

PeopleSoft®

EnterpriseOne 8.9
パッケージ管理
PeopleBook

2003 年 9 月

PeopleSoft EnterpriseOne 8.9
パッケージ管理 PeopleBook
SKU AC89JPK0309

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

パッケージ管理の概要	1
インプリメンテーション・チームの役割	2
CNC コンサルタントと CNC 管理者	3
カスタム・ソリューション・コンサルタントとアプリケーション開発者	3
アプリケーション・コンサルタントおよびアプリケーション・プロジェクト・リーダー	3
ハードウェア、ネットワーク、サードパーティ・ソフトウェアのコンサルタントおよび管理者	3
『パッケージ管理』ガイド	3
パッケージ	4
デプロイメント・プログラム	4
ワークステーション・インストールの理解	4
パッケージ・デプロイメント指示	5
サーバー・パッケージの理解	5
複数階層デプロイメントの理解	5
パッケージ管理プランニングおよびセットアップ	6
推奨パス・コードの処理	6
ジャストインタイム・インストールの無効化	7
推奨のパッケージ名	7
本稼働用環境の整合性保証	8
開発プロセスの理解	9
標準開発プロセスの処理	9
短期的なカスタマイズ	10
デプロイメント方法	11
PTF/Cum アップデート・パッケージおよび非 PTF/Cum アップデート・パッケージ	11
部分パッケージおよび JITI を使用するサイトへの推奨	11
JITI によるフル・パッケージを使用するサイトに対する推奨事項	11
開発者に対する推奨事項	11
複数階層デプロイメントおよびジャストインタイム・インストール	12
デプロイメント方法の比較	12
各種修正の配布	13
データの配布	15
オフライン処理ユーザーの専用データベースの配布	17
修正規則	18
アップグレードで保持される修正と置換される修正	19

修正用一般規則	19
対話型アプリケーション	19
レポート	21
アプリケーション・テキストの変更	22
テーブル・スペック	22
コントロール・テーブル	23
ビジネス・ビュー	23
イベント・ルール	24
データ構造体	25
ビジネス関数に関する規則	25
バージョン	26
オブジェクト管理	27
オブジェクト移動の理解	27
データ辞書項目のレプリケート	29
ビルド・プロセスで作成されたファイルの理解	29
ワークステーション・ビルド	29
サーバー・パッケージ・ビルド	30
UNIX サーバー・ビルド	30
Windows NT サーバー・ビルド	32
AS/400 サーバー・ビルド	33
複製オブジェクトとセントラル・オブジェクトの関連付け	35
オブジェクトのバックアップおよび復元	36
データ・ソース間でのレコードのコピー	38
パッケージ・ビルド	41
パッケージおよびデプロイメントの理解	41
フル・ワークステーション構成および部分ワークステーション構成の理解	42
ジャストインタイム・インストレーションの理解	43
パッケージ・タイプの理解	44
機能の理解	46
パッケージの作成およびデプロイメントの概要	46
サーバー・パッケージの理解	49
Sun オペレーティング・システム・ユーザーの考慮事項	54
パッケージの組立て	55
Package Assembly Director の理解	55
パッケージのアクティブ化	83
パッケージのビルド	84
パッケージの検証	85
検証ログの表示	86
既存パッケージの改訂	87
パッケージのコピー	88
パッケージの削除	89
パッケージ・ビルドの定義	89

ビルド・プロセスの理解	91
Package Build Definition Director の理解	91
パッケージ・ビルドの定義	93
パッケージの検証	108
パッケージ・ビルド・オプションの改訂	109
親パッケージの圧縮(アップデート・パッケージのビルド後)	109
ビルド済みパッケージのコピー	109
パッケージ・ビルドの履歴表示およびビルドの再投入	112
パッケージへの機能の組込み	112
機能のビルドとデプロイメント・プロセスの概要	113
Feature Based Deployment Director の理解	114
機能の作成	115
ファイル・セット・コンポーネントの定義	118
レジストリ・コンポーネントの定義	120
ショートカット・コンポーネントの定義	121
その他のパッケージ・ビルド・プロセスの定義	124
その他のインストール・プロセスの定義	126
初期化ファイル・コンポーネントの定義	128
ODBC データ・ソース・コンポーネントの定義	130
機能コンポーネントの検討	132
機能のコピー	132
パッケージ組立て時の機能の追加	134
パッケージ・ビルド定義時の機能の構成	138
機能を含んだパッケージのインストール	143
Package.INF ファイル内の機能項目の理解	143
パッケージ・ビルドの履歴およびログの表示	144
パッケージ・ビルドの履歴表示	144
ログの表示	146
パッケージ・ビルドの再投入	153
ビルド状況の変更	154
パッケージ INF ファイルの理解	156
INF ファイルの例	156
機能 INF ファイルの理解	161
機能 INF ファイルの例	161
配布	166
パッケージ・デプロイメントの理解	166
ERP 9.0 がインストールされていないワークステーションへの配布	166
ERP 9.0 がインストールされているワークステーションへの配布	166
サーバーへの配布	167
階層デプロイメント・ロケーションへの配布	167
CD からワークステーションへの配布	167
マシン、ロケーション、およびデプロイメント・グループの定義	167

マシンの定義.....	168
ロケーションの定義.....	173
パッケージ・デプロイメント・グループの定義.....	175
デプロイメント・グループの改訂.....	178
Package Deployment Director の処理.....	178
Package Deployment Director の理解.....	179
Package Deployment Director の使用.....	180
パッケージのアクティブ化.....	195
スケジュール済みパッケージのインストール.....	195
サーバー・パッケージの配布.....	197
プッシュ・インストールの使用.....	199
プッシュ・インストールの理解.....	200
プッシュ・インストール用エンタープライズ・サーバーの準備.....	201
プッシュ・インストール用ワークステーションの準備.....	201
プッシュ・インストール用パッケージのスケジュール.....	209
プッシュ・インストール・バッチ・アプリケーションのスケジュール.....	210
Push Package Installation Results レポートの実行.....	215
CD からの ERP 9.0 ワークステーションのインストール.....	216
CD ライター・ロケーションの定義.....	216
CD ライター・ロケーションへのパッケージの配布.....	219
ERP 9.0 インストール CD の作成.....	221
部分パッケージ・テンプレートの配布.....	221
部分パッケージ・テンプレートの組立て.....	222
部分パッケージ・テンプレートのビルド.....	224
部分パッケージ・テンプレートの配布.....	225
部分パッケージ・テンプレートのテスト.....	226
複数階層デプロイメントの用語.....	228
パッケージ・ビルド時間の比較.....	228

複数階層デプロイメント **229**

複数階層デプロイメントの理解.....	229
複数階層デプロイメントの機能.....	230
複数階層のインプリメンテーション.....	232
デプロイメント・サーバーの定義.....	233
新規デプロイメント・サーバーの定義.....	233
既存デプロイメント・サーバー定義の改訂.....	237
デプロイメント・ロケーション用パッケージのスケジュール設定.....	239
デプロイメント・ロケーションへのソフトウェアの配布.....	239
Package Deployment を使用したソフトウェアの配布.....	240
複数階層デプロイメントでのサーバー・パッケージの配布.....	241
〈Multitier Deployment〉バッチ・プロセスを使用したソフトウェアの配布.....	241
デプロイメント・ロケーションへの〈Workstation Installation〉プログラムのコピー.....	244
デプロイメント・ロケーションからのワークステーション・パッケージのインストール.....	244

サーバー・パッケージの複数階層デプロイメント	245
スマート・デプロイメント	246
package.inf ファイルの自動更新	246
複数階層デプロイメント・ケーススタディの理解	248

パッケージ管理の概要

『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』の一連のマニュアルは、CNC コンサルタント、システム管理者、ネットワーク/サーバー管理者を対象に作成されています。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を完了と、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(CNC)コンサルタントを対象としており、次の項目について説明しています。
 - ミドルウェアの処理
 - データ・ソースの理解と確認
 - パス・コードおよび環境の作成
 - OCM(オブジェクト構成マネージャ)の処理
 - 各処理モードの理解
 - 標準カスタマー構成
- 『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おり、次の項目について説明しています。
 - データ・レプリケーションのセットアップ
 - プリンタのセットアップ
 - サーバーの処理
 - ユーザー・プロファイルおよびセキュリティのセットアップ
 - データ辞書の管理と処理
 - トランザクション処理の理解
 - メディア・オブジェクトおよびイメージングの処理
 - ユニバーサル・テーブル・ブラウザの使用
 - 命名規則の理解
 - jde.ini ファイルの処理
- 『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。『パッケージ管理』ガイドでは次の項目について説明しています。
 - パッケージ管理プランニングおよびセットアップ
 - 修正規則
 - オブジェクト管理
 - パッケージのビルド
 - デプロイメント
 - 複数階層デプロイメント

- 『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おり、次の項目について説明しています。
 - スナップショット(マルチクライアント・インストーラ)
 - サーバー管理
 - ワークステーションとサーバーのトラブルシューティング

これらのガイドでは、関連するタスクに従って内容を構成されていますが、必要なタスクに関する情報が複数のガイドに記載されていることもあります。たとえば、パス・コード、環境およびデータ・ソースのセットアップに関する情報は、『CNC インプリメンテーション』ガイドに記載されていますが、パッケージのビルドおよび配布に関する情報は、『パッケージ管理』ガイドに記載されています。

これらのガイドは、次のタスクを除くすべての CNC 関連タスクで中心的な役割を果たします。

- 初期インストール。詳細については、『インストール・ガイド』を参照してください。
- アップグレードと PTF/Cum アップデート。詳細については、『アップグレード・ガイド』を参照してください。
- ネットワーク・インフラストラクチャおよびサードパーティ・ソフトウェアのセットアップおよび保守管理。サードパーティのソフトウェアおよびハードウェアに関する情報は、それぞれの製造販売元提供のドキュメンテーションを参照してください。

コンフィギュレーション・プランニングとセットアップを行う際は、インストール・プロセスに関する深い知識/技術は必要ありませんが、これらのガイドを利用するには、インストールで実行される内容について理解しておく必要があります。

インプリメンテーション・チームの役割

通常、ソフトウェアのインプリメンテーションでは、カスタマーとコンサルタントはいずれも次のような役割を果たします。

- CNC コンサルタントおよび CNC 管理者
- カスタム・ソリューション・コンサルタントおよびアプリケーション開発者
- アプリケーション・コンサルタントおよびアプリケーション・プロジェクト・リーダー
- ハードウェア、ネットワーク、サードパーティ・ソフトウェアのコンサルタントおよび管理者

通常、コンサルタントとカスタマーが共同でインプリメンテーション作業を行います。コンサルタントは、次の役割を実行します。

- CNC コンサルタント
- カスタム・ソリューション・コンサルタント
- アプリケーション・コンサルタント
- ハードウェア、ネットワーク、およびサードパーティ・ソフトウェアのコンサルタント

カスタマーは、コンサルタントの役割に対応する、次の役割を実行します。

- CNC 管理者
- アプリケーション開発者
- アプリケーション・プロジェクト・リーダー

- ハードウェア、ネットワーク、およびサードパーティ・ソフトウェアの管理者

インプリメンテーションが終了すると、一般にコンサルタントの役割は小さくなります。そのため、カスタマーは、各自が役割を果たすのに十分なトレーニングを受ける必要があります。

CNC コンサルタントと CNC 管理者

CNC コンサルタントと CNC 管理者は、ソフトウェアをインストールし、環境、ユーザー、セキュリティ、分散処理、データ・レプリケーションを設定します。また、バージョン管理のセットアップとさまざまな CNC 構成のテストにも責任を負います。さらに、全社的なソフトウェアのデプロイメントを統括します。

カスタム・ソリューション・コンサルタントとアプリケーション開発者

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によってビジネス上の問題を解決します。その主な役割は、アップグレードを念頭において修正内容を設計し、カスタマイズされたソフトウェアの開発、テスト、および導入作業を行うことです。CNC 管理者がソフトウェアをビルドして配布するためのバージョン管理業務を実施する間に、カスタム・ソリューション・コンサルタントは、ビジネスに合ったアプリケーション開発サイクルを記載した社内手順の開発をサポートする必要があります。

アプリケーション・コンサルタントおよびアプリケーション・プロジェクト・リーダー

ソフトウェアインストール、構成、および配布が終了した後も、アプリケーション・コンサルタントは製品のエキスパートとしての役割を担い、問題が発生した場合にトラブルシューティングを行うことがあります。アプリケーション・コンサルタントは CNC コンフィギュレーションのインプリメントは行いませんが、ソフトウェアで分散処理、データ・レプリケーション、環境などがどのように処理されるかを理解しておく必要があります。これは、アプリケーションの問題が CNC のセットアップに影響するためです。

ハードウェア、ネットワーク、サードパーティ・ソフトウェアのコンサルタントおよび管理者

ソフトウェアのインプリメンテーション作業では、PeopleSoft から提供されないタスクも多く発生します。このようなサービスは、サードパーティのコンサルタントが提供します。サードパーティのコンサルタントは、CNC コンサルタント、ネットワーク設計者、カスタム修正コンサルタントなどの役割を果たすこともあります。

『パッケージ管理』ガイド

このガイドでは、ERP 9.0 ツールによるカスタム修正の開発/配布プロセスをセットアップし、保守する方法について説明します。

また、ERP 9.0 開発ツールによって行われたカスタム修正を配布できる環境をセットアップする方法について説明します。また、修正規則の適用、オブジェクトの転送、開発オブジェクトのチェックアウト、およびデータ辞書の処理に関する情報も提供します。

このような環境が既に設定されており、これらの概念を十分理解している場合は、パッケージの作成およびスケジュール方法に関する情報や、ワークステーションおよびサーバーへのパッケージの配布方法について説明している章より前のページは省略してもかまいません。

パッケージ

パッケージの目的は、ソフトウェア修正をワークステーションに配布できるようにグループ化することです。パッケージは、ワークステーションに展開するコンポーネントがサーバーのどこにあるかを説明します。パッケージには、ERP 9.0 の実行に必要なすべて（新規ワークフローへの初期インストール向け）、または既存のアプリケーションに対するアップデートまたは変更のみを含めることができます。

パッケージには次の 3 つのタイプがあります。

- フル・パッケージ
- 部分パッケージ
- アップデート

フル・パッケージは、すべての ERP 9.0 アプリケーションを含んでおり、前の標準パッケージと同じものです。ERP 9.0 アプリケーションのフル・スイートを必要とするユーザーは、このタイプのパッケージを受け取る必要があります。

部分パッケージは、ERP 9.0 の最小構成です。このタイプのパッケージに必要なディスク・スペースは、フル・パッケージに比べて非常に少なくなります。このタイプのパッケージをインストールするユーザーは、初期インストールで使用しない ERP 9.0 を多数受け取るのではなく、必要なアプリケーションのみを実行時にロードすることができます。

アップデート・パッケージでは、変更されたオブジェクトで既存のフル・パッケージまたは部分パッケージを更新、追加、またはリフレッシュすることができます。アップデート・パッケージの配布先は、既に ERP 9.0 がインストールされているワークステーションのみです。アップデート・パッケージ内のオブジェクトは、ワークステーション上の同じオブジェクトを置換します。ワークステーション上の他のオブジェクトはいずれも変更されません。このタイプのパッケージの利点は、ソフトウェアの変更または機能拡張を迅速に配布できることです。

デプロイメント・プログラム

ERP 9.0 では、複数のデプロイメント方法があります。それぞれの方法に特有の目的および利点があります。デプロイメント方法は主に、配布するパッケージのタイプに応じて選択します。ERP 9.0 を配布する際には、次のタイプのプログラムを使用することができます。

- ワークステーション・インストール
- パッケージ・デプロイメント指示
- 複数階層デプロイメント

ワークステーション・インストールの理解

〈Package Deployment (パッケージ・デプロイメント)〉プログラム(P9631)は、フル・パッケージおよび部分パッケージを配布するために使用します。このプログラムを使って、アップデート・パッケージを配布することはできません。

P9631 は、パッケージで指定されている項目を取り込みます。フル・パッケージまたは部分パッケージは、部品表や指示付きのキットのようなものです。必要なコンポーネントをすべて取り込むために、〈Package Deployment〉プログラムは、パッケージをローカル・コンピュータに配布します。このプログラムは常にユーザーがワークステーション上で起動します。

パッケージ・デプロイメント指示

〈Package Deployment (パッケージ・デプロイメント)〉プログラム(P9631)では、システム管理者が、ユーザーやグループがパッケージを使用できる日付と時刻を指定できます。また、パッケージを必須にするか任意にするかを指定できます。パッケージが必須である場合、パッケージを受け取るユーザーは、そのパッケージをインストールするまで ERP 9.0 にアクセスできません。

スケジュールされたパッケージを受け取るユーザーは、指定日に ERP 9.0 にログオンした直後にパッケージをインストールできます。パッケージをインストールする場合は、インストール・ルーチンが起動し、パッケージがロードされます。また、時間をずらして後でインストールすることもできれば、パッケージが必須でない場合は、インストール自体を中止することもできます。

その他の方法として、管理者はプッシュ・インストール機能を使用することにより、ユーザーとの対話型処理を必要としないで、指定時刻にデプロイメント・サーバーからワークステーションへ自動的にプッシュされるパッケージをスケジュールできます。

サーバー・パッケージの理解

サーバー・パッケージは、共通の SAR 番号またはバージョン制御トラッキング番号を付けてグループ化された 1 組の ERP 9.0 オブジェクトです。オブジェクトは、スペック・レコード、ソース・ファイルおよびヘッダー・ファイルから構成されます。コンパイル済みオブジェクトはエンタープライズ・サーバー上で作成されます。パッケージを定義すると、複数のサーバーおよび複数のパス・コードにパッケージをインストールできます。

アプリケーション開発はいずれもワークステーション上で行われ、オブジェクトは、デプロイメント・サーバーか、エンタープライズ・サーバーのいずれかのセントラル・ロケーションに保管されます。エンタープライズ・サーバー上には複数のビジネス・アプリケーションが存在します。開発者およびシステム管理者は、サーバー・パッケージと呼ばれるオブジェクトのグループをセントラル・オブジェクト・データ・ソースから、レプリケートしたサーバー・パス・コードにプッシュできます。

サーバー・パッケージはワークステーション・パッケージとは異なりますが、そのビルドと配布には、同じアプリケーション、つまり〈Package Assembly (パッケージ・アセンブリ)〉(P9601)、〈Package Build (パッケージ・ビルド)〉(P9621)、および〈Package Deployment (パッケージ・デプロイメント)〉(P9631)が使用されます。ワークステーション用のアップデート・パッケージを作成する場合は、対応するサーバー・パッケージを作成し、ワークステーションに配布するのと同じオブジェクトをエンタープライズ・サーバーに配布するようにする必要があります。

複数階層デプロイメントの理解

複数階層デプロイメントを使用すると、複数のデプロイメント・ロケーションおよび複数のデプロイメント・サーバーから ERP 9.0 ワークステーションにソフトウェアをインストールできます。ERP 9.0 ソフトウェア・インストールを毎日実行するワークステーションがサイトに 50 台以上ある場合、あるいは ERP 9.0 ソフトウェアを WAN に配布している場合には、複数階層デプロイメントを検討することをお勧めします。

パッケージ管理プランニングおよびセットアップ

修正を管理するには、変更されたオブジェクトをトラッキングするための実際的なバージョン制御プランを作成する必要があります。変更されたオブジェクトをトラッキングすることにより、多くのソフトウェアの問題を回避できます。

開発のプランニングとトラッキングを容易にし、バージョン制御を簡素化するには、パッケージのビルドおよび配布は必要最小限にする必要があります。開発の変更を数多く行う場合は、設定したスケジュールに従ってパッケージをビルドおよび配布する必要があります。このスケジュールにより、オブジェクトの完成予定日とパッケージのビルドおよび配布日が関係者全員に確実に知らされます。

バージョン制御のインプリメントに際しては、情報技術スタッフが必要になることがあります。PeopleSoft は数百名の開発者を抱えており、複雑な CNC(コンフィギュラブル・ネットワーク・コンピューティング)構成を採用しています。この複数の開発者に対してバージョン制御を管理するために、当社の製品バージョン制御グループは次のような構成になっています。

- 部門の調整を監督するマネージャ 1 名
- パッケージのビルドおよびオブジェクト転送を調整し、問題をトラブルシューティングする監督者 1 名
- サーバー・パッケージをビルドするサーバー・スペシャリスト 2 名
- ワークステーション・パッケージをビルドし、オブジェクト転送を実行し、品質保証のためにパッケージ・リリース前の自動テストを実行する技術者 4 名
- ワークステーション・パッケージおよびサーバー・パッケージをビルドし、ビルド・エラーを解消する夜間オペレータ 1 名

推奨パス・コードの処理

開発プロジェクトの計画を立てない場合は、PY9、PD9、JD9 の 3 つのパス・コード(セントラル・オブジェクトのセット)だけが必要です。広範囲にわたるソフトウェアの修正を行う場合は、開発パス・コード(DV9)も作成する必要があります。

パス・コードは、個別にバージョン制御管理を行う必要があります。そのため、実際に必要なパス・コードだけを作成する必要があります。広範囲にわたるソフトウェアの修正を行う場合でも、次の 4 つのパス・コードだけを使用してください。

- PY9** このパス・コードには、オブジェクトを本稼働に転送する前のカンファレンス・ルーム・パイロット時にテストされる実用オブジェクト・セットが含まれています。このパス・コードは、簡単な修正を配布したり、本稼働に転送するオブジェクトをわずかに変更するために使用します。また、オブジェクトを本稼働用パス・コードに転送する前に、開発パス・コードで行った修正をテストするためにも使用できます。
- PD9** 本稼働用パス・コード。ジャストインタイム・インストールおよび本稼働用サーバー・オブジェクトは、このパス・コードから配布されます。PY9 でソフト・ウェア変更をテストした後、PD9 に転送し、エンタープライズ・サーバーおよびワークステーションにその変更を配布します。
- JD9** ソフトウェアに付属しているオリジナルのオブジェクト・セット。このパス・コードには、書類で提示した以外の変更を加えないでください。このパス・コードは、標準ソフトウェアを、他のパス・コードにインプリメントしたカスタム・ソフトウェアと比較するために使用します。このパス・コードのコピーを必ず作成し、変更内容をすべて取り消す必要がある場合に、変更されていない ERP 9.0 のコピーを使用できるようにしてください。

DV9 標準開発で使用するパス・コード。開発したオブジェクトのテストが無事終了したら、〈Object Transfer (オブジェクトの転送)〉プログラムを使って PY9 パス・コードにオブジェクトを転送し、パッケージ・ビルド・デプロイメント・プロセスを介してユーザーに配布します。

パス・コードはいずれも同じオブジェクト・ライブラリアン・テーブルと同じシステム・データソースを共有し、通常は、同じデータ辞書も共有します。パス・コードごとに異なるテーブルは、セントラル・オブジェクトおよびスペック・テーブル (F987 で始まるテーブル)、バージョン・リスト・テーブル (F983051)、処理オプション・テキスト・テーブル (F98306)、およびユーザー一時変更テーブル (F98950) だけです。

ジャストインタイム・インストールの無効化

JITI は、次のいずれかの方法で無効化することができます。

- 〈Work with Environments (環境の処理)〉プログラム (P0094) を使用して、該当する環境での JITI を無効にする。JITI を無効にすると、その環境にサインオンしたユーザーは、スペックを所持しているアプリケーションにしかアクセスできません。本稼働用セントラル・オブジェクトを新しい変更によって更新する場合は、この方法を累積インストール・プロセス中に使用することをお勧めします。
- アプリケーション・セキュリティを使用して、特定のアプリケーション、エンドユーザー、またはグループ・プロファイルの JITI を無効にする。ユーザーが、自己のワークステーションにスペックが存在しないアプリケーションにアクセスすると、システムは、最初に P0094 を調べ、JITI が環境全体で無効にされていないかどうかを判別します。JITI が有効な場合は、アプリケーション・セキュリティで、JITI の使用がユーザーに許可されているかどうかを判別します。アプリケーション・セキュリティには、[not allowed to install (インストール不可)] フィールドがあります。

JITI は、環境に対して無効にできますが、データ辞書項目をグローバル・テーブルにコピーする際には、依然として JITI が使用されます。データ辞書の JITI は、データ辞書の構造により、無効にすることができません。

参照

- アプリケーション・セキュリティのセットアップについては、『システム・アドミニストレーション』ガイドの「セキュリティ」

推奨のパッケージ名

パッケージごとに A と B の 2 つのバージョンを維持管理し、パッケージを作成する際にそれらを交互に使用できるようにすることをお勧めします。この方法には、1 つのパッケージの最新バージョンを作成している時でも、ユーザーはそのパッケージを利用できるという利点があります。たとえば、PRODA を作成している間は、ユーザーが PRODB を使用できるようにしておきます。その後 PRODA をリリースしたら、PRODB に次のパッケージを作成します。

フル・パッケージおよび部分パッケージを両方とも使用する場合は、1 つのパス・コードにつき 4 つのパッケージが用意されます。この設定では、次の表に示されているように、本稼働用フル・パッケージ (A および B) が 2 つ、本稼働用部分パッケージ (A および B) が 2 つ用意されます。

PD9FA 標準本稼働用フル A

PD9FB 標準本稼働用フル B

PD9PA 標準本稼働用部分 A

PD9PB 標準本稼働用部分 B

アップデート・パッケージには次のような名前が付けられます。

PD9UA 本稼働用更新パッケージ 1

PD9UB 本稼働用更新パッケージ 2

PD9UA 本稼働用更新パッケージ 3

PD9UB 本稼働用更新パッケージ 4

本稼働用環境の整合性保証

オブジェクトを PD9 パス・コードに転送した直後から、エンドユーザーはそのオブジェクトにアクセスできますそのため、本稼働用パス・コードに転送する前に、修正されたオブジェクトをテストする必要があります。

オブジェクトを本稼働用パス・コードに転送すると、次のような場合に、オブジェクトはエンドユーザーに即座に配布されます。

- JITI(ジャストインタイム・インストール)がユーザーに対してセットアップされている場合は、ユーザーがオブジェクトへのアクセスをはじめて試みたときに、ユーザーのワークステーションにオブジェクトが自動的に配布される。

注:

JITI は、部分パッケージを使用しているユーザーだけでなく、スペックなしのアプリケーションを含むアップデート・パッケージを受け取ったユーザーにも影響します。JITI の詳細については、「ジャストインタイム・インストールの理解」を参照してください。

- ビジネス関数ビルドを含むアップデート・パッケージや部分パッケージをビルドした場合、システムはビジネス関数をビルドし、そのビジネス関数を、パス・コードのチェックイン・ロケーションにある他のすべてのビジネス関数とグローバルにリンクします。

本稼働用パス・コードに転送した修正が、エンドユーザーに即時リリースされないようにするために、スペックなしのアプリケーションを含む部分パッケージやアップデート・パッケージの使用を避けてください。また、パッケージのビルド時に、ビジネス関数のグローバル・ビルドには自動的に新しい関数が含まれるため、配布の準備が整うまでビジネス関数を本稼働用パス・コードに転送しないでください。変更を本稼働用パス・コードに転送しても、フル・パッケージやアップデート・パッケージを作成するまで、これらの変更は使用できません。

開発プロセスの理解

この項目では、推奨する開発プロセスについて説明します。ここには、長期的な機能拡張および短期的なカスタマイズに関する情報が記載されています。

標準開発プロセスの処理

次のリストに、標準開発サイクルの実行方法の概要を示します。

DV9 パス・コードで、次のタスクを実行します。

- 修正の実行
- 修正のテスト
- PY9 へのオブジェクトの転送

PY9 パス・コードで、次のタスクを実行します。

- パッケージの作成
- 修正のテスト
- サーバー・オブジェクトをエンタープライズ・サーバーの CRP パス・コードに配布し、テストする。
- パッケージのスケジュール設定
- PD9 へのオブジェクトの転送

PD9 パス・コードで、次のタスクを実行します。

- パッケージの作成
- パッケージのスケジュール設定
- サーバー・オブジェクトをエンタープライズ・サーバーの PD9 パス・コードに配布し、テストする。

▶ 標準開発プロセスを処理するには

1. オブジェクトを DV9 パス・コードからチェックアウトし、修正し、テストしてからチェックインします。
2. SAR システム（または、採番システム）を使用して、変更をトラッキングします。
オブジェクトをチェックインするときは、常に SAR 番号を使用します。
3. オブジェクトをロジック・サーバーに配置する必要がある場合は、ロジック・サーバーの DV9 パス・コードに転送します。
4. オブジェクトをサーバー上のオブジェクトと比較してテストします。
5. 〈Object Transfer〉で SAR 番号を使用して、オブジェクトを PY9 パス・コードに転送します。

チェックアウト・ログを使用して、転送を確認します（任意）。これらのオブジェクトは本稼働用ではありませんが、PY9 パス・コードでテスト・パッケージをビルドするために使用できます。

6. フル・パッケージ、部分パッケージ、またはアップデート・パッケージをビルドします。

7. 新規にビルドされ、リリースされていないパッケージを PY9 パス・コードでテストします。

パッケージのテストは、サーバー・プロセスではなく、ワークステーション・プロセスと比較することで行います。このパッケージは、おそらく PY9U1 (CRP パス・コード用のアップデート・パッケージ番号 1) という名前になりますが、まだユーザーにはリリースされていないので、テスト・パッケージにすぎません。

8. アップデート・パッケージをテスト・マシンに配布するようにスケジュールし、CRP オブジェクトと CRP データが存在する環境でテストします。
9. サーバー・オブジェクトをエンタープライズ・サーバーの PY9 パス・コードに配布し、テストします。

必要に応じて、ワークステーション・パッケージをビルドし、スケジュールするのと同時に、サーバー・パッケージをビルドし、配布をスケジュールすることができます。これらのパッケージを同時にビルドすると時間を節約できますが、サーバーの負荷が増大します。

10. 新しいパッケージを CRP ユーザーに配布するようにスケジュールします。
11. 〈Object Transfer〉で SAR 番号を使用して、オブジェクトを PD9 パス・コードに転送します。

チェックアウト・ログを使用して、転送を確認します (任意)。これで、オブジェクトは本稼働用環境に配置され、PD9 パス・コードでパッケージをビルドするために使用できます。

注:

ジャストインタイム・インストレーションが有効な場合、ユーザーは直ちにオブジェクトにアクセスできます。

12. クライアント・ワークステーション用のフル・パッケージ、部分パッケージ、またはアップデート・パッケージをビルドします。
 13. サーバー・パッケージ・ビルドを実行します。
- サーバー・パッケージをすぐに転送することもできますが、ワークステーションでのテストを完了してから転送することもできます。
14. 新しいパッケージをエンドユーザーのワークステーションに配布するようにスケジュールします。
 15. サーバー・オブジェクトをエンタープライズ・サーバーの PD9 パス・コードに配布し、テストします。

必要に応じて、ワークステーション・パッケージをビルドし、スケジュールするのと同時に、サーバー・パッケージをビルドし、配布をスケジュールすることができます。

短期的なカスタマイズ

DV9 パス・コードで主要な機能拡張が行われているアプリケーションに対して、簡単な変更を加えないければならないことがあります。DV9 パス・コードでオブジェクトが修正されているときに、簡単な変更を行ってすぐに配布すると、予期せぬ結果が生じる場合があります。そのため、PY9 と DV9 の両方のパス・コードで変更を行い、その変更を PY9 パス・コードを介して配布する必要があります。この方法により、DV9 パス・コードでの主要な機能拡張に悪影響を及ぼさずに、ユーザーに変更を速やかに配布できます。

デプロイメント方法

ソフトウェアの変更を行った後、その変更をエンタープライズのワークステーションに配布する方法は、通常作成するパッケージのタイプやユーザーのニーズといった要因によって異なります。

PTF/Cum アップデート・パッケージおよび非 PTF/Cum アップデート・パッケージ

パッケージの配布で PTF/Cum アップデートを使用する場合は、追加、再作成、およびユーザーへの再リリースを行うためのパッケージを 1 つ用意します。配布する修正があるたびに新しいパッケージは作成しません。累積パッケージを使用するには、次の手順を実行します。

1. 新規修正を追加するためにパッケージ状況を変更します。
2. 変更されたオブジェクト、または新規オブジェクトをパッケージに追加します。
3. パッケージを再配布します。

非 PTF/Cum アップデートを採用する場合は、オブジェクトを追加または変更するたびに異なるパッケージを作成し、配布します。たとえば、毎週 1 つの修正を 10 週間にわたって配布する場合は、各週のソフトウェアの変更だけを含む 10 種類のパッケージが必要になります。

部分パッケージおよび JITI を使用するサイトへの推奨

部分パッケージと JITI を使用するサイトでは、非 PTF/Cum アップデート・パッケージを採用することをお勧めします。新規のアップデート・パッケージを週ごとにビルドし、ビジネス関数の変更など、JITI を通じて配布できない修正をそれに含める場合は、その週の部分パッケージをリビルドし、部分パッケージに最新の変更を含める必要もあります。新しいワークステーションや完全にリフレッシュされたワークステーションにパッケージを配布するときは、アップデート・パッケージではなく、この部分パッケージをインストールできます。

JITI によるフル・パッケージを使用するサイトに対する推奨事項

JITI によるフル・パッケージを使用するサイトでは、累積アップデート・パッケージを採用することをお勧めします。変更を配布する週ごとに、そのオブジェクトを既存のアップデート・パッケージに追加してから、パッケージをリビルドし、スケジュールを設定します。

この方法の利点は、フル・パッケージを毎週リビルドする必要がないことです。この方法を使用して、新しいワークステーションや完全にリフレッシュされたワークステーションにパッケージを配布する場合は、フル・パッケージをインストールしてから、1 つの PTF/Cum アップデート・パッケージをインストールする必要があります。

この方法の欠点は、アップデート・パッケージが大きくなり、配布にかかる時間が長くなる可能性があることです。新しいワークステーションのフル・パッケージをリビルドし、既存のユーザーが新規のアップデート・パッケージをインストールできるように確定する必要があります。

開発者に対する推奨事項

開発者はフル・パッケージを使用する必要があります。アップデート・パッケージは開発者にも配布できますが、部分パッケージが含まれるワークステーションは開発オブジェクトを受け取らないので、開発者は部分パッケージを使用しません。

フル・パッケージまたはアップデート・パッケージをインストールする場合、コンパイラを搭載しているワークステーションは、開発オブジェクトを自動的に受け取ります。開発者は、ソース・コード、ヘッダーおよびライブラリを受け取ります。インストール・プロセスでは、オブジェクトをビジネス関数ライブラリと DLL (ダイナミック・リンク・ライブラリ) 内に作成することにより、開発者が行ったビジネス関数の変更を保存します。

複数階層デプロイメントおよびジャストインタイム・インストール

複数階層デプロイメントを使用して、パッケージをインストールしている遠隔ロケーションでは JITI を使用できます。ただし、JITI を使用すると、処理時間が非常に長くなる場合があります。そのような場合は、最初にフル・パッケージをインストールし、アップデート・パッケージを使用して、ソフトウェアの変更を配布できます。

デプロイメント方法の比較

各デプロイメント方法には長所および制約があります。貴社のニーズに合ったデプロイメント方法を決定するのに役立つと思われる、各方法の要点を次に説明します。

- 新規マシンにロードする新規ユーザーは、〈Install Manager (インストール・マネージャ)〉を使用して、フル・パッケージまたは部分パッケージ、および前回のパッケージ・ビルド以来ユーザーにロードを指示しているアップデート・パッケージをロードする必要があります。したがって、特定のパッケージをインストールした後、どのアップデート・パッケージを適用するかをトラッキングし続ける手作業トラッキング・システムが必要です。
- 〈Push Installation (プッシュ・インストール)〉機能を使用していない場合、すべてのアップデート・パッケージはワークステーションに対してスケジュールされた〈Package Deployment〉プログラム (P9631) を使用する必要がある。
- ERP 9.0 がすでにマシンにロードされている場合は、フル・パッケージと部分パッケージでも〈Package Deployment〉を使用できます (〈Push Installation〉機能を使用すると、ERP 9.0 がインストールされていないマシンに配布できます)。
- ワークステーションのディスク領域にさほど余裕がない場合は、ERP 9.0 の最小構成を提供し、適用可能な ERP 9.0 アプリケーションだけをロードできる部分パッケージの配布を検討すること。この状況では、〈Install Manager〉または〈Package Deployment〉を使用して部分パッケージを配布できます。
- 〈Silent Installation (インストールの自動モード)〉を使用し、コマンド行の引数を介して〈Workstation Installation〉要求を投入する。最初のインストールでは、この自動モードを使用しないでください。
- 〈Multitier Deployment (複数階層デプロイメント)〉プロセスを使用し、複数のデプロイメント・ロケーションからインストールする。ERP 9.0 ソフトウェアのインストールを毎日実行するワークステーションが 50 台以上ある場合は、この方法を検討してください。
- サーバー・パッケージ内のオブジェクトをセントラル・オブジェクト・データ・ソースからエンタープライズ・サーバーにプッシュする必要がある場合、〈Package Deployment〉プログラム (P9631) を使用します。

各種修正の配布

配布する変更に適したパッケージを選ぶために、各タイプのパッケージを通じて配布できるオブジェクトと変更のタイプを認識しておく必要があります。

部分パッケージには次の項目が含まれています。

- 連結ビジネス関数のすべての DDL
- ソフトウェアへのログインとアクセスに必要なアプリケーション専用のスペック
- すべてのアイコン
- ヘルプ、データ、および ERP 9.0 ファンデーション

次の表は、フル・パッケージ、スペック付きアップデート・パッケージ、およびスペックなしのアップデート・パッケージにおいて、どのタイプの変更がインストールされるかを示しています。

修正	フル・パッケージ	スペック付きアップ デート・パッケージ	スペックなしのアップ デート・パッケージ
アプリケーション			
埋込みイベント・ルール	X	X	X
用語一時変更 (FDA テキスト)	X	X	X
データ構造体	X	X	X
処理オプション (レポート)	X	X	X
ビジネス関数に関する規則			
C 言語のソース/インクルード/オブジェクト (コンパイラがある場合)	X	X	
連結ビジネス関数 DLL	X	X	
データ構造体	X	X	
テーブル・イベント・ルール	X	X	
イベント・ルール・ビジネス関数	X	X	
バッチ・アプリケーション			
レポート	X	X	X
レポート内の埋込みイベント・ルール	X	X	X
レポート・データ構造体	X	X	X
レポート用語一時変更	X	X	X

修正	フル・パッケージ	スペック付きアップ デート・パッケージ	スペックなしのアップ デート・パッケージ
レポート処理オプション	X	X	X
バージョンおよび処理オプション値(処理オプションに依存)	X	X	X
バージョン内の埋込みイベント・ルール	X	X	X
処理オプション・テンプレート	X	X	X
ビジネス・ビュー			
フィールドの追加または変更	X	X	X
テーブル			
構造(スペック)	X	X	
インデックス	X	X	
結合	X	X	
その他			
サーバーに保管されたヘルプ			
ワークステーションに保管されたヘルプ	X	X	
汎用テキスト・データ構造体	X	X	
データ辞書項目			
[Foundation(ファンデーション)]コード(フル・パッケージおよび部分パッケージでは必須、アップデート・パッケージでは任意)	X	X	X
外国語	X	X	X
非 ERP 9.0 オブジェクト(カスタム項目は、どのパッケージ・タイプでも配布可能)	X	X	X
レプリケートしたローカル・データ(フル・パッケージおよび部分パッケージでは必須、アップデート・パッケージでは任意)	X	X	X
新規アイコン	X	X	

JITI タイプのアップデート・パッケージでは、次の変更がインストールされます。

アプリケーション

- 埋込みイベント・ルール
- 用語一時変更 (FDA テキスト)
- データ構造体
- 処理オプション (レポート)

バッチ・アプリケーション

- レポート
- レポートの埋込みイベント・ルール
- レポート・データ構造体
- レポート用語一時変更
- レポート処理オプション
- バージョンおよび処理オプション値 (処理オプションに依存)
- バージョンの埋込みイベント・ルール

ビジネス・ビュー

- フィールドの追加または変更

その他

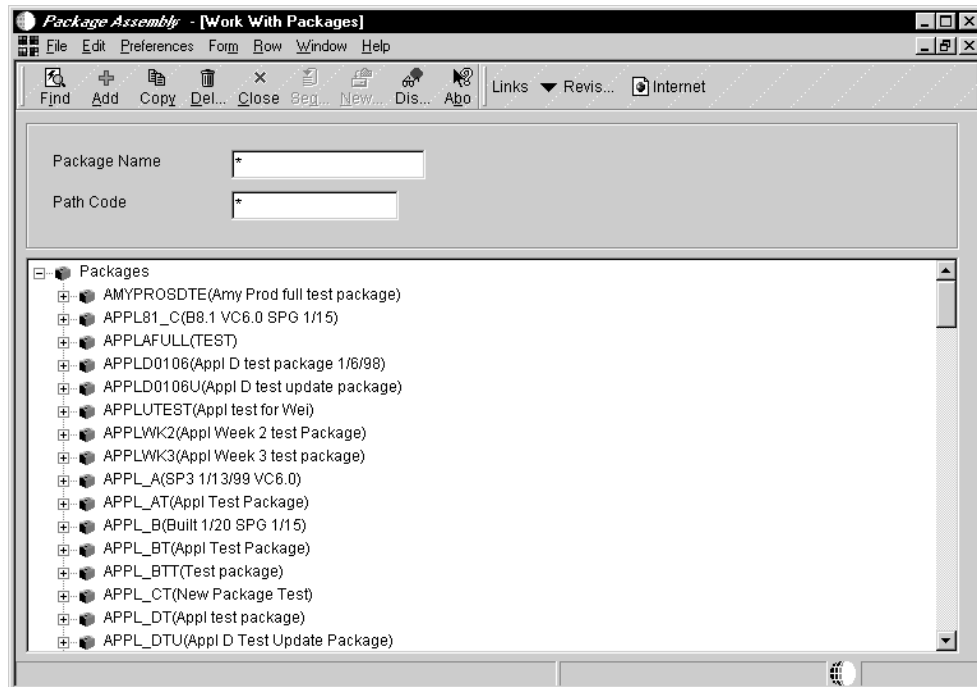
- [Foundation]コード (フル・パッケージおよび部分パッケージでは必須、アップデート・パッケージでは任意)
- 外国語
- 非 ERP 9.0 オブジェクト (カスタム項目は、どのパッケージ・タイプでも配布可能)
- レプリケートしたローカル・データ (フル・パッケージおよび部分パッケージでは必須、アップデート・パッケージでは任意)

データの配布

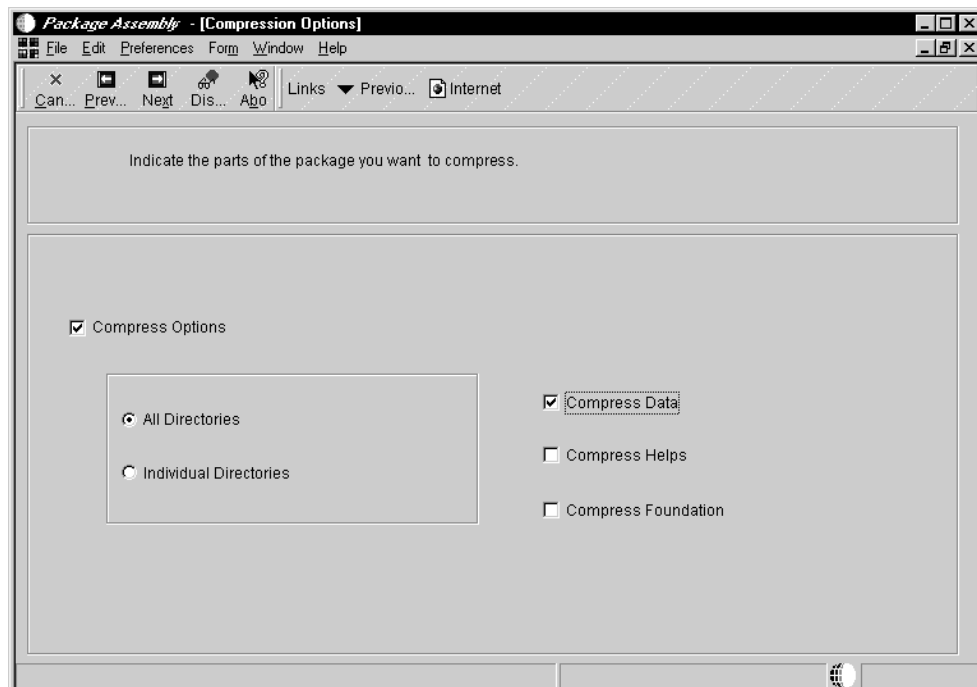
この項目では、新規レプリケーション・ローカル・データベース、およびオフライン処理ユーザーの専用データベースを配布する方法について説明します。

▶ 新規レプリケーション・ローカル・データベースを配布するには

〈Package and Deployment Tools〉メニュー (GH9083) から 〈Package Assembly〉を選択します。



1. 〈Work With Packages(パッケージの処理)〉でパッケージを選択し、[Row(ロー)]メニューから〈Define Build(ビルドの定義)〉を選択します。
2. 〈Compression Options〉フォームに達するまで、〈Package Assembly Director〉フォームで処理を進めます。



3. 〈Compression Options(圧縮オプション)〉フォームで、[Compress Options(圧縮オプション)]と[Compress Data(データの圧縮)]オプションのみをオンにします。

これらのオプションによりデータベースが再圧縮され、このパッケージをインストールしたユーザー全員が最新のデータを受け取るようになります。

注:

圧縮が正常に行われたことを確認するには、
jdedwardsoneworld/B9/pathcode/package/data フォルダにある JZ ファイルのサイズと日付をチェックします。このファイルが 1KB 以下の場合は、圧縮に失敗しました。

オフライン処理ユーザーの専用データベースの配布

オフライン処理ユーザーには、一般的なユーザーが必要とするよりもはるかに大きい J.D. Edwards サポート対象のデータベースが必要です。そのため、オフライン処理ユーザーには、データベース用の 1 つの項目を持つアップデート・パッケージをビルドする必要があります。

これらのユーザーは、ネットワークに接続し、ジャストインタイム・インストールによってアプリケーションをインストールすることができないので、フル・パッケージをインストールする必要があります。したがって、オフライン処理ユーザーは、ジャストインタイム・インストールをオフにした環境を使用する必要があります。オフライン処理ユーザーには、標準フル・パッケージをインストールした後で、オフライン処理ユーザー用に作成されたアップデート・パッケージをインストールするように指示してください。

▶ オフライン処理ユーザーの専用データベースを配布するには

〈Package and Deployment Tools〉メニュー(GH9083)から〈Package Assembly〉プログラム(P9601)を選択します。

