなど)
- データのロケーション
- バッチ・アプリケーションの実行場所

標準的な構成

ここでは、ワークステーションの観点から標準的な構成について説明します。つまり、ここで示すマッピングは、ワークステーションとサーバー間のマッピングであり、サーバー同士のマッピングではありません(たとえば、サーバー・マップのマッピングは取り上げません)。製品は標準的な構成で出荷されます。

環境の理解

システム構成における基本的な環境について理解する必要があります。ここでは次の内容について説明します。

- パス・コード
- 目的
- オブジェクト・マッピング

提供されているデフォルトの環境では、次のデータ・ソースを共有しています。

- System - B9
- Object Librarian - B9
- Data Dictionary - B9

UNIX および Windows NT 環境の共有データ・ソース

UNIX および Windows NT 環境で共有されるデータ・ソースは次のとおりです。

- Central objects
- Versions tables
- Business data
- Control tables
- OneWorld Local

本稼働用環境(PD9)

パス・コード	PD9
セントラル・オブジェクト・データ・ソース	Central Objects - PD9
バージョン・テーブル・データ・ソース	Versions - PD9
ビジネス・データ・データ・ソース	Business Data - PROD
自動採番データ・ソース	Control Tables - Prod
メニュー/UDC データ・ソース	Control Tables - Prod

カンファレンス・ルーム・パイロット環境(PY9)

パス・コード	PY9
セントラル・オブジェクト・データ・ソース	Central Objects - PY9
バージョン・テーブル・データ・ソース	Versions - PY9
ビジネス・データ・データ・ソース	Business Data - CRP
自動採番データ・ソース	Control Tables - CRP
メニュー/UDC データ・ソース	Control Tables - CRP

テスト環境(TS9)

パス・コード	PY9
セントラル・オブジェクト・データ・ソース	Central Objects - PY9
バージョン・テーブル・データ・ソース	Versions - PY9
ビジネス・データ・データ・ソース	Business Data - TEST
自動採番データ・ソース	Control Table - Test
メニュー/UDC データ・ソース	Control Table - Test

開発環境(DV9)

パス・コード	DV9
セントラル・オブジェクト・データ・ソース	Central Objects - DV9
バージョン・テーブル・データ・ソース	Versions - DV9
ビジネス・データ・データ・ソース	Business Data - TEST
自動採番データ・ソース	Control Table - Test
メニュー/UDC データ・ソース	Control Table - Test

プリステイン環境(JD7334)

パス・コード	JD7334
セントラル・オブジェクト・データ・ソース	Central Objects - JD7334
バージョン・テーブル・データ・ソース	Version - JD7334
ビジネス・データ・データ・ソース	Business Data - JDE
自動採番データ・ソース	Business Data - JDE
メニュー/UDC データ・ソース	System Local - JD7334

プランナ環境(JDEPLAN)

パス・コード	PLANNER
セントラル・オブジェクト・データ・ソース	-
プランナ・テーブル・データ・ソース	OneWorld Planner - B9
バージョン・テーブル・データ・ソース	OneWorld Local
ビジネス・データ・データ・ソース	OneWorld Business Data
自動採番データ・ソース	OneWorld Local
メニュー/UDC データ・ソース	OneWorld Local

デプロイメント環境(DEP9)

パス・コード	PLANNER
セントラル・オブジェクト・データ・ソース	-
バージョン・テーブル・データ・ソース	OneWorld Local
ビジネス・データ・データ・ソース	OneWorld Business Data
自動採番データ・ソース	OneWorld Local
メニュー/UDC データ・ソース	OneWorld Local