1. 〈Work With Packages (パッケージの処理)〉でパッケージを選択し、[Row]メニューから[Define Build]を選びます。
2. 〈Compression Options〉フォームに達するまで、〈Package Assembly Director〉フォームで処理を進めます。
3. 〈Compression Options〉で、[Compress Data]オプションをオンにします。
4. オフライン処理トランザクションを実行するすべてのワークステーションに配布するようにパッケージをスケジュールします。

修正規則

高機能かつ柔軟性に優れた開発ツールを使って、ビジネス・ソリューションやアプリケーションの特定部分を、カスタム修正を行わずにカスタマイズできます。これを「modless modifications (修正なき修正)」と呼んでいます。この修正は、開発者の支援を受けずに容易に実行できます。「modless modifications」は次の項目に対して実行できます。

- ユーザー一時変更
- ユーザー定義コード
- メニューの改訂
- すべてのテキスト
- 処理オプション値
- データ辞書属性
- ワークフロー・プロセス

このような柔軟性により、作業効率が向上すると共に、次のような操作を実行することもできます。

- グリッド・レコードを Microsoft Excel スプレッド・シートなど他のアプリケーションにエクスポートできる。
- グリッドのソート順序を変更できる。
- グリッドのフォントおよび色を変更できる。
- 処理オプションを使用して主要なシステム関数を制御できる。

J.D. Edwards では、できるかぎり柔軟で堅牢なツールの作成に努めていますが、ERP 9.0 をより広範囲にわたって修正しなければならないこともあります。「modless modifications」のような修正と、次のリリース・レベルへのシームレスで適切なアップグレードを可能にするために、ソフトウェアを修正する際は、必ず推奨された規則と標準に従ってください。

アップグレードをスムーズに行うには、カスタム修正を行う前にアップグレードのための準備を整える必要があります。修正プランを適切に立案すると、アップグレードの後の作業を最小限に抑えることができます。プランを作成することで、一般にソフトウェアのアップグレードに必要な時間が短縮されるため、業務への影響を抑え、アップグレードの費用を削減できます。

ERP 9.0 は、カスタム修正をサーバーにチェックインするときにすべてのカスタム修正をトラッキングします。アップグレードを実行する前に、〈Object Librarian Modifications (オブジェクト・ライブラリアン修正)〉レポート(R9840D)を実行し、変更されたオブジェクトの一覧を確認できます。

ERP 9.0 は、コントロール・テーブル(メニュー、ユーザー定義コード、バージョン、データ辞書など)およびトランザクション・テーブル(住所録マスター(F0101)など)から構成されています。J.D. Edwards は、修正可能なデータを含むコントロール・テーブルを提供しています。また、トランザクション・テーブルには貴社のビジネス・データが含まれます。

アップグレード時には、コントロール・テーブルとトランザクション・テーブルの両方に自動マージ処理が実行されます。コントロール・テーブルは新しいソフトウェアのデータとマージされ、トランザクション・テーブルは既存データを変更せずに新しいスペックに変換されます。ビジネス・ビュー、テーブル、データ構造体、処理オプション、イベント・ルール、アプリケーションなどのオブジェクト・スペック・マージでは、ERP 9.0 で定義されているルールに従って、スペックがマージされるか、置換されます。

アップグレードで保持される修正と置換される修正

ERP 9.0 ソフトウェアのカスタム修正が必要な場合は、スムーズで適切なアップグレードを行うために、次の一般定義に従ってください。これらの定義は、アップグレード・プロセスによって保持される修正、および置換される修正を示しています。

- 保持

アップグレードの際に、アップグレード CD に含まれている新規アプリケーションに修正を自動的にマージし、カスタマイズが失われないことを意味します。貴社のスペックとアップグレード版のスペックとの間に矛盾がある場合、貴社のスペックが使用されます。矛盾がない場合は両方のスペックがマージされます。

- 置換

アップグレードの際にカスタマイズとアップグレード版の新規アプリケーションがマージされず、新規ソフトウェアによって既存のカスタマイズ内容が上書きされることを意味します。この場合、アップグレードが完了した後に再度カスタマイズ内容を追加する必要があります。

アップグレード・プロセスの前に〈Object Librarian Modifications〉レポート(R9840D)を実行し、修正したオブジェクトを認識してください。

修正用一般規則

次の一般修正規則は、すべての ERP 9.0 オブジェクトに適用されます。

- 新規オブジェクトを追加する場合は、システム・コード 55～59 を使用すること。

ERP 9.0 では予約済みシステム・コードを使用して、さまざまなアプリケーションや業種別グループを区別します。システム・コード 55～59 を使用して開発したオブジェクトは上書きされません。

- ZJDE または XJDE で始まるカスタム・バージョン名や新規バージョン名を作成しないこと。

これらのプレフィックスは、J.D. Edwards がソフトウェアに添付する標準バージョン・テンプレート用として確保されています。これらのプレフィックスを使用すると、名前の競合が発生した場合にカスタム・バージョンが保持されません。オリジナルのバージョンをコピーして、新しいテンプレートやバージョンを作成できます。

- アップグレードする場合は、最新のセントラル・オブジェクト・セットからパッケージを作成し、開発サーバー、セントラル・オブジェクトおよびオブジェクト・ライブラリアンのデータ・ソースのバックアップを行うこと。これにより、比較やトラブルシューティングを行う場合に、これらのスペックにアクセスできます。

対話型アプリケーション

既存のアプリケーションのコントロール、グリッド・カラム、およびハイパー・アイテムは削除せずに、非表示または無効にしてください。これらの項目は計算用または変数として使用されている場合があります。これらの項目を削除すると、主要なシステム関数が無効になる場合があります。

アップグレードで保持される修正と置換される修正

下表に、アップグレード時に保持される対話型アプリケーション要素と、置換される要素を示します。

オブジェクト	保持	置換	コメント
新規アプリケーション	X		<p>アプリケーションを作成するには、最初から作成する方法と、アプリケーション設計ツールの「コピー」機能を使用して、既存のアプリケーションをコピーする方法があります。コピー機能を使用すると、イベント・ルールを含むすべてのアプリケーション・スペックをコピーできます。</p> <p>コピー機能を使って既存のアプリケーションをコピーして修正した場合、アップグレードの際にオリジナルのアプリケーションに対して行っていた変更があっても、コピーされた新しいアプリケーションにはそれらは適用されません。</p>
既存のフォームに追加された新規ハイパー項目	X		
既存のフォームに追加された新規コントロール	X		
既存のフォームに追加された新規グリッド・カラム	X		
スタイル変更	X		スタイル変更にはフォントと色が含まれます。新規コントロールには、標準ベース定義が設定されています。スタイルを変更する場合は、アプリケーションに追加した新規コントロールのスタイルも変更する必要があります。
コード・ジェネレーター時変更	X		
データ辞書一時変更	X		
ロケーション変更とサイズ変更	X		今後のソフトウェア・リリースで、カスタム・コントロールを配置したのと同じ位置に新規コントロールが配置される場合もあります。その場合、新規コントロールは、カスタム・コントロールの上に重なって表示されます。これによってアプリケーションのイベント・ルールや機能に影響が及ぶことはありません。アップグレード後、アプリケーション設計ツールを使用すれば、これらのコントロールの位置を変更できます。
タブまたはカラムの順序変更	X		アップグレード・プロセスでは、新しいコントロールをカスタム・タブの一番最後に追加します。アップグレード後にタブの配列を確認することができます。
既存のアプリケーションのカスタム・フォーム		X	既存のアプリケーションにカスタム・フォームを追加する代わりに、システム・コード 55～59 を使ってカスタム・アプリケーションを作成し、そのカスタム・アプリケーションにカスタム・フォームを配置してください。その後で、カスタム・アプリケーション内でカスタム・フォームを呼び出す既存のアプリケーション・フォーム・エグジットやロー・エグジットを追加できます。アプリケーション内のフォームからではなく、ロー・エグジットから外部アプリケーションを呼び出しても、システムのパフォーマンスには影響しません。

レポート

〈Report Design Aid(レポート設計ツール)〉のスペックには、次の規則が適用されます。

既存のレポート上にあるオブジェクトは削除しないでください。代わりに、非表示にします。これらのオブジェクトが計算用または変数として使用されている場合があります。これらのオブジェクトを削除すると、主要なシステム関数が無効になる場合があります。

アップグレードで保持される修正と置換される修正

下表に、アップグレード時に保持されるレポート要素と、置換される要素を示します。

オブジェクト	保持	置換	コメント
新規レポート	X		レポートを作成する方法には、最初から作成する方法と、レポート設計ツールの「コピー」機能を使用して、既存のレポートをコピーする方法があります。後者の方法では、イベント・ルールを含む、レポートのすべてのスペックをコピーすることができます。 コピー機能を使って既存のレポートをコピーして修正した場合、アップグレードの際にオリジナルのレポートに対して行っていた変更があっても、コピーされた新しいレポートにはそれらは適用されません。
既存のレポートに追加された新規の固定情報	X		
既存のレポートに追加された新しい変数(文字型)	X		
既存のレポートに追加された新規変数(数値)	X		
既存のレポートに追加された新規日付変数	X		
既存のレポートに追加された新規ランタイム変数	X		
既存のレポートに追加された新規データベース変数	X		
既存のレポートに追加された新規データ辞書変数	X		
スタイル変更	X		スタイル変更にはフォントと色が含まれます。新規コントロールには、標準ベース定義が設定されています。デフォルトのスタイルを変更する場合は、レポートに追加した新しいコントロールのスタイルも変更する必要があります。
位置とサイズの変更	X		今後のソフトウェア・リリースで、コントロールのように、カスタム・オブジェクトを配置したのと同じ位置に新しいオブジェクトが配置される場合もあります。その場合、カスタム・オブジェクトと新しいオブジェクトは横並びで表示されます。これによってレポートのイベント・ルールや機能に影響が及ぶことはありません。アップグレード後、レポート設計ツールを使ってオブジェクトの位置を変更できます。

オブジェクト	保持	置換	コメント
データ辞書一時変更	X		
既存のレポートのカスタム・セクション		X	カスタム・セクションを既存レポートに追加する代わりに、〈Report Interconnect (レポート・インターコネクト)〉を使用して、システム・コード 55～59 を使用している新規のカスタム・レポートに接続します。レポート・インターコネクションでレポートを呼び出しても、システムのパフォーマンスには影響しません。

アプリケーション・テキストの変更

下の表に、アップグレード時に保持されるアプリケーション・テキストと、置換されるテキストを示します。

アップグレードで保持される修正と置換される修正

下表に、アップグレード時に保持されるアプリケーション・テキスト要素と、置換される要素を示します。

オブジェクト	保持	置換	コメント
アプリケーション設計ツールで行った一時変更	X		
レポート設計ツールで行った一時変更	X		
対話型用語一時変更で行った一時変更	X		
バッチ用語一時変更で行った一時変更	X		

テーブル・スペック

アップグレード時には、あるリリース・レベルのテーブル・スペックが次のレベルにマージされます。

アップグレードで保持される修正と置換される修正

下表に、アップグレード時に保持されるテーブル・スペック要素と、置換される要素を示します。

オブジェクト	保持	置換	コメント
新規テーブル	X		
ERP 9.0 テーブルへのカスタム・インデックス	X		
既存の ERP 9.0 テーブルに追加または削除されたカラム		X	これには、フィールド長、フィールド・タイプおよび小数点位置の変更が含まれます。 新規カラムを既存のテーブルに追加する代わりに、システム・コード 55～59 を持つタグ・テーブルを使用します。

カスタム・タグ・ファイルでは、データ辞書のデータ項目の変更を確認してください。今後のリリースで、サイズなど、データ項目の一定の属性が変更される場合があります。これにより、データ整合性や、データベースでのデータの保存方法に影響が及ぶことがあります。このため、〈Table Conversion (テーブル変換)〉ツールを使用して、タグ・ファイル・データを新しいリリース・レベルに変換しなければならないことがあります。基本ファイルでは、アップグレード処理の際に、データ辞書が新規リリース・レ

ベルに自動的に適用されます。アップグレードでは、カスタム・タグ・ファイルに対するカスタム・インデックスが保持されます。

コントロール・テーブル

コントロール・テーブルには、ユーザー定義コード(UDC)、メニューおよびデータ辞書項目が存在します。アップグレード時には、テーブル変更処理を使用して、前のリリース・レベルから次のリリース・レベルへコントロール・テーブルがマージされます。この処理ではデータ・マージの基準として、ERP 9.0 のテーブルではなく、貴社のコントロール・テーブルが使用されます。

アップグレードで保持される修正と置換される修正

下表に、アップグレード時に保持されるコントロール・テーブル要素と、置換される要素を示します。

オブジェクト	保持	置換	コメント
データ辞書カスタム変更	X		これには、ロー、カラムおよび用語解説テキストの変更が含まれます。アップグレード処理では、処理基準として貴社のデータ辞書が使用されます。ERP 9.0 のデータ項目との競合が発生した場合は、貴社の変更が優先されます。新規データ項目は、システム・コード 55～59 を使用して作成してください。
ユーザー定義コード	X		アップグレード処理では、新規にハードコード化された ERP 9.0 の値がマージされます(ソフトウェア製造元が所有する値はシステム 90 以上と H90 以上です)。アップグレード処理では、ERP 9.0 ハードコード値が貴社のカスタム値と競合した場合には、すべて報告されます。
メニュー	X		ERP 9.0 基本メニューと矛盾する場合は、カスタム変更が優先されます。
既存のコントロール・テーブルに追加または削除されたカラム		X	

ビジネス・ビュー

既存のビジネス・ビューからはカラムを削除しないでください。アプリケーションで 사용되는ビジネス・ビューを変更すると、そのアプリケーションの実行時に予期しない結果が発生することがあります。カラムを非表示にする場合は、アプリケーション設計ツールまたはレポート設計ツールを使用して、アプリケーション設計レベルで行ってください。ビジネス・ビューからカラムをいくつか削除しても、システムのパフォーマンスはさほど向上しません。

アップグレードで保持される修正と置換される修正

下表に、アップグレード時に保持されるビジネス・ビュー要素と、置換される要素を示します。

オブジェクト	保持	置換	コメント
新規のカスタム・ビジネス・ビュー	X		
既存のビジネス・ビューに追加した新規カラム、結合またはインデックス	X		
ビジネス・ビューから削除されたカラム		X	

イベント・ルール

アップグレード処理の際に、カスタマイズされたイベント・ルールと、ソフトウェアがインストールする新しいイベント・ルールとの整合性が検証されます。矛盾が存在する場合、ERP 9.0 は、カスタム・イベント・ルールを無効にし、それらを新しいイベント・ルールの末尾に追加します。下表に、保持されるカスタム・イベント・ルールと無効にされるルールを示します。

アップグレードで保持される修正と置換される修正

下表に、アップグレード時に保持されるイベント・ルール要素と、置換される要素を示します。

オブジェクト	保持	置換	コメント
カスタム・アプリケーション、レポートおよびテーブルのカスタム・イベント・ルール	X		
カスタム・ビジネス関数のカスタム・イベント・ルール	X		
新規カスタム・コントロールのカスタム・イベント・ルール	X		
システム・イベント・ルールが添付されていない、システム・アプリケーション、レポート、テーブルのイベント	X		
システム・イベント・ルールが添付されていない、システム・ビジネス関数のイベント	X		
既存のイベント・ルールが添付されている、システム・アプリケーション、レポート、テーブルのイベント		X	アップグレード時に、カスタム・イベント・ルールは無効にされ、標準イベント・ルールの末尾に追加されます。マージ処理の際に、アップグレード処理で無効にされたイベント・ルールを報告する〈Object Librarian Modifications〉レポート(R9840D)が印刷されます。
イベント・ルールが添付されている、システム・ビジネス関数のイベント		X	アップグレード時に、カスタム・イベント・ルールは無効にされ、標準イベント・ルールの末尾に追加されます。マージ処理の際に、アップグレード処理で無効にされたイベント・ルールを報告する〈Object Librarian Modifications〉レポートが印刷されます。

システム・オブジェクトに対するカスタム・イベント・ルールを復元するには、イベント・ルールをハイライトしてイベント内の適切な場所にドラッグし、使用可能にします。アップグレードする前に、次のタスクを実行します。

- 修正済みアプリケーションを識別するために、〈Object Librarian Modifications〉レポートを実行します。
- 修正済みアプリケーションのイベント・ルールを印刷し、カスタム・イベント・ルールを復元する際にイベントのロジックを理解できるようにします。

データ構造体

アップグレードで保持される修正と置換される修正

下表に、アップグレード時に保持されるデータ構造体要素と、置換される要素を示します。

オブジェクト	保持	置換
カスタム・フォーム・データ構造体	X	
カスタム処理オプション・データ構造体	X	
カスタム・レポート・データ構造体	X	
カスタム・ビジネス関数データ構造体	X	
カスタム汎用テキスト・データ構造体	X	
既存のシステム・フォーム・データ構造体に対する修正		X
既存のシステム処理オプション・データ構造体に対する修正		X
既存のシステム・レポート・データ構造体に対する修正		X
既存のシステム・ビジネス関数データ構造体に対する修正		X
既存のシステム汎用テキスト・データ構造体に対する修正		X

システム・データ構造体に対するカスタム修正を次のリリース・レベルに移行させるには、〈Object Librarian Modifications〉レポート(R9840D)を実行して、修正済みデータ構造体の一覧を確認し、データ構造体を手作業で変更するときに、このレポートを手引きとして利用します。

ビジネス関数に関する規則

新規のカスタム・ビジネス関数(BSFN)では、カスタム修正を保管するために、新しい(カスタム)親 DLL を作成します。詳しくは、『開発ツール』ガイドを参照してください。

カスタム変更を次のリリース・レベルに移行するには、〈Object Librarian Modifications〉レポート(R9840D)を実行して修正済みビジネス関数の一覧を確認し、ビジネス関数を手作業で変更するときに、このレポートを手引きとして利用します。

ビジネス関数内で他のビジネス関数を呼び出す場合は、常に標準 API(jdeCallObject)を使用してください。この API に従わなければ、エラーが発生します。

既存の基本ビジネス関数のソース・コードが修正されているかどうかを判別するには、Microsoft WinDiff などのサードパーティのソース比較ツールを使用します。ビジネス関数内の API に対する修正を確認するには、オンライン・ヘルプの API に関する最新情報を参照してください。

アップグレードで保持される修正と置換される修正

下表に、アップグレード時に保持されるビジネス関数要素と、置換される要素を示します。

オブジェクト	保持	置換	コメント
新規のカスタム・ビジネス関数オブジェクト	X		
既存のシステム・ビジネス関数オブジェクトに対する修正		X	NER BSFN は修正できます。

バージョン

新規カスタム・バージョンの場合、XJDE および ZJDE で始まらない名前が付いた新規バージョンを作成します。詳細については、『基本操作』ガイドを参照してください。

アップグレードで保持される修正と置換される修正

下表に、アップグレード時に保持されるバージョン要素と、置換される要素を示します。

オブジェクト	保持	置換
非 JDE バージョン	X	
バージョン・スペック	X	
処理オプション・データ	X	
ZJDE および XJDE バージョンのすべてのスペック		X
XJDE バージョンのすべての処理オプション・データ		X

処理オプション・テンプレートを使用した JDE 以外のバージョンでは、処理オプション・データはコピーできますが、変換できません。実行時に警告メッセージが表示され、一部のデータが失われることがあります。

また、JDE テンプレートを使用したカスタム・バージョンのイベント・ルール修正と親テンプレートの不一致は調整されません。

オブジェクト管理

業界標準では、オブジェクトとはデータとデータを操作するために使用される構造と機能を備えた独立したエンティティです。オブジェクトは、ソフトウェア・スペックに基づいた再利用可能なエンティティを指します。スペックとはシステム・オブジェクトの完全な記述です。各オブジェクトには独自のスペックがあり、サーバー側とクライアント側の両方に保管されます。

システムでは、オブジェクトを次の 2 か所に保管します。

- セントラル・オブジェクトを保管する「セントラル・ストレージ」サーバー。開発者がセントラル・オブジェクトをチェックアウトすると、セントラル・オブジェクトはワークステーション上でレプリケートされ、開発者が再びチェックインすると、セントラル・オブジェクトに再変換されます。
- ランタイム・オブジェクトを保管するワークステーションとサーバー。ランタイム・オブジェクトとは、システムを実際に実行するレプリケートされたオブジェクトのことです。

ここでは、セントラル・オブジェクト・ロケーションからオブジェクト移動先にオブジェクトがどのように移動するかについて説明します。オブジェクトのチェックイン/アウト、プロジェクトへのオブジェクトの追加、およびチェックアウトせずにオブジェクトを取得する方法の詳細については、『開発ツール』ガイドの「オブジェクト管理ワークベンチ」を参照してください。

オブジェクト移動の理解

次の作業を実行すると、オブジェクトは常にセントラル・オブジェクト・ロケーションからオブジェクト移動先に移動します。

- オブジェクトのチェックイン/アウト
- プロジェクトへの既存オブジェクトの追加
- <Object Management Workbench (オブジェクト管理ワークベンチ)> プログラム(P98220)からのオブジェクトの取得の実行
- ワークステーション・インストール・プログラムの実行

ワークステーションのインストール時には、すべてのオブジェクト、およびレプリケートされたデータが収められたサポート対象のローカル・データベースが、パッケージからワークステーションにコピーされます。オブジェクトは、スペックを含めたパッケージに収められているものだけがコピーされます。パッケージにスペックが含まれていない場合、オブジェクトはレプリケートされませんが、代わりにジャストインタイム・インストールを通じてワークステーションにインストールされます。

特に記載がない限り、オブジェクトの移動は、オブジェクトのチェックイン/アウト、プロジェクトへのオブジェクトの追加、オブジェクト取得のいずれを行う場合でも同じです。次の表は、各タイプのオブジェクトをチェックイン、チェックアウト、追加、または取得したときに移動するオブジェクトとスペックを示しています。

テーブル(オブジェクト・タイプ TBLE)

次のオブジェクトが移動します。

- テーブルおよびテーブル・イベント・ルール・スペック
- ソース・ファイル(*.c)
- テーブル・ヘッダー・ファイル(*.h)
- オブジェクト・ファイル(*.obj)
- テーブル・イベント・ルール・インクルード・ファイル(*.hxx)

テーブル・ヘッダーは、データベース内に存在する実際のテーブルとは異なります。テーブル自体は、テーブル設計ツールを使用して作成したときに生成されます。〈Workstation Installation〉プログラムは、そのテーブルがサポート対象のデータベースに保管されている場合、テーブルをワークステーションにコピーします。

ビジネス・ビュー(オブジェクト・タイプ BSVW)

スペックが移動します。

C ビジネス関数(オブジェクト・タイプ BSFN、ソース言語 C)

次のオブジェクトが移動します。

- スペック
- ソース・ファイル(*.c)
- ヘッダー・ファイル(*.h)
- オブジェクト・ファイル(*.obj)

ビジネス関数データ構造体(オブジェクト・タイプ DSTR)

スペックが移動します。

埋込みイベント・ルール

埋込みイベント・ルールをチェックアウトすることはできません。埋込みイベント・ルールは、イベント・ルールが埋め込まれているオブジェクトをチェックアウトしたときに移動します。例えば、埋込みイベント・ルールがテーブル、対話型アプリケーションまたはバッチ・アプリケーションに添付されている場合、これらのテーブルやアプリケーションを移動すると、埋込みイベント・ルールのスペックも一緒に移動します。

イベント・ルール・ビジネス関数(オブジェクト・タイプ BSFN、ソース言語 C)

イベント・ルール・ビジネス関数(オブジェクト・タイプ BSFN)はチェックアウトできます。イベント・ルール・ビジネス関数をチェックアウトする際、.h ファイルは include ディレクトリに、.c ファイルは source ディレクトリに、.obj ファイルは obj ディレクトリにそれぞれ移動され、ローカル・スペックは更新されます。

次のオブジェクトが移動します。

- スペック
- ソース・ファイル(*.c)
- ヘッダー・ファイル(*.h)
- オブジェクト・ファイル(*.obj)

メディア・オブジェクト・データ構造体(オブジェクト・タイプ GT)

データ構造体スペックが移動します。

対話型アプリケーション(オブジェクト・タイプ APPL)

次のスペックが移動します。

- アプリケーション
- フォーム
- フォーム・データ構造体
- 埋込みイベント・ルール

バッチ・アプリケーション(オブジェクト・タイプ UBE)

レポート・スペックとイベント・ルール・スペックが移動します。レポートとは別のバージョンをチェックインおよびチェックアウトする必要があります。

参照

- オブジェクトのチェックイン/アウト、プロジェクトへのオブジェクトの追加、およびオブジェクトの取得の詳細については、『開発ツール』ガイドの「オブジェクト管理ワークベンチ」

データ辞書項目のレプリケート

データ辞書はサーバーのリレーショナル・テーブル内に存在します。これは、サーバーおよびワークステーションにレプリケートする、すべてのデータ辞書変更を行うためのパブリッシャ・データ辞書です。パブリッシャ・データ辞書は次のテーブルに格納されます。

- データ項目マスター(F9200)
- データ・フィールド表示テキスト(F9202)
- データ項目名称記述(F9203)
- データ辞書 - エラー・メッセージ情報(F9207)
- データ・フィールド・スペック(F9210)
- データ辞書 - スマート・フィールド(F9211)
- メディア・オブジェクト(F00165)

ビルド・プロセスで作成されたファイルの理解

この節では、ワークステーションおよび各種サーバー・プラットフォームのビルド・プロセスで作成されるファイルについて説明します。

ワークステーション・ビルド

ワークステーション上のビジネス関数 DLL (ダイナミック・リンク・ライブラリ) は関連するビジネス関数によってグループ化されます。このグループ化では、各 DLL に含まれるプロシージャのサイズと数が制限されます。グループ化することで、メモリ割当エラーが防止され、同一の DLL から多数のプロシージャをエクスポートするときに生じる可能性があるプラットフォームの制限を避けることができます。

本稼働用環境 PD9/bin32 ディレクトリには、ワークステーション上で作成された DLL が格納されています。ビジネス関数ソース・ファイルはいずれも PD9/source ディレクトリに置かれています。

ビジネス関数ビルドで作成されたファイル

オブジェクト管理ワークベンチを使用して 1 つのビジネス関数を作成する場合、〈Business Function Builder(ビジネス関数ビルダー)〉プログラムは貴社のメイク・ファイル(*.mak)を使用して、ビジネス関数を各 DLL 内に作成し、次のファイルを作成します。

- ソース・ファイル(*.c)
- ヘッダー・ファイル(*.h)
- オブジェクト・ファイル(*.obj)

あるビジネス関数を別のビジネス関数から呼び出すには、jdecallobject API を使用する必要があります。

ビジネス関数イベント・ルールは次のファイルを作成します。

- OBJNAME.c
- OBJNAME.h
- OBJNAME.obj

テーブル・イベント・ルールは次のファイルを作成します。

- OBJNAME.c
- OBJNAME.hxx
- OBJNAME.obj

サーバー・パッケージ・ビルド

サーバー・パッケージ・ビルドは、デプロイメント・サーバーからエンタープライズ・サーバー・プラットフォームにパス・コード・オブジェクトを移動するときに使用します。サーバー・パッケージ・ビルド・プロセスは、〈Package Assembly〉中にフル・パッケージまたは部分パッケージを作成することで開始されます。ただし、サーバー・パッケージの場合、部分パッケージは使用できません。パッケージの組立を完了後に、サーバーの[Package Definition(パッケージ定義)]チェックボックスをオンにして、続く画面で、使用可能なサーバーのリストから関連サーバーを選択する必要があります。パッケージ定義を完了し、パッケージがアクティブ化されたら、そのパッケージをハイライトして、[Row]メニューから[Submit Build(ビルドの投入)]を選択します。これにより、サーバー・パッケージのビルドを開始します。このビルド・プロセスが正常終了した場合は、サーバー・パッケージを配布してパス・コード・オブジェクトをアクティブ化することができます。

サーバー・パッケージを組み立てるには、〈Package Assembly〉に表示されるファンデーション、データベース、およびオブジェクト情報を使用して、.c、.h、および.hxx ファイル用のビルド情報、スペック・ファイル、およびビジネス関数ソースを生成し、〈Package Assembly〉中に割り当てられるステージ領域に格納します。サーバー・パッケージ・ビルドで、これらのオブジェクトが生成されてステージ領域に格納されると、オブジェクトはパッケージ定義で指定した各サーバーに転送されます。その後、サーバー・パッケージを受け取るサーバーに対して、ビジネス関数ソース・コードをコンパイルし、対応するビジネス関数 DLL を生成するように指示されます。

UNIX サーバー・ビルド

ここでは、UNIX サーバー上でのビジネス関数のビルド時に生成されるファイルについて説明します。

ビジネス関数ビルドで作成されたファイル

ビジネス関数をビルドする場合、次のソース・ファイル・グループが実際にコンパイルされます。

- ビジネス関数イベント・ルール
- テーブル・イベント・ルール
- C ビジネス関数イベント・ルール

ビジネス関数をビルドする場合、次のファイル・タイプがビルド・プロセスに適用されます。

- ソース・ファイル(.c)
- ヘッダー・ファイル(.h, .hxx)

ビジネス関数を作成する場合、ビルド・プロセスは次のファイル・タイプを作成します。

- オブジェクト・ファイル(.o)
- メイク・ファイル(.mak)
- 共有ライブラリ(.sl, .so)

Windows NT ワークステーション用の DLL に相当する、ビジネス関数の共有ライブラリは統合されています。そのため、オブジェクト・ライブラリアン - 状況詳細テーブル(F9861)内の親 DLL ごとに、それぞれ 1 つの共有ライブラリが作成されます。カスタム・ビジネス関数を作成する場合は、ソフトウェア付属の親 DLL の代わりに独自の親 DLL を使用してください。

ビジネス関数の保管場所

UNIX プラットフォームでは、ビジネス関数の共有ライブラリは、関連するビジネス関数によってグループ化されます。このグループ化では、各共有ライブラリに含まれるプロシージャのサイズと数が制限されます。グループ化することで、メモリ割当エラーが防止され、共有ライブラリごとにエクスポートできるプロシージャのプラットフォーム固有の数量制限を避けることができます。環境 PD7334 ディレクトリの下にはソース・ディレクトリがあります。このソース・ディレクトリには、エンタープライズ・サーバー上で作成された各共有ライブラリのサブディレクトリが含まれています。ディレクトリ構造は次のとおりです。

PD7334

source

CAEC

CALLBSFN

CCORE

CDESIGN

CDIST

CFIN

CHRM

CMFG

JDBTRIG

各サブディレクトリには、共有ライブラリに属するビジネス関数のソース・ファイルが含まれます。共有ライブラリはすべて PD7334/bin32 ディレクトリにインストールされます。共有ライブラリには、プレフィックスの lib を伴う共有ライブラリのサブディレクトリ名が付けられ、拡張子として HPUX の場合は .sl、AIX の場合は .so が設定されて、たとえば、libccore.sl というようになります。

スペック・ファイル

スペック・ファイルは、フォーマット変換のためにサーバーに転送する前に、個々のファイルに統合する必要があります。スペック・ファイルは、同じプレフィックス (RDATEXT など) でグループ化された 2 つのファイルを合わせたものです。これら 2 つのファイルは異なる拡張子 (データ・ファイルは .ddb、インデックス・ファイルは .xdb) を持ち、2 つ合わせて完全なスペック・ファイルを構成します。

Windows NT サーバー・ビルド

ここでは、Windows サーバー上でのビジネス関数のビルド時に生成されるファイルについて説明します。

ビジネス関数ビルドで作成されたファイル

ビジネス関数をビルドする場合、次のソース・ファイル・グループが実際にコンパイルされます。

- ビジネス関数イベント・ルール
- テーブル・イベント・ルール
- C ビジネス関数イベント・ルール

ビジネス関数をビルドする場合、次のファイル・タイプがビルド・プロセスに適用されます。

- ソース・ファイル(.c)
- ヘッダー・ファイル(.h, .hxx)

ビジネス関数を作成する場合、ビルド・プロセスは次のファイル・タイプを作成します。

- オブジェクト・ファイル(.o)
- メイク・ファイル(.mak)
- DLLs(.dll)

ビジネス関数 DLL は、UNIX プラットフォームやワークステーション上の DLL と同様に統合されています。そのため、オブジェクト・ライブラリアン - 状況詳細テーブル(F9861)内の親 DLL ごとに、それぞれ 1 つの共有ライブラリが作成されます。カスタム・ビジネス関数を作成する場合は、ソフトウェア付属の親 DLL の代わりに独自の親 DLL を使用してください。

ビジネス関数の保管場所

Windows NT プラットフォームでは、ビジネス関数 DLL は、関連するビジネス関数によってグループ化されます。このグループ化では、各 DLL に含まれるプロシージャのサイズと数が制限されます。グループ化することで、メモリ割当エラーが防止され、DLL ごとにエクスポートされるプロシージャのプラットフォーム固有の制限を避けることができます。

環境 PD9 ディレクトリの下にはソース・ディレクトリがあります。このソース・ディレクトリには、エンタープライズ・サーバー上で作成された各 DLL のサブディレクトリが含まれています。

ディレクトリ構造は次のとおりです。

PD9

source

CAEC

CALLBSFN

CCORE

CDESIGN

CDIST

CFIN

CHRM

CMFG

JDBTRIG

各サブディレクトリには、DLL に属するすべてのビジネス関数ソース・ファイルが含まれます。DLL はすべて PD9¥bin32 ディレクトリにインストールされます。これらのファイルには、DLL サブディレクトリと同じ名前が付けられ、拡張子として.dll が設定されます。

スペック・ファイル

スペック・ファイルは、フォーマット変換のためにサーバーに転送する前に、個々のファイルに統合する必要があります。スペック・ファイルは、同じプレフィックス(RDATEXT など)でグループ化された 2 つのファイルを合わせたものです。これら 2 つのファイルは異なる拡張子(データ・ファイルは.ddb、インデックス・ファイルは.xdb)を持ち、2 つ合わせて完全なスペック・ファイルを構成します。

AS/400 サーバー・ビルド

ここでは、AS/400 サーバー上でのビジネス関数のビルド時に生成されるファイルについて説明します。

ビジネス関数ビルドで作成されたファイル

ビジネス関数をビルドする場合、サーバー・パッケージ・ビルドでは、QSYS ファイル・システム内でパッケージ名を含むライブラリに次のファイルが作成されます。

- *MODULES – オブジェクト名。
- *USRSPC – ユーザー領域には、各ビジネス関数 DLL に含まれる.c ファイルの情報を保持。
- *SRVPGM – サーバー・プログラムは、AS/400 上の DLL。
- *FILE – コンパイルされたビジネス関数に関するログのみを格納。

ビジネス関数ソース・メンバの保管場所

ERP 9.0 の新機能として、サーバー・パッケージ・ビルドの実行時には、ビジネス関数のソースとヘッダーのロケーションが、すべてのサーバー・プラットフォーム間で標準化されます。AS/400 のビジネス関数のソースおよびヘッダーは、統合ファイル・システム(Integrated File System: IFS)に転送されるようになりました。オブジェクトは、AS/400 用 IFS のサーバー・パッケージ・ディレクトリの下にある次のサブディレクトリに転送されます。

PD9

```
include
pack
source
CAEC
CALLBSFN
CCORE
CDESIGN
CDIST
.
spec
text
```

PD9/include – .h および.hxx ソース・ファイルが格納されるサブディレクトリ。これらのオブジェクトは、サーバーから取り込まれてビルドされます。

PD9/pack – 同じプラットフォームのサーバー間でサーバー・パッケージを移動するために使用される、圧縮されたオブジェクトが格納されます。

PD9/source – ビジネス関数 DLL 名を含むサブディレクトリを持ちます。各サブディレクトリには、コンパイルされて DLL にリンクされるビジネス関数の.c ソース・ファイルが格納されます。

PD9/spec – パス・コードのスペック・ファイルが格納されます。スペック・ファイルは、同じプレフィックス(RDATEXT など)でグループ化された 2 つのファイルを合わせたものです。サフィックス.xdb および.ddb は、スペック・ファイルを表します。

PD9/text – ビジネス関数 DLL とスペック・ファイルのビルドのテキスト・ファイルとステータス・ファイル(.txt および.sts)が格納されます。テキスト・ファイルには、サーバー・パッケージ・ビルドにより転送されたビジネス関数 DLL とスペック・ファイルに関する情報が格納されます。また、ビジネス関数 DLL 作成用のビルド指示も格納されます。スペック・ファイルのステータス・ファイルは、サーバー・パッケージ・ビルドで、親パッケージの spec サブディレクトリにスペックが正常に転送されたかどうかを

示します。ビジネス関数 DLL のステータス・ファイルは、どの.c ソース・ファイルが正常にコンパイルされリンクされたかを示します。

注:

Xe からの移行など、ソフトウェアのアップグレード後は、サーバー・パッケージ・アップデートのビルド・エラーを回避するために、サーバー・パッケージ・ビルドで既存の AS/400 サーバー・パス・コードをリビルドし、LINKBSFN プログラムを使用してビジネス関数を手作業で再リンクする必要があります。

スペック・ファイル

スペック・ファイルは、フォーマット変換のためにサーバーに転送する前に、個々のファイルに統合する必要があります。スペック・ファイルは、同じプレフィックス (RDATEXT など) でグループ化された 2 つのファイルを合わせたものです。これら 2 つのファイルは異なる拡張子 (データ・ファイルは .ddb、インデックス・ファイルは .xdb) を持ち、2 つ合わせて完全なスペック・ファイルを構成します。

複製オブジェクトとセントラル・オブジェクトの関連付け

次の表は、スペック・テーブル内に保管されている複製 (レプリケートされた) オブジェクトと、2 進ラージ・オブジェクト (BLOB) を含むリレーショナル・データベース内に保管されているセントラル・オブジェクトの相互関係を示しています。

レプリケートされたオブジェクト	セントラル・オブジェクトおよび記述
DDTABL	テーブル・ヘッダー・テーブル (F98710)。テーブルごとに 1 つのレコードを格納します。
DDCLMN	テーブル・カラム・テーブル (F98711)。テーブル内のカラムごとに 1 つのレコードを格納します。
DDPKEYH	プライマリ・インデックス・ヘッダー・テーブル (F98712)。テーブル・インデックスごとに 1 つのレコードを格納します。
DDPKEYD	プライマリ・インデックス詳細テーブル (F98713)。インデックス内のカラムごとに 1 つのレコードを格納します。
BOBSPEC	ビジネス・ビュー・スペック・テーブル (F98720)。ビジネス・ビューごとに 1 つのレコードを格納します。
GBRLINK	イベント・ルール - リンク・テーブル (F98740)。アプリケーション、レポートまたはテーブルでのイベント・ルールが設定されたイベントごとに 1 つのレコードを格納します (イベント・ルール・ビジネス関数リンクは、ビジネス関数 F9862 に保管されます)。
GBRSPEC	イベント・ルール - スペック・テーブル (F98741)。イベント・ルールの行ごとに 1 つのレコードを格納します。
DSTMPL	データ構造体テンプレート・テーブル (F98743)。各ビジネス関数、処理オプション、フォーム・インターコネクション、レポート・インターコネクション・データ構造体に対して 1

	つのレコードを格納します。
FDATEXT	フォーム設計ツール・テキスト情報テーブル(F98750)。アプリケーション用一時変更テキストを格納します。
FDASPEC	フォーム設計ツール・スペック情報テーブル(F98751)。アプリケーション内の各カラム、グリッド・ライン、ボタン、ハイパー項目、コントロールなどに対して 1 レコード。
ASVRHDR	フォーム設計ツール/ソフトウェア・バージョン・リポジトリ・ヘッダー情報(F98752)。各アプリケーションに対して 1 レコード(アプリケーションに処理オプションがある場合、その情報もレコード内に保管されます)。
ASVRDTL	フォーム設計ツール/ソフトウェア・バージョン・リポジトリ詳細情報(F98753)。各フォームに対して 1 レコード(データ構造体への参照も含みます)。
RDATEXT	レポート設計ツール・テキスト情報(F98760)。バッチ・レポート用一時変更テキスト。
RDASPEC	レポート設計ツール・スペック情報(F98761)。バッチ・レポートおよびバージョン内の各セクション、カラム、ソート、定数などに対して 1 レコード。
JDEBLC	JDEBLC - ビジネス関数情報(F98762)。BSFN 内の関数ごとに 1 レコード。
CGTYPE	スペック・フォーマット内だけに保管される。コード・ジェネレータ・フォーム・タイプ。
DDDICT: ジャストインタイム・インストールされた各データ辞書項目に対して 1 レコード	<ul style="list-style-type: none"> • F9200、データ項目マスター • F9202、データ・フィールド表示テキスト • F9203、データ項目名称記述 • F9207、データ辞書 - エラー・メッセージ情報 • F9210、データ・フィールド・スペック(ERP 9.0)
DDINDEX: B73.2 で使用中止 - 辞書用インデックス	
DDTEXT: データ辞書テキスト	
NEXTID	ネクスト ID マスター・テーブル(F98701)。各ワークステーションに割り当てられたネクスト ID のローカル・レコード。
GLBLTBL	データ辞書およびテーブル・スペックからのキャッシュ情報。ランタイム・テーブルおよび一時変更情報を格納します。最初に動的に作成されたテーブルが使用されます。
SMRTTMPL	フィールド情報に必要なデータ構造体。

オブジェクトのバックアップおよび復元

ワークステーションおよびサーバー上の開発オブジェクトは、必要に応じてバックアップできます。

次のバックアップを行うことをお勧めします。

- 開発者ワークステーションのバックアップ: サーバー容量の制限により、開発者は ERP 9.0 ディレクトリをサーバーにバックアップしません。代わりに、処理するオブジェクトを 8 時間ごと、または再作業しなくてよい頻度でチェックインします。

開発者がファイル・サーバーに全パス・コード・ディレクトリをバックアップするディスク領域が確保できない場合、チェックイン・プロセスをバックアップ方法として使用してください。推奨開発プロセスを実行すると、開発者は、未完成または動作不良のアプリケーションを DEV パス・コードにチェックインできることがわかります。

- エンドユーザー・ワークステーションのバックアップ: エンドユーザーの場合、トランザクション・テーブルにアップロードされていないオフライン処理されたトランザクション以外のレプリケートしていないデータをマシン上に保管しないでください。その日の作業を終了する前にすべてのオフライン処理ユーザーがトランザクションをアップロードするようにしてください。このアップロードを実行できない場合は、ローカル・データベースをファイル・サーバーのサブディレクトリにバックアップする必要があります。サブディレクトリの名前はユーザーの名前にします。
- 開発サーバーのバックアップ: 弊社の IT 部門は、開発ファイル・サーバー(通常、デプロイメント・サーバー)と共に必要なデータベース(セントラル・オブジェクト、オブジェクト・ライブラリアンおよびデータ辞書)のバックアップをとります。開発者が特定のオブジェクトをバックアップから復元する必要がある場合、データベース管理者は「Restore(復元)」と呼ばれるパス・コードにエクスポートを復元します。その後、開発者はオブジェクトを Restore からチェックアウトし、オブジェクトが正常に動作することを確認してから、標準開発パス・コードにチェックインします。
- デプロイメント・サーバーのバックアップ: 多くの場合、サーバー全体を日次バックアップする必要はありません。ただし、状況によっては、次のディレクトリを日次バックアップする必要がある場合があります。
 - DEV パス・コード。オブジェクトを修正、新規パッケージをビルド、またはワークステーションのインストール時にデータベースを更新した場合
 - ヘルプ。ヘルプ・ファイルを修正した場合
 - MEDIA OBJ。デプロイメント・サーバーにメディア・オブジェクトが存在する場合。システム・データやその他の重要なデータがデプロイメント・サーバーに保管されている場合、Oracle サーバーまたは SQL サーバー内の ERP 9.0 データ・ソースは、日次バックアップをする必要があります。
- エンタープライズ・サーバーのバックアップ:
 - DBMS を日次バックアップする。データベース・ベンダー提供のバックアップ・ツールを使用することをお勧めします。
 - ERP 9.0 ディレクトリ全体をバックアップして、ソフトウェアの全オブジェクトのバックアップをとります。また、PROD および DEV パス・コードと JDE.INI ファイルもバックアップします。バージョン制御管理者が、〈Server Package(サーバー・パッケージ)〉アプリケーションへのアクセス権限をもつ開発者、およびサーバー上で実行される新規パッチ・バージョンを作成するエンドユーザーによって修正されたオブジェクトを配布すると、パス・コードは更新されます。

参照

- バックアップおよび復元については、『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

データ・ソース間でのレコードのコピー

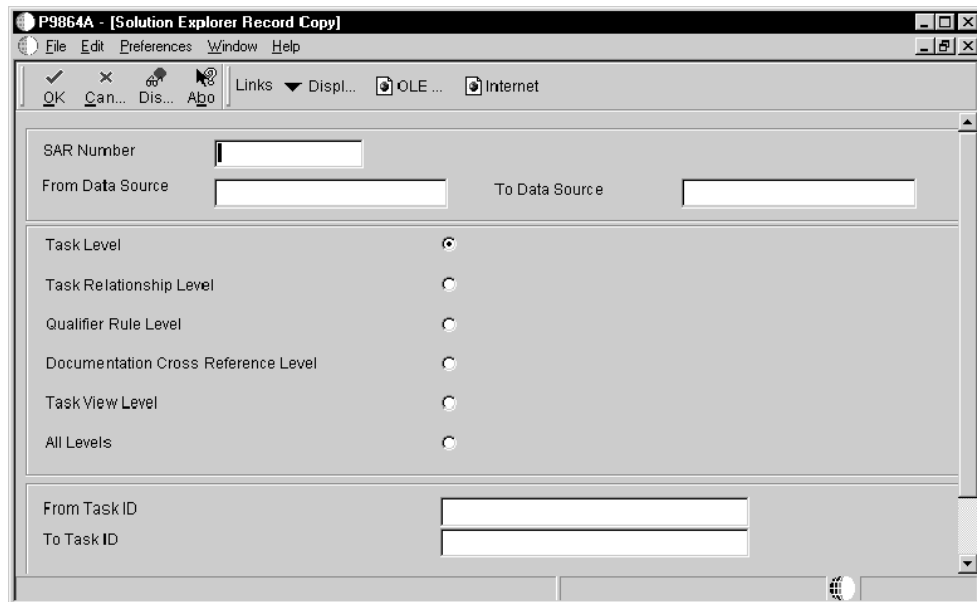
ERP 9.0 では、〈Object Management Workbench〉プログラム(P98220)を使用して、コントロール・テーブル・レコードを 1 つのデータ・ソースから別のデータ・ソースにコピーします。ただし、Solution Explorer レコードをコピーするには、〈Solution Explorer Record Copy (Solution Explorer レコードのコピー)〉プログラム(P9864A)を使用する必要があります。例えば、本稼働環境や CRP 環境用に別の 1 組の UDC、メニュー、データ辞書などが必要になることがあります。そのような場合は、〈Solution Explorer Record Copy〉プログラムを使用して、CRP 環境で行った変更をそれぞれ本稼働環境にコピーできます。

Solution Explorer のコントロール・テーブル・レコードは、データ・ソース間でコピーできます。コントロール・テーブル・レコードは一括してコピーすることもできれば、次のレコードを個別にコピーすることもできます。

- タスク
- タスク・リレーションシップ
- 修飾子ルール
- ドキュメンテーション・インデックス
- タスク・ビュー

▶ データ・ソース間でレコードをコピーするには

〈Content Management〉タスク・ビューから〈Solution Explorer Record Copy〉プログラム(P9864A)を選択します。



1. 〈Solution Explorer Record Copy〉フォームで、次のフィールドに値を入力します。

- SAR 番号

移動するタスク・グループの SAR 番号を入力します。タスクが SAR に関連付けられていない場合は、このフィールドに任意の値を入力します。

- データ・ソース(From)

レコードのコピー元データ・ソースを入力します。

- データ・ソース(To)

レコードのコピー先データ・ソースを入力します。

2. 次のオプションのいずれかをクリックし、フォーム最下部に表示される関連フィールドに入力します。

- タスク・レベル

1 つのタスクまたは一定範囲のタスクを転送する場合は、[Task Level(タスク・レベル)]を選択します。ERP 9.0 は、[From Task ID]から[To Task ID]までのすべてのタスクを転送します。単一のタスクを転送する場合は、[From Task ID]フィールドと[To Task ID]フィールドの両方にそのタスク ID を入力します。

[Task Level]から[Solution Explorer]タスクをコピーすると、レコードはすべてのタスク・リレーションシップおよびタスク・ビューにコピーされ、ソース・データ・ソース内に表示されます。

- タスク・リレーションシップ・レベル

親タスク(タスク・ノード)に関連付けられたタスクを転送する場合は、[Task Relationship Level(タスク・リレーション・レベル)]を選択します。このレベルのタスクを転送するには、次のフィールドに値を入力します。

- Parent Task - 転送するノードまたはタスクの親タスクを入力します。
- Child Task - 子タスクを入力する。子タスクがノードの場合、ERP 9.0 はノードとその子すべてを転送します。

c. Task View - このタスク・リレーションシップをコピーしたいタスク・ビューを入力する。このタスク・リレーションシップが別のタスク・ビューに存在している場合でも、指定したタスク・ビューだけにタスク・リレーションシップがコピーされます。

- 修飾子ルール・レベル

1 つまたは複数の修飾子ルールを転送する場合は、[Qualifier Rule Level(修飾子ルール・レベル)]を選択します。ERP 9.0 は、指定された範囲内のすべての修飾子ルールを転送します。

- ドキュメンテーション相互参照レベル

タスク ID に関連付けられたドキュメンテーション・インデックス・レコードを転送する場合は、[Documentation Cross Reference Level(ドキュメンテーション相互参照レベル)]を選択します。一定範囲のドキュメンテーション・インデックスを転送するには、タスク ID の範囲を入力します。指定した範囲内のすべてのインデックスが転送されます。

- タスク・ビュー・レベル

タスク・ビューに関連付けられたすべてのタスク、タスク・リレーションシップ、修飾子ルール、およびドキュメンテーション・インデックスを転送する場合は、[Task View Level(タスク・ビュー・レベル)]を選択します。タスク・ビューを指定してこれらのレコードを転送すると、指定したビュー内のタスクのみ表示されます。

- すべてのレベル

すべての[Solution Explorer]タスク、タスク・リレーションシップ、修飾子ルール、ドキュメンテーション・インデックス、およびバリエーションを転送する場合は、[All Level(全レベル)]を選択します。ソース・データ・ソースから新しいレコードをコピーする前に、ターゲット・データ・ソースの内容を削除したい場合は、[Clear To Data Source]をチェックしてください。

パッケージ・ビルド

アプリケーション、ビジネス関数、およびその他のオブジェクトの変更に伴って、社内のユーザーがその変更を利用できるようにすることが必要です。また、このソフトウェアを搭載した新規ワークステーションのセットアップが必要になる場合もあります。

パッケージを使用すると、ソフトウェアの変更および新規アプリケーションをユーザーに配布したり、ワークステーション上に ERP 9.0 を新規インストールできます。パッケージを定義して作成した後は、〈Client Workstation Installation (クライアント・ワークステーション・インストール)〉プログラムまたは〈Package Deployment (パッケージ・デプロイメント)〉プログラム(P9631)を使用して配布できます。

パッケージおよびデプロイメントの理解

ワークステーションやエンタープライズ・サーバー、ロジック・サーバー、アプリケーション・サーバーなどは、次のようなさまざまな理由によって、更新やセットアップが必要になる場合があります。

- 新規ユーザーやグループのためにワークステーションをセットアップします。
- すべてのユーザーまたは選択したユーザーだけにカスタム・ソリューションを配布する必要があります。
- 開発目的で作成した新規パス・コードを配布する必要があります。
- ソフトウェアの修正を適用する必要があるユーザーで構成されたグループに迅速に配布する必要があります。
- 一部のワークステーションでディスクの空き領域が少なくなり、ERP 9.0 の最小構成を作成する必要があります(この状況を避けるために、すべての本稼働用ユーザーが部分パッケージを使用することをお勧めします)。
- ERP 9.0 ツールセットによって開発したカスタム修正を使用して、サーバーをアップデートする必要があります。

これらのニーズをすべて満たすソリューションが提供されています。まず、ワークステーションに配布する必要があるコンポーネントのロケーションを示すパッケージを作成できます。これらの要素を配布するには、パッケージを定義および作成する必要があります。

3 つのパッケージ・タイプ、つまり、フル、部分、およびアップデートのいずれかを自由に選択できます。部分パッケージはワークステーション専用です。選択するパッケージ・タイプは、ニーズによって異なります。

パッケージを定義およびビルドしたら、配布の準備が整います。パッケージ・タイプに応じて、〈Client Workstation Installation〉プログラムや〈Package Deployment〉プログラム(P9631)を使用してパッケージを配布できます。

〈Package Deployment〉を使用すると、パッケージを受け取るワークステーションとサーバー、およびパッケージがいつから使用できるかを指定できます。パッケージは企業内のすべてのマシン、特定のマシン・グループ、または個々のマシンに配布できます。また、デプロイメント・サーバーからワークステーションにプッシュされるパッケージをスケジュールできます。プッシュ・インストールでは、ワークステーション・ユーザーの操作は不要です。

フル・ワークステーション構成および部分ワークステーション構成の理解

パッケージの組立ておよびビルドを開始する前に、ERP 9.0 で使用できる 2 種類のワークステーション構成(フルおよび部分)を理解しておく必要があります。これらの構成は、主にパッケージをワークステーションにインストールするのに必要なディスク領域に関係しています。これら 2 種類の構成を理解しておく、社内のワークステーションに適したパッケージ・タイプ(フル、または部分)を確定する上で役立ちます。

フル・ワークステーション構成では、大量のディスク領域が必要なため、インストールにかかる時間が長くなります。通常のフル・ワークステーション・インストールでは、1.4GB を超えるディスク領域が必要であり、ネットワーク・トラフィックに応じてインストールに 10～30 分を要します。

部分ワークステーション構成は ERP 9.0 の最小構成を意味し、フル・ワークステーション構成と比べて、ディスク領域をごくわずかし使用しません。また、部分ワークステーション構成では、ERP 9.0 のインストール時間も非常に短くて済みます。通常の部分ワークステーション構成は約 165MB のディスク領域を使用し、ネットワーク・トラフィックに応じてインストールに 3～10 分かかります。ただし、ERP 9.0 は、ユーザーの必要に応じてアプリケーションを最小構成に追加するので、WAN を介して部分ワークステーション構成にアクセスする場合は、パフォーマンスが実用限度以下に低下することがあります。

フル・ワークステーション構成と部分ワークステーション構成の主な違いは、ワークステーションに存在するスペックの量です。フル・ワークステーションには、ERP 9.0 アプリケーションの完全パッケージが含まれ、それらの中には、ユーザーによって使用されないアプリケーションも存在します。フル・ワークステーションの利点は、すべてのアプリケーションがすぐに使用できることです。逆に欠点は、ワークステーションで大量のディスク領域を使用することです。

部分パッケージは保守管理が非常に容易なので、本稼働用ユーザーは、部分パッケージをインストールすることをお勧めします。本稼働用ユーザーは、ジャストインタイム・インストレーションのパフォーマンスがネットワーク構成で実用に耐えない場合にのみ、フル・パッケージをインストールしてください。

また、フル・パッケージには、部分パッケージにはない開発オブジェクトが含まれているため、開発ユーザーは、フル・パッケージをインストールすることをお勧めします。

フル・ワークステーション構成にも、部分ワークステーション構成にもそれぞれ利点があるので、企業にとってどちらかが最適な構成であるかを確定するには、自社のニーズを慎重に考慮する必要があります。フル・ワークステーション構成と部分ワークステーション構成の利点と欠点を次の表に示します。

フル・ワークステーション 利点:

- ERP 9.0 アプリケーションのフル・パッケージ(すべてのスペックおよびファンデーション)が含まれている。開発者は、ERP 9.0 のすべてのオブジェクトにローカル・アクセスできます。
- すべてのアプリケーションが同時にインストールされているので、初期クライアント・ワークステーション・インストール後のネットワーク・トラフィックが、部分ワークステーション構成の場合よりも少なくなります。

欠点:

- 部分ワークステーション構成よりも多くのディスク領域がワークステーションで必要です。
- 部分ワークステーション構成と比べて、初期インストールに時間が長くなります。

部分ワークステーション 利点:

- フル・ワークステーション構成と比べて、ワークステーションに必要なディスク領域が少なく済みます。
- 部分ワークステーションの場合、初期インストール時間が非常に短くなります。
- クライアント・ワークステーションへのインストール時に、ERP 9.0 の起動に必要なオブジェクトだけが配布される。他のすべてのアプリケーションはユーザーが最初に使用する時点でジャストインタイム・インストールされます。

欠点:

- 最初のユーザーがアプリケーションをロードする際に、ネットワーク・トラフィックがわずかに増加します。
- WAN によるジャストインタイム・インストレーションのパフォーマンスが LAN の場合よりも遅くなる可能性があります。
- 開発オブジェクトが配布されないため、開発者は部分パッケージを使用できません。

ジャストインタイム・インストレーションの理解

部分ワークステーションには常駐のアプリケーションがありません。代わりに、実行時にアプリケーションが取り込まれ、メイン・メニューからそのアプリケーションを最初に選択すると、ワークステーションにロードされます。ロードは一度だけ行われます。次に同じアプリケーションを選択した場合、そのアプリケーションはすでにマシンにロードされています。このプロセスは、「JITI(ジャストインタイム・インストール)」といいます。JITI はアプリケーションにのみ適用されます。JITI を介してビジネス関数をインストールすることはできません。

JITI は、アプリケーションがインストールされていないときに機能し、またワークステーションがスペックを含まない部分パッケージやアップデート・パッケージを受け取ったときに機能します。スペックを含むアップデート・パッケージおよび部分パッケージでは、JITI プロセスを必要としません。

メイン・メニューからオプションを選択すると、ランタイム・エンジンが、アプリケーション・スペックがワークステーションに存在するかどうかを確認します。ローカル・スペックが存在せず、〈Environment Master(環境設定マスター)〉プログラム(P0094)でジャストインタイム・インストレーションのフラグが「Y」に設定され、さらにセキュリティ・プロファイルにより、アプリケーションのインストールが可能な場合は、環境に関連付けられているパス・コードのセントラル・オブジェクト・データ・ソースからスペックが転送されます。このようにして、必要なアプリケーションは、ユーザーがパッケージを受け取った時や初期インストール時にインストールされるのではなく、「ジャストインタイム」でインストールされます。

アップデート・パッケージのインストール・プロセスもほぼ同じです。新しいスペックが添付されていない、変更されたアプリケーションを含むアップデート・パッケージを受け取ると、ERP 9.0 は、最初にそのアプリケーションのスペックがワークステーションに存在するかどうかを確認します。スペックが存在する場合、そのアプリケーションの旧バージョンがワークステーションから削除されます。次回、メイン・メニューでそのアプリケーションを選択すると、そのアプリケーションの新しいバージョンがロードされます。

パッケージにジャストインタイム・アプリケーションが含まれる場合、実行時に次の関連オブジェクトだけが配布されます。

- 対話型アプリケーション・スペックまたはバッチ・アプリケーション・スペック
- アプリケーションの埋込みイベント・ルール
- 処理オプション・テンプレート/データ構造体および関連ビジネス・ビュー

次のオブジェクトは配布されないため、それらのオブジェクトを修正した場合は、パッケージに含める必要があります。

- ビジネス関数およびそれらのデータ構造体
- 汎用テキストデータ構造体
- テーブルに添付されているテーブル・イベント・ルール
- イベント・ルール・ビジネス関数
- 新規アイコン

パッケージ・タイプの理解

フル・ワークステーション構成または部分ワークステーション構成にすることを確定した後、次のパッケージ・タイプを作成できます。

- フル・パッケージ
- 部分パッケージ
- アップデート

フル・パッケージ

フル・パッケージは、パッケージが基づくバス・コード用のセントラル・オブジェクトの、ある時点の静的なスナップショットです。フル・パッケージには、開発者が、ERP 9.0 で開発作業を行うのに必要な項目がすべて含まれています。具体的には、TAM スペック・ファイルの完全セット、DLL の完全セット、ビジネス関数とテーブル用のすべての.c ファイルと.h ファイル、そしてファンデーション、データ、ヘルプおよびパッケージの配置場所を定義する INF ファイルが含まれています。フル・ワークステーション構成を作成する場合に、このパッケージ・タイプを選択します。

フル・パッケージの主要な利点は、ライセンスを取得している ERP 9.0 のすべてのアプリケーションをユーザーが使用できることです。スペックがワークステーションに常駐するため、情報はローカルに処理され、ネットワーク・トラフィックが解消されます。これに対し部分パッケージの場合は、初回使用時にジャストインタイム・インストールを行ってオブジェクトをダウンロードする必要があります。フル・パッケージを配布するときは、いくつかのスペックとテーブルがジャストインタイムでインストールされることがありますが、ネットワーク・パフォーマンスへの影響はほとんどありません。

フル・パッケージは、主に ERP 9.0 の初期インストール用であり、通常は、〈Client Workstation Installation〉プログラムによって配布されます。ERP 9.0 がすでにインストールされているコンピュータにフル・パッケージをインストールする場合は、〈Package Deployment〉プログラム(P9631)を使用することもできます。

フル・パッケージには、開発オブジェクト(ビジネス関数ソース、オブジェクト、および見出しファイル)を含めることもできます。開発オブジェクトをロードする必要があるユーザーには、そのためのオプションが配布時に提供されます。

部分パッケージ

このパッケージ・タイプは、ERP 9.0 の最小構成です。部分ワークステーション構成を作成する場合には、このパッケージ・タイプを使用します。部分パッケージには、基本的にユーザーが ERP 9.0 エクスプローラを起動できるスペックだけが含まれます。特に、部分パッケージの TAM ファイルのセットは最初は空であり、その後、パッケージに含まれているオブジェクトのスペックがシステムによって追加されます。また、部分パッケージには、DLL の完全セットと、データ、ヘルプ・ファイル、およびパッケージ・オブジェクトの配置場所を定義する INF ファイルも含まれています。

ユーザーがメイン・メニューからアプリケーションを最初に選択すると、アプリケーションは、ジャストインタイム・インストールによってロードされます。つまり、選択されたアプリケーションは、そのパス・コードのセントラル・オブジェクト・データ・ソースからロードされます。

フル・パッケージと同様に、部分パッケージは主に初期インストール用であり、通常は〈Client Workstation Installation〉アプリケーションを介して配布されますが、ソフトウェアが既にインストールされているコンピュータに部分パッケージをインストールする場合は、〈Package Deployment〉を使用することもできます。

アップデート・パッケージ

アップデート・パッケージでは、データベース、アプリケーション・オブジェクト、ヘルプ・ファイルなどのオブジェクトが変更された既存のフル・パッケージや部分パッケージをアップデート、追加、またはリフレッシュできます。このパッケージ・タイプは、ソフトウェアの変更と修正を迅速に配布するのに適しています。

パッケージにジャストインタイム・アプリケーションが含まれていない場合、アップデート・パッケージは特定パス・コードのセントラル・オブジェクトのある時点でのコピーです。アップデート・パッケージにジャストインタイム・アプリケーションが含まれている場合は、そのジャストインタイム・アプリケーションだけが動的であり、パッケージのその他のオブジェクトは静的です。動的とは、ユーザーがアプリケーションを最初に選択したときに、アプリケーションがセントラル・オブジェクト・データ・ソースから直接コピーされることを意味します。

ユーザーがアップデート・パッケージを受け取ってから ERP 9.0 にサインオンすると、パッケージ内のすべてのオブジェクトがユーザーのワークステーションにロードされます。アップデート・パッケージ内のオブジェクトは、ワークステーション上の同じオブジェクトを置換します。ワークステーション上の他のオブジェクトはいずれも変更されません。

パッケージ内のアプリケーションは、〈Security Workbench (セキュリティ・ワークベンチ)〉によって保護できます。〈Security Workbench〉は、ユーザーがアプリケーションのインストールを許可されているかどうかを確認します。許可されているユーザーだけがアプリケーションをロードできます。そのため、アップデート・パッケージには、複数の異なるユーザーやグループに適用されるさまざまなアプリケーションを含めることができますが、それらのアプリケーションは、許可されたユーザーまたはグループしか受け取ることができません。

フル・パッケージと同様に、アップデート・パッケージには開発オブジェクト(ビジネス関数ソース、オブジェクト、およびヘッダー・ファイル)を含めることができます。アップデート・パッケージの受信者には、配布時に開発オブジェクトをロードするためのオプションが提供されます。

ジャストインタイム・インストレーションの仕組みを考えると、部分パッケージまたはアップデート・パッケージにスペックのないアプリケーションだけが含まれている場合、WAN のパフォーマンスが遅くなる可能性があります。WAN のパフォーマンスを改善するには、各アプリケーションに対応するスペックをパッケージに含めます。

すべてのアップデート・パッケージにはそれに対応する「親パッケージ」が必要です。親パッケージとは、次のアップデート・パッケージが更新されるフル・パッケージまたは部分パッケージです。アップデート・パッケージ内のすべてのオブジェクトは親パッケージにマージされます。

アップデート・パッケージ内のビジネス関数オブジェクトは、親パッケージ内の対応するオブジェクトにリンクされ、新しい DLL が作成されます。また、アップデート・パッケージに含まれるスペックも、親パッケージ内のスペックにマージされます。

親パッケージの概念はワークステーションおよびサーバーに適用されます。ワークステーションの親パッケージは、デプロイメント・サーバーに常駐し、サーバーの親パッケージは、エンタープライズ・サーバーのビルド領域で保管されます。

機能の理解

パッケージには、ERP 9.0 オブジェクトのほかに、機能も追加できます。機能とは、アプリケーションやその他の機能をサポートするためにワークステーションやサーバーにコピーする必要があるファイルや構成オプションのセットをいいます。オブジェクトと同様に、機能はパッケージに含まれ、機能コンポーネントを必要とするワークステーションとサーバーに配布されます。

たとえばパッケージには、ActiveX コントロール、セールス・フォース・オートメーション機能用のサポート対象ローカル・データベース、オープン・データ・アクセスで使用する ODBC データ・ソース、Microsoft Windows レジストリ設定などを追加できます。

機能は、〈Package Deployment〉プログラム(P9631)を使用して定義します。そして機能をパッケージに追加するには、〈Package Assembly (パッケージ・アセンブリ)〉プログラム(P9601)と〈Package Build (パッケージ・ビルド)〉プログラム(P9621)を使用します。

パッケージの作成およびデプロイメントの概要

次に、パッケージを作成および配布する際のステップの概要を示します。

1. パッケージの組立て

このステップでは、ビルドするパッケージ・タイプを指定し、名前、パス・コード、およびパッケージ記述を入力します。次に、パッケージに含めるオブジェクトを指定して、パッケージを組み立てます。部分パッケージまたはアップデート・パッケージをビルドする場合、含めるオブジェクトを個別に指定することができます。

パッケージの組立てプロセスを簡素化するために、〈Package Assembly〉プログラム(P9601)には、〈Package Assembly Director (パッケージ・アセンブリ・ディレクタ)〉が組み込まれています。この〈Package Assembly Director〉は、パッケージに名前を設定し、パッケージに含めるオブジェクトを組み立てるステップを先導する一連のフォームを表示します。

2. パッケージ・ビルドの定義

パッケージを組み立てた後、パッケージをワークステーションおよびサーバーに配布する前にビルドを定義する必要があります。このステップでは、次の項目を指定します。

- ビルド・オプション
- ビルド・スペック・オプション
- ビジネス関数ビルド・オプション

- 圧縮オプション
- ビルド機能オプション

また、パッケージの対象をワークステーションにするか、サーバーにするか、またはこれら両方にするかを指定する必要もあります。パッケージがサーバー用の場合は、パッケージを受信するサーバーを指定する必要があります。

繰り返しますが、ビルドのプロセスを簡素化するために、〈Package Build〉プログラム(P9621)には〈Package Build Definition Director(パッケージ・ビルド定義ディレクタ)〉が組み込まれています。この〈Package Build Definition Director〉は、パッケージをビルドする場所、スペックを含めるかどうか、ビジネス関数を圧縮するか、ビルドするかなどを指定するステップを先導する一連のフォームを表示します。

3. パッケージ・ビルド

実際のビルド・プロセスでは、パッケージを組み立てて定義したときに指定した情報が取り込まれ、セントラル・オブジェクトのパッケージへのコピーと移行が行われます。また、パッケージ内に含まれるビジネス関数のグローバル・ビルドを実行し、パッケージを圧縮します。

4. ワークステーションに配布するパッケージのスケジュール

アップデート・パッケージをビルドする場合(または、フル・パッケージあるいは部分パッケージを、ERP 9.0 がインストールされているワークステーションに再配布する場合)、パッケージを配布する日付および時刻を指定する必要があります。パッケージをスケジュールするときは、パッケージのインストールを必須にするか、任意にするかを指示できます。

同時に、パッケージを受け取るユーザーの操作が不要なプッシュ・インストレーションを通じて、パッケージを配布するかどうかを指定することもできます。

デプロイメント・サーバーおよびエンタープライズ・サーバーへのパッケージの配布

変更済みオブジェクトをエンタープライズ・サーバーに移動するには、〈Package Deployment〉プログラム(P9631)を使用します。

パッケージ・ビルドの定義プロセスでサーバーを指定すると、対応するサーバー・パッケージが適切なフォーマットで自動的に作成されます。サーバーを指定しないで、ワークステーション・パッケージのみを定義した場合、対応するサーバー・パッケージを作成する必要があります。このプロセスはワークステーション・パッケージの作成とほとんど同じです。

参照

- 機能の追加については「パッケージへの機能の追加」
- パッケージの組立方法については「パッケージの組立て」
- パッケージ・ビルドを定義する方法については、「パッケージ・ビルドの定義」
- プッシュ・インストレーションについては「プッシュ・インストレーションの使用」
- パッケージの配布については「サーバー・パッケージの配布」

フル・パッケージのビルド方法

次に、フル・パッケージをビルドする方法の概要を示します。

1. パッケージ・ビルド・ディレクトリを作成します。
2. INF ファイルを作成します。
3. 次のディレクトリおよびファイルをチェックイン・ロケーションからパッケージ名ディレクトリにコピーします。
 - res
 - source(.c ファイル)
 - include(.h ファイル)
 - work
 - make
 - bin32
 - lib32
 - object(.obj ファイル)

注:

パッケージ・ビルドと共にビジネス関数をビルドする場合、bin32、lib32 および object(.obj)ファイルはコピーされません。これらのファイルは、〈BusBuild(自動ビジネス関数ビルダー)〉プログラムが作成します。

4. リレーショナル・データベース内の情報に基づいて、TAM(テーブル・アクセス管理)スペック・ファイルをビルドします。
5. 〈BusBuild〉プログラムを実行し、ビジネス関数をコンパイル、リンクして、bin32 ディレクトリに DLL、obj ディレクトリにオブジェクト、lib32 ディレクトリにライブラリを作成します
6. ディレクトリを圧縮します。

部分パッケージのビルドする方法

次に、部分パッケージをビルドする方法の概要を示します。

1. パッケージ・ビルド・ディレクトリを作成します。
2. INF ファイルを作成します。
3. 一連の空の TAM スペック・ファイルをビルドします。
4. 次のディレクトリおよびファイルをチェックイン・ロケーションからパッケージ名ディレクトリにコピーします。
 - res
 - work
 - make

- bin32
- lib32

注:

パッケージ・ビルドと共にビジネス関数をビルドする場合、bin32 ファイルと lib32 ファイルはコピーされません。これらのファイルは、〈BusBuild〉プログラムが作成します。

5. *LITE レコード内のオブジェクトのスペックをビルドします。
6. 〈BusBuild〉プログラムを実行し、ビジネス関数をコンパイル、リンクして、bin32 ディレクトリに DLL、obj ディレクトリにオブジェクト、lib32 ディレクトリにライブラリを作成します(任意)。

注:

部分パッケージのビルド時に〈BusBuild〉プログラムを実行しない場合は、この機能を別途実行する必要があります。

7. ディレクトリを圧縮します(任意)。

アップデート・パッケージのビルド方法

次に、アップデート・パッケージをビルドする方法の概要を示します。

1. パッケージ・ビルド・ディレクトリを作成します。
2. INF ファイルを作成します。
3. ソフトウェア・パッケージ詳細テーブル(F9631)内の各オブジェクトに関する情報をリレーショナル・データベースから取り出し、TAM スペック・ファイルに追加します。
4. 〈BusBuild〉プログラムを実行し、bin32 ディレクトリの DLL、obj ディレクトリのオブジェクト、および lib32 ディレクトリのライブラリを更新します。

サーバー・パッケージの理解

アプリケーション開発はすべてワークステーション上で行われます。開発オブジェクト自体は 1 台のデプロイメント・サーバー上に保管され、オブジェクト管理ワークベンチによって管理されます。エンタープライズ・サーバーで実行されるビジネス・アプリケーションもあります。ワークステーション上で開発される修正および機能拡張がサーバーに反映されるようにするために、これらの修正と機能拡張を含むサーバー・パッケージをビルドする必要があります。

ワークステーション・パッケージの作成に使用するのと同じ〈Package Assembly〉プログラム(P9601)および〈Package Build〉プログラム(P9621)を使用して、セントラル・オブジェクト・ロケーションからエンタープライズ・サーバーにオブジェクトをプッシュするサーバー・パッケージの組立て、定義およびビルドを実行できます。サーバー・パッケージは、スペック・レコード、ソース・ファイルおよびヘッダー・ファイルのグループです。コンパイル済みオブジェクトはエンタープライズ・サーバー上で作成されます。サーバー・パッケージの定義とビルドを終了したら、〈Package Deployment〉プログラム(P9631)を使用して、エンタープライズ・サーバー、ロジック・サーバーまたはアプリケーション・サーバーにインストールできます。

開発環境では、オブジェクトを作成または修正し、そのオブジェクトをエンタープライズ・サーバーに転送する必要があるときに、開発者は常にサーバー・パッケージを作成できます。本稼働用環境では、システム管理者はサーバー・パッケージを作成する必要があります。

サーバー・パッケージは、次の特徴を除いて、クライアント・ワークステーション・パッケージと本質的に同じものです。

- サーバー・パッケージには、ヘルプ情報とサポート対象ローカル・データベースが含まれません。
- 通常、ファンデーション・コードはサーバー・パッケージの一部としては配布されません。
- フォーム設計で使用される一部の TAM スペック(対話型エンジン・スペック)はサーバーで使用されないため、サーバー・パッケージに含まれません。
- 一部のビジネス関数がサーバーでビルドされないため、サーバー・パッケージに含まれません。
- サーバーに対する部分パッケージが使用できません。

次のような場合には、ワークステーション・パッケージに対応する別個のサーバー・パッケージを作成する必要があります。

- ビジネス関数がプラットフォーム全体で互換でない場合。たとえば、Windows NT ワークステーションでコンパイルされたビジネス関数は、UNIX サーバー上では実行できません。また、すべてのビジネス関数がサーバー上でコンパイルできるわけではありません。
- データ配置方法がプラットフォームによって異なる場合。ワークステーションでビルドされた TAM ファイルは、変換しないとサーバーで実行できません。また、整数型(数の認識および識別方法)がサーバーとワークステーションでは異なります。
- ASCII から EBCDIC への変換を AS/400 のために実行する必要がある場合。

さらに、サーバー・プラットフォームは、必ずしも互換であるとは限りません。例えば、各エンタープライズ・サーバー・プラットフォームには、プラットフォームに互換性のない独自スペックの内部ストレージがあります。また、各エンタープライズ・サーバー・プラットフォームでは異なるコンパイラが使用されており、オブジェクトが異なります。したがって、ビジネス関数はプラットフォームと互換性はありません。

サーバー・パッケージ・ビルド・プロセスの理解

サーバー・パッケージの作成は、クライアント・ワークステーション・パッケージの作成と変わりませんが、作成目的が異なります。サーバー・パッケージをビルドするのは、ERP 9.0 管理者が TAM スペックとビジネス関数をエンタープライズ・サーバー上でビルドできるようにすることが主要な目的です。

サーバー・パッケージ・ビルドをビルドする上で、〈Package Build〉プログラム(P9621)には次の利点があります。

- クライアント・ワークステーション・パッケージとの完全な整合性を提供します。
- 管理者が複数のサーバー上で同時にパッケージをビルドできます。
- 管理者がサーバー上で同時に別々のパッケージ・コンポーネントをビルドできます。
- ある 1 台のエンタープライズ・サーバー上でパッケージをビルドし、同じタイプの別のサーバーに配布できます。

- クライアント・ワークステーションからモニタリングできる履歴レコードを作成するため、管理者が、どのパッケージがビルドされたかを確定できます。
- 圧縮ファイルを作成し、デプロイメント・サーバーにそのファイルをロードして、マスターCDの作成をより容易にします。
- フル・パッケージおよびアップデート・パッケージがサポートされています。
- パッケージが適切に作成されなかった場合に備えて、再スタート機能を提供します。

エンタープライズ・サーバーには次のオブジェクトをインストールできます。

- ビジネス関数
- ビジネス・ビュー
- データ構造体
- テーブル（インストール時、データベース内にテーブルは作成されない。スペックおよびテーブル・ヘッダー・ファイルがサーバーにプッシュされるだけです）
- バッチ・アプリケーション・スペック（テンプレートおよびバージョン）
- アプリケーション・スペック・レコード

サーバー・パッケージは、クライアント・ワークステーション・パッケージと同じ方法で組み立てられ、定義されます。そのため、クライアント・ワークステーション・パッケージを組み立ててビルドすると同時に、〈Package Assembly〉プログラム(P9601)を使用してサーバー・パッケージを組み立て、〈Package Build〉プログラム(P9621)を使用してビルドすることができます。

サーバー・パッケージを組み立てて、パッケージ・ビルドを定義した後、次の処理が行われます。

1. ビルド・プロセスが開始されると、パック形式 TAM ファイルが作成されます。
2. ローカル・サーバーでサーバー・パッケージのバッチ・アプリケーションを投入します。
3. このバッチ・アプリケーションはビジネス関数を呼び出し、続いて、サーバー・パッケージ・エンジンが呼び出されます。
4. ビルド・エンジンは、〈Package Assembly〉プログラムと〈Package Build〉プログラムによって作成されたレコードを使用して、次の操作を実行します。
 - 各サーバー上のディレクトリを初期化します。
 - パック形式 TAM ファイルの転送を開始する。各パック形式ファイルが転送されると、サーバー上でアンパック・プロセスが開始されます。
 - すべてのビジネス関数ソースおよびヘッダー・ファイルをサーバーに転送する。この時点で、ビルド・エンジンは、オブジェクト・ライブラリアン・マスター・テーブル(F9860)を読み取って、各モジュールが属する DLL を確定します。JDBTRIG ライブラリの場合、モジュールが所属するトリガー・ライブラリを指示するために特殊機能が呼び出されます。この場合、オブジェクト・ライブラリアン・マスター・テーブルは使用されません。
 - すべてのビジネス関数のソース・ファイルが転送された時点で、サーバー上でビルド・マスター・プロセスを開始する。このビルド・マスターは個々のビルド・プロセスを同時に開始します。各 DLL には独自のビルド・プロセスがあります。JDE.INI ファイルは、同時に実行できるプロセス数を示します。
 - サーバーがパッケージ・ビルド・プロセス実行中に指定された場合、プロセスを別のサーバーに移動する。このプロセスでは、そのサーバー上のすべてのコンポーネントが転送およびビルドされます。

- ビルド・プロセスがすべてのサーバー上で開始された後、各サーバー上の各ビルド項目の状況を確認する。状況が変化すると、履歴レコードが更新されます。
- ビルドが完了すると、パッケージ・コンポーネントを圧縮し、圧縮ファイルをデプロイメント・サーバーに配置する。この操作は、パッケージのビルド時にファイルを圧縮するように指定している場合にのみ行われます。このプロセスはすべてのサーバーで繰り返されます。

サーバー・パッケージ・ビルド用 JDE.INI 設定

サーバー・パッケージにビジネス関数が含まれている場合、JDE.INI ファイルの BSFN BUILD セクションがパッケージに適用されます。次に、一般的な UNIX ベース・サーバーのエントリ例を示します。

```
[BSFN BUILD]
BuildArea=/u17/i5745669/b73.3/appdev/packages
DebugFlags=-g -y D_DEBUG -DJDEDEBUG
InliningFlags=
DefineFlags=-DKERNEL -DPRODUCTION_VERSION -DNATURAL_ALIGNMENT
CompilerFlags=-Aa +wl +z -c
OSReleaseLevel=+DAportable
LinkFlags=-b -z
LinkLibraries=
SimultaneousBuilds=0
DoCompression=1
```

設定	値	目的
Build Area=	/usr/jdedwardsoneworld/b9/packages	パッケージがビルドされるサーバー上のロケーションを示します。
Optimization Flags=	+O2 (HP 9000 用のデフォルト) -O2 (RS/6000 および Sun 用のデフォルト値)	ハードウェアによって異なります。このコンパイル・フラグは、ビジネス関数をリリース・モードでビルドするときに使用されます。このフラグは変更しないでください。
DebugFlags=	-g -y -D_DEBUG -DJDEDEBUG (HP9000 用のデフォルト値) -g qfulpath -qdbextra -D_DEBUG -DJDEDEBUG (RS/6000 用のデフォルト値) -g -D_DEBUG -DJDEDEBUG (Sun 用のデフォルト値)	ハードウェアによって異なります。このコンパイル・フラグは、ビジネス関数をデバッグ・モードでビルドするときに使用されます。このフラグは変更しないでください。
InliningFlags=	ブランク (デフォルト値)	AS/400 でインラインを使用するかどうかを示します。AS/400 でインラインをオンにする場合は "Yes" を入力し、オフにする場合は "No" を入力します。AS/400 以外のサーバーでは、このフラグはブランクにします。
DefineFlags=	-DKERNEL -DPRODUCTION_VERSION - DNATURAL_ALIGNMENT -D_HPUX-SOURCE (HP9000 用のデフォルト値) -DKERNEL -DPRODUCTION VERSION -	HP、RS、および SUN 用のソースのカーネル本稼働用バージョンを示します。

	DNATURAL_ALIGNMENT (RS/6000 用のデフォルト値) -DKERNEL -DPRODUCTION_VERSION - DNATURAL_ALIGNMENT -D_SUN-SOURCE (Sun 用のデフォルト値)	
CompilerFlags=	-Aa +w1 +z -c (HP9000 用のデフォルト値) -qalign=natural -qflag=l:l -c (RS/6000 用のデフォルト値) -qspill=1024 -misalign -KPIC (Sun 用のデフォルト値)	ハードウェアによって異なります。 この spill フラグは、ビジネス関数をコンパイルするときにスタック領域を設定します。通常、1024 の値で、転送されたビジネス関数をコンパイルするのに十分な領域が確保されます。
OSReleaseLevel=	+DAportable -q32 (AIX 用)	コンパイルするパッケージのリリース・レベルを示します。このフラグは変更しないでください。
LinkFlags=	-b -z (HP9000 用のデフォルト値) -bl:/<システム・ディレクトリ名>/bin32/functlist.imp -bM:SRE - bexpall -brtl -lc -bnentry -L. L/usr/<システム・ディレクトリ名>/lib -ljdelib -ljdekrnl -ljdenet -bloadmap:loadmap (RS/6000 用のデフォルト値) -G -L\$(ORACLE_HOME)/lib (Sun 用のデフォルト値)	ハードウェアによって異なります。このフラグは、ビジネス関数をリンクするために使用されます。このフラグは変更しないでください。
LinkLibraries=	ブランク (デフォルト値)	ビジネス関数がリンクされるライブラリを示します (Windows NT および AS/400 サーバーのみに適用されます)。
SimultaneousBuilds=	0 (無制限) (デフォルト値) 整数 (同時にビルドされる数)	一度にビルドできる DLL の数を示します。「0」(ゼロ)は、すべての DLL が同時にビルドされることを意味します。
Qname=	<キュー名>	AS/400 専用です。DLL のビルド作業で、デフォルト・キュー以外のキューに移動する場合、キュー名を指定します。

参照

- JDE.INI ファイルの詳細については、『システム・アドミニストレーション』ガイドを参照してください。

ワークステーションの JDE.INI の設定

以下は、ワークステーション用の jde.ini ファイルに任意で入力できる設定です。

スペックおよびビジネス関数のビルド

ビジネス関数とスペックの両方を含んだフルまたは部分クライアント・パッケージを作成する場合、パッケージの作成に使用するコンピュータ上の jde.ini ファイルの [INSTALL (インストール)] セクションに次の設定を追加します。

```
WaitForBusbuild=Y
```

この設定を追加すると、スペックとビジネス関数が同時ではなく順にビルドされるため、ビルド・プロセスを短縮できます。

サーバー・パッケージの圧縮

サーバーで作成するパッケージを圧縮するには、ワークステーション上の jde.ini ファイルに [BSFN BUILD (BSFN ビルド)] セクションを追加し、次のエントリを作成します。

```
DoCompression=1
```

この設定は、サーバー上で作成されたパッケージを圧縮し、同じタイプの他のサーバー(すべての AIX UNIX サーバー、NT サーバーなど)に配布します。サーバーのタイプが異なる場合は、該当するタイプのサーバーでパッケージ作成する必要があります。

サーバーで圧縮パッケージが作成されると、圧縮されたファイルは、`%bin32` や `%specs` などのサブディレクトリに格納されます。このサブディレクトリは、デプロイメント・サーバー上の `%package %package name %server type %` の下にあります。この場合、`package name` はパッケージ名、`server type` はパッケージが圧縮されたサーバーのタイプです。圧縮プロセスでは、`compressed.inf` という名前の新しいファイルが `server type` ディレクトリで作成されます。このファイルには、圧縮されたファイルの配布に必要な情報が収められます。次の表は、サーバーの各タイプで作成される圧縮ファイルのタイプを示しています。

NT	UNIX	AS/400
.cab	.z	SRVPG、SPECS、MODULE、USRSPC

パッケージを他のエンタープライズ・サーバーに配布する際、圧縮された `.inf` ファイルを読み込み、その情報に従ってデプロイメント・サーバー上のパッケージ・ディレクトリからエンタープライズ・サーバーに圧縮ファイルがコピーされます。

Sun オペレーティング・システム・ユーザーの考慮事項

Sun Solaris コンパイラでは、コンパイルするソース・コード・ファイルの終わりに改行文字を置く必要があります。コンパイラがこの改行文字を検索できない場合、その行は却下され、警告メッセージが表示されます。行が却下されてメッセージが表示されると、パッケージ・ビルドに失敗する場合があります。

Solaris オペレーティング・システムを実行する Sun サーバーでカスタム修正を開発した場合は、それらの修正を含むパッケージの組立て、定義、およびビルドを実行する前に、このような改行文字が

コンパイル済みのソース・コードに存在することを確認する必要があります。このステップにより、パッケージ・ビルド・プロセスを正常に終了できます。

改行文字が自動的に追加される場合は、それを手動で追加する必要はありません。UNIX 環境で vi または emacs などのエディタを使用してソース・コードを編集する場合、これらのエディタは自動的に改行文字を追加します。また、ビジネス関数のソース・コード・ファイルには、いずれも改行文字が含まれていますが、

PC ワークステーションにインストールされているエディタは、通常、改行文字を追加しません。そのため、ソース・コードを PC ワークステーションで編集し、ファイルをサーバーに転送してコンパイルする場合は、改行文字がソース・コードに存在することを確認してください。

パッケージの組立て

パッケージの組立てがパッケージのビルドおよび配布における最初のステップです。このステップでは、ビルドするパッケージのタイプを選択し、パッケージ名および詳細な記述を入力して、パッケージに含めるオブジェクトを組み立てる必要があります。このパッケージ名と記述は、ワークステーション・インストールでユーザーがインストールするパッケージを選択するときに表示されます。

このプロセスは、〈Package Assembly〉プログラム(P9601)からアクセスする〈Package Assembly Director〉を使用して実行します。パッケージを組み立てる際、ビルド状況は、常に「In Definition (定義中)」か、「Definition Complete (定義完了)」になります。パッケージを組み立てると、ビルド状況を定義できます。

Package Assembly Director の理解

〈Package Assembly Director〉フォームは、パッケージ・コンポーネントを検索するロケーションを指定または確認するためのプロセス、およびパッケージに含めるオブジェクトを指示するプロセスをガイドします。このフォームでは、常に前後のフォームに移動できます。また、いつでも組立てプロセスをキャンセルすることができます。

次の表に、〈Package Assembly Director〉の各フォームの機能を示します。

〈Package Assembly Directory (パッケージ・アセンブリ・ディレクトリ)〉フォーム	〈Package Assembly Director〉に関する基本情報を確認します。
〈Package Information (パッケージ情報)〉フォーム	パッケージ名、記述、および対応するパス・コードを入力します。
〈Package Type Selection (パッケージ・タイプの選択)〉フォーム	フル・パッケージ、部分パッケージ、アップデート・パッケージのうちのどのパッケージを作成するかを示します。 アップデート・パッケージを作成する場合、アップデート・パッケージの基となる「親パッケージ」も指示する必要があります。例えば、APPL_B というオリジナルの部分パッケージを更新するためにパッケージを作成する場合、アップデート・パッケージを親パッケージとして「APPL_B」と入力します。 また、アップデート・パッケージや部分パッケージには、オブジェクト・スペックを含めることもできます。スペックを含めない場合は、ユーザーがアプリケーションを最初に選択したときに、ジャストインタイム・インストールによってそのアプリケーションがインストールされます。正確に言うと、パッケージを受け取ってからユーザーがアプリケーションを最初に選択したときに、アプリケーション・スペックがセントラル・オブジェクト・データ・ソースから自動的に取

り込まれるということです。

〈Foundation Component(ファンデーション・コンポーネント)〉フォーム

ERP 9.0 ファンデーションのロケーションを入力します。ERP 9.0 ファンデーションは、ERP 9.0 のすべてのアプリケーションを実行するのに必要なコードです。ERP 8.0 ファンデーションは、すべてのフル・パッケージおよび部分パッケージに必要です。フル・パッケージおよび部分パッケージのファンデーション・パスを指定しない場合は、デフォルトのファンデーション・パスが使用されます。アップデート・パッケージは、親パッケージ以外のファンデーションを指定しない限り、親パッケージのファンデーションを使用します。

〈Help Component(ヘルプ・コンポーネント)〉フォーム

ヘルプ・ファイルのロケーションを入力します。ヘルプ・コンポーネントは、パッケージ用のヘルプ・ファイルが検索される場所を確定します。フル・パッケージと部分パッケージでは、ヘルプ・ロケーションを指定しないと、デフォルトのヘルプ・ファイル・パスが使用されます。アップデート・パッケージにはヘルプは必要ありません。

ヘルプ・コンポーネントがパッケージに含まれていても、ヘルプ・ファイルがユーザーのワークステーションにインストールされるとは限りません。ヘルプ・コンポーネントには、パッケージ用のヘルプ・ファイルを検索する場所が指定されています。

〈Database Component(データベース・コンポーネント)〉フォーム

パッケージに含めるデータベースのロケーションを指定します。フル・パッケージと部分パッケージでは、データベース・ロケーションを指定しないと、デフォルトのデータベース・パスが使用されます。アップデート・パッケージにはデータベースは必要ありません。

〈Default Object Component(デフォルト・オブジェクト・コンポーネント)〉フォーム(フル・パッケージのみ)

デプロイメント・データ・ソースを確認する時に使用します。フル・パッケージをビルドする際、パッケージに含めるオブジェクトは、指定したパッケージのパス・コードに関連付けられたデプロイメント・データ・ソースから取り込まれます。

〈Object Component(オブジェクト・コンポーネント)〉フォーム(部分パッケージおよびアップデート・パッケージのみ)

パッケージに含めるオブジェクトを個別に指定します。次のオブジェクトのいずれかを追加することができます。

- 対話型アプリケーションまたはパッチ・アプリケーション
- ビジネス関数モジュール
- ビジネス・ビュー
- データ構造体
- メディア・オブジェクト・データ構造体
- テーブル定義

〈Features Component(機能コンポーネント)〉フォーム

機能をパッケージに組み込む時に使用します。機能とは、アプリケーションやその他の機能をサポートするためにワークステーションやサーバーにコピーする必要がある、レジストリ設定などの一連のファイルや構成オプションのことを示します。

〈Language Component(言語コンポーネント)〉フォーム

デフォルト言語(英語)以外の言語スペックをパッケージに含めるために使用します。

〈Package Component Revisions(パッケージ・コンポーネントの改訂)〉フォーム

前のフォームで入力した情報を再確認します。また、このフォームで、一部またはすべての選択項目を修正できます。

デフォルト値の承認

〈Package Assembly Director〉では、多くのフォームでデフォルト値が設定されています。デフォルト値をそのまま使用する場合は、何も入力しないで次のフォームに進んでかまいません。

フォームは次の項目に基づいてデフォルト値を確定します。

- Foundation(ファンデーション): ファンデーションのデフォルト・ロケーションは、パッケージのパス・コードの下のサーバー共有パスです。
- Help(ヘルプ): ヘルプのデフォルト・ロケーションは、パッケージのパス・コードの下のサーバー共有パスです。
- Database(データベース): データベースのデフォルト・ロケーションは、パッケージのパス・コードの下のサーバー共有パスです。
- Objects(オブジェクト): フル・パッケージでのデフォルト・ロケーションは、デプロイメント・データ・ソースです。
- Language(言語) デフォルト言語は英語です。