リモート環境

リモート環境の名前は、3文字のロケーション・コードに Java アプリケーションの場合は J、Windows ターミナル・サーバーの場合は W が付きます。

次に例を示します。

xxxJPD9 - ロケーション xxx、PD9 環境、Windows ターミナル・サーバー

xxxWTS9 - ロケーション xxx、TS9 環境、Windows ターミナル・サーバー

リモート環境のパス・コードとデータ・ソースは、基本的な環境と同じです。

PD9 環境

パス・コード PD9

目的 PD9 は、ワークステーションにテスト/リリース済みのパッケージをインストールするエンドユーザーのための本稼動用環境です。バッチ・アプリケーションはサーバーで実行します。さまざまな分散データやロジック、処理モードについて、本稼動用環境が複数必要になる場合もあります。

PD9 - オブジェクト・マッピング

マッピング	説明
Business Data - PROD	デフォルトのオブジェクト・マッピングです。他のデータ・ソースによってマップされていないテーブルには、このデータ・ソースをデフォルトのマッピングとして使用します。このマッピングには次のテーブルが含まれます。 <ul style="list-style-type: none">F00165 - メディア・オブジェクト
Central Objects - PD9	F98950(ユーザー一時変更)を含むセントラル・オブジェクト・テーブルにマップします。
Control Tables - Prod	自動採番テーブルにマップします。
Data Dictionary - B9	データ辞書テーブルと、次のデータ構造体にあるメディア・オブジェクト・テキストにマップします。 <ul style="list-style-type: none">GT92002 - データ辞書 - 用語解説
Logic Data Source	バッチ・アプリケーションが実行されるマシンにマップします。
Object Librarian - B9	オブジェクト・ライブラリアン・テーブルと、次のデータ構造体にあるメディア・オブジェクト・テキストにマップします。 <ul style="list-style-type: none">GT9860A - オブジェクト・ライブラリアン汎用テキスト構造GT9862A - ビジネス関数注記GT98DSA - データ構造注記 - 構造、項目GT98DSB - データ構造注記 - 構造、項目GT98TMPL - メディア・オブジェクト・テンプレート
OneWorld Local - PD9	ユーザー定義コード・テーブルにマップします。
System - B9	システム・テーブルにマップします。
Versions - PD9	次のテーブルを含むバージョン・テーブルにマップします。 <ul style="list-style-type: none">F983051 - バージョン・リストF98306 - 処理オプション・テキスト

TS9 環境

パス・コード PY9

TS9 環境は、次のことをテストするために使用します。

- カンファレンス・ルーム・パイロット(CRP)段階のソフトウェア
- 開発パス・コードで修正して PY9 に転送済みであるけれども、本稼動用パス・コードへの転送とエンドユーザーへのリリースはされていないカスタマイズ内容

目的

この環境では、外部システムからのデータ移行をテストすることもできます。テスト環境を使用する権限があれば、本稼動環境に影響することなくデータやバージョンを変更することができます。この環境を使ってエンドユーザーのトレーニングを行うこともできます。

この環境ではアプリケーションのテストやデータ移行のテストを行います。インストール・ワークベンチを実行しても、手動で入力しない限りビジネス・データはこの環境にはありません。月次や四半期ごとなど定期的に本番環境のデータを PD9 からコピーして、最新のデータを含めるようにしてください。

TS9 - オブジェクト・マッピング

マッピング	説明
Business Data - Test	デフォルトのオブジェクト・マッピングです。他のデータ・ソースからマップされていないテーブルは、このデータ・ソースをデフォルトのマッピングとして使用します。このマッピングには次のテーブルがあります。 <ul style="list-style-type: none">• F00165 - メディア・オブジェクト
Central Objects - PY9	F98950(ユーザー一時変更)を含むセントラル・オブジェクト・テーブルにマップします。
Control Table - Test	自動採番テーブル、ユーザー定義コード・テーブル、メニューにマップします。
Data Dictionary - B9	データ辞書テーブルにマップします。また、次のデータ構造体のデータ辞書メディア・オブジェクト・テキストにもマップします。 <ul style="list-style-type: none">• GT92002 - データ辞書 - 用語解説
Logic Data Source	バッチ・アプリケーションを実行するマシンにマップします。
Object Librarian - B9	オブジェクト・ライブラリアン・テーブルと、次のデータ構造体にあるメディア・オブジェクト・テキストにマップします。 <ul style="list-style-type: none">• GT9860A - オブジェクト・ライブラリアン汎用テキスト構造• GT9862A - ビジネス関数注記• GT98DSA - データ構造注記 - 構造、項目• GT98DSB - データ構造注記 - 構造、項目• GT98TMPL - メディア・オブジェクト・テンプレート
System - B9	システム・テーブルにマップします。
Versions - PY9	次のバージョン・テーブルにマップします。 <ul style="list-style-type: none">• F983051 - バージョン・リスト• F98306 - 処理オプション・テキスト