デフォルト値が設定されたフォームでは、フィールドを変更またはクリアしても[Default(デフォルト)]ボタンをクリックすることにより、オリジナルのデフォルト値を復元できます。また、〈Package Component Revisions〉フォームの[Form(フォーム)]メニューにも、デフォルト値を復元する[Set Default(デフォルトの設定)]オプションがあります。

フル・パッケージまたは部分パッケージをビルドしており、そのパッケージにオブジェクトを指定する必要がない場合、パッケージを最も速く定義する方法は、ファンデーション、データベース、ヘルプおよび言語のデフォルト・ロケーションを承認することです。この方法はフル・パッケージおよび部分パッケージでのみ有効です。アップデート・パッケージでは、デフォルト値を承認してオブジェクトを含めないと、空のパッケージが作成されます。

〈Package Assembly Director〉では、フォームが表示されたときに[Next(次へ)]をクリックすると、そのフォームのデフォルト値を承認できます。必要に応じて、最後の〈Package Component Revisions〉フォームでいつでも値を変更できます。

パッケージ・ビルドに使用するパス・コードの検証

パッケージを組み立てる前に、パッケージのビルド元となるパス・コードが適切に構成されているかどうかを検証できます。この検証プロセスでは、パッケージの投入前に環境、マシン、およびテーブルがテストされます。環境の検証により、構成エラーが原因でパッケージ・ビルドに失敗する可能性がなくなります。また、検証を行うことでパッケージ・ビルドの所要時間を大幅に短縮できます。

この検証プロセスの対象事項は次のとおりです。

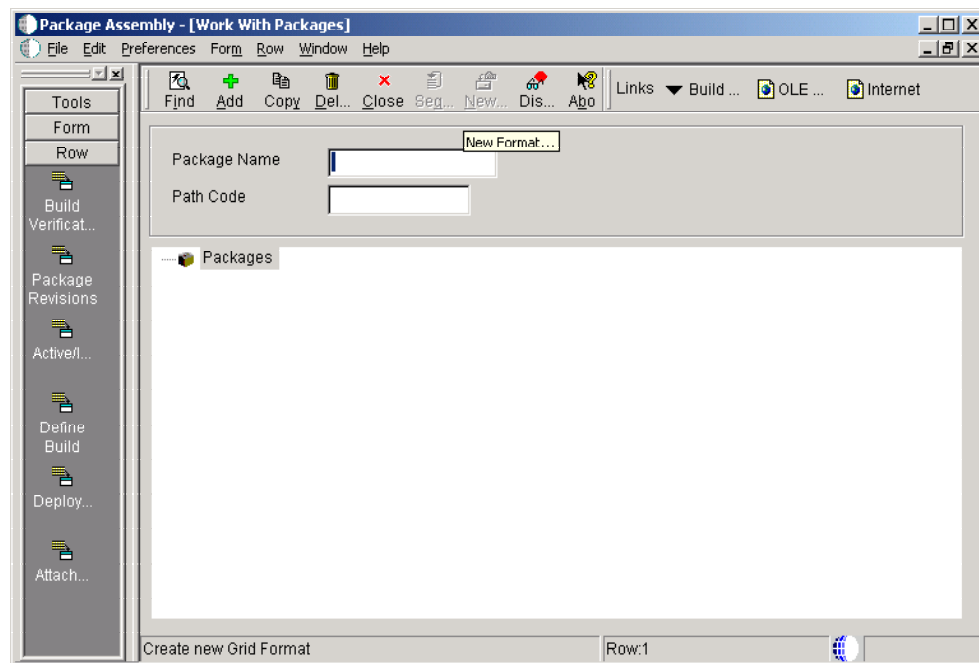
- 空きディスク領域が十分にあるかどうか。
- セントラル・オブジェクトとパッケージ・ビルドのテーブルにアクセスできるかどうか。
- ユーザーにデプロイメント・サーバーとエンタープライズ・サーバー上でディレクトリを作成するための許可が付与されているかどうか。
- 必要なサービス・パックがインストールされているかどうか。
- 必要な MDAC がインストールされているかどうか。

- マシン・テーブルがセットアップされているかどうか。
- 必要なバージョンのコンパイラがインストールされているかどうか。
- エンタープライズ・サーバー・ポートにアクセスできるかどうか。
- jde.ini ファイルのデバッグ・レベルが、クライアントとエンタープライズ・サーバーに適した値に設定されているかどうか。

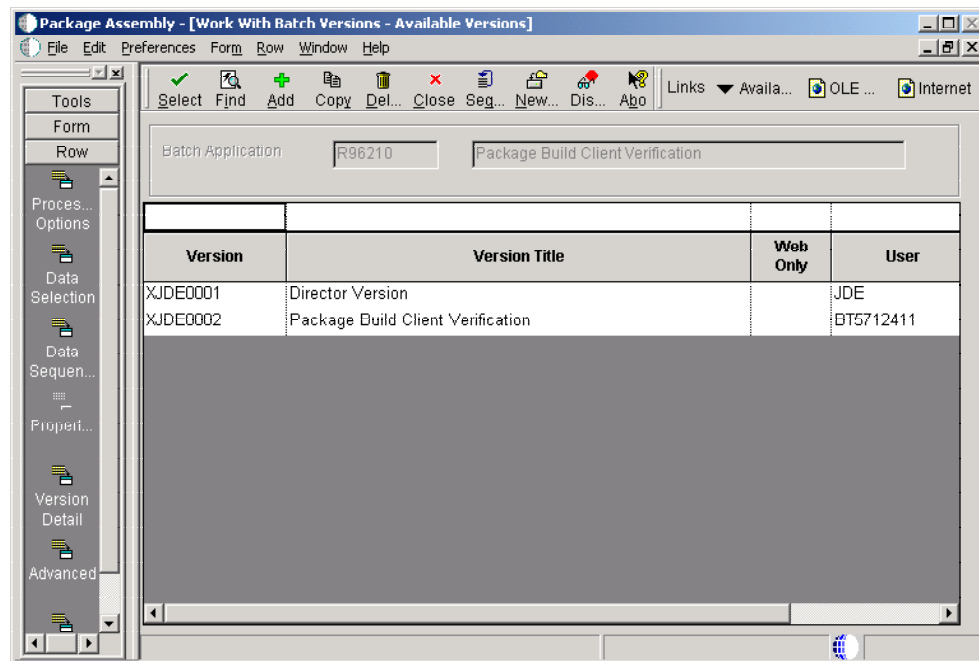
▶ パッケージ・ビルドに使用するパス・コードを検証するには

〈Package and Deployment Tools〉メニュー(GH9083)から〈Package Assembly〉を選択します。

〈Work with Packages(パッケージの処理)〉フォームが表示されます。



1. [Form(フォーム)]メニューから[Build Verification(ビルドの検証)]を選択します。



2. 〈Work with Batch Versions – Available Versions (バッチ・バージョンの処理 – 使用可能なバージョン)〉フォームで、[Select (選択)] をクリックします。
3. 〈Version Prompting (バージョン・プロンプト)〉フォームで [Submit (投入)] をクリックします。
4. 〈Processing Options (処理オプション)〉フォームで、次のフィールドに値を入力します。
 - 1. Indicate if this is a Client or Server (これがクライアントであるかサーバーであるかを指定してください。)

クライアントの場合はblank、サーバーの場合は"1"を入力します。
 - Path Code (パス・コード)

検証するパス・コードを入力します。
 - Server Name (サーバー名)

マシンがサーバーの場合は、サーバー名を入力します。
 - Compiler Flag (コンパイラ・フラグ)

サーバー上にコンパイラが存在する場合は"1"を入力します。
5. [OK] をクリックします。
6. 〈Printer Selection (プリンタ選択)〉で、使用するプリンタを選んで [OK] をクリックします。

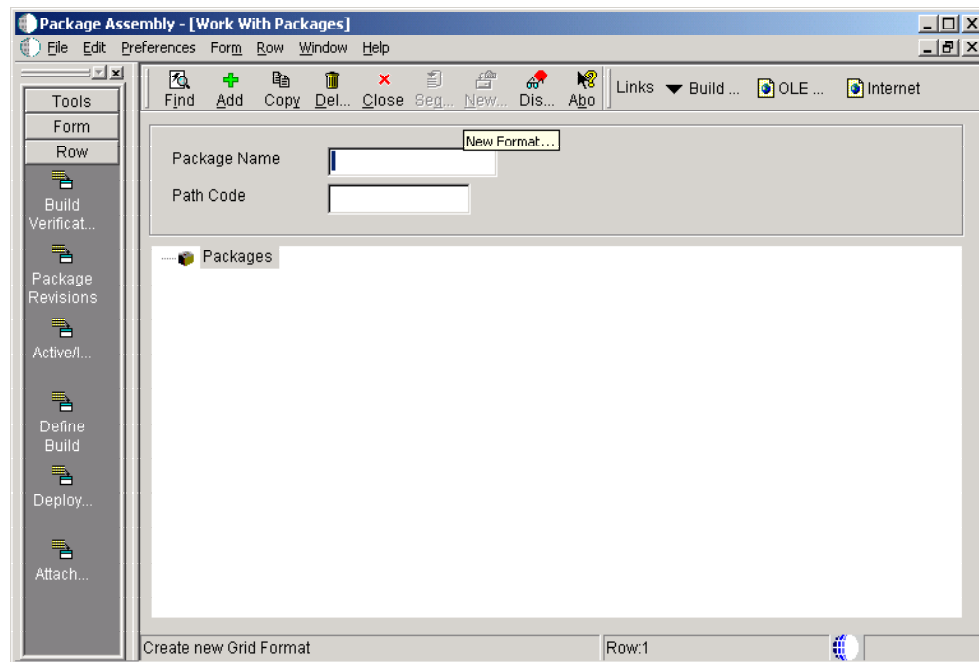
パッケージ・アセンブリ・ディレクトリの実行

〈Package Assembly Director〉の各メイン・フォームにデフォルト情報を入力するには、以下のタスクを実行します。各フォーム上の情報をカスタマイズする方法の詳細については、以下の該当するタスクを参照してください。

▶ パッケージ・アセンブリ・ディレクトリを実行するには

〈Package and Deployment Tools〉メニュー(GH9083)から〈Package Assembly〉を選択します。

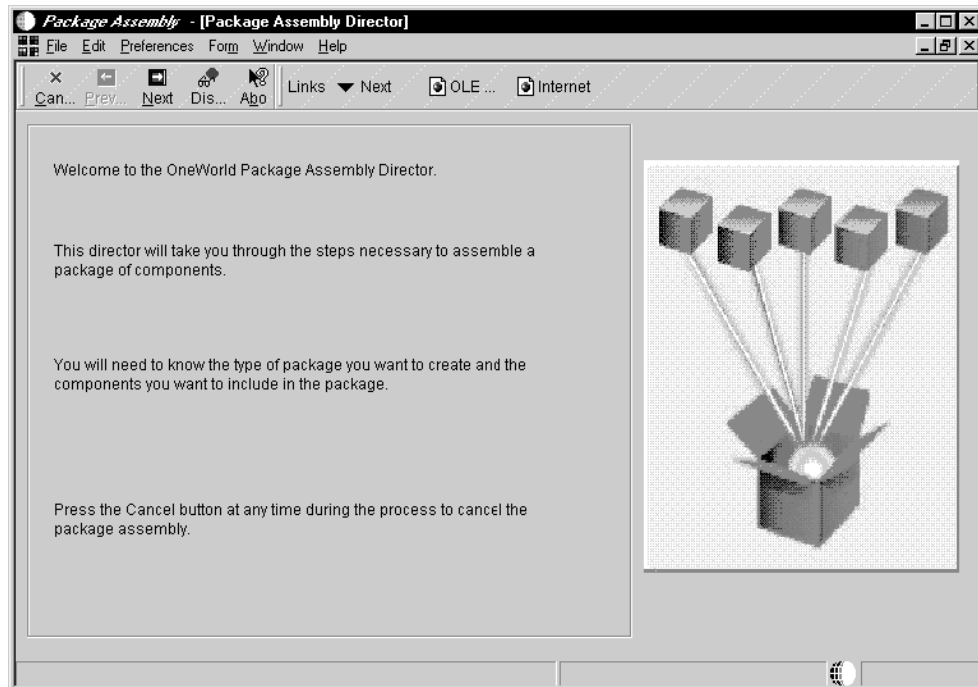
〈Work with Packages(パッケージの処理)〉フォームが表示されます。



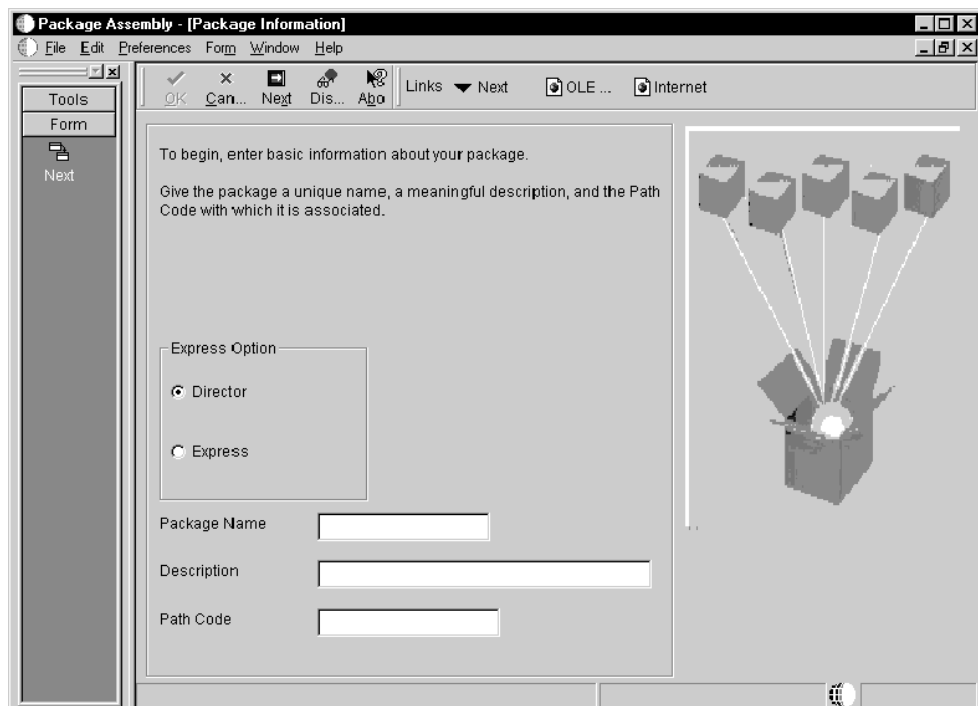
以前に組み立てたパッケージがある場合は、このフォームに表示されます。このフォームでは、他の親/子フォームと同様に、プラス(+)またはマイナス(-)記号をクリックすると、パッケージの一覧を拡大/縮小表示して、画面に表示する情報の量を調節できます。

組立て済みのパッケージがある場合は、それぞれのパッケージ名の下にパッケージ・プロパティ(パッケージ・タイプ、現行状況など)や、選択されているファンデーション、データベース、ヘルプ、言語などを表示できます。

1. 新規パッケージを組み立てるには、[Add(追加)]をクリックします。



2. 〈Package Assembly Director〉フォームで、[Next]をクリックします。



3. 〈Package Information〉フォームが表示されます。次のフィールドに値を入力します。

- パッケージ名
- 記述

- パス・コード

注:

B9 からは、このフィールドに“*ALL”と入力することで、単一のファンデーション・パッケージをビルドして全パス・コードに配布できるようになりました。この方法を使用すると、どのパス・コードにでもインストール可能なサービス・パックについて、アップデート・パッケージを作成することができます。[Path Code]フィールドに“*ALL”と入力すると、スペック、ヘルプ、ビジネス関数など、パッケージ内のオブジェクトの選択、ビルド、配布はできなくなり、選択、ビルド、配布できるのはファンデーションのみとなります。パッケージは、B9/all_packages など、リリース・パスの下にある all_packages ディレクトリにビルドされます。このパッケージは、どのパス・コードにでも配布できます。

この方法を使用するには、事前に<Path Code Master(パス・コード・マスター)>に項目 *ALL を追加しておく必要があります。このパス・コードをセットアップする方法については、『CNC インプリメンテーション』ガイドを参照してください。

4. 次のいずれかのオプションをクリックします。

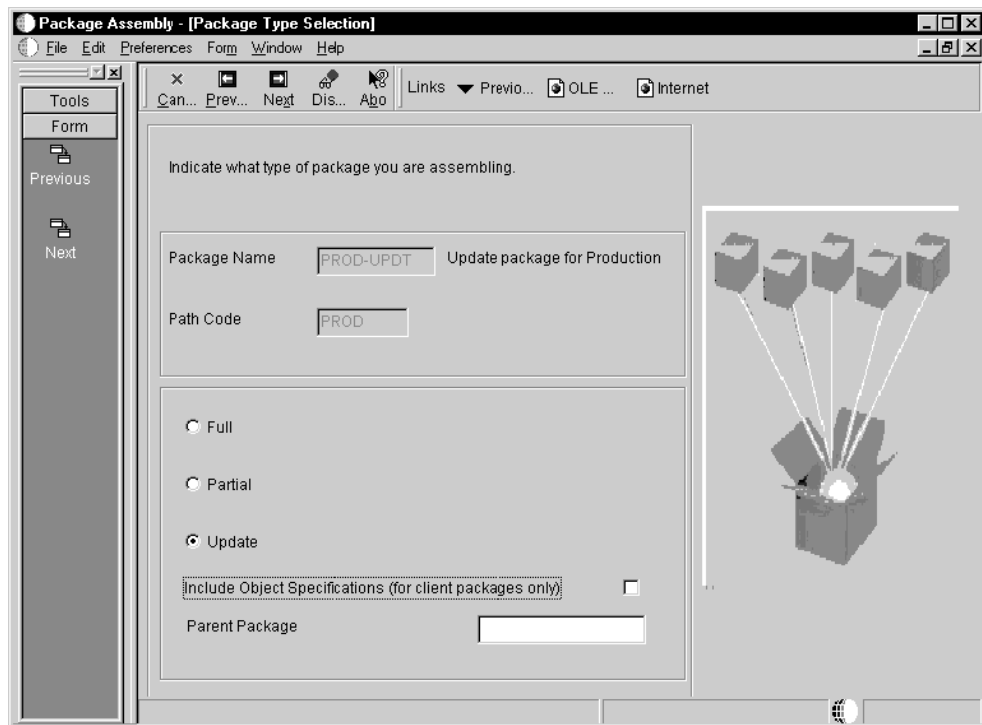
- Director

パッケージ・アセンブリをカスタマイズするには、このオプションをクリックし、このタスクの他のステップを実行します。

- Express

パッケージ・アセンブリでデフォルト値を承認する場合は、このオプションをクリックし、「パッケージ・アセンブリの選択の検討」に進みます。デフォルト値を承認しない場合は、ステップ 5 に進みます。

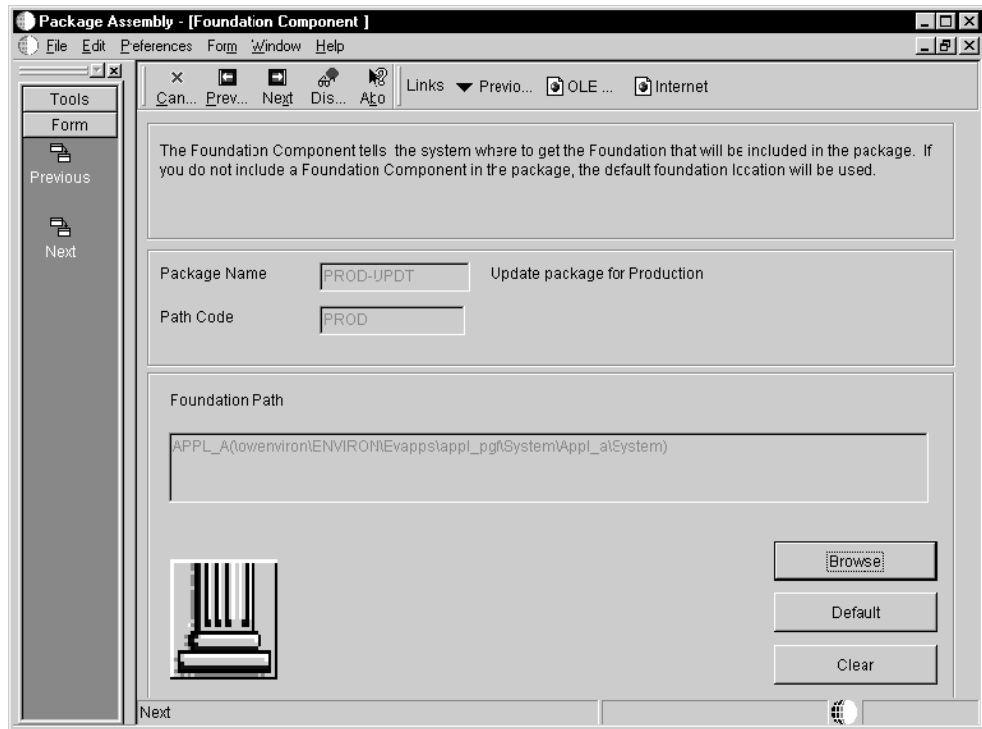
5. [Next]をクリックします。



〈Package Type Selection〉フォームが表示されます。次のフィールドに値を入力します。

- Full, Partial, or Update (フル、部分またはアップデート)
ファンデーション・パッケージを*ALL パス・コードにビルドする場合は、アップデート・パッケージが自動的に選択されます。
- Parent Package (親パッケージ)
(アップデート・パッケージのみ)
- Include Object Specifications (オブジェクト・スペックの組込み)
(クライアント・パッケージのみ)

6. [Next]をクリックします。



7. フル・パッケージと部分パッケージでは、デフォルト・ロケーションを承認する場合は[Next]をクリックし、別のファンデーション・ロケーションを指定する場合は[Browse(ブラウズ)]をクリックします。

ファンデーションの入力については、「ファンデーション・ロケーションの入力」を参照してください。

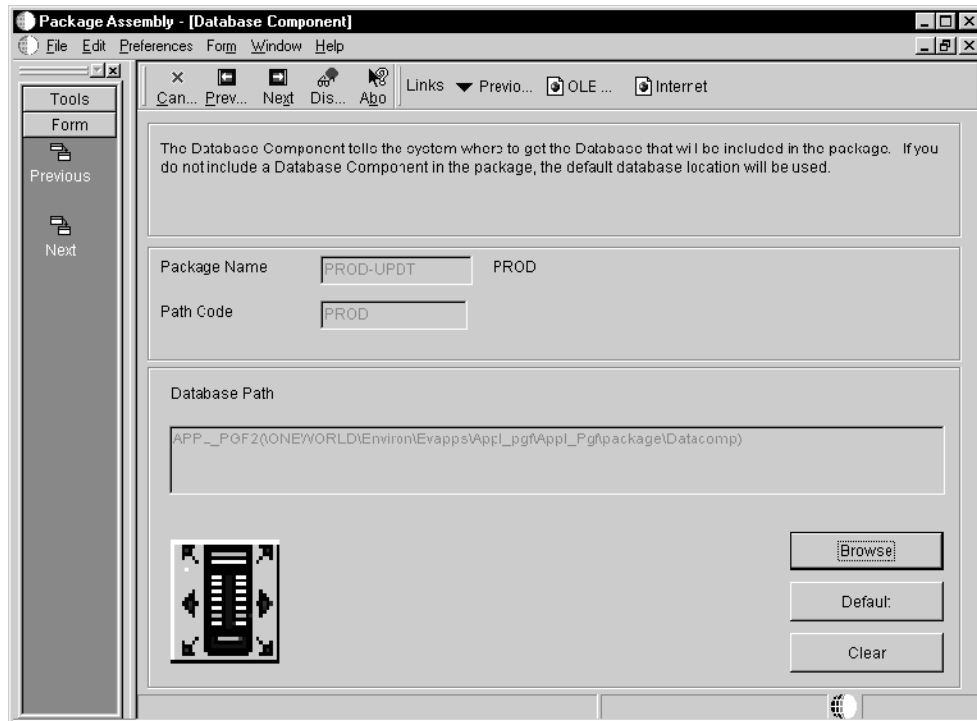
8. アップデート・パッケージでは、ファンデーションを組み込む場合を除き、[Clear]をクリックします。
9. [Next]をクリックします。



10. フル・パッケージと部分パッケージでは、デフォルト・ロケーションを承認する場合は[Next]をクリックし、別のヘルプ・ロケーションを指定する場合は[Browse]をクリックします。

ヘルプ・ロケーションの入力の詳細については、「ヘルプ・ファイル・ロケーションの入力」を参照してください。

11. アップデート・パッケージでは、ファンデーションを組み込む場合を除き、[Clear]をクリックします。
12. [Next]をクリックします。

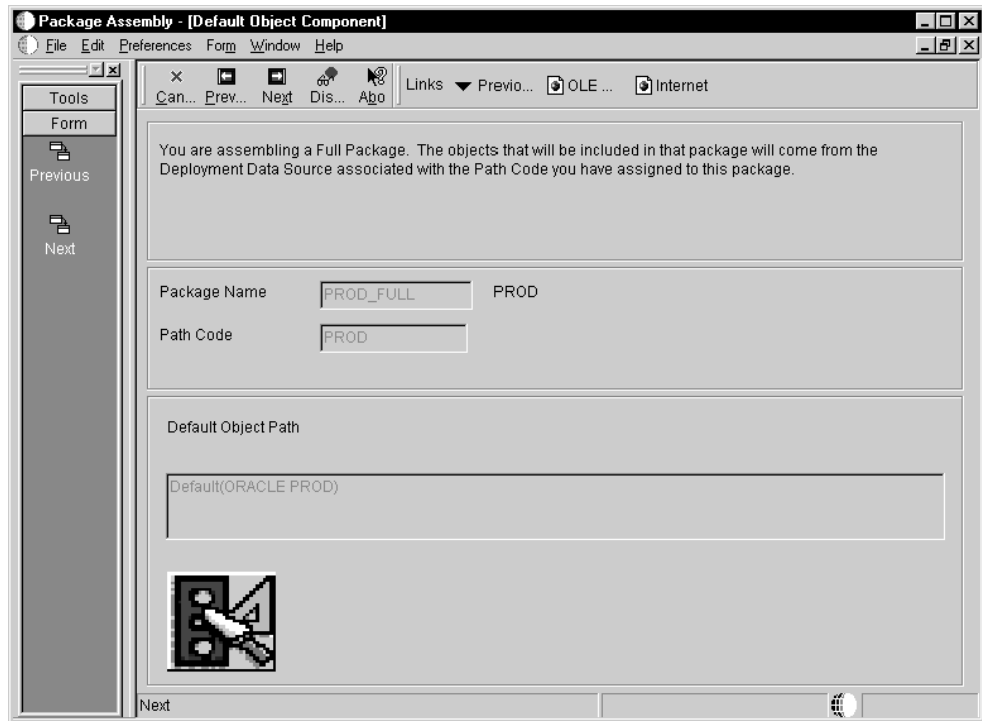


13. フル・パッケージと部分パッケージでは、デフォルト・ロケーションを承認する場合は[Next]をクリックし、別のデータベース・ロケーションを指定する場合は[Browse]をクリックします。

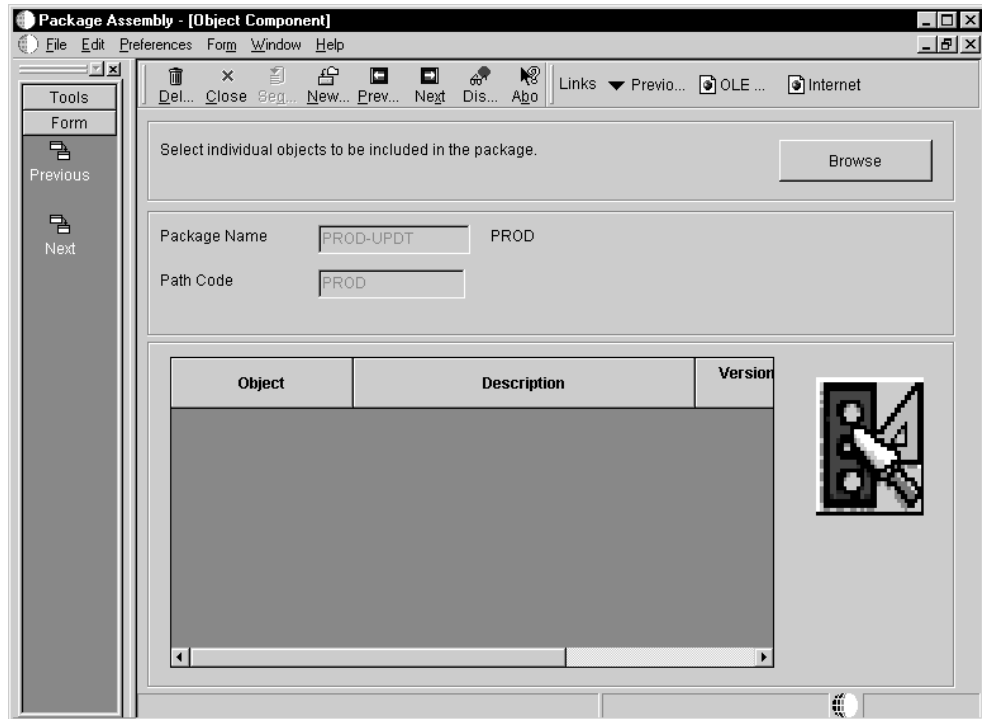
データベース・ロケーションの入力の詳細については、「データベース・ロケーションの入力」を参照してください。

14. アップデート・パッケージでは、ファンデーションを組み込む場合を除き、[Clear]をクリックします。
15. 次のどちらかの手順を完了します。

- フル・パッケージを組み立てている場合は、[Next]をクリックします。
フル・パッケージの場合、デフォルトのオブジェクト・パスに関連付けられているデプロイメント・データ・ソースからパッケージがビルドされます。適切なロケーションがフォームに表示されていることを確認して、次のステップに進みます。
- 部分パッケージまたはアップデート・パッケージを組み立てる場合は、〈Database Component〉フォームで[Next]をクリックして、次のステップに進みます。



部分パッケージまたはアップデート・パッケージを組み立てるときは、〈Object Component〉フォームが表示されます。このフォームでは、パッケージに含めるオブジェクトを個別に指定できます。以前に組み立てたパッケージを改訂する場合、追加済みのオブジェクトも表示されます。



16. 〈Object Component〉フォームで、オブジェクトを追加するために、[Browse]をクリックします。

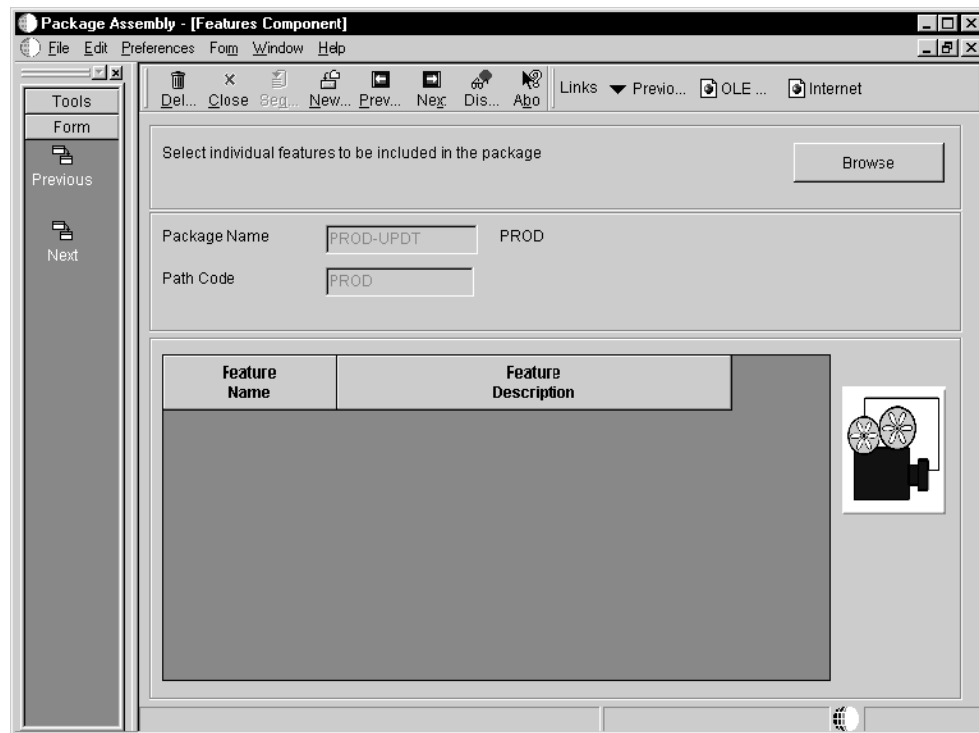
〈Object Component Selection(オブジェクト・コンポーネントの選択)〉フォームが表示されます。

17. 〈Object Component Selection〉フォームで、パッケージに含めるオブジェクトを選択し、[Close(閉じる)]をクリックして、〈Object Component〉フォームに戻ります。

オブジェクトの追加については、「パッケージへのオブジェクトの追加」を参照してください。

18. 〈Object Component〉フォームで、[Next]をクリックします。

〈Features Component〉フォームが表示されます。このフォームでは、パッケージに組み込む機能を指定できます。以前に組み立てたパッケージを改訂する場合は、追加済みの機能が表示されます。



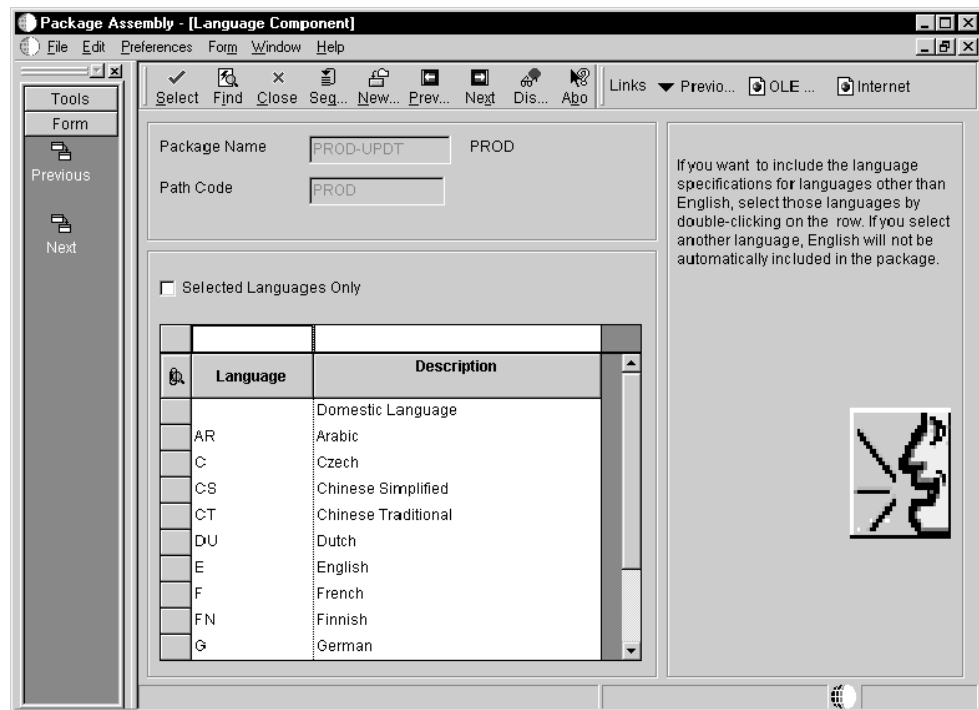
19. 機能を追加するには、[Browse]をクリックして〈Feature Component Selection(機能コンポーネントの選択)〉フォームを表示します。
20. 〈Feature Component Selection〉フォームで、[Find(検索)]をクリックして機能のリストを表示します。1 つ以上の機能を選択し、[Select(選択)]ボタンをクリックして、パッケージに組み込む機能を追加します。

複数の機能を選ぶには、Ctrl キーまたは Shift キーを押しながら機能をクリックし、[Select]をクリックします。

21. [Close]をクリックして、〈Features Component〉フォームに戻ります。

機能の追加の詳細については、「パッケージへの機能の追加」を参照してください。

22. 〈Feature Component(機能コンポーネント)〉フォームで、[Next]をクリックします。



23. 言語として英語だけを構成する場合は、[Next]をクリックします。
24. パッケージの言語スペックに言語を追加するには、グリッド行に表示されたその言語のロー・ヘッダーをダブルクリックします。

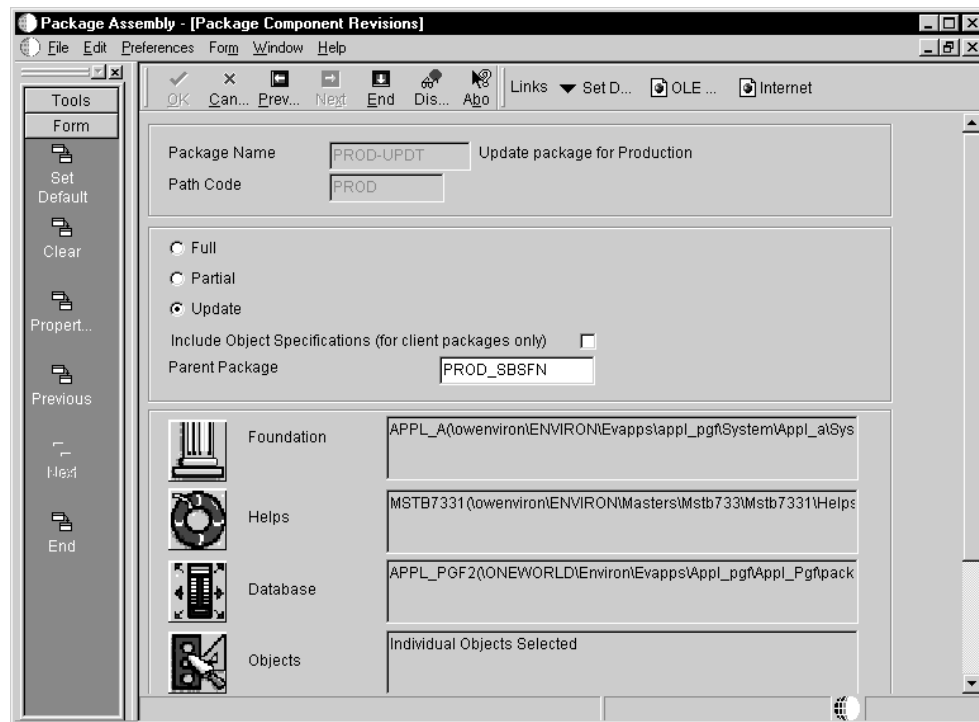
言語をパッケージに追加する場合、その言語だけが組み込まれます。たとえば、フランス語を追加する場合、英語はデフォルト言語であっても組み込まれません。フランス語と英語など、2つの言語を組み込む場合は、両方の言語の詳細レコードを選択する必要があります。複数言語対応のインストールの詳細については、『インストール・ガイド』を参照してください。

25. [Next]をクリックします。

パッケージ・アセンブリ選択の検討

〈Package Component Revisions〉フォームでは、現在のファンデーション、ヘルプ、データベース・ロケーション、およびパッケージ内のオブジェクトと機能を一覧表示でき、また選択されている言語があれば、その言語も表示できます。

▶ パッケージ・アセンブリ選択を検討するには



1. 〈Package Component Revisions〉フォームで、パッケージ・コンポーネントを変更するには、変更するコンポーネントのアイコンをクリックします。
該当するパッケージ・コンポーネントのフォームが表示されます。
2. パッケージの組立てが完了したら、[End]をクリックして、〈Package Assembly Director〉を終了します。
3. 「組み立てたパッケージのアクティブ化」に進みます。

フィールド記述

記述

用語解説

記述

パッケージの記述

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。

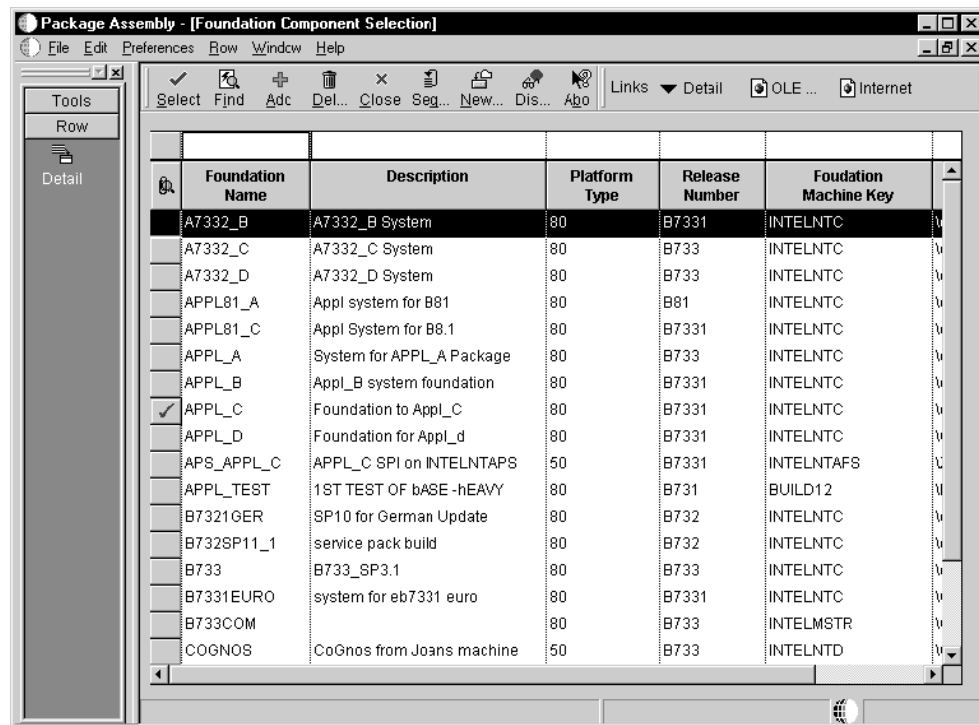
オブジェクト・スペックを含める	<p>更新パッケージに個々のオブジェクトを含むと、これらのオブジェクトのスペックセットを含むオプションも含まれます。スペックを含むと、ビルドされた後にパッケージにスペックのスナップショットが含まれます。</p> <p>パッケージが展開されると、パッケージを受け取ったユーザーはこれらのオブジェクトのスペックも受け取ります。</p>
親パッケージ	<p>更新パッケージには、オブジェクトのサブセットのみ含まれるため、更新パッケージの基準となる、または関連する親パッケージを指定してください。この情報は、ビジネス関数のビルドの仕方を決定するためシステムにより使用されます。</p>
選択した言語のみ	<p>このオプションがオンの場合は、パッケージの組込みに選択された言語のみがグリッドに表示されます。</p>
言語	<p>ユーザー定義コードテーブルの有効なコードのリスト</p>

ファンデーション・ロケーションの入力

デフォルトのファンデーション・ロケーションは、パッケージのパス・コードに関連付けられているリリースによって確定します。このロケーションは通常、パス・コードと同じディレクトリ・レベルのシステム・ディレクトリになります。次の手順でファンデーション・ロケーションを入力すると、〈Foundation Component〉フォームに表示されるデフォルトのロケーションが、入力したロケーションに変更されます。

▶ ファンデーション・ロケーションを入力するには

1. 次のいずれかの方法で、〈Foundation Component Selection (ファンデーション・コンポーネントの選択)〉フォームを表示します。
 - 〈Foundation Component〉フォームで、[Browse]をクリックします。
 - 〈Package Component Revisions〉フォームで、[Foundation (ファンデーション)] アイコンをクリックします。



2. パッケージで使用する既存のファンデーションを検索します。または、新しいファンデーションを追加する場合は、[Add]をクリックします。

Package Assembly - [Foundation Item Revisions]

File Edit Preferences Window Help

Links Displ... OLE ... Internet

General Information

Foundation Name: [] Description: []

Service Pack Number: [] Release: []

Build Information

Platform Type: []

Build Type: []

Foundation Build Status: 10 Foundation does not exist

Date Built: []

Time of Build: 00:00:00

Build Location

Foundation Machine Key: []

Foundation Path: []

Browse...

3. 〈Foundation Item Revisions〉フォームで、次のフィールドに値を入力してから、[OK]をクリックします。

- 基盤名
- 記述
- サービスパック番号
- リリース No.
- プラットフォーム・タイプ
- 作成タイプ
- ファンデーション・ビルド状況
- 作成日付
- 作成時刻
- 基盤マシン・キー
- 基盤パス

フィールド記述

記述	用語解説
基盤名	すべての OneWorld アプリケーションを実行するのに必須のコード。フルまたは部分パッケージに必須の ID です。フルまたは部分パッケージでは、ファンデーション行を入力しないとデフォルト情報が使用されます。デフォルト情報は、パッケージのパス・コードに関連付けられているリリースを通して決定されます。通常、パス・コードと同じディレクトリ・レベルにあるシステム・ディレクトリです。ファンデーションは、ビルド時に圧縮してください。
サービスパック番号	サービスパックは、ONEWORLD ソフトウェアの主要リリースおよび累積 PTF の間で提供される、ファンデーション・コードへのアップデート版です。
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。 --- フォーム固有 --- このファンデーションに関する ONEWORLD の主なリリース
プラットフォーム・タイプ	ホストマシン・タイプ
作成タイプ	ソフトウェアビルドに使用するコンパイラ構成を示します。
ファンデーション・ビルド状況	OneWorld ファンデーションの現在のビルドプロセス状況を説明します。
作成日付	ソフトウェアのビルド完了日付
作成時刻	ソフトウェアビルドが完了する時刻

基盤マシン・キー

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。

--- フォーム固有 ---

このフォームでは、マシンキーはカスタム基盤が常駐するデプロイメントサーバー名です。

基盤パス

World の場合: サーバー共用パス・フィールドは、環境が現サーバーのロケーションを決定するのに使用されます。

J.D. Edwards ERP の場合: このフィールドはこのパス・コードへの共有ディレクトリを示します。

ファイルサーバー上に保管されているオブジェクトは、このパス上にあります。

--- フォーム固有 ---

このフォームでは、この項目の正確なコピー元のパスを入力します。指定されたディレクトリのファイルがすべて、パッケージに含まれます。ソースマシンのキーおよびソースを使用し、項目の保管場所を定義します。

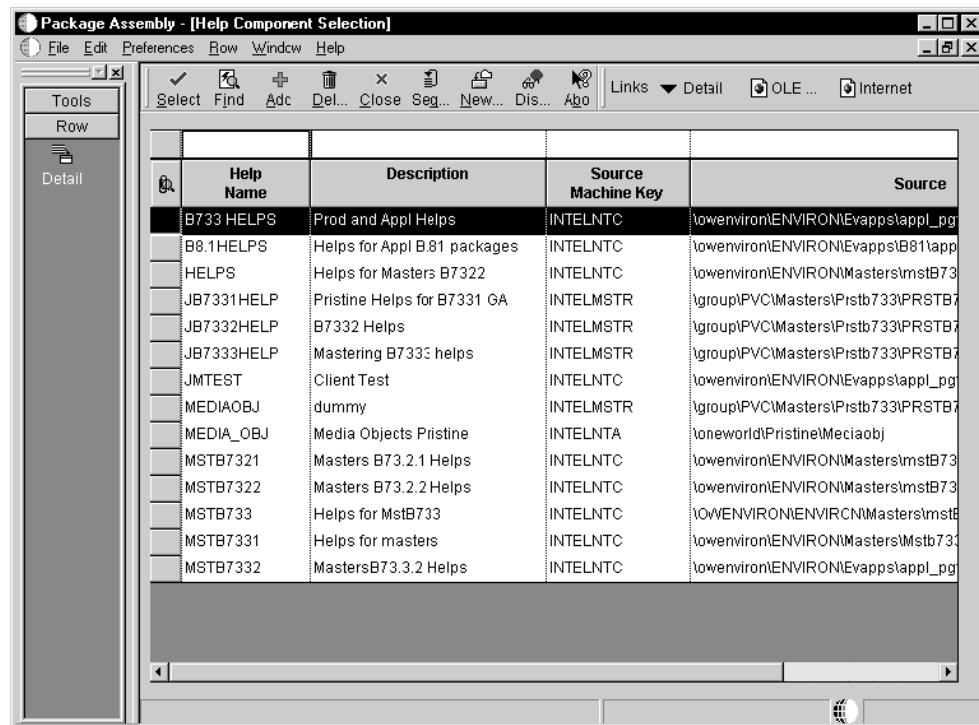
ヘルプ・ファイル・ロケーションの入力

どのパッケージ・タイプでも、ヘルプ・ロケーションを入力しないと、パッケージ・ビルドでは、〈Release Master〉で指定されているデフォルト・ロケーションにヘルプ(リリースごとに 1 組のヘルプ)をマップする INF を作成します。ヘルプ・ファイルのロケーションがパッケージで定義されている場合、パッケージ・ビルドでは、〈Help Item Revisions(ヘルプ項目の改訂)〉で指定されているロケーションからパッケージ・ディレクトリにヘルプ・ファイルをコピーします。

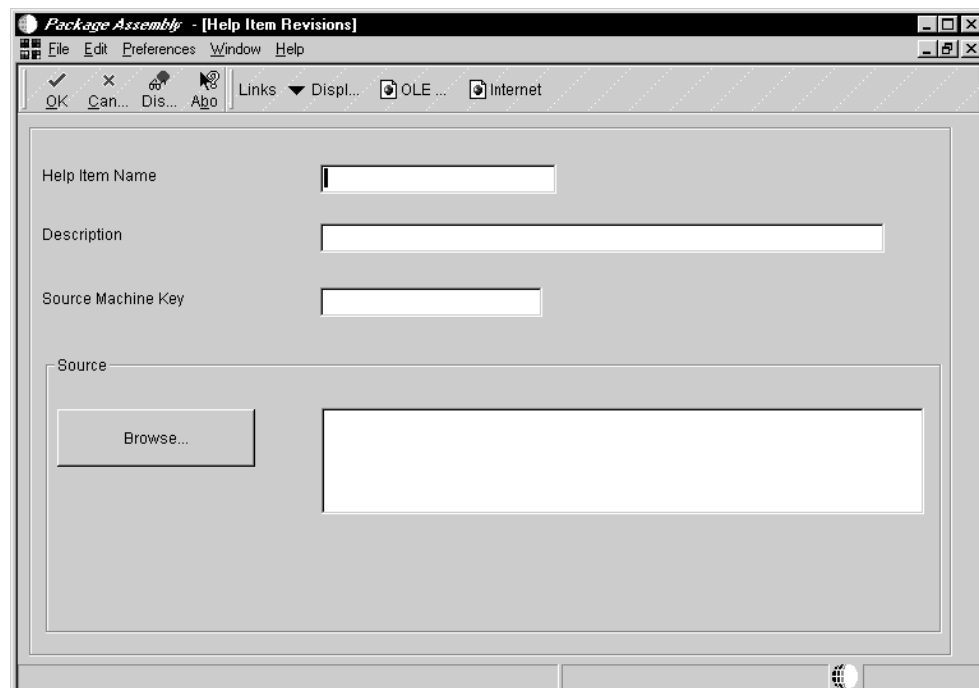
パッケージ INF は、ヘルプ・ファイルがデフォルト・ロケーションではなく、パッケージ自体に配置されていることを示します。ただし、〈Workstation Installation(ワークステーション・インストール)〉プログラムは、ユーザーのクライアントまたはサーバーのデプロイメント設定に基づいて、ヘルプをワークステーションにインストールします。〈Workstation Installation〉プログラムはパッケージ INF ヘルプ・ロケーションおよびデプロイメント設定のヘルプ・ロケーションに応じて、JDE.INI ファイルを更新します。

▶ ヘルプ・ロケーションを入力するには

1. 次のいずれかの方法で、〈Help Component Selection(ヘルプ・コンポーネントの選択)〉フォームを表示します。
 - 〈Help Component〉フォームで、[Browse]をクリックします。
 - 〈Package Component Revisions〉フォームで、[Helps(ヘルプ)]アイコンをクリックします。



2. パッケージで使用する既存のヘルプ・ロケーションを検索します。または、新しいヘルプ・ロケーションを追加する場合は、[Add]をクリックします。



3. 〈Help Item Revisions〉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。

- ヘルプ項目名

- 記述
- ソース・マシン・キー
- ソース

フィールド記述

記述	用語解説
ヘルプ項目名	すべての OneWorld アプリケーションを実行するのに必須のコード。フルまたは部分パッケージに必須の ID です。フルまたは部分パッケージでは、ファンデーション行を入力しないとデフォルト情報が使用されます。デフォルト情報は、パッケージのパス・コードに関連付けられているリリースを通して決定されます。通常、パス・コードと同じディレクトリ・レベルにあるシステム・ディレクトリです。ファンデーションは、ビルド時に圧縮してください。
サーバー共有パス	<p>World の場合: サーバー共有パス・フィールドは、環境が現サーバーのロケーションを決定するのに使用されます。</p> <p>J.D. Edwards ERP の場合: このフィールドはこのパス・コードへの共有ディレクトリを示します。</p> <p>ファイルサーバー上に保管されているオブジェクトは、このパス上にあります。</p>
ソース	<p>World の場合: サーバー共有パス・フィールドは、環境が現サーバーのロケーションを決定するのに使用されます。</p> <p>J.D. Edwards ERP の場合: このフィールドはこのパス・コードへの共有ディレクトリを示します。</p> <p>ファイルサーバー上に保管されているオブジェクトは、このパス上にあります。</p>

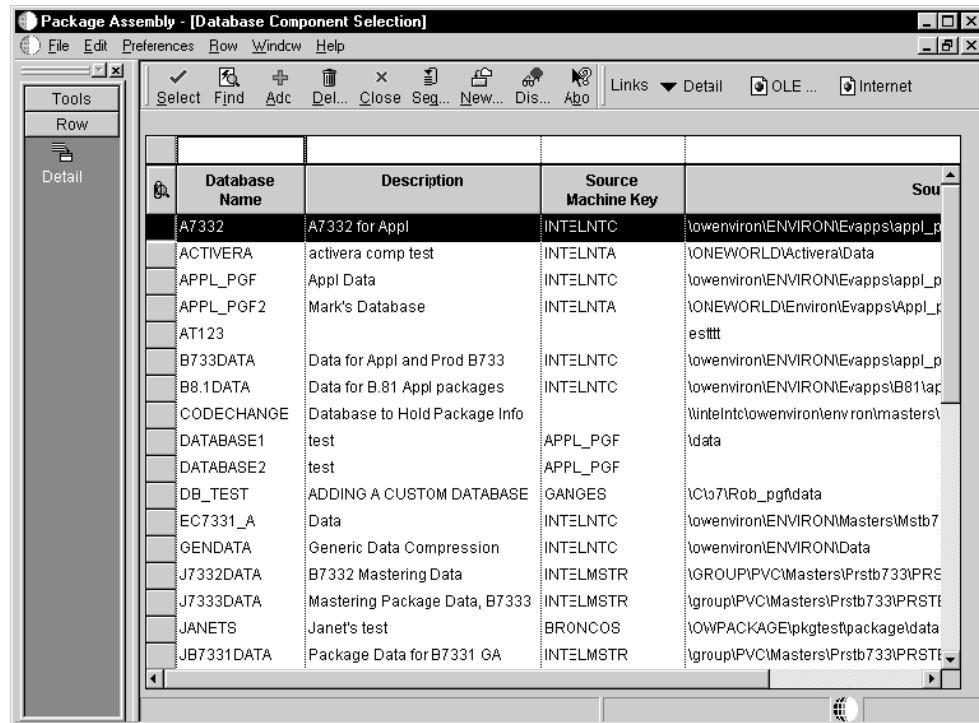
データベース・ロケーションの入力

フル・パッケージと部分パッケージでは、データベース・ロケーションを指定しないと、デフォルトのデータベース・パス(¥pathcode¥Packages¥Data)が使用されます。アップデート・パッケージにはデータベースは必要ありません。

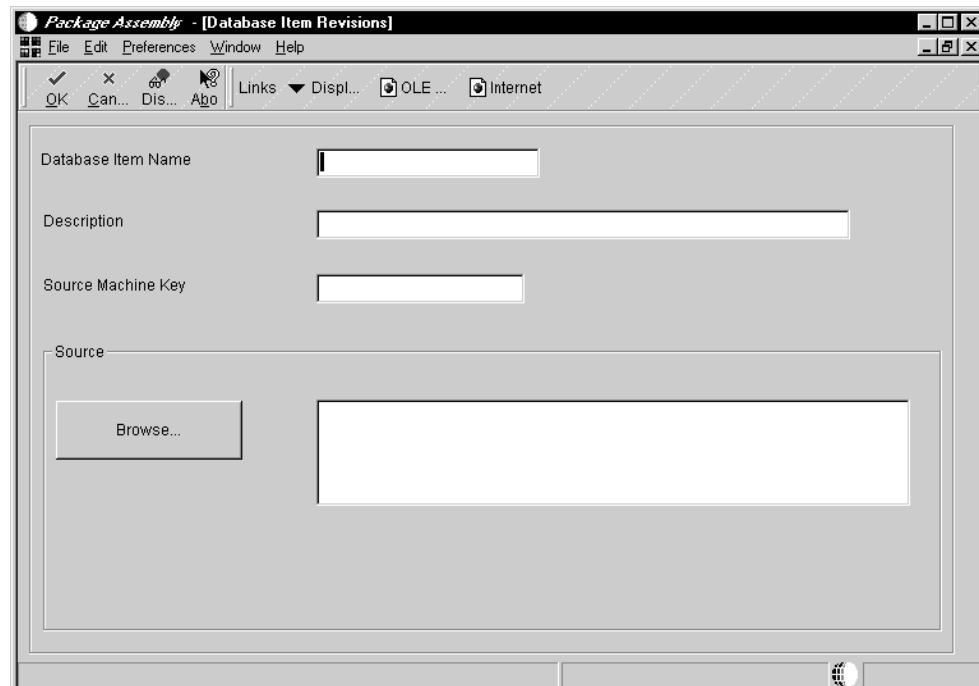
ここでは、〈Database Component〉フォームに表示されるデータベース・ロケーションをデフォルト・ロケーションから変更する方法について説明します。

▶ データベース・ロケーションを入力するには

- 次のいずれかの方法で、〈Database Component Selection(データベース・コンポーネントの選択)〉フォームを表示します。
 - 〈Database Component〉フォームで、[Browse]をクリックします。
 - 〈Package Component Revisions〉フォームで、[Database(データベース)]アイコンをクリックします。



2. パッケージで使用する既存のデータベース・ロケーションを検索します。または、新しいデータベース・ロケーションを追加する場合は、[Add]をクリックします。



3. <Database Item Revisions(データベース項目の改訂)>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。

- データベース項目名
- 記述
- ソース・マシン・キー
- ソース

フィールド記述

記述	用語解説
データベース項目名	すべての OneWorld アプリケーションを実行するのに必須のコード。フルまたは部分パッケージに必須の ID です。フルまたは部分パッケージでは、ファンデーション行を入力しないとデフォルト情報が使用されます。デフォルト情報は、パッケージのパス・コードに関連付けられているリリースを通して決定されます。通常、パス・コードと同じディレクトリ・レベルにあるシステム・ディレクトリです。ファンデーションは、ビルド時に圧縮してください。
ソース・マシン・キー	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	World の場合: サーバー共有パス・フィールドは、環境が現サーバーのロケーションを決定するのに使用されます。 J.D. Edwards ERP の場合: このフィールドはこのパス・コードへの共有ディレクトリを示します。 ファイルサーバー上に保管されているオブジェクトは、このパス上にあります。

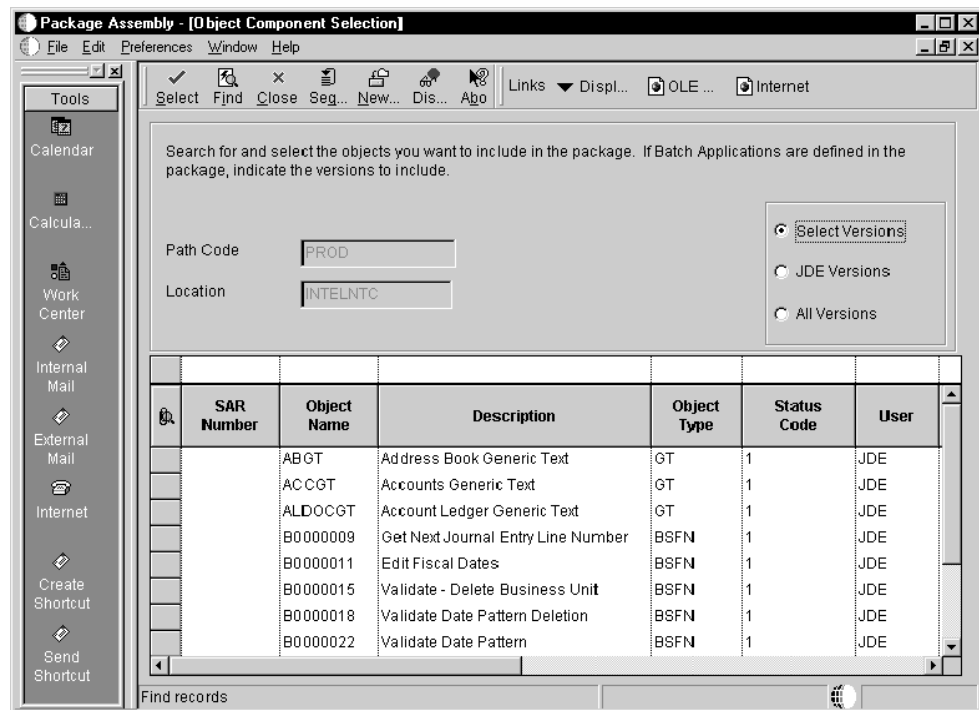
パッケージへのオブジェクトの追加

アップデート・パッケージまたは部分パッケージを組み立てる場合、パッケージに含めるオブジェクトを個別に選択できます。オブジェクトの追加を完了すると、それらのオブジェクトは〈Object Component〉フォーム、〈Package Component Revision〉フォーム、および〈Work with Packages〉フォームに表示されます。

ここでは、部分パッケージまたはアップデート・パッケージにオブジェクトを追加する方法について説明します。オブジェクトをはじめて追加する場合も、以前に組み立てたパッケージを改訂する場合も手順は同じです。

▶ オブジェクトをパッケージに追加するには

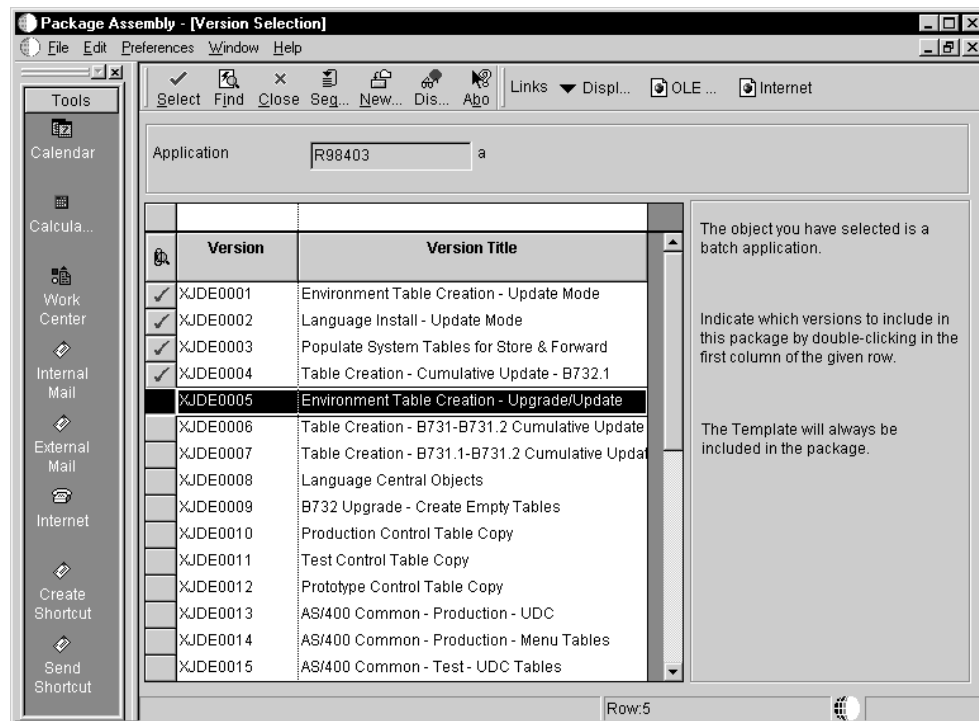
1. 次のいずれかの方法で、〈Object Component Selection〉フォームを表示します。
 - 〈Object Component〉フォームで、[Browse]をクリックします。
 - 〈Package Component Revisions〉フォームで、[Objects(オブジェクト)]アイコンをクリックして〈Object Component〉フォームを表示してから、[Browse]をクリックします。



2. パッケージに含める既存のオブジェクトを検索します。
3. バッチ・アプリケーションを追加する場合は、次のオプションのいずれかをクリックし、[Select]をクリックします。

- Select Versions (バージョンの選択)
特定のバージョンのオブジェクトのみを選択する場合にクリックします。
- JDE Versions (JDE バージョン)
J.D. Edwards バージョンのオブジェクトをすべて選択する場合にクリックします。
- All Versions - オブジェクトのすべてのバージョンを選択します。
全バージョンのオブジェクトを選択する場合にクリックします。

バッチ・アプリケーションを追加するときに[Select Versions (バージョンの選択)]をクリックすると、〈Version Selection (バージョンの選択)〉フォームが表示されます。



4. 〈Version Selection〉フォームで、パッケージに組み込む特定のバージョンをリストから選択します。
選択した各バージョンの左側にチェックマークが表示されます。
5. バージョンの選択を完了したら、[Close]をクリックして〈Object Component Selection〉フォームに戻ります。
6. パッケージへのオブジェクトの追加が完了するまで、このプロセスを繰り返します。
7. 完了したら、[Close]をクリックして、〈Object Component Selection〉フォームにアクセスした場所に戻ります。

フィールド記述

記述

SARNo.

用語解説

ソフトウェア・アクション・リクエスト(SAR)の略語。この番号は、各 SAR 番号ごとにかかった費用および活動をトラッキングするために、JDE プロジェクト管理システムで使用されます。

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
記述	ソフトウェア・バージョン・リポジトリ・ファイル内のレコード記述。メンバ記述は基本メンバの記述と一致させてください。
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。
状況コード	このコードは、開発中のソフトウェアの状況を示します。
ユーザー	ユーザー・プロファイルを識別するコード。
マージオプション	マージオプションは、得意先の ONEWORLD オブジェクトが J.D. EDWARDS の ONEWORLD オブジェクトとマージされるかどうかを指定します。マージオプションは、そのパスにチェックインされたすべてのオブジェクトが、パスワードと同じマージオプションを所有するようにパスワードレベルで設定できます。
修正日付	ドリームライター・バージョン、ソフトウェアバージョンリポジトリレコードなどのオブジェクトが最後に変更された日付。
更新時刻	前回オブジェクトをチェックインした時間
システムコード	システム・コードを示すユーザー定義コード(98/SY)
レポートシステム	レポートおよびアプリケーション一時変更のシステム番号を示すユーザー定義コード(98/SY)。
すべてのバージョン	ソフトウェアのバージョン識別名
JDE バージョン	ソフトウェアのバージョン識別名

バージョン ID	アプリケーションやレポートの実行方法の指定に使用するユーザー定義のスペックです。バージョンを使用することで、ユーザー定義の処理オプション値やデータ選択、順序オプションなどをグループ化して保存します。対話型バージョンは(通常、タスク・レベルで)アプリケーションと関連付けられています。バッチバージョンはバッチ・プログラムまたはレポートと関連付けられています。バッチ・プログラムを実行する場合はバージョンを選択する必要があります。
バージョン・タイトル	バージョン番号の横に表示されるバージョンの記述。バージョン・タイトルとレポート・タイトルは異なります。このフィールドでは、バージョンの用途を記述します。たとえば、ピッキング・リストを作成するアプリケーションのバージョンの 1 つは[ピッキング・リスト - 会計]、もう 1 つは[ピッキング・リスト - 在庫管理]とすることができます。

パッケージへの機能の追加

機能とは、アプリケーションやその他の機能をサポートするために、ワークステーションやサーバーにコピーされる、レジストリ設定などの一連のファイルや構成オプションを意味します。オブジェクトと同様に、機能はパッケージに組み込まれ、機能コンポーネントを必要とするワークステーションとサーバーに配布されます。

パッケージへの機能の追加が完了すると、それらの機能は、〈Package Assembly Director〉の〈Feature Component〉フォーム、〈Package Component Revision〉フォーム、および〈Work with Packages〉フォームに表示されます。

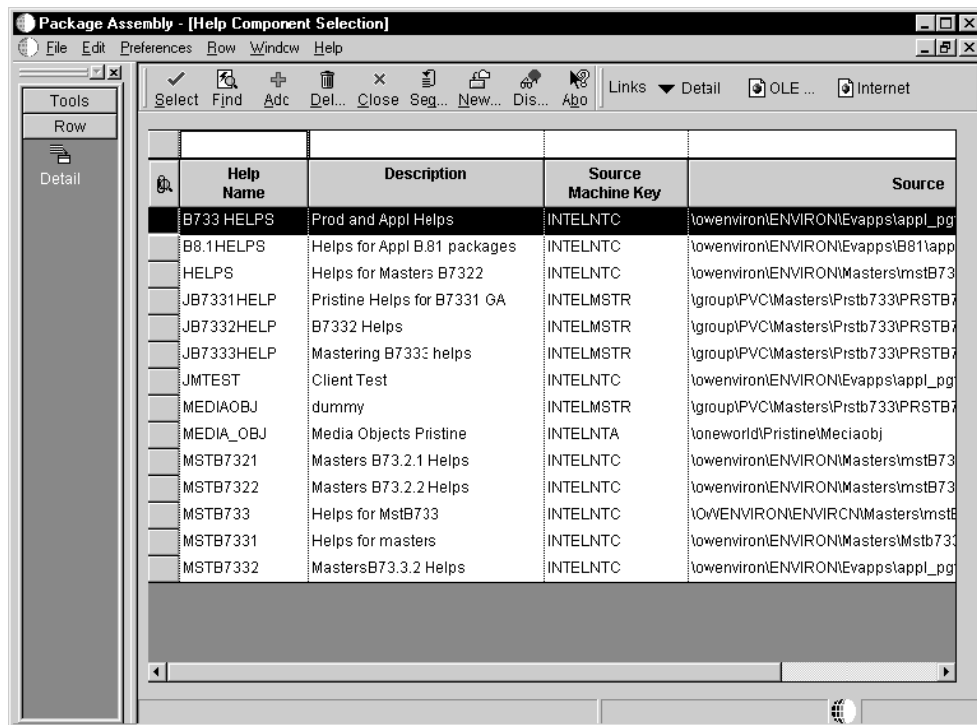
このタスクでは、機能をパッケージに追加する方法について説明します。機能をはじめて追加する場合も、以前に組み立てたパッケージを改訂する場合も手順は同じです。

参照

- 機能の追加については、「パッケージへの機能の組み込み」

▶ 機能をパッケージに追加するには

1. 次のいずれかの方法で、〈Feature Component Selection〉フォームを表示します。
 - 〈Feature Component〉フォームで、[Browse]をクリックします。
 - 〈Package Component Revisions〉フォームで、[Features(機能)]アイコンをクリックして〈Feature Component〉フォームを表示し、[Browse]をクリックします。



2. パッケージに含める既存の機能を検索して選択し、[Select]をクリックします。
複数の機能を選択するには、[Control(コントロール)]キーか[Select]キーを押します。
3. 含める機能が定義されていない場合は、[Add]をクリックすると、機能定義を作成できます。
〈Feature Based Deployment Director(機能ベース・デプロイメント・ディレクタ)〉が起動します。この機能を使用して新機能を作成できます。
4. 機能をパッケージに追加し終えるまで、ステップ 2 と 3 を繰り返します。
5. 作業を完了したら、[Close]をクリックして、〈Feature Component Selection〉フォームを表示したフォームに戻ります。

パッケージのアクティブ化

組立てを完了した後も、パッケージ状況は「Assembly(アセンブリ)」のままです。定義中のパッケージは非アクティブになっています。パッケージをアクティブにするには、パッケージ状況を明示的に「Assembly-Definition Complete(アセンブリ定義完了)」に変更する必要があります。組み立てられたパッケージは、その状況が「Assembly-Definition Complete」に変更されるまでビルドできません。この状況は、パッケージの組立てが完了し、ビルド定義プロセスを開始できることを示します。

ビルド定義プロセスを開始する前であればいつでもパッケージ状況を変更できます。つまり、組み立てたパッケージを改訂する必要がある場合は、パッケージ状況を「Assembly-Definition Complete」に変更した後でも、それを「In Definition」に戻すことができます。

▶ パッケージをアクティブにするには

〈Package and Deployment Tools〉メニュー(GH9083)から〈Package Assembly〉プログラム(P9601)を選択します。

1. 〈Work with Packages〉フォームで、アクティブにするパッケージを選択します。

現行状況が「In Definition」のパッケージは、オープン・ボックスのアイコンで表示され、状況が「Assembly-Definition Complete」のパッケージは、クローズ・ボックスのアイコンで表示されます。

2. [Row]メニューから[Active/Inactive(アクティブ/非アクティブ)]を選択します。

状況が「Complete(完了)」のパッケージを「In Definition」に戻す場合も、これと同じ手順を使用できます。

注:

パッケージ・ビルドを定義する準備が整ったら、「パッケージ・ビルドの定義」に記載されているステップを実行してください。

パッケージのビルド

パッケージをアクティブにした後に、ビルドを投入してパッケージのビルドを開始できます。この処理には、パッケージ内のオブジェクト数に応じて数時間かかることがあります。

パッケージのビルド中は、セントラル・オブジェクト・データ・ソースからパッケージにコピーされたスペック・レコードの数がカウントされます。親パッケージ内の当初のスペック・レコード数と共に、コピーされた各オブジェクトのスペック・レコード数を示すリストがレポートに表示されます。この数を比較して、パッケージが正常にビルドされたことを確認できます。クライアント・パッケージの場合は、コピーされたレコードの数はスペック・ファイル数と同じになるはずですが、サーバー・パッケージの場合は、コピーされたレコードの合計数から GBRSPEC を除いた数で、当初のレコード数と同じになります。

▶ パッケージ・ビルドを開始するには

1. パッケージ・ビルドを開始する準備が整ったら、〈Work With Package Build Definition(パッケージ・ビルド定義の処理)〉で、[Row]メニューから[Submit Build]を選択します。
2. 次のオプションのいずれかをクリックして、[OK]をクリックします。
 - On Screen(画面)
 - To Printer(プリンタ)

フォームが自動的に閉じて、パッケージのビルドが開始されます。ビルドの時間は、パッケージに含まれる項目数とサイズによって異なります。ビルドが終了すると、指定した出力先に応じて、レポートが画面に表示されるか、印刷されます。

この処理中には、セントラル・オブジェクトからパッケージにコピーされたスペック・レコードの数もカウントされます。

3. パッケージ内にすべての要素が正しくビルドされたことを確認するために、レポートを検討します。次のいずれかが問題を示します。

- エラー・メッセージ
- クライアント・パッケージの場合は、コピーされたレコードの数はスペック・ファイル数と異なっていないかどうかを調べます。
- クライアント・パッケージの場合は、コピーされたレコードの数が当初の数と異なっていないかどうかを調べます。

ただし、GBRSPEC の場合、コピーされたレコードの数は当初の数と一致しません。

4. レポートに上記の問題がある場合は、エラー・ログで詳細を検討してください。

注:

パッケージ・ビルドが正常に終了した場合は、「配布」に記載されている手順に従って、パッケージ・デプロイメントをスケジュールできます。

パッケージの検証

組立てを完了した後、パッケージが正常にビルドされたかどうかを検証できます。この検証を使用すると、ビルドの投入前にパッケージをテストしたり、パッケージ・ビルドに失敗した場合にビルド・プロセスのエラーを解決することができます。

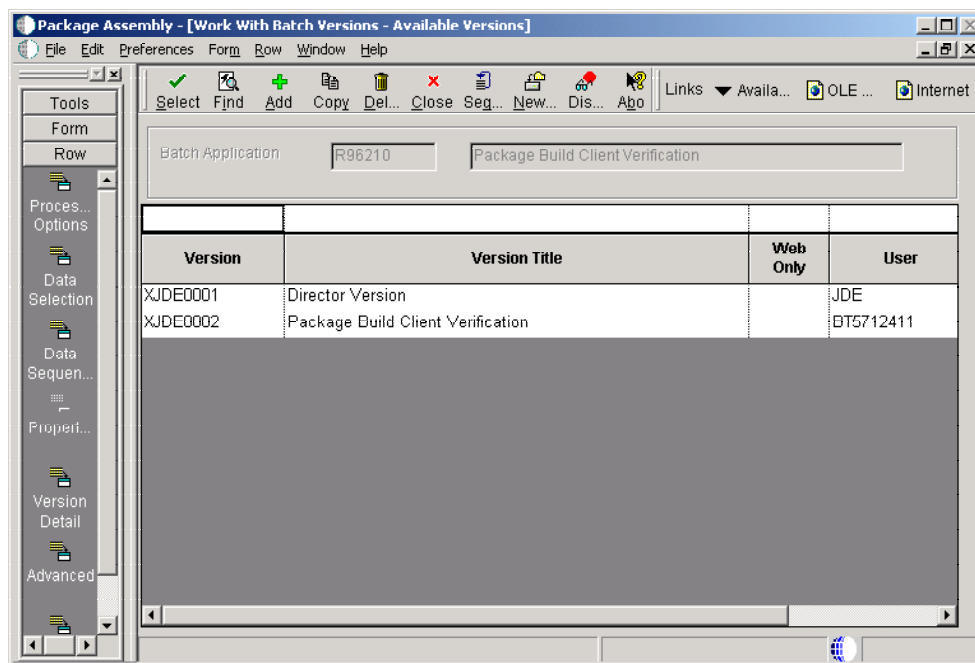
この検証プロセスの対象事項は次のとおりです。

- 空きディスク領域が十分にあるかどうか。
- セントラル・オブジェクトとパッケージ・ビルドのテーブルにアクセスできるかどうか。
- ユーザーにデプロイメント・サーバーとエンタープライズ・サーバー上でディレクトリを作成するための許可が付与されているかどうか。
- 必要なサービス・パックがインストールされているかどうか。
- 必要な MDAC がインストールされているかどうか。
- マシン・テーブルがセットアップされているかどうか。
- 必要なバージョンのコンパイラがインストールされているかどうか。
- エンタープライズ・サーバー・ポートにアクセスできるかどうか。
- jde.ini ファイルのデバッグ・レベルが、クライアントとエンタープライズ・サーバーに適した値に設定されているかどうか。

▶ パッケージを検証するには

〈Package and Deployment Tools (パッケージとデプロイメント・ツール)〉メニュー(GH9083)から〈Package Assembly〉(P9601)または〈Package Build〉(P9621)を選びます。

1. [Row]メニューから[Build Verification]を選びます。



2. 〈Printer Selection〉で、使用するプリンタを選んで[OK]をクリックします。

環境とパッケージの構成が検証されます。

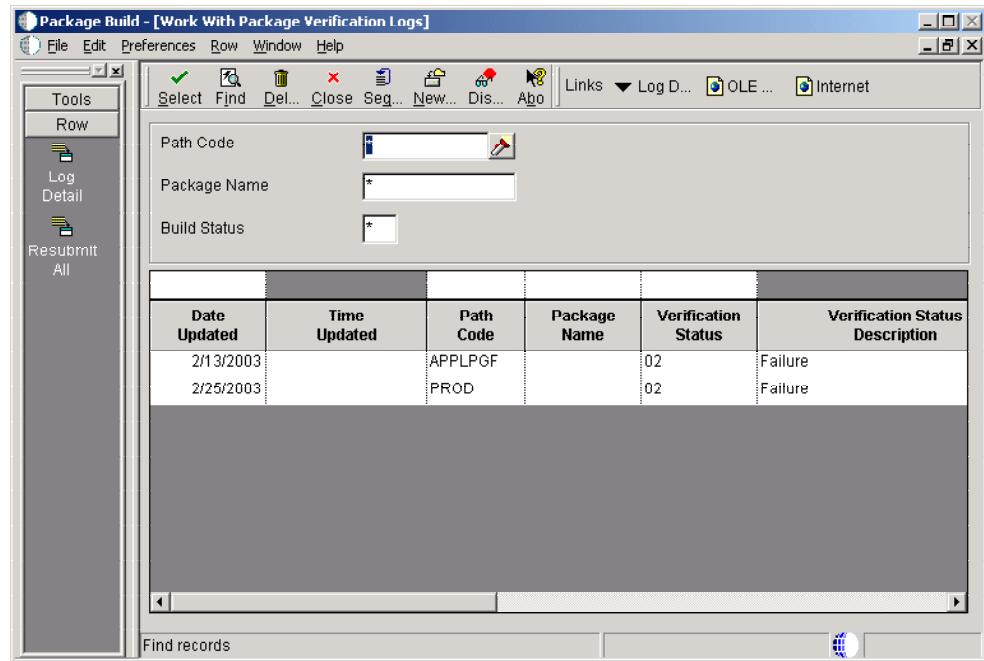
検証ログの表示

〈Build Verification〉を実行すると、検証プロセスの詳細レポートが作成されます。このレポートの各行には、チェックされた項目、テストの正否、およびテストが失敗した場合はエラー訂正の推奨手順が示されます。検証ログはすべて保管され、〈Package Assembly〉プログラム(P9601)または〈Package Build〉プログラム(P9621)の[Form]メニューから[View Verification (検証の表示)]を選択してアクセスできます。

▶ 検証ログを表示するには

〈Package and Deployment Tools〉メニュー(GH9083)から〈Package Assembly〉(P9601)または〈Package Build〉(P9621)を選択します。

1. [Form]メニューから[View Verification]を選びます。



2. [Path Code]、[Package Name(パッケージ名)]または[Build Status(ビルド状況)]フィールドに値を入力して[Find]をクリックします。

指定した条件と一致する検証ログと検証状況である[Success(成功)]または[Failure(失敗)]を示すリストが表示されます。

3. 検証ログを選んで、[Row]メニューの[Log Detail(ログ詳細)]をクリックします。

既存パッケージの改訂

パッケージを組み立てた後でも、〈Package Component Revision〉フォームを使用すると、パッケージに含まれているコンポーネントをいずれも容易に改訂できます。パッケージを改訂するときは、〈Package Assembly Director〉のすべてのフォームを使用する必要はありません。

はじめる前に

- パッケージ状況が「In Definition」であることを確認します。状況が「Assembly-Definition Complete」のパッケージを改訂しようとする、エラー・メッセージが表示されます。パッケージ状況を変更するには、〈Work with Packages〉フォームの[Row]メニューから[Active/Inactive]を選択します。

▶ 既存パッケージを改訂するには

〈Package and Deployment Tools〉メニュー(GH9083)から〈Package Assembly〉を選択します。

1. 〈Work with Packages〉フォームで、[Find]をクリックするか、またはパッケージ・ツリーを展開して、改訂するパッケージを見つけます。
2. 改訂するパッケージを選択し、[Row]メニューから[Package Revisions(パッケージの改訂)]を選択します。

3. 〈Package Component Revisions〉フォームで、必要な変更を行います。
4. パッケージ定義の改訂が完了したら、[OK]をクリックして〈Work with Packages〉フォームに戻ります。

パッケージのビルド情報が存在する場合は、変更により既存のビルド情報が削除されるという警告が表示されます。

5. 次のいずれかをクリックします。
 - [OK] – 改訂を承認し、既存のビルド情報を削除する。改訂を承認する場合は、ビルド情報を更新して、行った変更を反映する必要があります。
 - [Cancel(キャンセル)] – 改訂を削除し、既存のビルド情報を保存します。

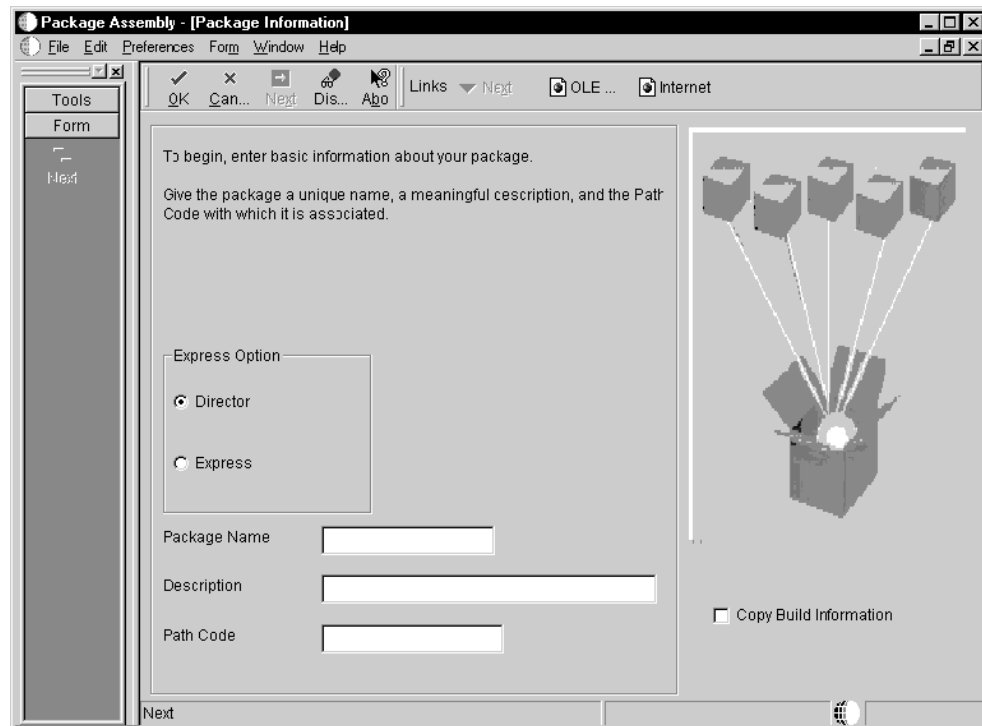
パッケージのコピー

パッケージを新規作成するよりも、組み立てたパッケージをコピーするほうが適切な場合があります。また、ビルドされているパッケージ(「Build Complete(ビルド完了)」状況のパッケージ)をコピーすると、オリジナルのパッケージの定義レコードもコピーされ、新しいパッケージに追加されます。旧パッケージのディレクトリ(サーバー・ディレクトリを含む)も新規パッケージにコピーされます。

▶ パッケージをコピーするには

〈Package and Deployment Tools〉メニュー(GH9083)から〈Package Assembly〉プログラム(P9601)を選択します。

1. 〈Work with Packages〉フォームで、コピーするパッケージの名前を検索して選択し、[Copy(コピー)]をクリックします。



2. 〈Package Information〉フォームで、新規パッケージの名前、記述およびパス・コードを入力します。

コピーするパッケージがビルドされている場合(「Build Complete」状況のパッケージ)は、[Copy Build Information(ビルド情報のコピー)]オプションを利用できます。このオプションをオンにすると、オリジナル・パッケージのすべてのパッケージ定義レコード、ディレクトリおよびサーバー・ディレクトリが新規パッケージにコピーされます。

3. [OK]をクリックして、〈Work with Package〉フォームに戻ります。このフォームの[Row]メニューから[Package Revisions]を選択すると、パッケージ・コンポーネントを検討し、変更できます。

フィールド記述

記述	用語解説
ビルド情報のコピー	パッケージをコピーする場合は、パッケージ定義のほかにコピーする先のディレクトリも指定するかどうか指示してください。パッケージ・ディレクトリをコピーする場合、新しいパッケージは元のパッケージとまったく同じため、ユーザーに展開する前に新しくビルドする必要はありません。

パッケージの削除

〈Work With Packages〉フォームには、ビルド・プロセスが実行されていない既存パッケージを削除するオプションが含まれています。ビルド実行中のパッケージを削除することはできません。

パッケージを削除すると、そのパッケージに関するすべての情報が、ソフトウェア・パッケージ・ヘッダー・テーブル(F9603)とソフトウェア・パッケージ詳細テーブル(F9631)から削除されます。さらに、パッケージがビルドされている場合は、そのパッケージに関するすべてのスペックがすべてのサーバーから削除され、ソフトウェア・パッケージ・ビルド・ヘッダー・テーブル(F96021)とソフトウェア・パッケージ・ビルド詳細テーブル(F9622)内の情報もすべて削除されます。また、既存のデプロイメント・レコードも削除されます。

▶ パッケージを削除するには

〈Package and Deployment Tools〉メニュー(GH9083)から〈Package Assembly〉を選択します。〈Work with Packages(パッケージの処理)〉フォームが表示されます。

1. 削除するパッケージの名前を検索して選択します。
2. [Delete(削除)]をクリックします。
3. 警告メッセージが表示された場合、[OK]をクリックすることにより削除を確認します。

パッケージ・ビルドの定義

パッケージを組み立てた後、ワークステーションに配布する前にパッケージ・ビルドを定義する必要があります。ビルド・プロセスでは、パッケージで定義したパス・コードのセントラル・オブジェクト・データ・ソースを読み取ります。この情報はリレーショナル・フォーマットからレプリケートされたオブジェクトに変換され、パッケージ自体におさめられます。

B9(リリース名)ディレクトリ構造の例を次に示します。アスタリスクで始まるディレクトリは、開発者が開発オブジェクトをチェックインするディレクトリを表しています。

PD9(パス・コード名)

PACKAGE

PackageA(パッケージ名)

bin32

include

lib32

make

obj

res

source

spec

work

*bin32

*include

*lib32

make

*obj

*res

*source

work

パッケージ名ディレクトリの下位のディレクトリは、パッケージのビルド時に自動作成されます。
source および include ディレクトリの情報は、開発者が開発オブジェクトをチェックインするデプロイメント・サーバー上のパス・コードの下位ディレクトリからコピーされます。他のすべてのディレクトリの情報は、セントラル・オブジェクト・データ・ソースから提供されます。bin32、lib32、obj の各ディレクトリは、ビジネス関数ビルド・プロセスの出力と共に自動作成されます。

ビルド・プロセスの理解

パッケージのビルド・プロセスは、数時間かかることがあります。このため、できるだけ就業時間後に実際のパッケージ・ビルドを開始することをお勧めします。パッケージをビルドする場合に必要なタスクのチェックリストを次に示します。

- オブジェクトの転送

ビルドに組み込むすべてのオブジェクトが適切なパス・コードに転送されていることを確認します。

- パッケージ用データベースに最新のレプリケートされたデータが存在することの確認

- パッケージのビルド

オブジェクトが転送されたパス・コードを使用してパッケージをビルドします。

- 相互参照ビルドの実行(任意)

相互参照情報が変更済みオブジェクトを反映していることを確認するために、相互参照ビルドを実行します。このプロセス実行には最高で 15 時間を要します。相互参照ビルドが終了する前に、パッケージを配布できます。

- ソフトウェアを次の場所に配布する

- ワークステーションおよびサーバー
- 階層デプロイメント・ロケーション

参照

- オブジェクトの転送方法については「オブジェクト移動の理解」

- レプリケートされたデータの検証については、『システム・アドミニストレーション』ガイドの「データ・レプリケーション」

- 相互参照ビルドの実行については、『開発ツール』ガイドの「相互参照機能」

- パッケージの配布については「配布」

Package Build Definition Director の理解

〈Package Assembly Director〉と同様に、〈Package Build Definition Director〉では、プロセスを先導する一連のフォームが表示されるため、ビルド定義プロセスを簡単かつ迅速に実行することができます。また、〈Package Assembly Director〉の場合と同様に、[Next]または[Previous(前へ)]をクリックすると、いつでも前後のフォームに移動できます。また、[Cancel]をクリックするとビルド定義プロセスを中止することができます。

次の表に、〈Package Build Definition Director〉の各フォームの機能の概要を示します。

〈Package Build Definition Director(パッケージ・ビルド定義ディレクタ)〉フォーム	〈Package Build Definition Director〉に関する基本情報を確認します。
〈Package Selection(パッケージの選択)〉フォーム	ビルドする定義済みパッケージを選択します。パッケージ状況は、「Assembly Definition Complete」であることが必須です。
〈Package Build Location(パッケージ・ビルド・ロ	クライアント・ワークステーション、1 つ以上のサーバー、またはクライアント