PY9 環境

パス・コード PY9

カンファレンス・ルーム・パイロット(CRP)環境 PY9 は、本稼動のための準備環境です。CRP では、固定情報テーブルや会社固定情報、会計期間パターン、品目マスターなどのマスター・テーブルとともに実際のデータを使用します。

目的

本稼動の前にこれらのテーブルを本稼動用環境にコピーします。インストール・ワークベンチを実行しても、手動で入力しない限りビジネス・データはこの環境にはありません。必要に応じて本稼動環境の PD9 からデータをコピーして、テスト・データを最新のデータにしてください。

PY9 - オブジェクト・マッピング

マッピング 説明

Business Data - CRP デフォルトのオブジェクト・マッピングです。他のデータ・ソースからマップされていないテーブルは、このデータ・ソースをデフォルトのマッピングとして使用します。このマッピングには次のテーブルがあります。

- F00165 - メディア・オブジェクト

Central Objects - PY9 F98950(ユーザー一時変更)を含むセントラル・オブジェクト・テーブルにマップします。

Control Tables - CRP 自動採番テーブルにマップします。

Data Dictionary - B9 データ辞書テーブルと、次のデータ構造体にあるメディア・オブジェクト・テキストにマップします。

- GT92002 - データ辞書 - 用語解説

Logic Data Source バッチ・アプリケーションを実行するマシンにマップします。

Object Librarian - B9 オブジェクト・ライブラリアン・テーブルと、次のデータ構造体にあるメディア・オブジェクト・テキストにマップします。

- GT9860A - オブジェクト・ライブラリアン汎用テキスト構造
- GT9862A - ビジネス関数注記
- GT98DSA - データ構造注記 - 構造、項目
- GT98DSB - データ構造注記 - 構造、項目
- GT98TMPL - メディア・オブジェクト・テンプレート

OneWorld Local - PY9 ユーザー定義コード・テーブルにマップします。

System - B9 システム・テーブルにマップします。

Versions - PY9 次のテーブルを含むバージョン・テーブルにマップします。

- F983051 - バージョン・リスト
- F98306 - 処理オプション・テキスト

DV9 環境

パス・コード DV9

目的

開発環境 DV9 は、オブジェクト開発のためのテスト環境です。この環境は TS9 で使用されるテスト・データを共有しています。開発者はこの環境でオブジェクトを修正し、PD9 パス・コードに転送する前にこの環境でテストを行います。

PD9 にオブジェクトを転送すると、まだエンドユーザーにリリースしていない最新の本稼動(PROD)パッケージをインストールして、GRP 環境または TST 環境で必要なテストを行うことができます。