ケーション))フォーム	とサーバーの両方のうち、どれを対象にしたパッケージをビルドするかを指定します。
〈Server Selection(サーバーの選択))フォーム	サーバー・ロケーションを指定するために、このフォームを使用します。(サーバー・パッケージをビルドする場合は、サーバー・ロケーションが必要です)。
〈Build Specification Options(ビルド・スペック・オプション))フォーム	パッケージにすべてのスペック・テーブルを含めるか、選択したテーブルのみを含めるかを指定します。ビルドが失敗し、パッケージ・エラー・ログにリビルドする必要がある各スペック・ファイルが示されている場合、このフォーム上に表示される各スペックをビルドするためのオプションを使用すると便利です。
〈Individual Specification Selection(各スペックの選択))フォーム	選択したテーブルのみをパッケージに組み込む場合に使用します。
〈Business Function Options(ビジネス関数オプション))フォーム	ビジネス関数をビルドするために使用します。ビルド・モード、ビルド・プロセスを中断する重要度レベル、ビジネス関数ドキュメンテーションをビルドするかどうか、およびビルド前に出力先をクリアするかどうかを指定することもできます。
〈Compression Options(圧縮オプション))フォーム	パッケージの圧縮を指定し、ディレクトリ(すべてまたは個別)、データ、ヘルプおよびファンデーションを圧縮するかどうかを指定します。
〈Individual Directory Selection(各ディレクトリの選択))フォーム	圧縮するディレクトリを個別に選択する場合に使用します。
〈Build Features(ビルド機能))フォーム	パッケージに追加した機能のファイルセットと圧縮情報を入力します。機能とは、アプリケーションやその他の機能をサポートするために、ワークステーションやサーバーにコピーする必要がある、レジストリ設定などの一連のファイルや構成オプションを意味します。
〈Package Build Revisions(パッケージ・ビルドの改訂))フォーム	パッケージで指定したオプションの見直しや変更を行う場合に使用します。

パッケージ・ビルド時のビジネス関数のビルド

ビルド定義プロセスの中で、ビジネス関数をビルドするかどうかを指定できます。ビジネス関数のビルドを指定した場合、パッケージのビルド・プロセス時にビジネス関数のグローバル・ビルドが実行されます。パッケージ・ビルドの一環としてビジネス関数をビルドするときは、パッケージのビルド後に、[Global Build(グローバル・ビルド)]メニューから[Build(ビルド)]を選択して、〈BusBuild〉プログラムを手動で実行するのと同じプロセスが実行されます。

ソース情報とヘッダー情報がパッケージ(source ディレクトリと include ディレクトリ)から取り込まれ、コンパイルされて、bin32、obj、lib32 の各ディレクトリに保存されます。ビジネス関数は、ワークステーション上ではなく、パッケージ内にビルドされます。[Compress Package(パッケージの圧縮)]オプションをオンにすると、ビジネス関数はビルド後に圧縮されます。

ビジネス関数のビルドでは、パス・コード、ファンデーション、および出力先に次のガイドラインが適用されます。

- ビジネス関数のビルド時には、パッケージで定義したパス・コードを使用します。
- ファンデーションは、パッケージに含まれているファンデーションと同じです。アップデート・パッケージでは、親パッケージのファンデーションです。

- ビルド出力先は、パッケージ自体の bin32、obj および lib32 ディレクトリです。
- フル・パッケージまたは部分パッケージ、またはビジネス関数を含むアップデート・パッケージをビルドするときは、必ずビジネス関数をビルドします。ビジネス関数をビルドしない場合、パッケージに含まれる連結 DLL は現行のものではありません。

アップデート・パッケージでは、ビジネス関数が個別にビルドされます。個々のビジネス関数がビルドされると、そのオブジェクトおよび同じ連結 DLL にある他のすべてのオブジェクトに対してグローバル・リンクが実行されます。グローバル・リンクは、該当するパッケージのパス・コードのチェックイン・ロケーションにあるすべてのオブジェクトに影響します。

同様に、部分パッケージで定義したビジネス関数も個別にビルドされ、チェックイン・ロケーションにあるすべてのオブジェクトにリンクされます。部分パッケージでは、パッケージ自体で定義されている連結 DLL だけでなく、すべての連結 DLL が提供されます。

注:

ビルド・プロセスでは、ビルドからの出力によってチェックイン・ロケーションのディレクトリを更新しません。

参照

- 〈BusBuild〉プログラムについては『開発ツール』ガイドの「ビジネス関数」

パッケージ・ビルドの定義

ここでは、〈Package Build Definition Director〉を使用してパッケージ・ビルドを定義する方法について説明します。

〈Package Build Definition Director〉には、[Package Assembly]メニュー項目からアクセスすることもできれば、[Package Build(パッケージ・ビルド)]メニュー項目からアクセスすることもできます。

このディレクトリに[Package Assembly]メニュー項目からアクセスした場合は、パッケージ名とその他の情報が自動的に入力されていますが、〈Work With Package Build Definition(パッケージ・ビルド定義の処理)〉フォームからアクセスした場合は、ビルドするパッケージの名前を指定する必要があります。

〈Pack Build Definition Director〉を起動する前に〈Work with Package Build Definition〉フォームを使用すると、以前にビルドされたパッケージに関する情報を検討できます。例えば、パッケージのプロパティ、ビルド・オプション、ビジネス関数オプション、圧縮オプションなどを検討できます。このフォームでは、他の親/子フォームと同様に、プラス(+)記号またはマイナス(-)記号をクリックして、フォームに表示するパッケージ情報の量を調節できます。

はじめる前に

- パッケージを組み立て、組み立てたパッケージ状況が「Assembly-Definition Complete」であることを確認します。詳細については、「パッケージの組立て」を参照してください。
- パッケージ・ビルド・プロセス時に生成される〈Package Build〉レポート(R9621)および〈Server Package Build(サーバー・パッケージ・ビルド)〉レポート(R9622)に対して、〈Object Configuration Manager (OCM)(オブジェクト構成マネージャ(OCM))〉マッピング機能が正しく

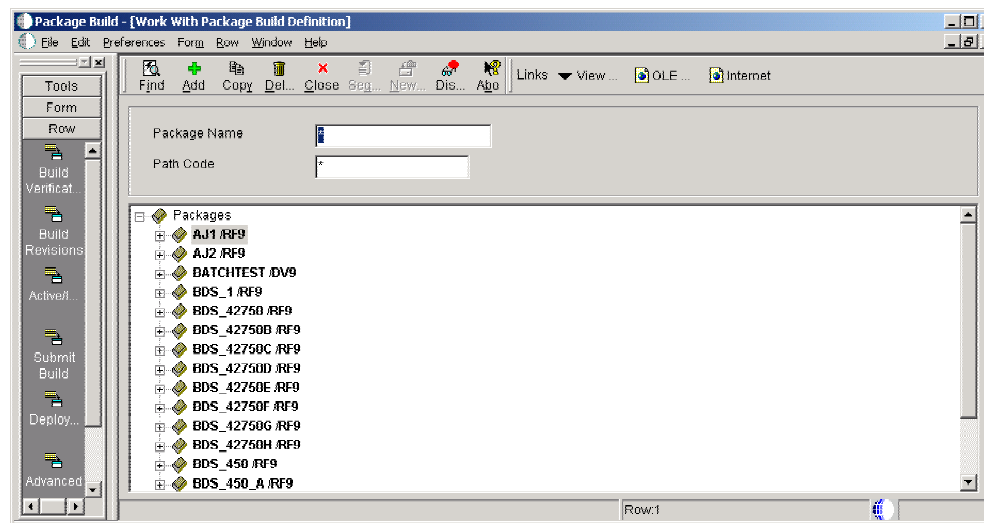
設定されていることを確認します。たとえば、これらのレポートをローカルで実行する場合、OCM マッピングが、パッケージ・ビルドが実行される環境の「Local(ローカル)」をポイントしていることを確認してください。OCM マッピングの設定の詳細については、『CNC インプリメンテーション』ガイドの「オブジェクト構成マネージャ(OCM)」を参照してください。

▶ パッケージ・ビルドを定義するには

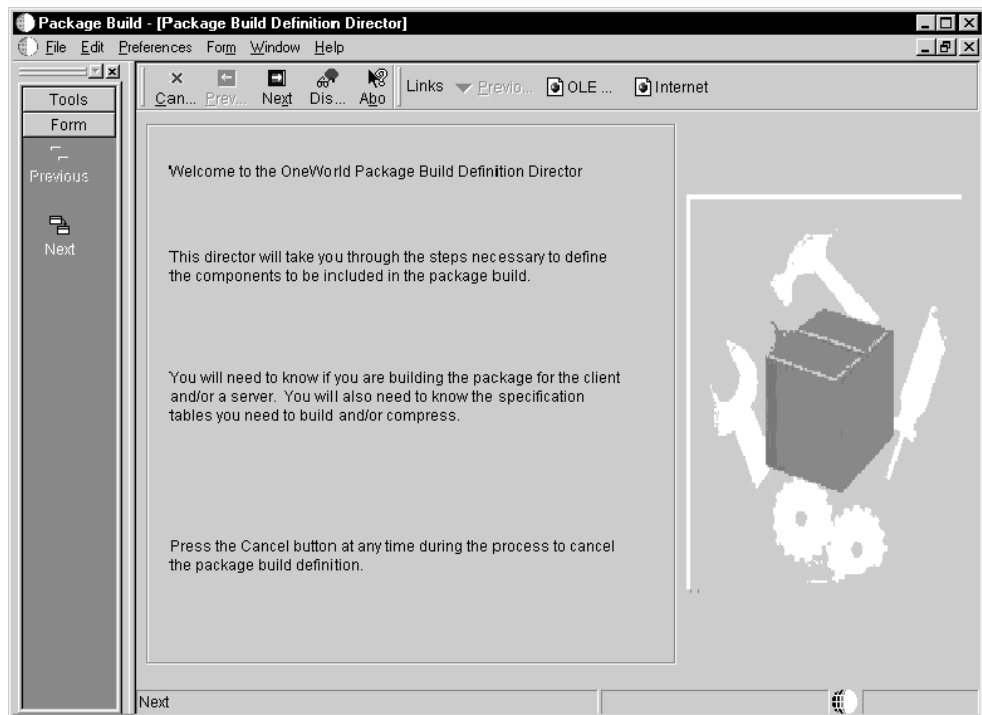
次の操作のいずれかを実行します。

〈Package and Deployment Tools〉メニュー(GH9083)から〈Package Assembly〉を選択します。〈Work With Packages〉で、「Definition Complete」状況が設定されている定義済みパッケージを選択し、[Row]メニューから[Build Director(ディレクトリの作成)]をクリックします。

〈Package and Deployment Tools〉メニュー(GH9083)から〈Package Build〉を選択します。

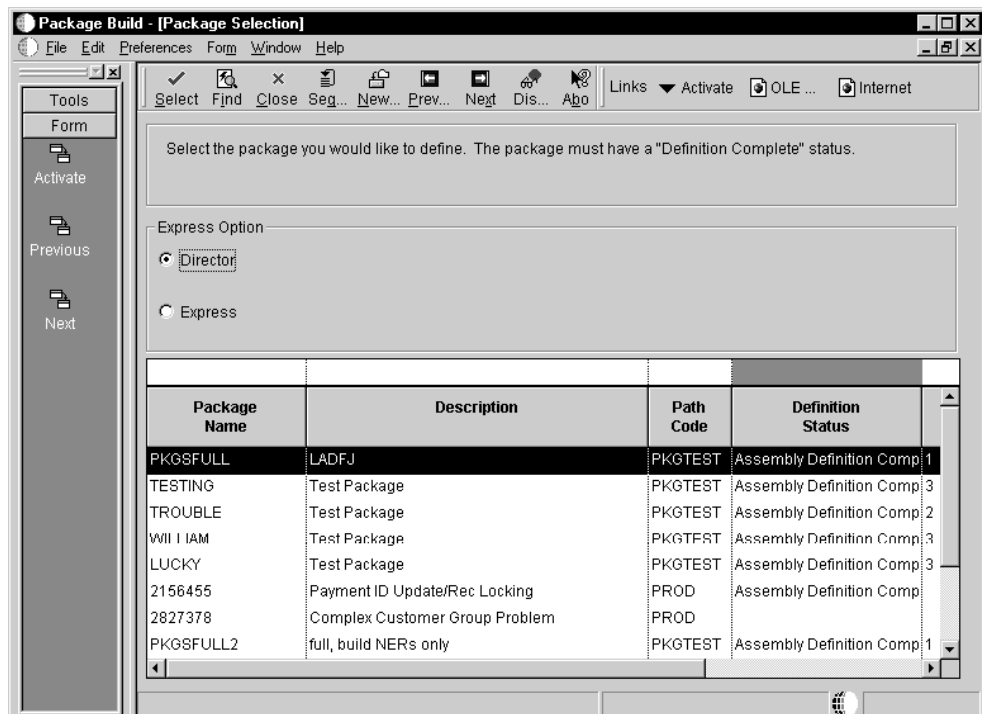


1. 〈Work with Package Build Definition〉フォームで、[Add]をクリックし、〈Package Build Definition Director〉を起動します。



〈Work With Packages〉からディレクトリにアクセスした場合は、ステップ 2 をとばしてください。

2. 〈Package Build Definition Director〉フォームで、[Next]をクリックします。



3. 〈Package Selection〉フォームで、ビルドする定義済みパッケージを検索して選択します。

パッケージ定義の状況が「In Definition」の場合は、パッケージをビルドする前に状況を「Definition Complete」に変更する必要があります。この場合、パッケージを選択してから、[Row]メニューから[Activate(アクティブ化)]を選択します。

4. [Express Option(高速オプション)]で、次のオプションのいずれかをオンにします。

- Director

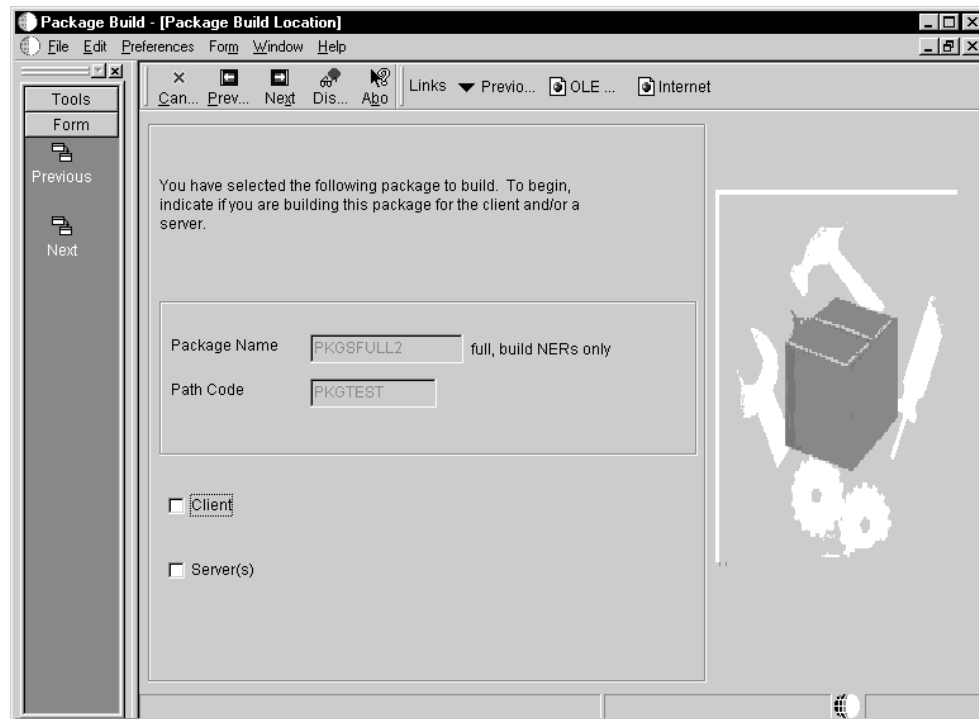
パッケージ・ビルドをカスタマイズする場合は、このオプションをオンにします。[Director(ディレクタ)]をオンにすると、〈Package Build Definition(パッケージ・ビルド定義)〉の各フォームを移動できます。

- Express

デフォルトのビルド・パラメータを受け入れる場合は、このオプションをオンにします。[Express(エクスプレス)]を選択すると、パッケージ・ビルドのデフォルトのオプションを受け入れて、パッケージ・ビルド定義の各フォームをスキップできます。

5. 次のいずれかを実行します。

- [Express]を選択した場合は、[Next]をクリックし、「パッケージ・ビルド選択を検討するには」に進んでください。
- [Director]を選択した場合は、[Next]をクリックして、このタスクの残りのステップを完了します。



6. 〈Package Build Location〉フォームで、次のオプションの一方または両方をクリックしてオンにし、[Next]をクリックします。

- Client(クライアント)
- Server(s)(サーバー)

部分パッケージは、サーバー用としてはビルドできません。

ワークステーション専用のパッケージをビルドしている場合は、ステップ 10 に進んでください。

〈Package Build Location〉フォームで[Server(s)(サーバー)]オプションをクリックした場合は、〈Server Selection〉フォームが表示されます。このフォームを使用すると、パッケージをビルドするサーバーを選択できます。

Server Name	Data Source Type	Database Name
A733AS2A	I	A733AS2A
A733AS4001A	I	A733AS1A
A733AS4002A	I	A733AS2A
ADEV ALPHANT	O	owsn1 devl
ADEV AS400	I	DEM073OCM
ADEV AS400 UBE	I	DEM073OCM
<input checked="" type="checkbox"/> ADEV HP9000	O	owsn1 devl
<input checked="" type="checkbox"/> ADEV HP9000 UBE	O	owsn1 devl
ADEV INTELNT	O	owsn1 devl
ADEV INTELNT UBE	O	owsn1 devl
ADEV RS6000	O	owsn1 devl
ADEV RS6000 4.3	O	owsn1 devl
ADEV SUN	O	owsn1 devl

7. サーバーを選択するには、[Find]をクリックしてから、サーバーの詳細行のロー・ヘッダーをダブルクリックします。

チェック・マークは、そのサーバーが選択されていることを示します。複数のサーバーを選択できます。

8. [Next]をクリックします。

9. 〈Build Specification Options〉フォームで、次のオプションをクリックしてオンにします。

- Build Specification Options

10. 次のオプションのうち 1 つをオンにします。

- All Specification Tables (全スペック・テーブル)
- Individual Specification Tables (各スペック・テーブル)

[All Specification Tables]を選択した場合は、〈Individual Specification Selection〉フォームに表示されているすべてのテーブルがパッケージに組み込まれます。すべてのスペック・テーブルをビルドする場合は、[Next]をクリックし、〈Business Function Options〉フォームに値を入力します。

11. 次のフィールドに値を入力します。

- Stop-Build Option (ビルド停止オプション)

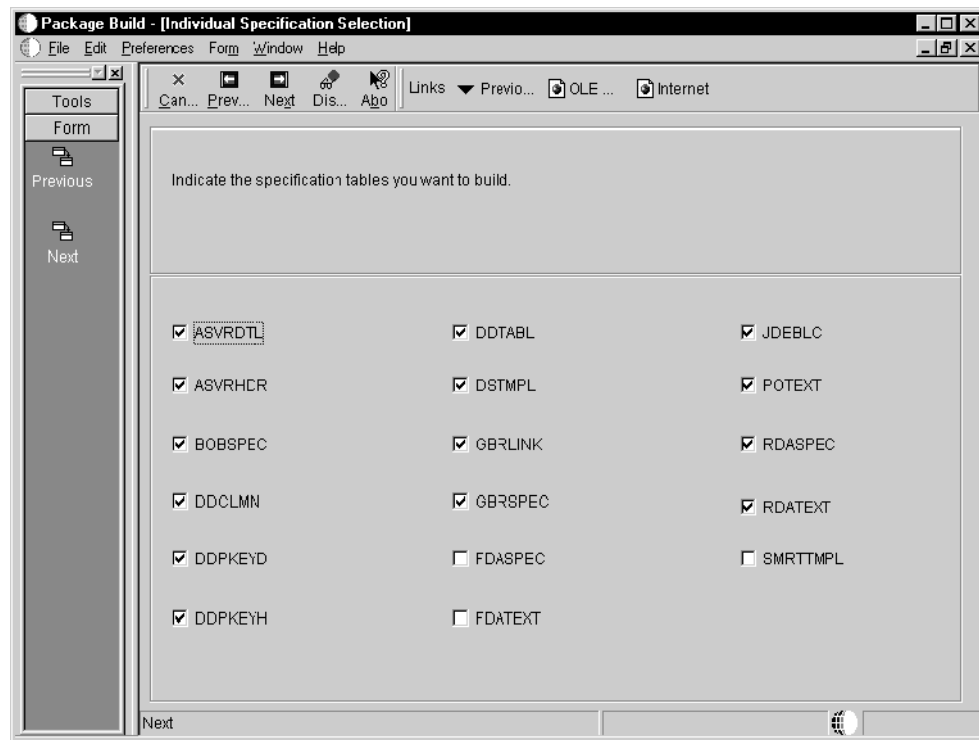
[Stop-Build Option]フィールドでは、B9 がビルドを停止する条件を指示できます。エラーが発生してもビルドを継続する、スペック・エラーが発生した場合にビルドを停止する、ビジネス関数エラーが発生した場合にビルドを停止する、またはエラーが存在する場合は圧縮しない、のいずれかを指定できます。

- Replace JDE.INI (JDE.INI の置換)

このフィールドは、アップデート・パッケージ専用です。

12. [Next]をクリックします。

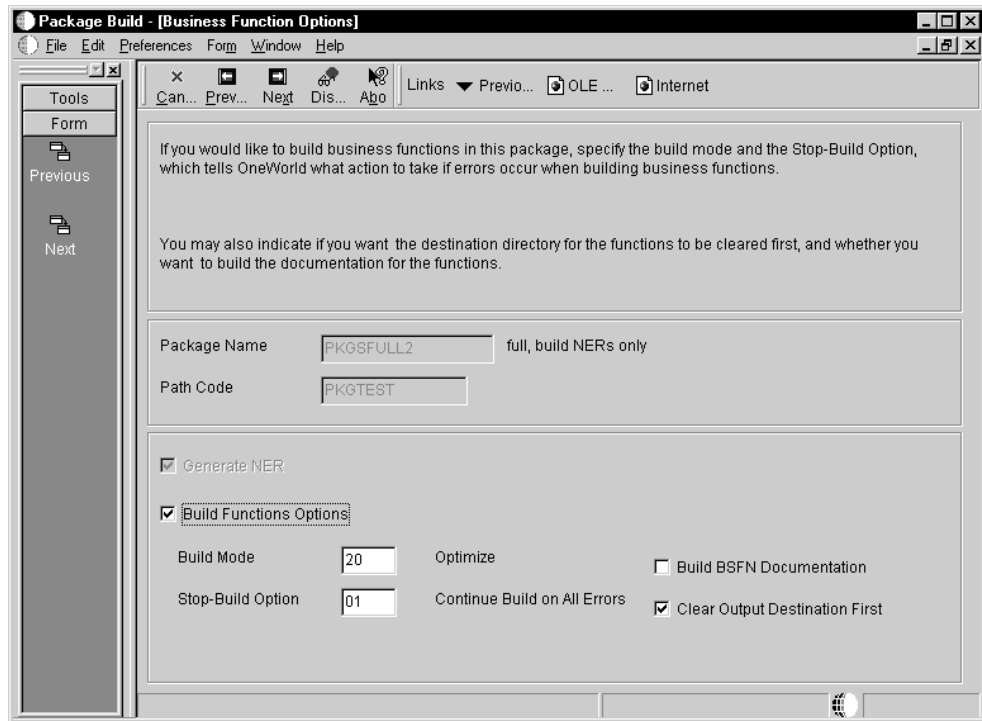
各スペック・テーブルのビルドを選んだ場合は、〈Individual Specification Selection〉フォームが表示されます。



13. ビルドしないスペック・テーブルを指示するには、該当するオプションをクリックしてオフにします。
複数のオプションをオフにできます。

14. [Next]をクリックします。

ビジネス関数、またはテーブル・イベント・ルールが設定されたテーブルを含むフル・パッケージやアップデート・パッケージでは、〈Business Function Options〉フォームが表示されます。

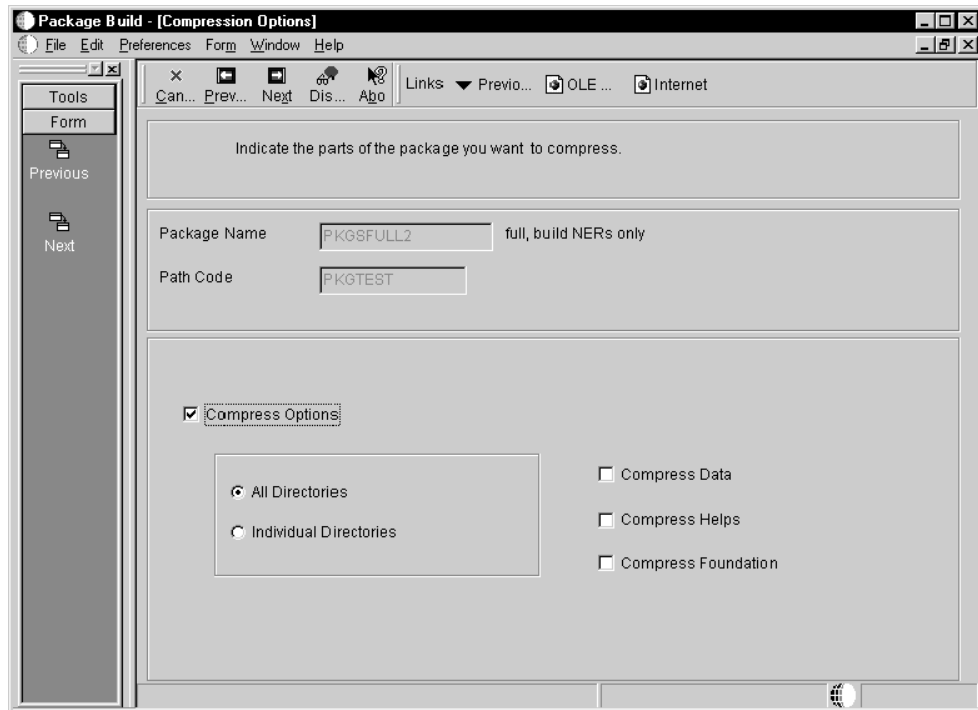


15. 〈Business Function Options〉フォームが表示されます。次のフィールドに値を入力します。

- Generate NER(NER の生成)
イベント・ルール・ビジネス関数を生成しない場合は、このオプションをオフにしてください。
- Build Functions Options(ビジネス関数ビルド・オプション)
- Build Mode(ビルド・モード)
[Debug(デバッグ)]、[Optimize(最適化)]および[Performance(パフォーマンス)]の3つのビルド・モードから選べます。[Debug]および[Performance]は、J.D. Edwards の開発者専用です。ユーザーは、[Optimize]モードを選択してください。
- Stop-Build Option(ビルド停止オプション)
- Build BSFN Documentation(ビジネス関数ドキュメンテーションのビルド)
このオプションは、オフになっていることを確認してください。
- Clear Output Destination(出力先のクリア)
このオプションは、フル・パッケージに適用されます。

16. [Next]をクリックします。

フル・パッケージまたは部分パッケージをビルドしている場合は、〈Compression Options〉フォームが表示されます。アップデート・パッケージをビルドしている場合、〈Compression Options〉フォームは表示されません。



17. 〈Compression Options〉フォームで、次のオプションをクリックしてオンにします。

- Compress Options(圧縮オプション)

18. 次のオプションのいずれかを選択します。

- All Directories(全ディレクトリ)
- サーバー・パッケージの場合

ビルド中のパッケージをサーバーに配布する場合は、次の場合にのみ[Compress Options]を選んでください。

- ワークステーション用とサーバー用に同じパッケージをビルドしており、ワークステーション・パッケージ用に圧縮ファイルを作成する必要がある場合。
- エンタープライズ・サーバー上でパッケージをビルドし、それを別のエンタープライズ・サーバーに配布する予定の場合。

サーバー・パッケージを圧縮する必要がある場合は、[Compress Options]をクリックし、[All Directories]を選択します。そうすると、エンタープライズ・サーバー上のディレクトリが圧縮され、デプロイメント・サーバー上の次のディレクトリ(server.ini ファイルが置かれているディレクトリ)内のパッケージにコピーされます。

〈パスコード〉/〈パッケージ〉/〈パッケージ名〉/〈オペレーティング・システム・タイプ〉

ディレクトリは、使用しているエンタープライズ・サーバーのタイプと互換性があるファイル・タイプに圧縮されます。NT サーバーの場合、ファイル・タイプは.cab、UNIX サーバーの場合は.z、AS/400 の場合はファイルに拡張子が付きません。

- クライアント・パッケージの場合

ディレクトリを圧縮すると、パッケージに含まれているアプリケーション・オブジェクトが自動的に圧縮されます。B9 は、ファンデーション、ヘルプ、データおよびアプリケーション・オブジェクトが圧縮されているかどうかを示すエントリをパッケージ INF ファイルで作成します。

- Compress Data (データの圧縮)

[Compress Data] を選ぶと、このパッケージに関連付けられている J.D. Edwards サポート対象ローカル・データベースが圧縮されます。

- Compress Helps (ヘルプの圧縮)

[Compress Helps (ヘルプの圧縮)] は、パッケージに関連付けられているヘルプ・ファイルを圧縮します。

- Compress Foundation (ファンデーションの圧縮)

[Compress Foundation] を選ぶと、パッケージに関連付けられているファンデーションが圧縮されます。

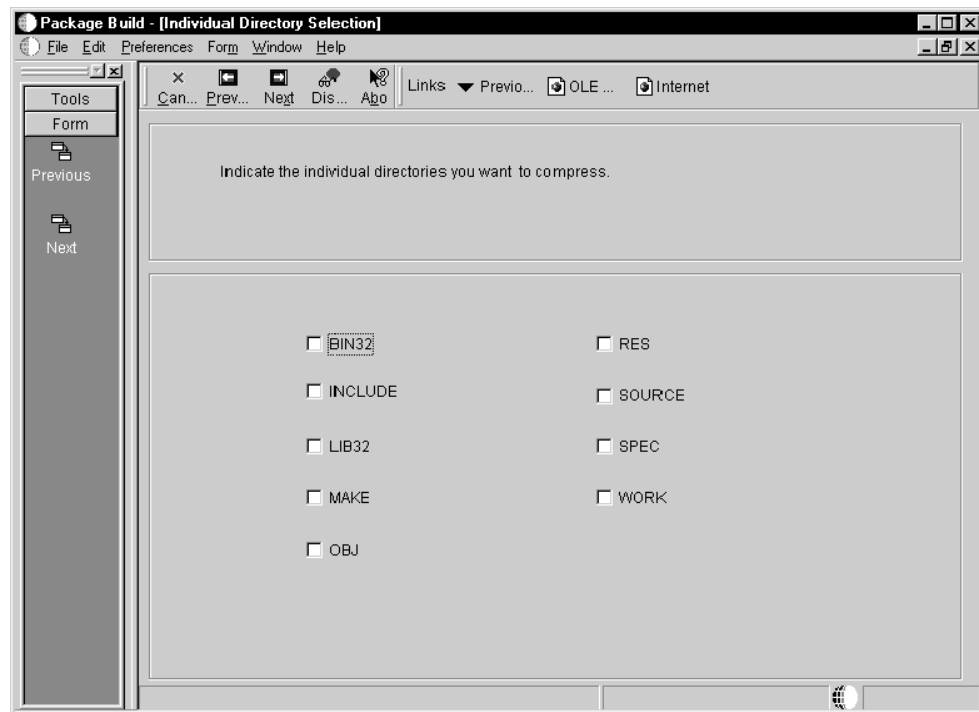
[All Directories] オプションの圧縮をオンにしている場合、〈Individual Directory Selection〉フォームに表示されているすべてのディレクトリが圧縮されます。すべてのディレクトリを圧縮する場合、次のステップはとばしてください。

重要

DoCompression が「1」に設定され、圧縮が有効になっていることを確認してください。この設定が「1」に設定されていない場合は、サーバー上で圧縮ファイルが作成されません。詳細については、「サーバー・パッケージ・ビルド用 JDE.INI 設定」を参照してください。

19. [Next] をクリックします。

各ディレクトリを圧縮するよう指定した場合は、〈Individual Directory Selection〉フォームが表示されます。

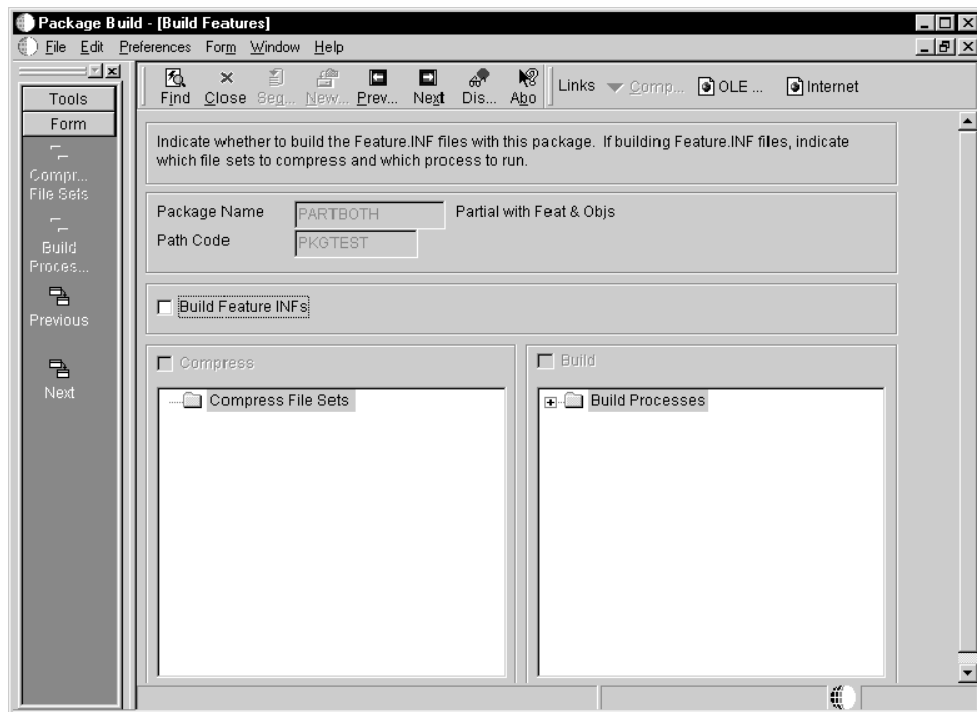


20. 〈Individual Directory Selection〉フォームで、圧縮するディレクトリを指示するには、該当するオプションをクリックしてオンにします。

複数のオプションをオンにできます。

21. [Next]をクリックします。

パッケージに機能を追加しない場合は、次のタスクに進んでください。



22. 〈Build Features(ビルド機能)〉フォームで、feature.inf ファイルをパッケージと共にビルドする場合は、次のオプションをクリックしてオンにします。

- Build Feature INFs(ビルド機能 INF)

このオプションをオンにすると、[Compress(圧縮)]オプションと[Build]オプションが使用可能になります。[Compress]オプションと[Build]オプションの詳細については、「パッケージ・ビルド定義時の機能の構成」を参照してください。

23. [Next]をクリックします。

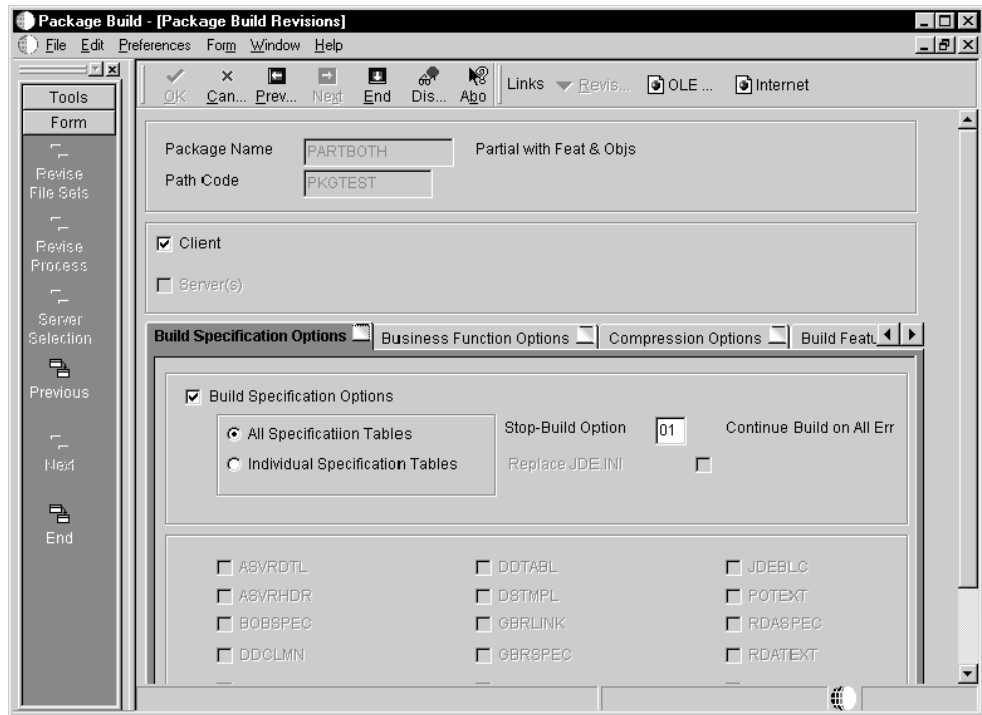
▶ パッケージ・ビルド定義を検討するには

パッケージ・ビルド定義の最後に、〈Package Build Revisions〉フォームが表示されます。

このフォームでは、パッケージのために指定した現在のビルド・オプション、ビジネス関数オプション、圧縮オプション、機能オプションを一覧表示できます。

1. これらのオプションを変更するには、変更するオプションのタイプに対応するタブをクリックします。

選択したオプションに対応するタブだけが、このフォームに表示されます。



2. 〈Package Build Revisions〉フォームで、ビルド・オプション、ビジネス関数オプション、圧縮オプション、機能オプションの検討または変更を完了したら、[End]をクリックして〈Package Build Definition Director〉を終了します。
3. 〈Work With Package Build Definition〉フォームで、[Row]メニューから[Active/Inactive]を選択してパッケージをアクティブ化します。
4. パッケージ・ビルドを開始する準備が整ったら、[Row]メニューから[Submit Build(ビルドの投入)]を選択します。
5. 次のオプションのいずれかをクリックして、[OK]をクリックします。

- On Screen
- To Printer

フォームが自動的に閉じて、パッケージのビルドが開始されます。ビルドの時間は、パッケージに含まれる項目数とサイズによって異なります。ビルドは、小さいパッケージの場合は5分程度で終了しますが、すべてのアプリケーションを含むフル・パッケージの場合は数時間かかることがあります。ビルドが終了すると、指定した出力先に応じて、レポートが画面に表示されるか、印刷されます。

6. パッケージ内にすべての要素が正しくビルドされたことを確認するために、レポートを検討します。レポートにエラーがある場合、エラー・ログで詳細を検討してください。

パッケージ・ビルドが正常に終了した場合は、「配布」に記載されている手順に従って、パッケージ・デプロイメントをスケジュールできます。

フィールド記述

記述	用語解説
サーバー名	データ・ソースの名前。
データ・ソースタイプ	データベースのタイプ
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。
ビルド・スペック・オプション	作成オプションを選択して、パッケージ定義をコピーしてセントラルデータソースからオブジェクトをワークステーションで複製されたフォーマットに変換します。また、ビルドプロセスに関連するオプションを入力します。
すべてのスペック	パッケージにすべてのスペックテーブルを作成するか、個々に選択するか指定します。
ビルド停止オプション	パッケージ・ビルド・プロセス中にエラーが発生した場合、システムが行う操作を指定します。
JDE.INI の置換	パッケージとともに新しい JDE.INI ファイルを取得するかどうか示します。使用している JDE.INI ファイルが変更されていないかぎり、チェック・マークを付けないでください。たとえば、更新したりリリース・マスターを構成したりする場合は、使用している JDE.INI ファイルを変更することがあります。
NER の生成	クライアントまたはサーバー・ビルドを再投入すると、NER を再ビルドするかどうか示されます。NER のビルドが完了すると、再ビルドせずに時間を節約できます。
関数のビルド・オプション	
ビルド・モード	ソフトウェアビルドに使用するコンパイラ構成を示します。
ビルド停止オプション	ビジネス関数ビルド・プロセス中にエラーが発生した場合、システムが行う操作を指定します。
BSFN ドキュメンテーションの作成	パッケージビルドプロセス中にビジネス関数ドキュメンテーションをビルドするかどうか示します。フルパッケージの場合、ドキュメンテーションはすべてのビジネス関数に対してビルドされます。部分または更新パッケージの場合、パッケージに含まれるこれらの関数に対してのみビルドされます。
出力先をまずクリア	ビジネス関数を作成する場合、パッケージ作成時に古いビジネス関数が含まれないよう、通常は、出力ディレクトリ (BIN32、LIB32、および OBJ) をクリアしてください。出力をクリアせずに関数の処理に失敗すると、グローバルリンクプロセスで、古いライブラリとオブジェクトを使用して連結 DLL ファイルが作成されます。この場合、グローバルパッケージが正しく作成されないことがあります。
パッケージの圧縮	パッケージに含めるアプリケーションを圧縮する際に選択するオプション。圧縮プロセスのオプションを指定する場合にも選択します。

すべてのディレクトリ	パッケージのすべてのディレクトリを圧縮するか、個々に選択するか指定します。
各ディレクトリ	パッケージのすべてのディレクトリを圧縮するか、個々に選択するか指定します。
データの圧縮	パッケージが作成された後にパッケージのデータを圧縮するかどうか指定します。
ファンデーションの圧縮	パッケージを作成したあとにパッケージのファンデーションファイルを圧縮するかどうか指定します。
ヘルプの圧縮	パッケージを作成した後にパッケージのヘルプファイルを圧縮するかどうかを指定します。

パッケージ・ビルド処理オプション(P9621)

〈Package Build Definition Director〉プログラム(P9621)には、次の処理オプションがあります。

- 1. Enter a value to determine how changes will occur(変更の反映方法を決定する値を入力してください。)

サーバーごとにビルド定義を変更できるかどうかを指定します。

この処理オプションに“1”を入力すると、個々のサーバーごとにビルド定義を変更できます。このオプションをブランクにしておくと、ビルド・オプションの改訂やその他の情報が、パッケージをビルドしているすべてのサーバーに適用されます。

具体的には、“1”を入力した場合は、〈Work With Package Build Definition〉フォームで、プロパティ、ビルド・オプション、ビジネス関数オプション、および圧縮オプションをサーバーごとに変更できます。このオプションをブランクにした場合、個々のサーバーに対しての変更はできませんが、パッケージは変更できます。これらの変更は、そのビルドに対応するすべてのサーバーに適用されます。

- 2. Mark this processing option with a 1 if this process is for Mastering purposes.(これがマスタリング・プロセスの場合は、この処理オプションに“1”を入力してください。)

プロセスが全ユーザーを対象としている場合はブランク、マスタリング・プロセスの場合は“1”を入力します。

- 3. Mark this processing option with a 1 if the Build Verification UBE is to be run prior to building all packages.(全パッケージのビルド前に〈Build Verification UBE〉を実行する場合は“1”を入力してください。)

パッケージのビルド前に〈Build Verification〉プログラムを自動的に実行する場合は、このフィールドに“1”を入力します。このオプションを選択すると、パッケージのビルド前に〈Package Build〉の構成がその都度検証されます。ビルドの検証に失敗すると、パッケージUBEは実行されません。〈Build Verification〉プログラムについては、「パッケージの検証」を参照してください。

パッケージの検証

組立てを完了した後、パッケージが正常にビルドされたかどうかを検証できます。この検証を使用すると、ビルドの投入前にパッケージをテストしたり、パッケージ・ビルドに失敗した場合にビルド・プロセスのエラーを解決することができます。

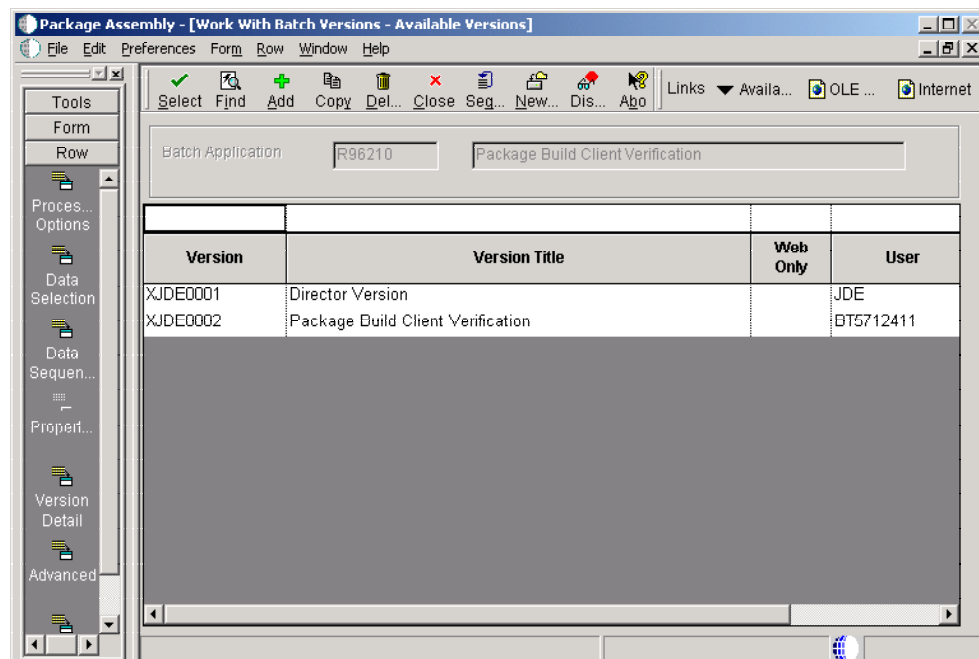
この検証プロセスの対象事項は次のとおりです。

- 空きディスク領域が十分にあるかどうか。
- セントラル・オブジェクトとパッケージ・ビルドのテーブルにアクセスできるかどうか。
- ユーザーにデプロイメント・サーバーとエンタープライズ・サーバー上でディレクトリを作成するための許可が付与されているかどうか。
- 必要なサービス・パックがインストールされているかどうか。
- 必要な MDAC がインストールされているかどうか。
- マシン・テーブルがセットアップされているかどうか。
- 必要なバージョンのコンパイラがインストールされているかどうか。
- エンタープライズ・サーバー・ポートにアクセスできるかどうか。
- jde.ini ファイルのデバッグ・レベルが、クライアントとエンタープライズ・サーバーに適した値に設定されているかどうか。

▶ パッケージを検証するには

〈Package and Deployment Tools〉メニュー(GH9083)から〈Package Assembly〉(P9601)または〈Package Build〉(P9621)を選択します。

1. [Row]メニューから[Build Verification]を選択します。



2. 〈Printer Selection〉で、使用するプリンタを選んで[OK]をクリックします。

環境とパッケージの構成が検証されます。

パッケージ・ビルド・オプションの改訂

パッケージ・ビルド・オプションを入力した後、これらのオプションはいずれも〈Package Build Revision (パッケージ・ビルドの改訂)〉フォームを使用して容易に改訂できます。ビルド・オプションを改訂するときは、〈Package Build Definition Director〉のすべてのフォームを使用する必要はありません。

▶ パッケージ・ビルド・オプションを改訂するには

〈Package and Deployment Tools〉メニュー(GH9083)から〈Package Build〉を選択します。

1. 〈Work With Package Build Definition〉フォームが表示されます。このフォームで、[Find]機能を使用するか、またはプラス(+)記号をクリックしてツリー構造を展開し、改訂するパッケージを検索します。
2. パッケージがアクティブな場合(そのパッケージ・アイコンが閉じている場合)は、パッケージを選択し、[Row]メニューから[Active/Inactive]を選択します。

パッケージが非アクティブになっていないと、改訂を実行できません。

3. パッケージを選択し、[Row]メニューから[Build Revisions(ビルドの改訂)]を選択します。

〈Package Build Revisions〉フォームが開き、選択したパッケージのビルド情報がすべて表示されます。

4. パッケージ・ビルド・オプション、ビジネス関数オプション、または圧縮オプションを変更するには、該当するタブをクリックします。完了したら[OK]をクリックします。
5. 〈Work With Package Build Definition〉フォームで、[Row]メニューから[Active/Inactive]を選択してパッケージをアクティブ化します。
6. ビルド・プロセスを開始する準備が整ったら、[Row]メニューから[Submit Build]を選択します。

親パッケージの圧縮(アップデート・パッケージのビルド後)

アップデート・パッケージをビルドしたときは、親パッケージの cab ファイルのバックアップが作成されます。親パッケージを配布する前に、親パッケージを再圧縮する必要があります。

ビルド済みパッケージのコピー

パッケージのビルド後、パッケージ・レコード、ヘッダーおよび詳細ファイルを含むパッケージ定義をコピーできます。既存パッケージに類似したパッケージを作成する場合に、コピー機能を使用すると便利です。このような場合は、パッケージ定義をコピーした上で、必要に応じてパッケージ・レコード、ヘッダーおよび詳細ファイルを修正することができます。

パッケージをコピーするときは、次の 2 つの方法のいずれかを選択できます。

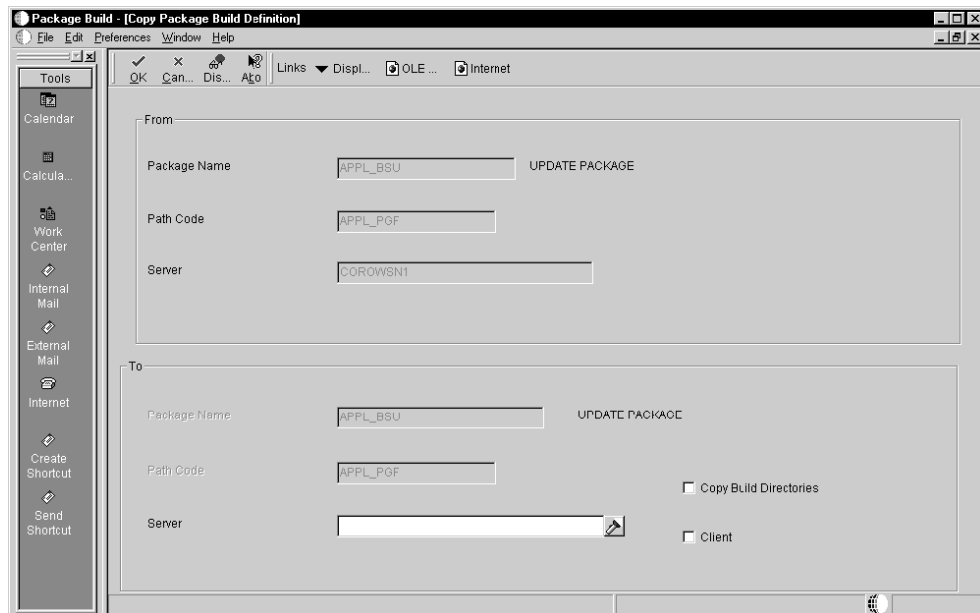
- 特定のサーバーに対するビルド定義をコピーする。あるサーバーから、別のサーバーまたはワークステーションに同一のパッケージをコピーする場合、この方法を使用してください。

- パッケージ全体のビルド定義をコピーする。既存パッケージに基づいて新規パッケージを作成する場合、この方法を使用してください。

▶ パッケージ・ビルド定義をサーバーからコピーするには

〈Package and Deployment Tools〉メニュー(GH9083)から〈Package Build〉を選択します。

1. 〈Work With Package Build Definition〉フォームが表示されます。コピーするパッケージを検索します。該当するパッケージの下で、定義をコピーするサーバーを選択します。
2. [Copy]をクリックします。



3. 〈Copy Package Build Definition (パッケージ・ビルド定義のコピー)〉フォームが表示されます。パッケージをコピーするサーバー名を入力します。

コピー元のパッケージ名、パス・コード、サーバー名と共に、コピー先のパッケージ名とパス・コードが表示されます。コピー先のパッケージ名とパス・コードはコピー元と同じです。

4. 次のオプション・フィールドに値を入力します。

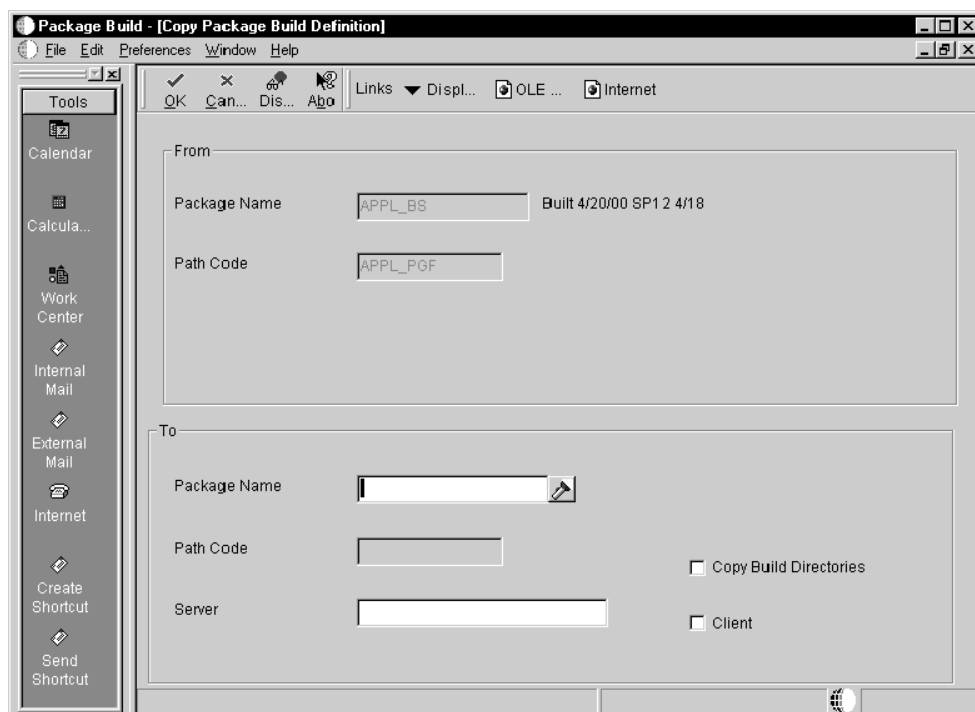
- Copy Build Directories (ビルド・ディレクトリのコピー)
パッケージ・ビルド・ディレクトリをコピーする場合にオンにします。
- Client
サーバーの代わりにクライアント・ワークステーションにコピーする場合にオンにします。

5. [OK]をクリックして〈Work With Package Build Definition〉フォームに戻ります。

▶ パッケージを新規パッケージにコピーするには

〈Package and Deployment Tools〉メニュー(GH9083)から〈Package Build〉を選択します。

1. 〈Work With Package Build Definition〉フォームが表示されます。このフォームで、コピーするパッケージを検索して選択します。
2. [Copy]をクリックします。



3. 〈Copy Package Build Definition〉フォームが表示されます。新規パッケージ名およびサーバー名を入力します。

コピー元のパッケージ名、パス・コード、サーバー名が表示されます。

4. 次のオプション・フィールドに値を入力します。

- Copy Build Directories(ビルド・ディレクトリのコピー)
パッケージ・ビルド・ディレクトリをコピーする場合にクリックします。
- Client
サーバーの代わりにクライアント・ワークステーションにコピーする場合にクリックします。

5. [OK]をクリックして〈Work With Package Build Definition〉フォームに戻ります。

パッケージ・ビルドの履歴表示およびビルドの再投入

パッケージをビルド・プロセスに投入した後、〈Package Build History (パッケージ・ビルドの履歴)〉アプリケーション(P9622)を使用すると、ビルド状況をトラッキングできます。このアプリケーションでは、ビルド・プロセスに関連するログを表示し、ビルド・プロセスでエラーが発生したかどうかを確定することもできます。

ビルドが正常に終了しなかった場合、パッケージを再投入してビルドが停止した時点からのビルドを再開することができます。また、スペックとオブジェクトの状況をリセットして、パッケージのビルドをやり直すこともできます。

参照

- パッケージ・ビルドの履歴の検討、ログの表示、ビルドの再投入、およびパッケージ・ビルド状況の変更については、「パッケージ・ビルドの履歴およびログの表示」

パッケージへの機能の組み込み

機能とは、アプリケーションやその他の機能をサポートするために、ワークステーションやサーバーにコピーされる、レジストリ設定などの一連のファイルや構成オプションを意味します。B9 オブジェクトと同様に、機能はパッケージに組み込まれ、機能コンポーネントを必要とするワークステーションとサーバーに配布されます。

以下に、パッケージをビルドするときに組み込む機能の例をいくつか示します。

- ActiveX コントロール。B9 の〈Application Design Aid (アプリケーション設計ツール)〉を使用すると、ActiveX コントロールを B9 アプリケーションに組み込むことができます。ActiveX コントロールを B9 と共に配布する場合は、それらのコントロールをワークステーションにコピーする手段が必要です。
- ODA (オープン・データ・アクセス) データ・ソース。ODA では、ODA を使用するワークステーションやサーバー上で、追加の ODBC データ・ソースを作成する必要があります。
- セールス・フォース・オートメーション・データベース。セールス・フォース・オートメーション機能では、別個の J.D. Edwards サポート対象ローカル・データベースをワークステーションにインストールして、オフライン操作時にワークステーションをネットワークから切断できるようにする必要があります。また、マシンがオフラインで使用されることを示すレジストリ設定を記述する必要があります。
- BMC Patrol、GenCorba、GenCom などのサードパーティ・インターフェイスまたは製品。これらの製品とインターフェイスを使用するには、それぞれに対応する追加のコンポーネントがワークステーションとサーバーで必要です。追加のサードパーティ製品やインターフェイスをサポートするためにソフトウェアの機能が拡張するにつれて、これらの製品にはそれぞれ独自のサポート・ファイルが設定されます。

過去のリリースでは、ワークステーションやサーバーに機能コンポーネントを追加するには、カスタム・プログラミングが必要でした。このリリースでは、〈Package Assembly Director〉や〈Package Build Definition Director〉などの標準のツールを使用して、機能を含むパッケージを作成し、〈Package Deployment Director (パッケージ・デプロイメント・ディレクタ)〉や複数階層デプロイメントを使用して配布できるようになりました。

機能コンポーネントはオブジェクトではないため、機能コンポーネントをパッケージに組み込むプロセスは、通常のパッケージ・ビルド・プロセスとわずかに異なります。特に、最初に機能を定義しておかなければパッケージに追加することができません。

機能のビルドとデプロイメント・プロセスの概要

次に、機能を定義してパッケージに追加するプロセスの概要を示します。

機能の定義

機能をパッケージに追加する前に、まず〈Feature Based Deployment Director〉を使用して機能を定義する必要があります。機能を定義する際には、機能の名前とタイプを指定し、概要を入力し、インストール・パラメータを指定します。

〈Feature Based Deployment Director〉のフォームでは、次の操作を実行できます。

- ファイルセットの作成
- レジストリ設定の定義
- Windows ショートカットの定義
- 初期化ファイル情報の入力
- ODBC データ・ソースの追加
- 機能ビルド順序の指定
- サードパーティ製品に関する情報の入力

機能の定義の詳細については、「機能の理解」を参照してください。

パッケージ組立て時の機能の選択

機能の定義が済んだら、いつでもパッケージに含めることができます。〈Package Assembly Director〉を使用して、一般のパッケージと同様にパッケージを組み立てます。パッケージを組み立てるときは、機能固有のフォームを使用して、組み入れる機能を指定できます。

機能を含んでいるパッケージの組立てについては、「パッケージ組立て時の機能の選択」を参照してください。

パッケージ・ビルド定義時の機能の構成

機能を含んだパッケージを組み立てたら、〈Package Build Definition Director〉を使用してパッケージのビルドを定義できます。このディレクタのフォームでは、パッケージの中で圧縮するファイルセットを選ぶことができ、また機能をビルドする前後に実行するプロセスを指定できます。

機能を含んだパッケージのビルドについては、「パッケージ・ビルド定義時の機能の構成」を参照してください。

パッケージの配布

パッケージのビルドが済んだら、〈Package Deployment Director〉を使用して、デプロイメントのスケジュールを設定できます。この手順は、機能を含まないパッケージをスケジュールするときの手順と同じです。

パッケージ・デプロイメントのスケジュール設定については、「配布」を参照してください。

〈Workstation Installation〉と〈Deployment Server Installation〉の実行

ワークステーションとデプロイメント・サーバー向けのパッケージを配布し終わったら、〈Workstation Installation〉アプリケーションと〈Deployment Server Installation〉(デプロイメント・サーバー・インストール)アプリケーションを使用してパッケージをインストールします。

機能を含んだパッケージのインストールについては、「機能を含んだパッケージのインストール」を参照してください。

Feature Based Deployment Director の理解

〈Feature Based Deployment Director〉では、パッケージに組み込んでワークステーションやサーバーに配布できるように機能を定義することができます。このディレクトリのフォームでは、機能の名前とタイプおよび各種機能コンポーネントを指定できます。

このリリースでは、CLIENT の Platform 値は常に 80 でなければなりません。将来のリリースでは別のプラットフォームを選択できるようにする予定です。

機能定義プロセスの全体を通じて、[Next]や[Previous]をクリックすると、いつでも前後のフォームに移動できます。また、プロセスのどの段階でも、[Cancel]をクリックすることにより、いつでも機能定義を取り消せます。

次の表に、〈Feature Based Deployment Director〉の各フォームと各フォームの機能の概要を示します。

〈Welcome(はじめに)〉 フォーム	〈Feature Base Deployment Director〉の導入説明が表示されます。
〈Feature Information(機能情報)〉フォーム	<p>機能の名前、タイプ、概要を入力します。また、インストール時にその機能が必須かどうかを示します。その場合は、インストール・プログラムの実行時に自動的に機能がインストールされます。</p> <p>任意機能の場合は、インストール・プログラムの実行時にその機能が初期選択されるかどうか(つまり、デフォルトでオンになるかどうか)を指定できます。任意機能が初期選択されない場合(つまり、デフォルトでオフになる場合)は、手動で選択しないとインストールできません。</p> <p>また、次の項目を適用する機能コンポーネントも選択する必要があります。</p> <ul style="list-style-type: none">• ファイルセット• レジストリ• ショートカット• ODBC データ・ソース• その他のパッケージ・ビルド・プロセス• その他のインストール・プロセス• 初期化ファイル(INI) <p>このフォームでの選択内容によって、次に表示されるフォームが確定します。機能の定義を続けるには、少なくとも 1 個の機能コンポーネントを選択する必要があります。</p>
〈File Set Definition(ファイルセット定義)〉フォーム	パッケージに含めるファイルセットを選択できます。ファイルセットとは、機能が正しく動作するためにワークステーションまたはデプロイメント・サーバーにインストールされる必要のあるファイルの集合のことです。
〈Registry Definition(レジストリの定義)〉フォーム	定義している機能で、Windows レジストリ内の項目の追加、修正、または削除が必要な場合に使用します。このフォームで入力したレジストリ情報は、機能が含まれているパッケージに提供されます。

〈Shortcut Definition (ショートカットの定義)〉フォーム	インストール・プロセスの一環として、Windows デスクトップ上にショートカットを作成する場合に使用します。ショートカット情報を入力すると、機能のインストール後、ショートカットがデスクトップ上に自動的に作成されます。
〈Additional Package Build Processes (その他のパッケージ・ビルド・プロセス)〉フォーム	機能を含んだパッケージをインストールする前後にバッチ・アプリケーションや実行可能プログラムを実行するかどうかを指定します。
〈Additional Install Processes (その他のインストール・プロセス)〉フォーム	機能インストールの一環としてサードパーティ製品をインストールする必要がある場合に使用します。
〈Initialization File (INI) Definition (初期化ファイル (INI)の定義)〉フォーム	インストールの一環として、JDE.ini ファイルなどの初期化ファイルに変更を加える必要がある場合に使用します。たとえば、ERP 9.0 の開発者は、多くの場合、デバッグ・ログを手動でオフにします。そのため、パッケージを開発者に配布する場合は、デバッグ・ログを自動的にオフにできます。
〈ODBC Data Source Definition (ODBC データ・ソースの定義)〉フォーム	機能を含んだパッケージのインストール時に、補足的な ODBC データ・ソースを追加する必要がある場合に使用します。
〈Features Summary (機能の要約)〉フォーム	前のフォームで機能に関して入力した情報を検討する場合に使用します。フォームの左側に、機能に関する主要情報がツリー構造で表示されます。フォーム右側のオプションにより、前に表示した任意のフォームにアクセスでき、必要に応じて変更を加えることができます。

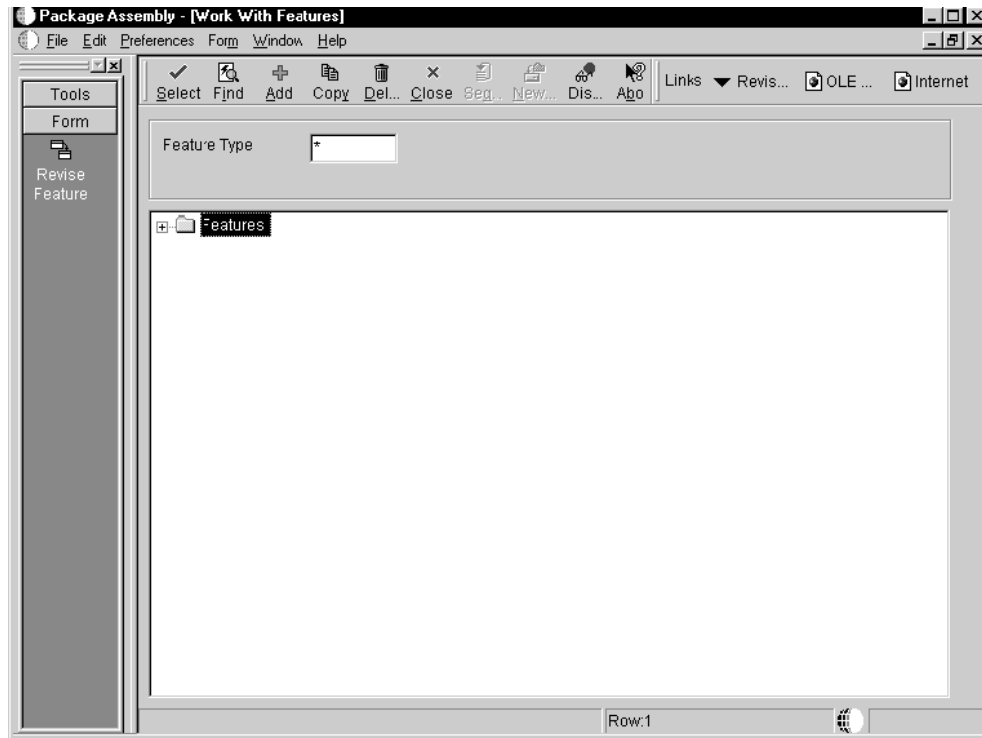
機能の作成

機能を定義してコンポーネントを追加するには、以下のタスクを実行します。

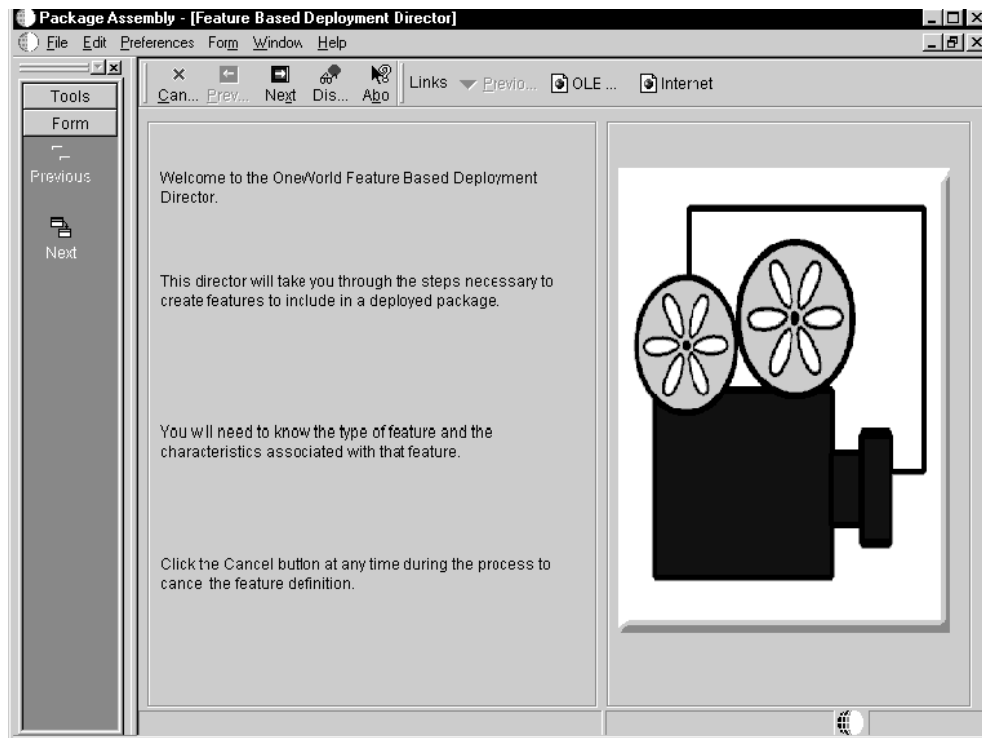
▶ 機能を作成するには

〈Package and Deployment Tools〉メニュー(GH9083)から〈Package Assembly〉を選択します。

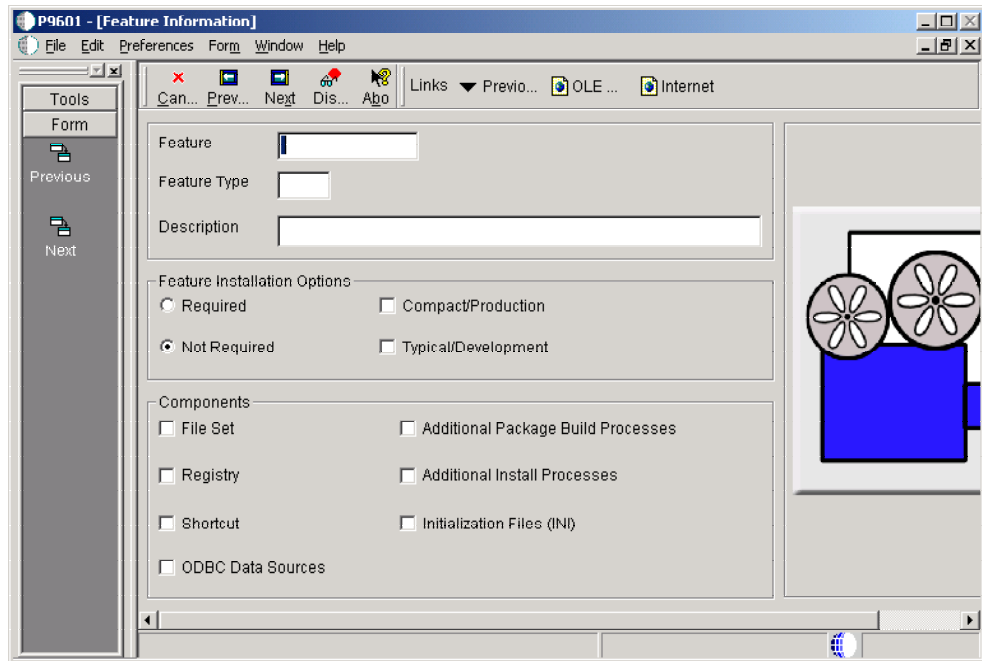
1. 〈Work with Packages〉フォームで、[Form]メニューから[Features]を選択します。



2. 〈Work With Features(機能の処理)〉フォームで、[Add]をクリックします。



3. 〈Feature Based Deployment Director〉フォームで、[Next]をクリックします。



4. 〈Feature Information〉フォームが表示されます。次のフィールドに値を入力します。

- Feature (機能)
- Feature Type (機能タイプ)
- Description (記述)

次のオプションのうち 1 つをオンにします。

- Required (必須)

圧縮/本稼動および通常/開発の両方のインストールでこの機能のインストールを必須にします。パッケージのインストール時には、この機能の組み込みを一時変更できません。

- Not Required (任意)

この機能のインストールを必須にしません。この機能がインストールされるかどうかは、次に選択するオプション ([Compact/Production] と [Typical/Development]) によって決まります。パッケージのインストール時には、機能の組み込みを一時変更できません。

次の一方または両方のオプションをオンにします。[Required] を選んだ場合は、次のオプションが両方とも自動的に選択されます。

- Compact/Production

このオプションをオンにすると、この機能はデフォルトで圧縮/本稼動インストールに組み込まれます。[Not Required] も選択している場合は、このオプションをパッケージのインストール時に一時変更できます。

- Typical/Development

このオプションをオンにすると、この機能はデフォルトで通常/開発インストールに組み込まれます。[Not Required]も選択している場合は、このオプションをパッケージのインストール時に一時変更できます。

パッケージに適用する次のオプションと機能コンポーネントのいずれかをオンにします。この後に表示されるフォームは、ここで選択した機能コンポーネントによって異なります。

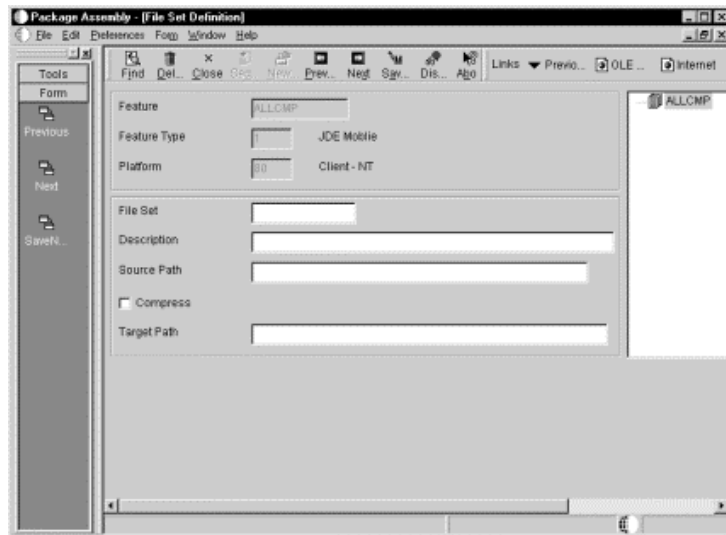
- ファイルセット
- レジストリ
- ショートカット
- ODBC データ・ソース
- その他のパッケージ・ビルド・プロセス
- その他のインストール・プロセス
- 初期化ファイル(INI)

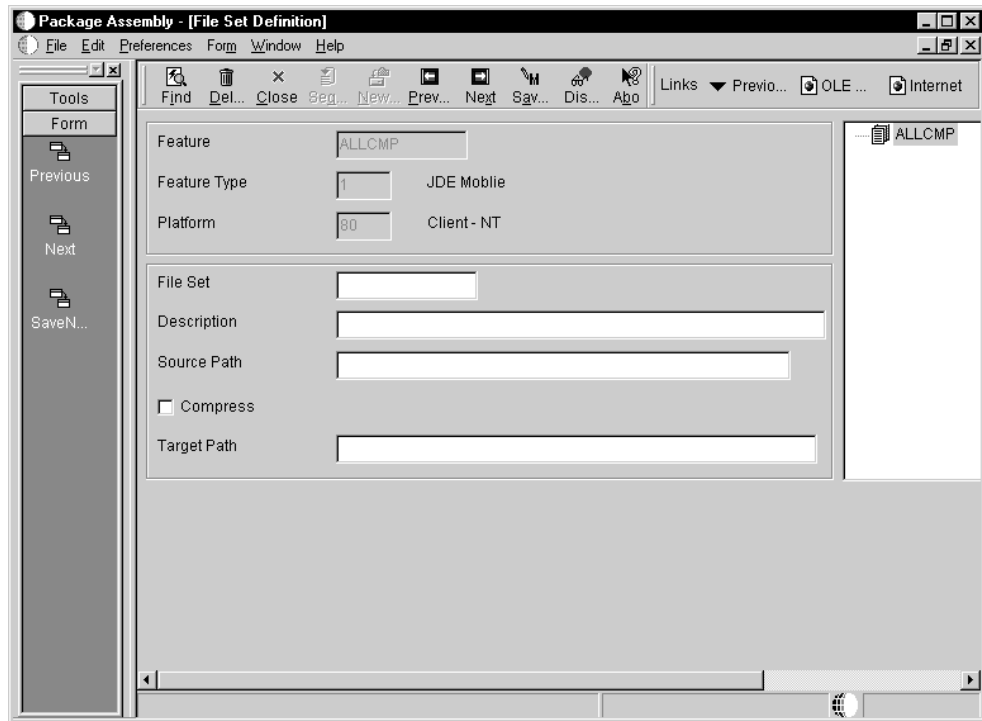
5. [Next]をクリックします。

ファイル・セット・コンポーネントの定義

ファイルセット・コンポーネントを選択した場合は、〈File Set Definition〉フォームが表示されます。このフォームでは、機能が正しく動作するようにワークステーションまたはサーバーにインストールする必要があるファイルセットに関する情報を入力します。

► ファイルセット・コンポーネントを定義するには





1. 〈File Set Definition〉上で、次のフィールドに入力します。

- File Set(ファイルセット)
コメントまたは覚え書き用の自由形式テキスト・フィールド
- File Set Description(ファイルセットの記述)
ファイル・グループの説明
- Source Path(ソース・パス)
ファイルセットのソース・ロケーションを特定するパス
- Compress
ファイルを圧縮するオプション
- Target Path(ターゲット・パス)
ファイルセットのターゲット・ロケーションを特定するパス

ソース・パスは、パッケージにコピーするファイルセットが置かれている場所を ERP 9.0 に示し、ターゲット・パスはパッケージのインストール時にファイルセットをコピーする場所を示します。機能はファイルセットをいくつでも持つことができますが、各ファイルセットは 1 つしかターゲット・パスを持つことができません。

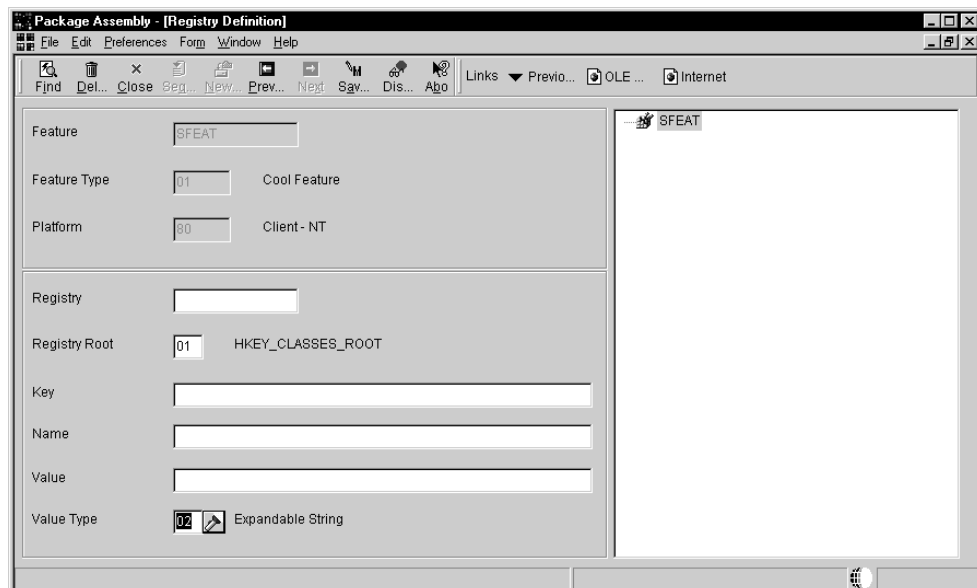
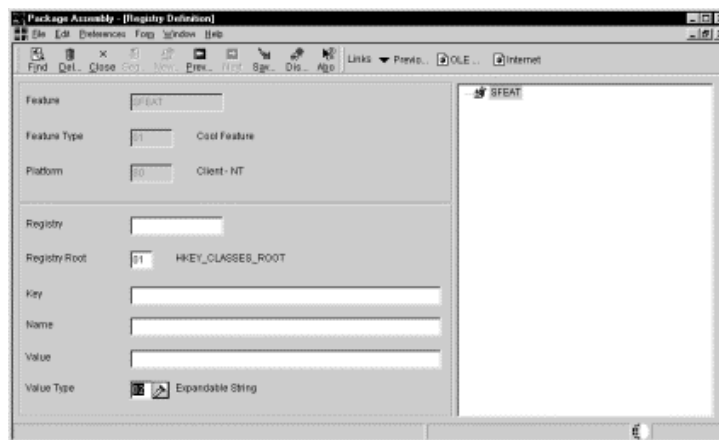
このフォームでは、以前に定義したファイル・セットの修正や削除も行えます。既存のファイルセットは、フォームの右側にツリー構造で表示されます。ファイルセットを修正する場合は、ツリー構造上でファイルセットを選択し、そのファイルセットに関するフィールドを修正します。ファイルセットを削除するには、ファイルセットを選択し、[Delete]をクリックします。

2. ファイルセット情報の追加を完了したら、[Form]メニューから[Save Node(ノードの保存)]を選択します。
3. [Next]をクリックします。

レジストリ・コンポーネントの定義

レジストリ・コンポーネントを選択した場合は、〈Registry Definition〉フォームが表示されます。機能インストールの一環として Windows レジストリに追加する情報を入力するには、このフォームを使用します。

▶ レジストリ・コンポーネントを定義するには



1. 〈Registry Definition〉上で、次のフィールドに入力します。

- Registry(レジストリ)
レジストリ修正の識別子

- Registry Root(レジストリ・ルート)
レジストリ内のルート・キー
- Key(キー)
レジストリ値のキー
- Name(名前)
レジストリ値の名前。
- Value(値)
レジストリ値
- Value Type
レジストリで、値が保管されるデータ・タイプ。

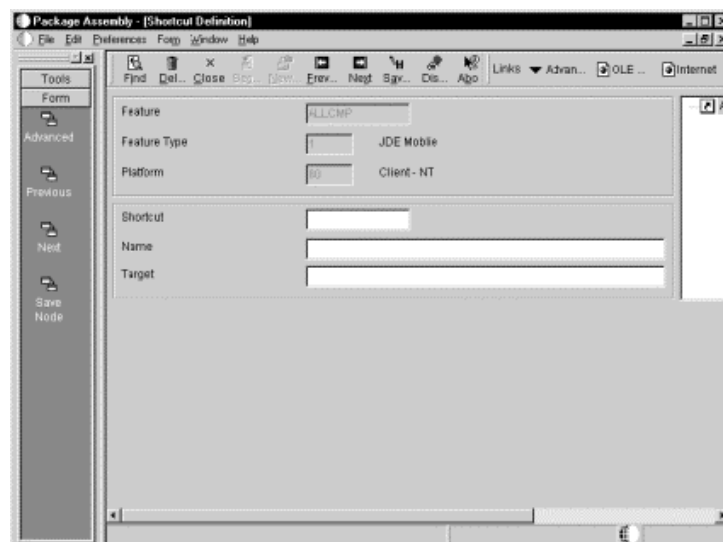
このフォームでは、既存のレジストリ定義の修正や削除も行うことができます。既存のレジストリ定義は、フォームの右側にツリー構造で表示されます。レジストリ定義を修正するには、ツリー構造で項目を選択し、レジストリ定義用のフィールドを修正します。レジストリ定義を削除するには、項目を選択し、[Delete]をクリックします。

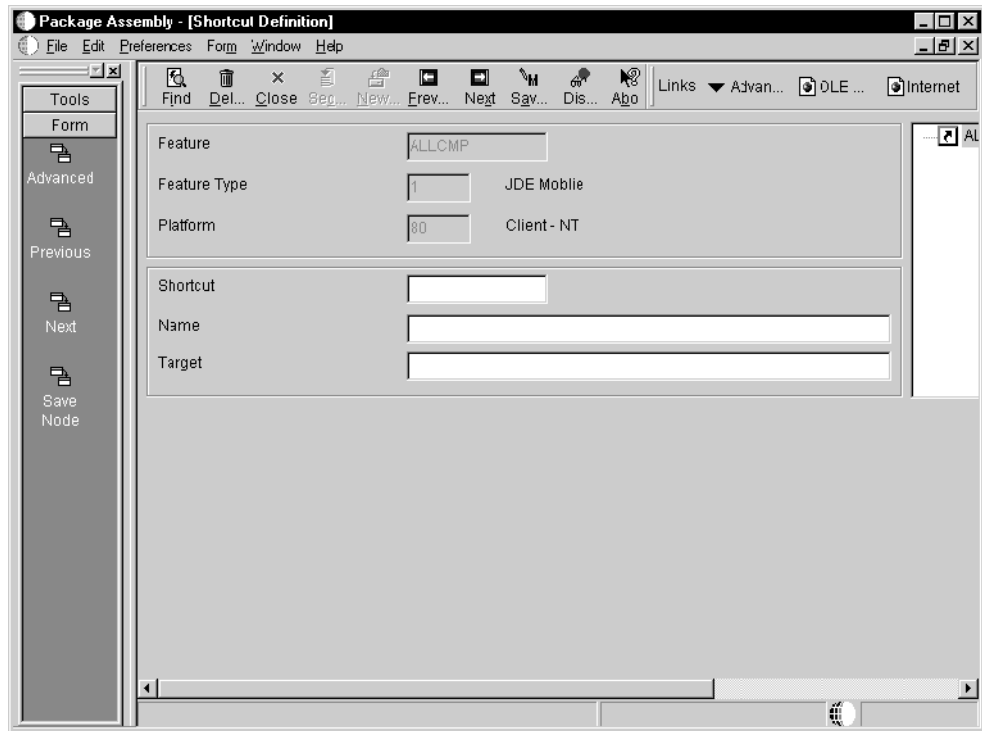
2. レジストリ情報の追加を完了したら、[Form]メニューから[Save Node]を選択します。
3. [Next]をクリックします。

ショートカット・コンポーネントの定義

ショートカット・コンポーネントを選択した場合は、〈Shortcut Definition〉フォームが表示されます。このフォームは、機能のショートカットを Windows デスクトップに追加したい場合に使用します。

▶ ショートカット・コンポーネントを定義するには





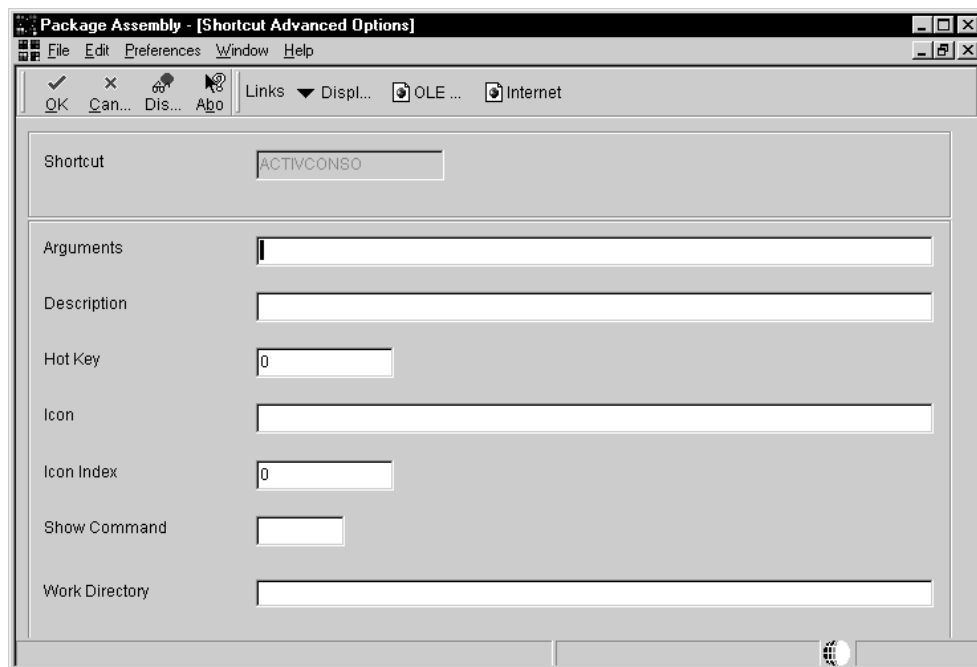
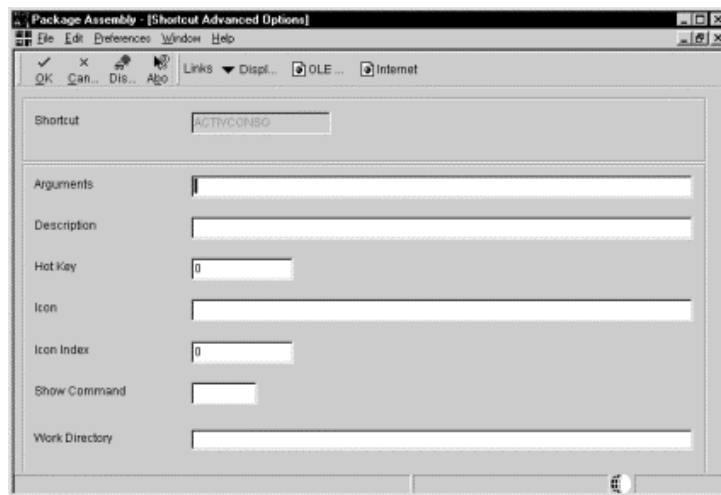
1. 〈Shortcut Definition〉上で、次のフィールドに入力します。

- Shortcut
ユーザーのコンピュータに対して固有のショートカットを識別する名前
- Name
ショートカットの名前
- Target (ターゲット)
ターゲット・ファイルのパスとファイル名

標準インターネット・プロトコル (http://、FTP://、FILE// など) を使用してショートカットを入力すると、インターネット・ショートカットが自動的に作成されます。

このフォームでは、既存のショートカット定義の修正や削除も行うことができます。既存の定義は、フォームの右側にツリー構造で表示されます。ショートカット定義を修正するには、ツリー構造で項目を選択し、ショートカット定義用のフィールドを修正します。ショートカット定義を削除するには、項目を選択し、[Delete] をクリックします。

2. 上級ショートカット・オプションを入力するには、[Form] メニューから [Advanced (上級)] を選択します。



3. 〈Shortcut Advanced Options(ショートカット上級オプション)〉フォームが表示されます。次のフィールドのいずれかに値を入力します。

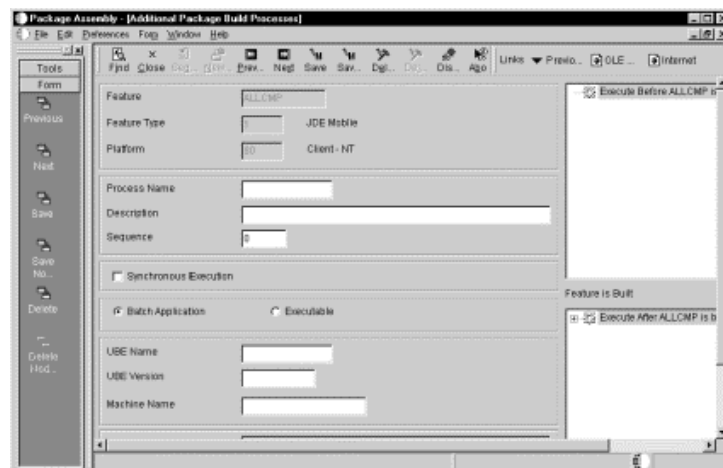
- Arguments(引数)
ショートカットに関してコマンド行に入力するパラメータ
- Description
ショートカットの説明
- Hot Key(ホット・キー)
押すとショートカットを自動的に起動するキー・シーケンス

- Icon (アイコン)
相対ターゲット・パスに基づくアイコン・ファイルのパスと名前
 - Icon Index (アイコン・インデックス)
ショートカットのアイコン・インデックス
 - Show Command
ショートカット起動後のウィンドウのサイズ。たとえば、ウィンドウを最小化/最大化できます。
 - Work Directory (作業ディレクトリ)
ショートカットのディレクトリ・パスまたは作業ディレクトリの識別子
4. [OK]をクリックして入力内容を保存し、〈Shortcut Definition〉フォームに戻ります。
 5. ショートカット情報を入力し、[Form]メニューから[Save Node]を選択します。
 6. すべてのショートカットを追加します。
 7. [Next]をクリックします。

その他のパッケージ・ビルド・プロセスの定義

その他のパッケージ・ビルド・プロセスを選択した場合は、〈Additional Package Build Processes〉フォームが表示されます。このフォームでは、パッケージのビルド時に実行する UBE や実行可能プログラムを指定します。

▶ その他のパッケージ・ビルド・プロセスを定義するには



1. 〈Additional Package Build Processes〉上で、次のフィールドに入力します。
 - Process Name (プロセス名)
ビルド・プロセスの名前
 - Description
ビルド・プロセスの説明
 - Sequence
パッケージ・ビルド時に実行する他のプロセスとの関係で、このプロセスを実行する順序を特定する番号
 - Synchronous Execution (同期実行)
このプロセスが完了するまで待ってからパッケージ・ビルド・ジョブを続行するかどうかを指示するオプション
 - Batch Application または Executable
プロセスがアプリケーションか、実行可能プログラムかを指定します。
2. [Batch Application (バッチ・アプリケーション)] を選択した場合は、次のフィールドに入力します。
 - UBE Name (UBE 名)
UBE の名前
 - UBE Version (UBE バージョン)
UBE のバージョン番号
 - Machine Name (マシン名)

UBE を実行するサーバーまたはワークステーションの名前

3. 実行可能プログラムを選択した場合は、次のフィールドに入力します。

- Executable Name (実行可能プログラム名)
サードパーティ・ソフトウェアをインストールするために起動する実行可能プログラムの名前
- Target Path (ターゲット・パス)
ターゲット・ファイルのパスとファイル名
- Parameters (パラメータ)
サードパーティ・ソフトウェアのインストールで、セットアップ・プログラムが使用する実行可能パラメータ