DV9 - オブジェクト・マッピング

マッピング

説明

Business Data - PROD	デフォルトのオブジェクト・マッピングです。他のデータ・ソースによってマップされていないテーブルには、このデータ・ソースをデフォルトのマッピングとして使用します。このマッピングには次のテーブルがあります。 <ul style="list-style-type: none">F00165 - メディア・オブジェクト
Central Objects - DV9	次のセントラル・オブジェクト・テーブルにマップします。 <ul style="list-style-type: none">F98950 - ユーザー一時変更
Control Tables - Production	自動採番テーブルにマップします。
Data Dictionary - B9	データ辞書テーブルと、次のデータ構造体にあるメディア・オブジェクト・テキストにマップします。 <ul style="list-style-type: none">GT92002 - データ辞書 - 用語解説
Logic Data Source	バッチ・アプリケーションを実行するマシンにマップします。
Object Librarian - B9	オブジェクト・ライブラリアン・テーブルと、次のデータ構造体にあるメディア・オブジェクト・テキストにマップします。 <ul style="list-style-type: none">GT9860A - オブジェクト・ライブラリアン汎用テキスト構造GT9862A - ビジネス関数注記GT98DSA - データ構造注記 - 構造、項目GT98DSB - データ構造注記 - 構造、項目GT98TMPL - メディア・オブジェクト・テンプレート
OneWorld Local - DV9	<ul style="list-style-type: none">ユーザー定義コード・テーブルにマップします。
System - B9	システム・テーブルにマップします。
Versions - DV9	次のバージョン・テーブルにマップします。 <ul style="list-style-type: none">F983051 - バージョン・リストF98306 - 処理オプション・テキスト

JD9 環境

パス・コード JD9

デモ・データを使ってプリスティン(未変更)オブジェクトのテストを行います。この環境はトレーニングで使用することもできます。修正済みオブジェクトとプリスティン・オブジェクトを比較するために、この環境が必要です。

目的

カスタマー・サポートで再現できない問題がある場合は、このプリスティン環境で同じ現象が起きるかどうかをお尋ねします。このソフトウェア付属のデモ・データを使って、月次または四半期ごとなど定期的にこの環境のデータをリフレッシュしてください。

JD9 - オブジェクト・マッピング

マッピング	説明
Business Data - JDE	デフォルトのオブジェクト・マッピングです。他のデータ・ソースによってマップされていないテーブルには、このデータ・ソースをデフォルトのマッピングとして使用します。このマッピングには次のテーブルがあります。 <ul style="list-style-type: none">F00165 - メディア・オブジェクト
Data Dictionary - B9	データ辞書テーブルと、次のデータ構造体にあるメディア・オブジェクト・テキストにマップします。 <ul style="list-style-type: none">GT92002 - データ辞書 - 用語解説
Logic Data Source	バッチ・アプリケーションを実行するマシンにマップします。
Object Librarian - B9	オブジェクト・ライブラリアン・テーブルと、次のデータ構造体にあるメディア・オブジェクト・テキストにマップします。 <ul style="list-style-type: none">GT9860A - オブジェクト・ライブラリアン汎用テキスト構造GT9862A - ビジネス関数注記GT98DSA - データ構造注記 - 構造、項目GT98DSB - データ構造注記 - 構造、項目GT98TMPL - メディア・オブジェクト・テンプレート
OneWorld Local - JD9	ユーザー定義コード・テーブルにマップします。
System - B9	システム・テーブルにマップします。
Versions - JD9	次のバージョン・テーブルにマップします。 <ul style="list-style-type: none">F983051 - バージョン・リストF98306 - 処理オプション・テキスト

標準データ・ソースの理解

エンタープライズ・サーバーのデータベースが Oracle または Microsoft SQL Server でデプロイメント・サーバーより高性能の場合、パフォーマンスの向上のため、OneWorld Local を除くすべてのデータ・ソースをデプロイメント・サーバーではなくエンタープライズ・サーバーに置くことをお勧めします。

データ・ソース

データ・ソース名	所有者および目的
	所有者 CRPDTA
Business Data - CRP	ライブラリ CRPDTA CRP ビジネス・データ。本稼動の前に、このデータの多くを Business Data - PROD にコピーします。
	所有者 PRISTDTA
Business Data - JDE	ライブラリ PRISTDTA プリスティン・データは製品に同梱されています。
	所有者 PRODDTA
Business Data - PROD	ライブラリ PRODDTA 本稼動用ビジネス・データ
	所有者 TESDTA
Business Data - TEST	ライブラリ TSTDTA CRP で入力したテスト・データ、または他システムから移行したテスト・データ
	所有者 PY9
Central Objects - PY9	ライブラリ COPY9 PY9 パス・コードと関連付けられたセントラル・オブジェクト・データ・ソース。このパス・コードでオブジェクトの作成、テストを行った後、PD9 パス・コードに転送します。
	所有者 DV9
Central Objects - DV9	ライブラリ CODV9 DV9 パス・コードと関連付けられたセントラル・オブジェクト・データ・ソース。このパス・コードでオブジェクトの作成、テストを行った後、PD9 パス・コードに転送します。
	所有者 JD9
Central Objects - JD9	ライブラリ COJD9 JD9 パス・コードと関連付けられたセントラル・オブジェクト・データ・ソース

	所有者 PD9
Central Objects – PD9	ライブラリ COPD9 PD9 パス・コードと関連付けられたセントラル・オブジェクト・データ・ソース。DV9 パス・コードでテストした後、このデータ・ソースにオブジェクトを転送します。
	所有者 CRPCTL
Control Tables – CRP	ライブラリ CRPCTL CRP 環境で使用されるコントロール・テーブル
	所有者 PRODCTL
Control Tables – Prod	ライブラリ PRODCTL 本稼動環境で使用されるコントロール・テーブル
	所有者 TESTCTL
Control Table – Test	ライブラリ TESTCTL テスト環境で使用されるコントロール・テーブル
	所有者 DD9
Data Dictionary – B9	ライブラリ DD9 すべての環境で使用する唯一のデータ辞書
	所有者なし
LOCAL	ローカル・マシンを定義するデータ・ソース。これを使ってワークステーションへのレポートを一時変更します。
	所有者なし
マシン名	ロジック・ホストを定義する値。
	所有者 SVM9
マシン名 – Server Map	ロジック・サーバーのサーバー・マップ
	所有者 OL9
Object Librarian – B9	リリースごとのオブジェクト・ライブラリアン
	所有者なし
OneWorld Local	JDELocal MSDE にあるデータプランナでのみ使用されます。
	所有者なし
OneWorld Local – PY9	ユーザー定義コードを含むソース: Control Tables – CRP
	所有者なし
OneWorld Local – DV9	ユーザー定義コードを含むソース: Control Tables – Test

OneWorld Local – JD9	所有者なし ユーザー定義コードを含むソース: Business Data – JDE
OneWorld Local – PD9	所有者なし ユーザー定義コードを含むソース: Control Tables – Production
System – B9	所有者 SY99 ライブラリ SY99 すべての環境で使用する唯一のシステム・テーブルのセット。
Versions – PY9	所有者 PY9 ライブラリ COPY9 PY9 環境のバージョン・リストおよび処理オプション・テキスト・テーブルを含むソース
Versions – DV9	所有者 DV9 ライブラリ CODV9 DV9 環境のバージョン・リストおよび処理オプション・テキスト・テーブルを含むソース
Versions – JD9	所有者 JD9 ライブラリ JD9 JD9 環境のバージョン・リストおよび処理オプション・テキスト・テーブルを含むソース
Versions – PD9	所有者 PD9 ライブラリ COPD9 PD9 環境のバージョン・リストおよび処理オプション・テキスト・テーブルを含むソース

システム構成一覧表

次の表は、必要となるデータ、環境、セントラル・オブジェクト(パス・コード)、パッケージを示します。

要件	説明
データ	<p>次のようなデータがあります。</p> <ul style="list-style-type: none">カンファレンス・ルーム・パイロット(CRP)ビジネス・データカンファレンス・ルーム・パイロット(CRP)コントロール・テーブルデータ辞書(すべての環境で共有)プリスティン・データ本稼動用ビジネス・データ本稼動用コントロール・テーブル(PD9 環境で使用)オブジェクト・ライブラリアン(すべての環境で共有)システム(すべての環境で共有するテクニカル・データ)テスト用ビジネス・データテスト用コントロール・テーブル(TS9 および DV9 環境で使用)バージョン
セントラル・オブジェクト (パス・コード)	<p>次のようリリースごとのセントラル・オブジェクトまたはパス・コードが必要です。</p> <ul style="list-style-type: none">PY9 central objectsDV9 central objectsJD9 central objectsPD9 central objects
環境	<p>インストール時に次の環境を定義します。</p> <ul style="list-style-type: none">PY9 CRP にマップされたデータが含まれている CRP の オブジェクト(本番環境のユーザーにリリースしていないパッケージのテストが可能)。パス・コードは、CRP のデータにマップされたテーブル・オブジェクトがある PY9 です。DV9 テスト・データが含まれている開発オブジェクト。パス・コードは、テスト・データにマップされたテーブル・オブジェクトがある DV9 です。JD9 未修正のデモ・データが含まれているプリスティン・オブジェクト。パス・コードは、プリスティン・データにマップされたオブジェクトが含まれている JD9 です。PD9 本稼動用データが含まれている本稼動用オブジェクト。パス・コードは、本稼動用データにマップされたテーブル・オブジェクトがある PD9 です。TS9 テスト・データが含まれている CRP オブジェクト。パス・コードは、テスト・データにマップされたテーブル・オブジェクトがある PY9 です。

パッケージ

開発プロジェクトを予定していない場合、必要なパス・コードは PY9 と PD9 の 2 つだけです。広範囲にわたるソフトウェア修正を行う場合は、開発用のパス・コードを作成する必要があります。

パス・コードの数はできるだけ少なくしてください。追加する各パス・コードについて、バージョン管理などのメンテナンスに時間がかかるため、必要がある場合以外は追加しないでください。広範囲に及ぶソフトウェア修正を行う場合でも、次の 4 つのパス・コード(セントラル・オブジェクト)のみ必要です。

- PY9
このパス・コードには、本稼働環境に移行する前に CRP でテストするオブジェクトが含まれています。本稼働環境にすばやく移行するために短期間で行った修正を配布したり、小規模な変更を行うために使用します。また、本稼働環境パス・コードに移行した後のリスクを避けるために、開発パス・コードで行った変更をテストするためにも使用します。
- DV9
このパス・コードは標準的な開発のために使用します。テストが正常に完了した後、オブジェクト移行機能を使って PY9 パス・コードにオブジェクトを移行し、パッケージ・ビルドおよびワークステーションのインストールによってユーザーに配布します。
- JD9
製品に同梱されているプリスティン・オブジェクトです。このパス・コードにはペーパー・フィックス以外の修正を加えないでください。このパス・コードは、標準ソフトウェアと他のパス・コードで行ったカスタマイズを比較するために使用します。当初の環境が必要な場合に備えて、このパス・コードのコピーを保管しておいてください。
- PD9
本稼働用のパス・コードです。このパスからジャストインタイム・インストールが実行されます。本稼働用サーバーのオブジェクトもこのパスから配布されます。ソフトウェアの変更を PY9 でテストした後、PD9 に転送してエンタープライズ・サーバーとワークステーションに配布します。

すべてのパス・コードで同じオブジェクト・ライブラリアン・テーブルとシステム・データソースを共有し、通常はデータ辞書も共有します。パス・コードによって異なるのは、セントラル・オブジェクト/スペック(F987*)、バージョン・リスト(F983051)、処理オプション・テキスト(F98306)です。

各パッケージにはバージョン A と B があり、パッケージをビルドする際は A と B を交互に使用してください。

フル・パッケージと部分パッケージの両方を使用している場合、各パス・コードにつき 4 つのパッケージがあることとなります。この設定では、本稼働用にフル・パッケージ A、B およびと部分パッケージ A と B が提供されます。次に例を示します。

- PD9FA(標準本稼働用フル・パッケージ A)
- PD9FB(標準本稼働用フル・パッケージ B)
- PD9PA(標準本稼働用部分パッケージ A)
- PD9PB(標準本稼働用部分パッケージ B)