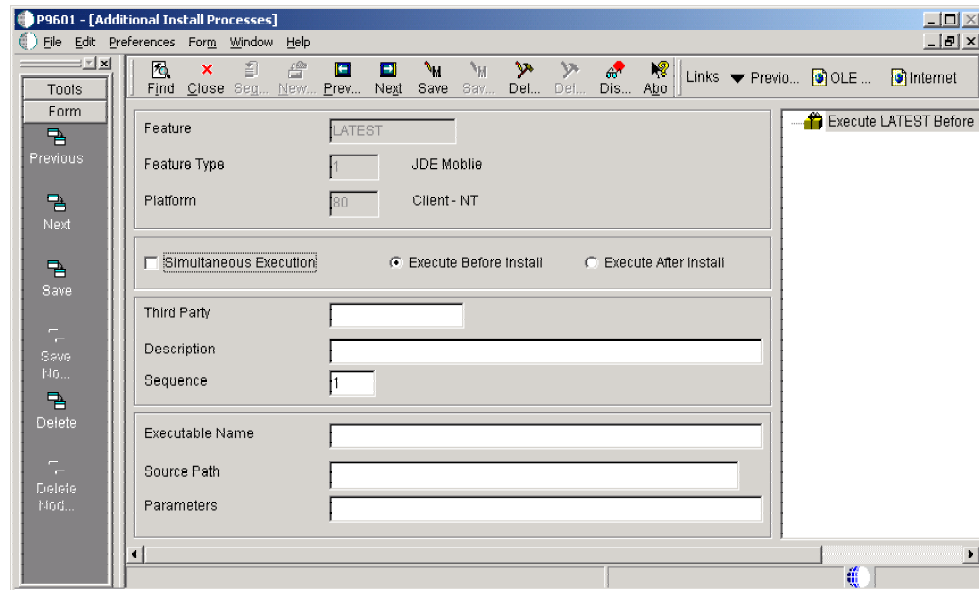
このフォームでは、以前に定義したプロセスを修正したり削除することもできます。既存のプロセスは、フォームの右側にツリー構造で表示されます。プロセス定義を修正するには、ツリー構造で項目を選択し、定義用フィールドを修正します。プロセス定義を削除するには、項目を選択した後、削除するプロセスの実行が機能インストールの前か後かに応じて、[Form]メニューから[Delete]または[Delete Node After]を選択します。プロセスは、機能をビルドする前後に実行できます。

4. プロセス情報の追加を完了したら、プロセスをいつ実行するかに応じて、[Form]メニューから[Save (保存)]または[Save Node After (後でノードを保存)]を選びます。
5. [Next]をクリックします。

その他のインストール・プロセスの定義

その他のインストール・プロセスを追加するオプションを選択した場合は、〈Additional Install Processes〉フォームが表示されます。このフォームでは、パッケージのインストール時に実行するサードパーティ・アプリケーションに関する情報を入力します。

▶ その他のインストール・プロセスを定義するには



1. 〈Additional Install Processes〉上で、次のフィールドに入力します。
 - Third Party (サードパーティ)
サードパーティ・コンポーネントの名前
 - Description
サードパーティ・ソフトウェアの説明
 - Sequence
他のインストール・プロセスとの関係で、このプロセスを実行する順序を特定する番号
2. 次の各オプションでは、相互に連動する2つのフィールドを選ぶ必要があります。必要な結果が得られる組合せを選択してください。
 - [Synchronous (同時実行)] および [Execute After Install (インストール後実行)]
[Simultaneous Execution] をオフ、[Execute After Install] をオンにします。J.D. Edwards クライアント・インストールをインストールしてから、サードパーティ・プロセスが実行されます。
 - [Synchronous] および [Execute Before Install (インストール前実行)]
[Simultaneous Execution] をオフ、[Execute Before Install] をオンにします。J.D. Edwards クライアント・インストールではサードパーティ・プロセスが実行され、クライアントのインストール後に完了します。
 - [Asynchronous (同時実行)] および [Execute After Install]
[Simultaneous Execution] をオン、[Execute After Install] をオンにします。J.D. Edwards クライアント・インストールが完了してから、サードパーティ・プロセスが開始されます。どちらのプロセスも、他方のプロセスが完了してから実行されます。

- [Asynchronous]および[Execute Before Install]

[Simultaneous Execution]をオン、[Execute Before Install]をオンにします。J.D. Edwards クライアント・インストールが始まるとすぐにサードパーティ・プロセスが開始されます。サードパーティ・プロセスの完了を待たずにクライアント・インストールが再開されます。

3. 次のフィールドに値を入力します。

- Executable Name (実行可能プログラム名)
サードパーティ・ソフトウェアを起動するプログラムの名前
- Target Path (ターゲット・パス)
実行可能ファイルへのパス。ファイル名は含めないでください。
- Parameters
サードパーティ・プログラムに渡す実行可能パラメータ

4. サードパーティ製品情報の追加を完了したら、[Form]メニューから[Save]を選択します。

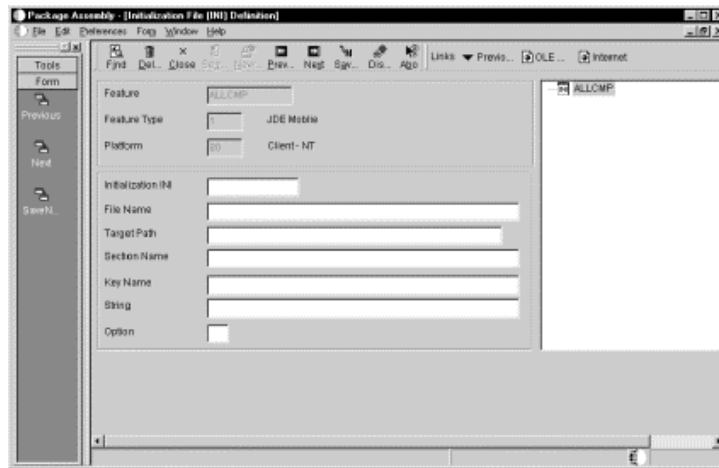
5. サードパーティ製品の定義を削除するには、項目をツリー構造で選択し、[Form]メニューから[Delete]を選びます。

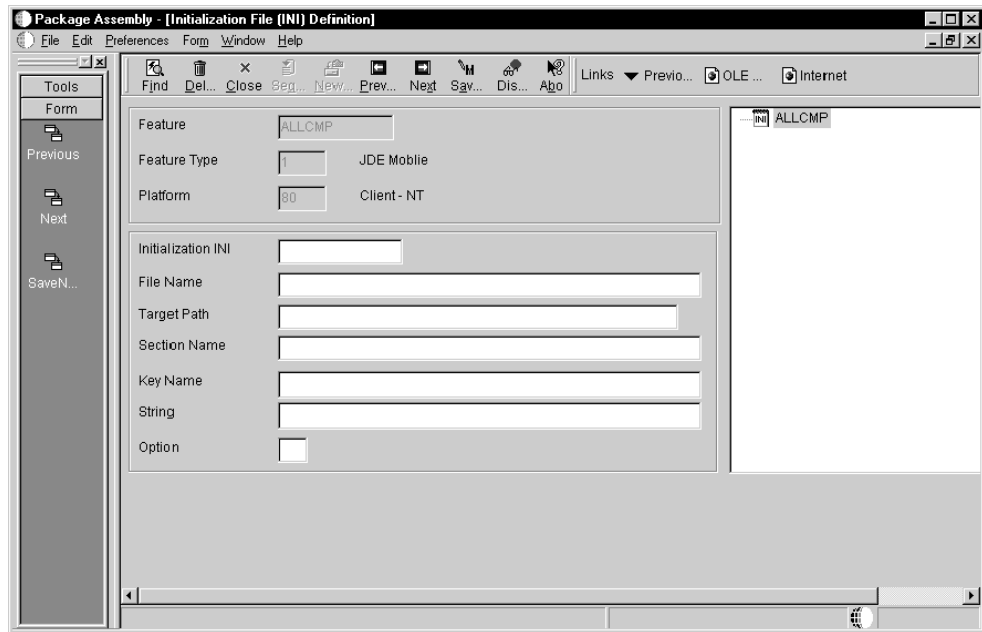
6. [Next]をクリックします。

初期化ファイル・コンポーネントの定義

初期化ファイル・コンポーネントを選択した場合は、〈Initialization File (INI) Definition〉フォームが表示されます。このフォームでは、機能インストールの一環として(JDE.ini など)の初期化ファイルに書き込む必要がある情報を入力します。

▶ 初期化ファイル・コンポーネントを定義するには





1. 〈Initialization Files (INI) (初期化ファイル(INI))〉上で、次のフィールドに入力します。

- Initialization INI (初期化 INI)
初期化ファイル・コンポーネントの識別子
- File Name (ファイル名)
初期化ファイルの名前
- Target Path (ターゲット・パス)
INI ファイルのパス
- Section Name (セクション名)
初期化ファイル内のアプリケーション・セクションの名前
- Key Name (キー名)
追加、修正、または除去する初期化ファイル内のキー
- String (ストリング)
初期化ファイル内のキーの値
- Option (オプション)
初期化ファイル内のキーに関連付けられているアクションを特定するオプション

このフォームでは、既存の初期化ファイル定義を修正したり削除することができます。既存の定義は、フォームの右側にツリー構造で表示されます。初期化ファイル定義を修正するには、ツリー構造で項目を選択し、定義用のフィールドを修正します。初期化ファイル定義を削除するには、項目を選択し、[Delete]をクリックします。

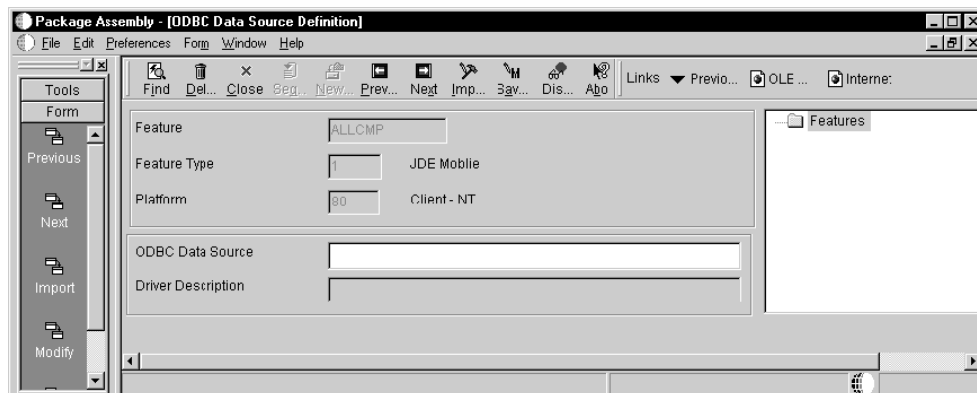
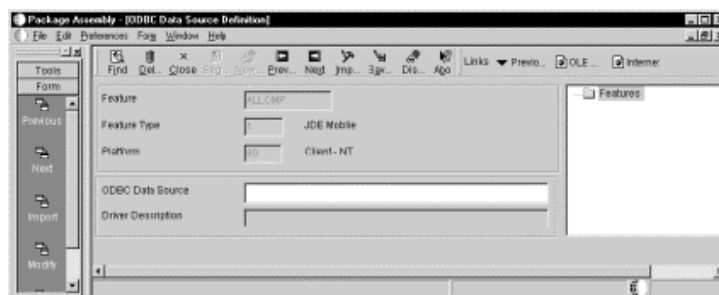
2. 初期化情報の追加を完了したら、[Form]メニューから[Save Node]を選択します。
3. [Next]をクリックします。

ODBC データ・ソース・コンポーネントの定義

ODBC データ・ソース・コンポーネントを選択した場合は、〈ODBC Data Sources Definition (ODBC データ・ソースの定義)〉フォームが表示されます。このフォームでは、機能をサポートするために追加する必要がある ODBC データ・ソースに関する情報を入力します。

ODBC データ・ソースは、新しく作成することも既存のデータ・ソースをインポートすることもできます。

▶ 新規 ODBC データ・ソースを作成するには

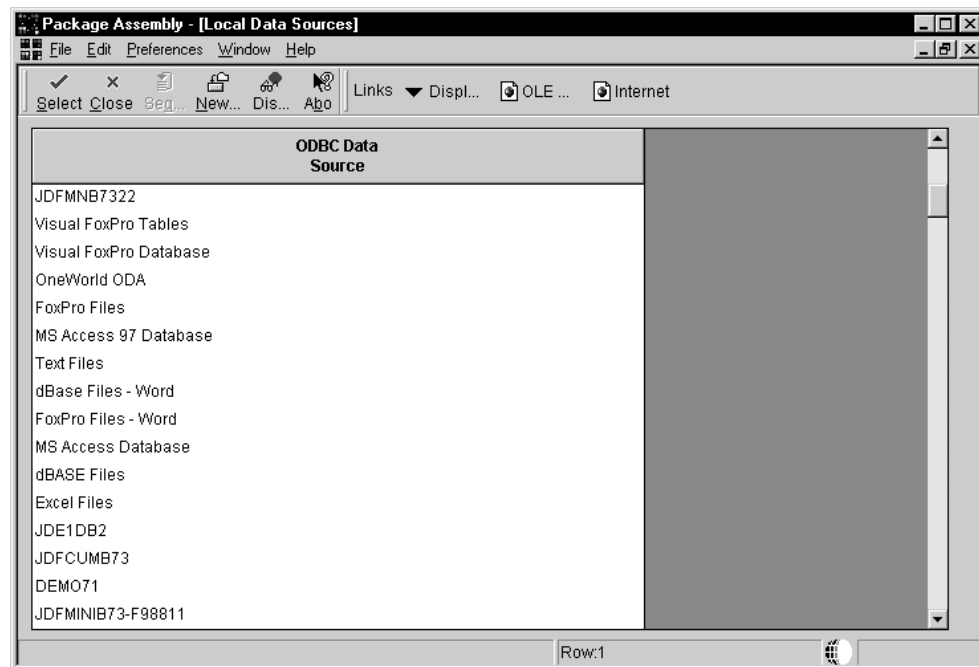
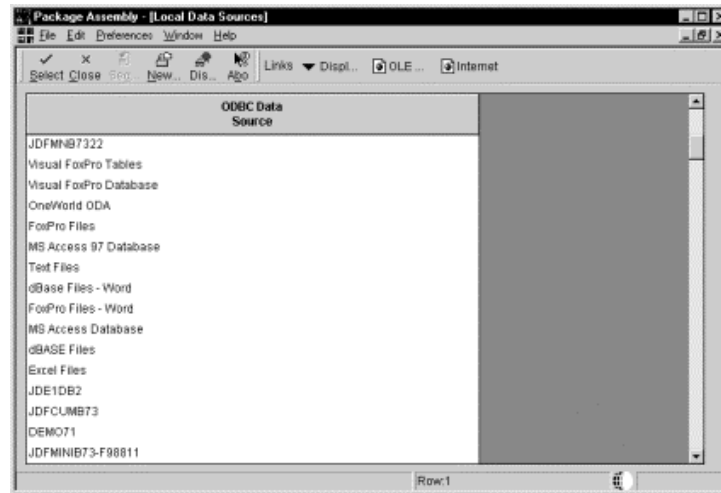


1. 〈ODBC Data Sources Definition〉上で、次のフィールドに入力します。
 - ODBC Data Source (ODBC データ・ソース)
ODBC データ・ソースの名前
2. [Form]メニューから[Save Node]を選択します。

Windows コントロール パネルのアプレットが起動され、ODBC データ・ソースのフォームが表示されます。
3. Windows ベースのフォームで、データ・ソース情報を入力します。
4. さらに別のデータ・ソースを作成するには、この手順を繰り返します。

▶ 既存の ODBC データ・ソースをインポートするには

1. 〈ODBC Data Sources Definition〉フォームで、[Form]メニューから[Import(インポート)]を選択し、〈Local Data Sources(ローカル・データ・ソース)〉フォームを表示します。このフォームには、マシンにローカルで存在する作成済みのデータ・ソースがすべて一覧表示されます。



2. 〈Local Data Sources〉フォームで、Ctrl キーまたは Shift キーを使用して、1 つまたは複数のデータ・ソースを選び、[Select]をクリックして、データ・ソースを機能に追加します。

〈ODBC Data Source Definition〉フォームが再表示されます。

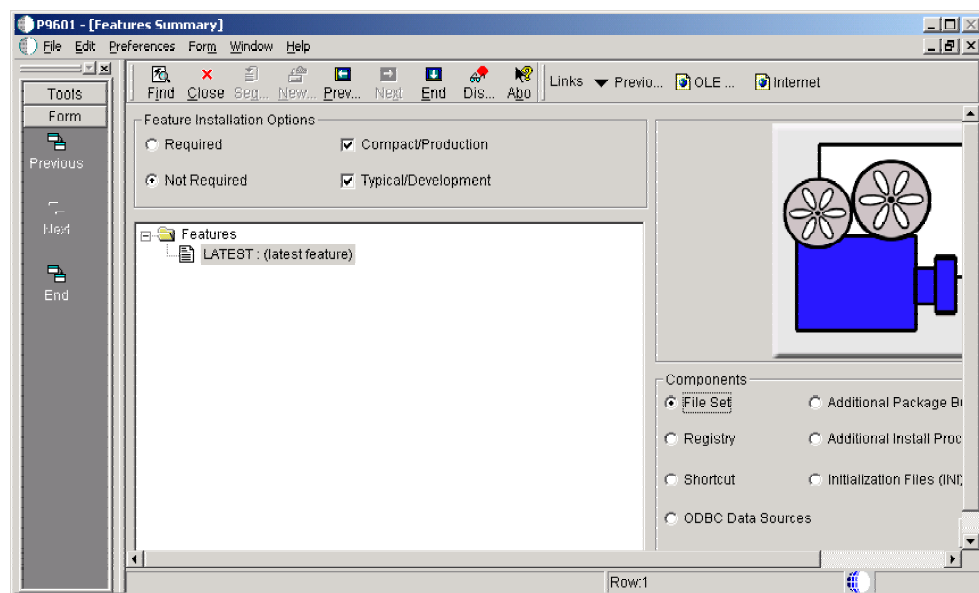
3. データ・ソース情報の追加を完了したら、[Form]メニューから[Save Node]を選択します。
4. [Next]をクリックします。

5. 既存のデータ・ソースを修正するには、データ・ソース名を入力し、[Form]メニューから[Modify (修正)]を選択します。〈ODBC Data Source Revisions (ODBC データ・ソースの改訂)〉フォームが表示されます。このフォームでデータ・ソースに変更を加えます。
6. 終了したら、[OK]をクリックして〈ODBC Data Source Definition〉フォームに戻ります。

機能コンポーネントの検討

選択したコンポーネントをすべて定義し終えたら、〈Feature Based Deployments〉フォームに入力した情報の検討や修正を行うことができます。機能コンポーネントの改訂の詳細については、「既存パッケージの改訂」を参照してください。

▶ 機能コンポーネントを検討するには



1. 〈Features Summary〉フォームの右ペインでコンポーネントを選択し、[Revise (改訂)]ボタンをクリックして、そのコンポーネントに関する情報を表示します。
2. 必要に応じ、選択したコンポーネントのフィールド値を変更し、[Save]ボタンをクリックします。
3. 別のコンポーネントを修正するには、前のステップを繰り返します。
4. 機能を定義し終えたら、〈Feature Summary〉フォームで[End]をクリックします。

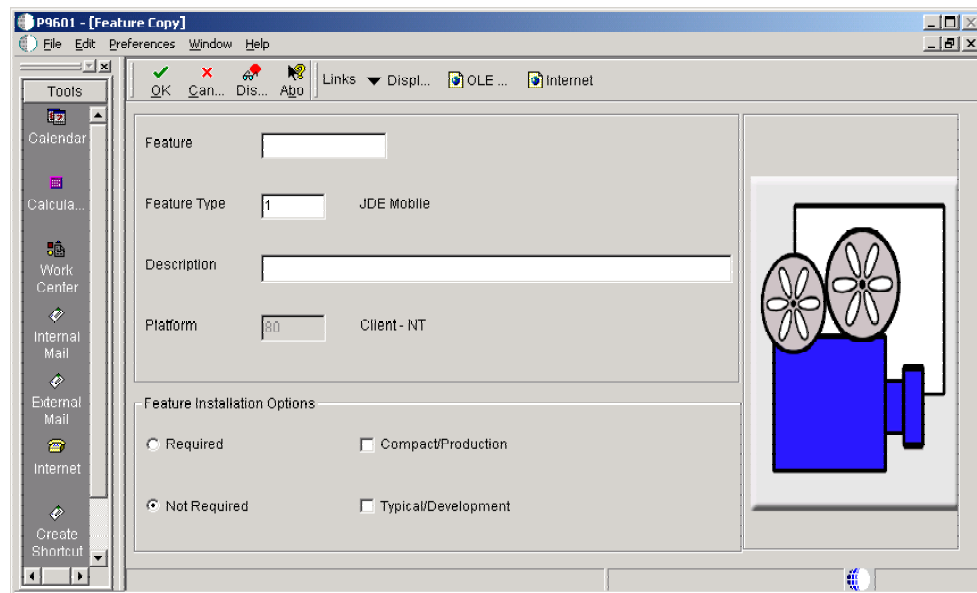
機能のコピー

〈Feature Based Deployment Director〉は、既存の機能をコピーして名前を変更し、新規機能として作成できるコピー機能を備えています。この機能は、既存の機能定義とよく似た機能定義を作成したい場合に特に有効です。

▶ 機能をコピーするには

〈Package and Deployment Tools〉メニュー(GH9083)から〈Package Assembly〉プログラム(P9601)を選択します。

1. 〈Work with Packages〉フォームで、[Form]メニューから[Features]を選択します。
2. 〈Work With Features〉フォームで、定義をコピーする機能を選択し、[Copy]をクリックします。



3. 〈Feature Copy(コピー機能)〉上で、次のフィールドに入力します。
 - Feature(機能)
 - Feature Type(機能タイプ)
 - Description
4. 次のオプションのうち 1 つをオンにします。
 - Required
圧縮/本稼動および通常/開発の両方のインストールでこの機能のインストールを必須にします。パッケージのインストール時には、この機能の組込みを一時変更できません。
 - Not Required
この機能のインストールを必須にしません。この機能がインストールされるかどうかは、次に選択するオプション([Compact/Production]と[Typical/Development])によって決まります。パッケージのインストール時には、機能の組込みを一時変更できません。
5. 次の一方または両方のオプションをオンにします。[Required]を選んだ場合は、次のオプションが両方とも自動的にオンになります。
 - Compact/Production
このオプションをオンにすると、この機能はデフォルトで圧縮/本稼動インストールに組み込まれます。[Not Required]もオンにしている場合は、このオプションをパッケージのインストール時に一時変更できます。

- Typical/Development

このオプションをオンにすると、この機能はデフォルトで通常/開発インストールに組み込まれます。[Not Required]もオンにしている場合は、このオプションをパッケージのインストール時に一時変更できます。

6. [OK]をクリックして〈Work With Features〉フォームに戻ります。
7. 新規機能定義を改訂するには、機能をハイライトして[Form]メニューから[Revise Feature (機能の改訂)]を選択します。

パッケージ組立て時の機能の追加

〈Package Assembly Director〉には、〈Feature Component〉というフォームが含まれており、このフォームを使用して、定義済みの機能をパッケージに追加できます。機能は新規パッケージにも、改訂用に開いている既存パッケージにも追加できます。

次のタスクでは、機能を新規パッケージまたは既存パッケージに追加する手順のみを説明します。パッケージの組立て、または組立て済みパッケージの修正に関する詳細は、「パッケージの組立て」を参照してください。

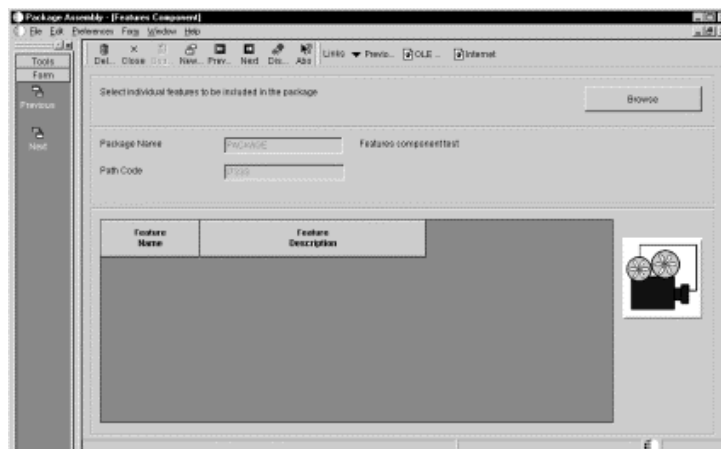
注:

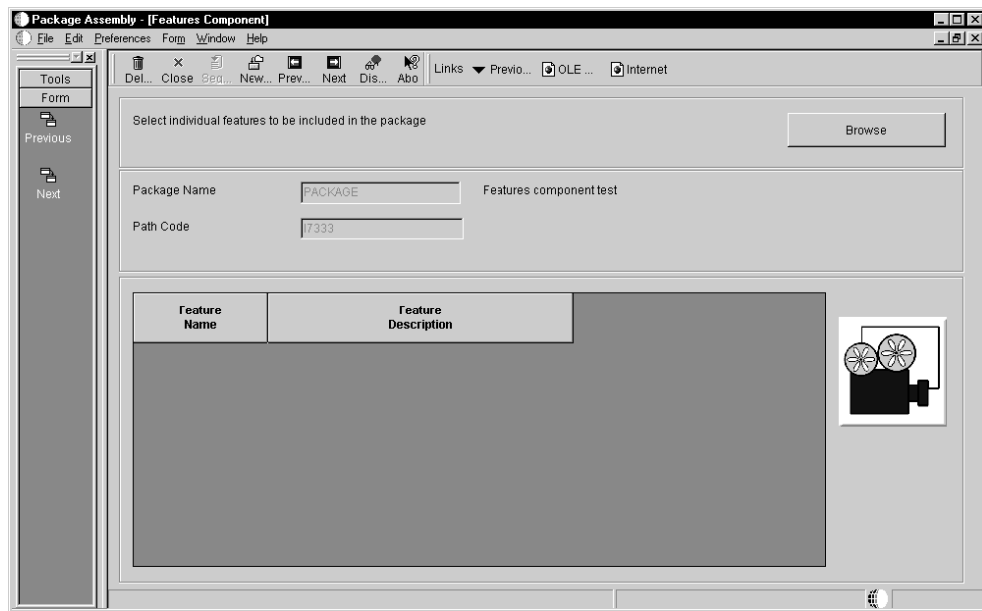
機能をパッケージに組み込むには、「機能の理解」に記載されている手順に従って、あらかじめ機能を定義しておく必要があります。

▶ パッケージ組立て時に既存の機能を新規パッケージに追加するには

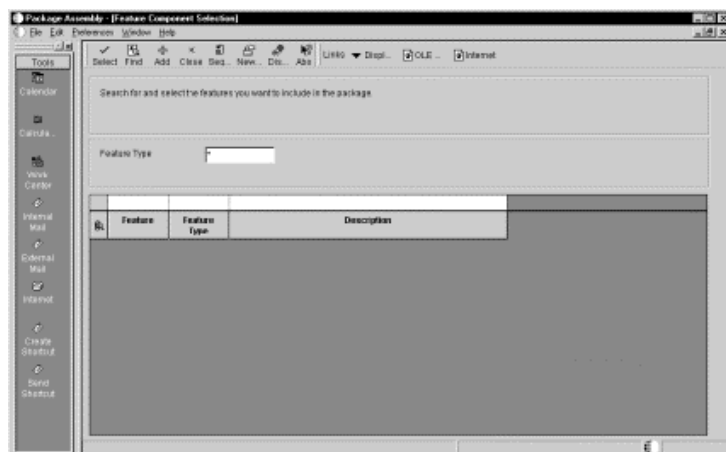
〈Package and Deployment Tools〉メニュー(GH9083)から〈Package Assembly〉プログラム(P9601)を選択します。

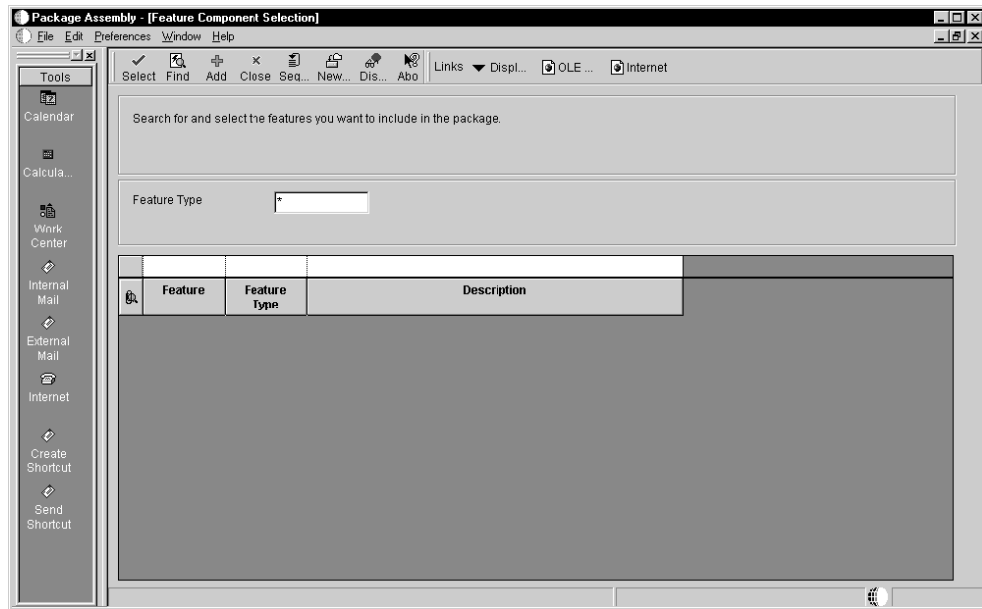
1. 〈Work with Packages〉フォームで、[Add]をクリックして新規パッケージを作成します。
2. 〈Package Assembly Directory〉の各フォームで、「パッケージの組立て」に記載されている手順を実行し、〈Feature Component〉フォームを表示させます。





3. 機能を追加するには、[Browse]をクリックします。





〈Feature Component Selection〉フォームで、[Find]をクリックして利用可能な機能を一覧表示します。

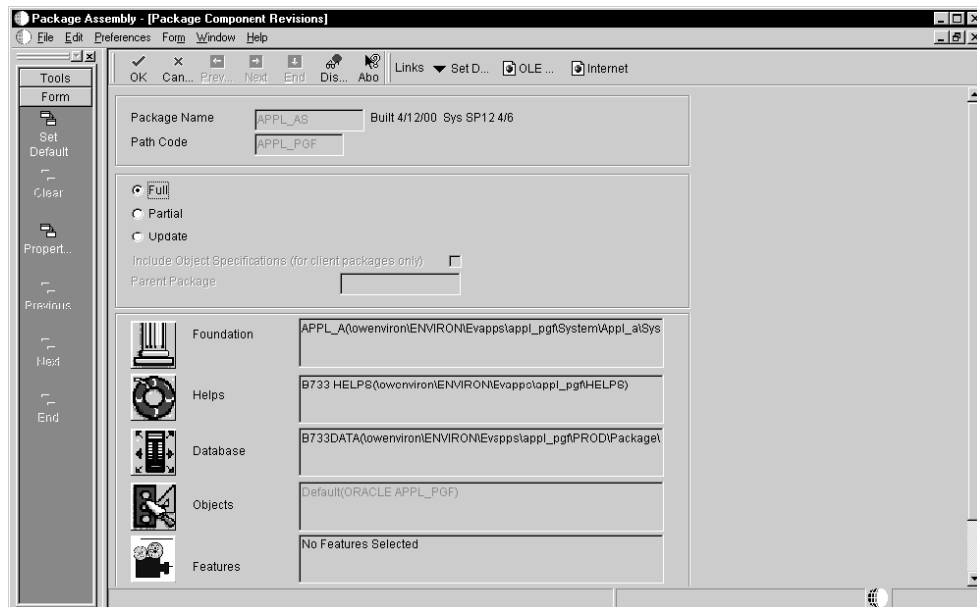
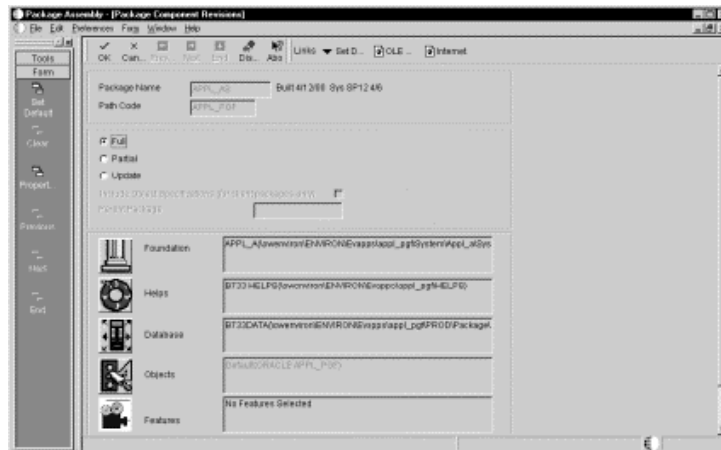
4. 次の方法のいずれかを使用して、パッケージに含める 1 つまたは複数の機能を選択します。
 - 機能を選択し、[Select]ボタンをクリックします(複数の機能を選択するには、Ctrl キーまたは Shift キーを使用します)。
 - 各機能をダブルクリックします。
5. 機能を追加し終わったら、[Close]をクリックして〈Features Component〉フォームに戻ります。選択した機能が表示されます。
6. [Next]をクリックし、残りのフォームに入力してパッケージの組立てを完了します。

▶ 機能を既存の組立て済みパッケージに追加するには

〈Package and Deployment Tools〉メニュー(GH9083)から〈Package Assembly〉プログラム(P9601)を選択します。

1. 〈Work with Packages〉フォームで、改訂用に使用できる既存パッケージに機能を追加するには、パッケージを選んで[Row]メニューから[Package Revisions]を選択します。

〈Package Component Revisions〉フォームが表示されます。



2. [Features]ボタンをクリックします。
3. 機能を追加するには、[Browse]をクリックします。
4. 〈Feature Component Selection〉フォームで、[Find]をクリックして、定義済み機能を表示します。
5. 次の方法のいずれかを使用して、追加する1つまたは複数の機能を選択します。
 - 機能を選択し、[Select]ボタンをクリックします(複数の機能を選択するには、Ctrl キーまたは Shift キーを使用します)。
 - 各機能をダブルクリックします。

選択した各機能の左の欄にチェックマークが表示されます。
6. 機能を追加したら、[Close]をクリックして〈Features Component〉フォームに戻ります。
7. [Close]をクリックして〈Package Component Revisions〉フォームに戻ります。
8. [OK]をクリックして〈Work With Packages〉フォームに戻ります。

機能の削除に関する注記

以前にパッケージに含めた機能を削除するには、〈Features Component〉上で機能を選択し、[Delete]をクリックします。

パッケージ・ビルド定義時の機能の構成

〈Package Build Definition Director〉には〈Build Features〉フォームがあり、このフォームを使用して、パッケージに含まれる機能の機能 INF ファイルを作成するかどうかを指定できます。機能内にファイルセット・コンポーネントを定義した場合は、それを圧縮するように選択できます。その他のパッケージ・ビルド・プロセスが機能に含まれている場合は、パッケージ・ビルド中にそれらを実行する前に、[Build Processes (ビルド・プロセス)]をクリックし、それらを選択しておく必要があります。

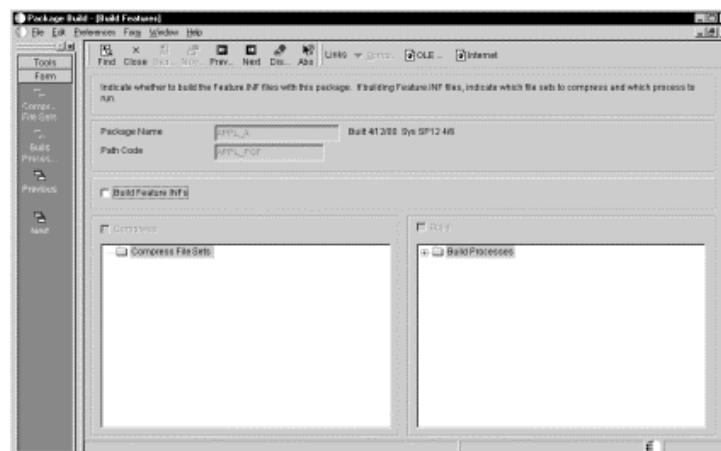
注:

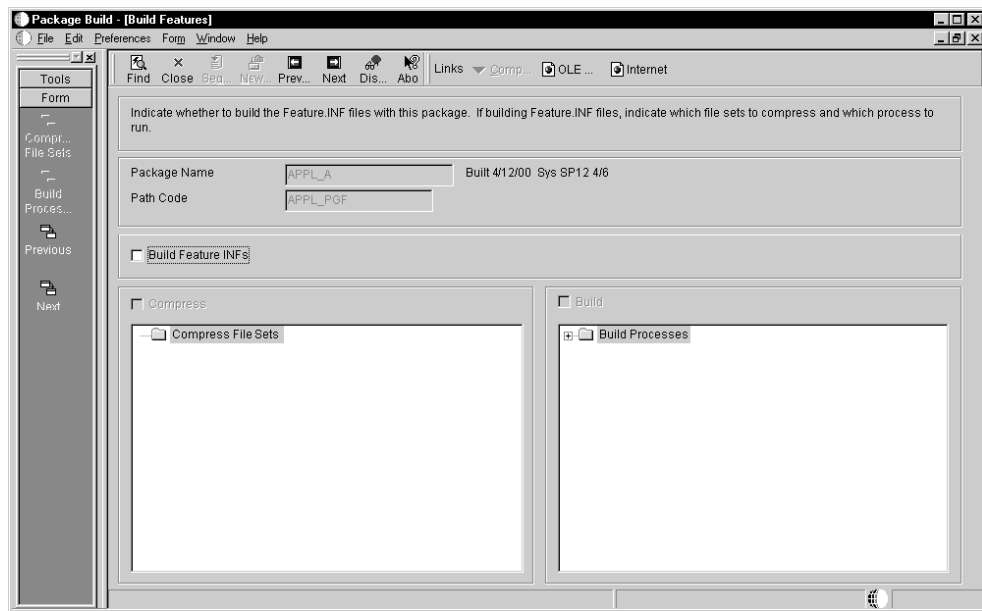
ビルド定義用の機能情報を入力するには、「パッケージ組立て時の機能の追加」に記載されている手順に従って、あらかじめ機能をパッケージに追加しておく必要があります。

▶ 新規パッケージ・ビルド定義用に機能を構成するには

〈Package and Deployment Tools〉メニュー(GH9083)から〈Package Build〉を選択します。

1. 〈Work with Package Build Definition〉フォームで、[Add]をクリックして〈Package Build Definition Director〉を起動します。
2. [Next]をクリックし、「パッケージ・ビルドを定義するには」に記載されているタスクを完了して、〈Build Features〉フォームに進みます。





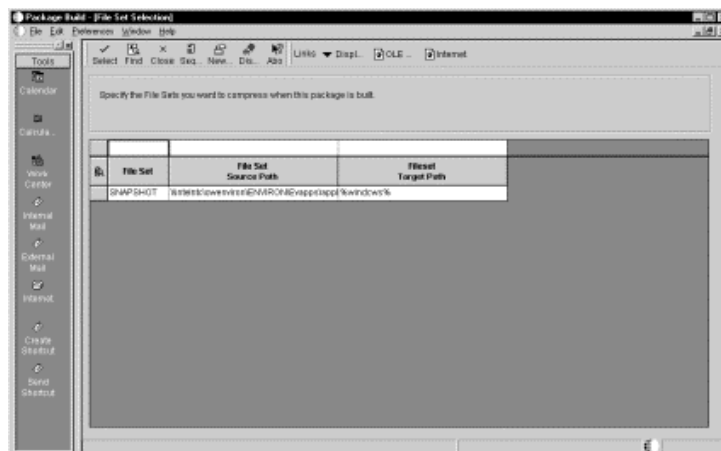
3. パッケージと共に feature.inf ファイルを作成する場合は、[Build Feature INF(ビルド機能 INF)]を選択します。

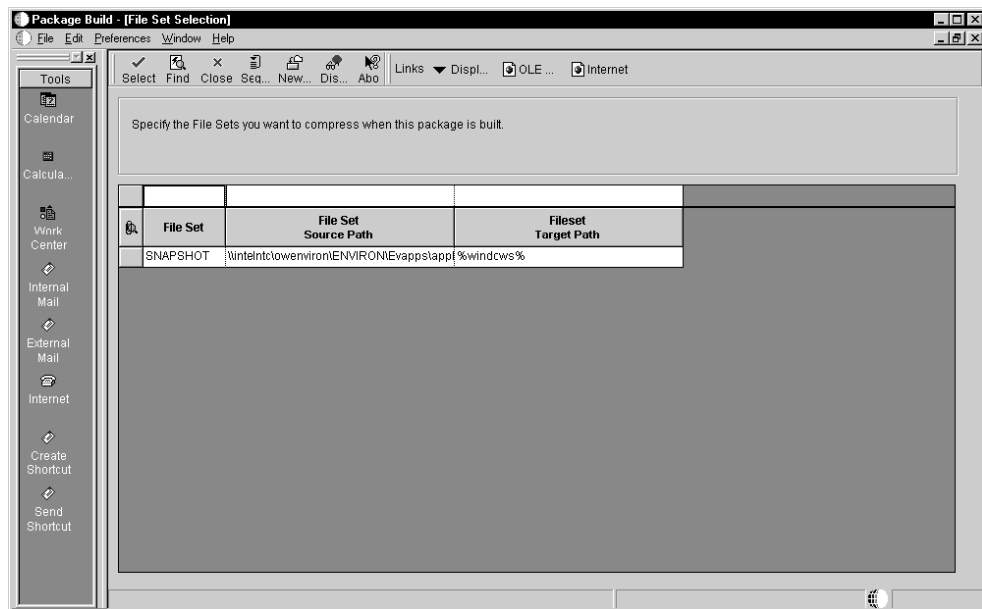
このオプションを選択すると、パッケージにファイルセットまたはその他のパッケージ・ビルド・プロセス・コンポーネントが含まれている場合は、[Compress]フィールドと[Build]フィールドが使用可能になります。

4. 次のタスクの一方または両方を実行します。
 - ファイルセットを圧縮するには
 - ERP 9.0 プロセスをビルドするには

▶ ファイルセットを圧縮するには

1. [Compress]を選び、[Form]メニューから[Compress File Sets(ファイルセットの圧縮)]を選びます。

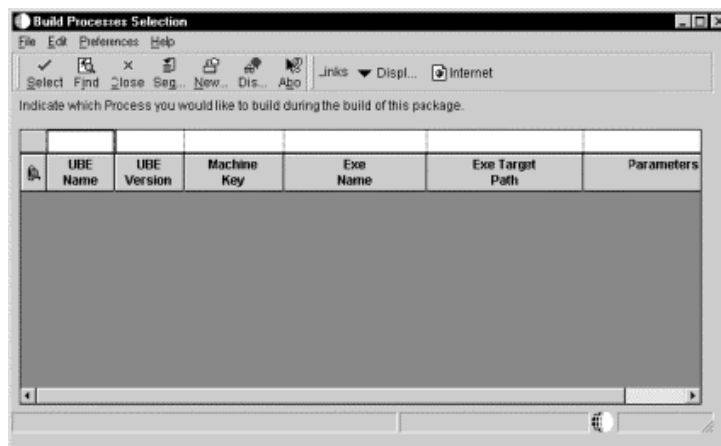


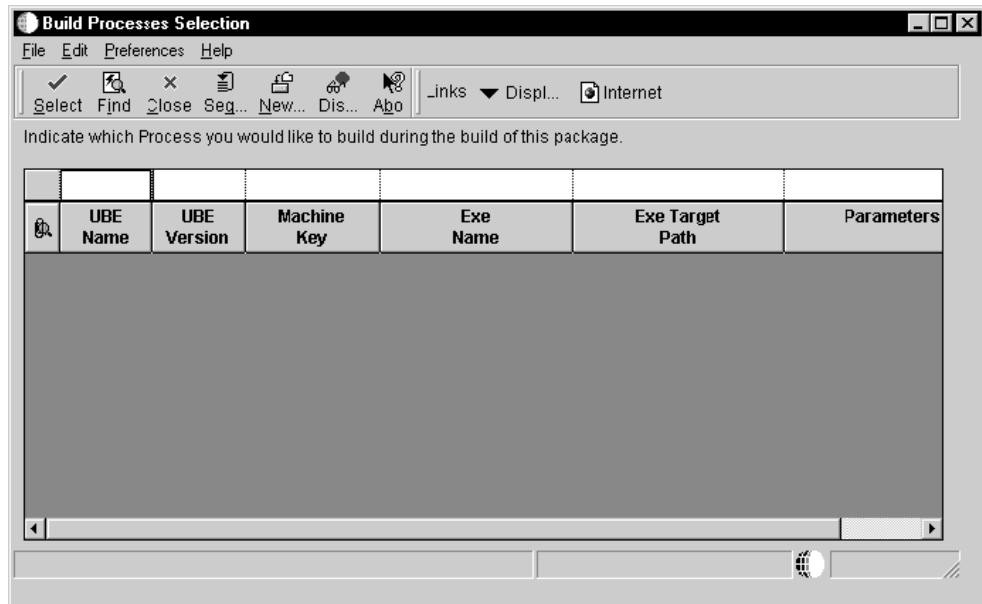


2. 〈File Set Selection (ファイルセットの選択)〉フォームで、ファイル・セットを選び、[Select]をクリックして、含める各機能を選択します。
3. ファイルセットを選択したら、[Close]をクリックします。
4. 「ERP 9.0 プロセスをビルドするには」に記載されているステップを実行するか、または[Next]をクリックして残りのフォームに進み、パッケージ・ビルドの定義を完了します。

▶ ERP 9.0 プロセスをビルドするには

1. プロセスをビルドするには、[Build]を選択し、[Select Build Processes (ビルド・プロセスの選択)]をクリックします。





2. 〈Build Processes Selection(ビルド・プロセスの選択)〉フォームで、プロセスを選択し、[Select]をクリックして、ビルドする各プロセスを選択します。
3. ビルドするプロセスを選択し終わったら、[Close]をクリックします。
4. [Form]メニューから[Build Processes]を選択し、パッケージ・ビルド時に実行する各プロセスを手動で選択します。

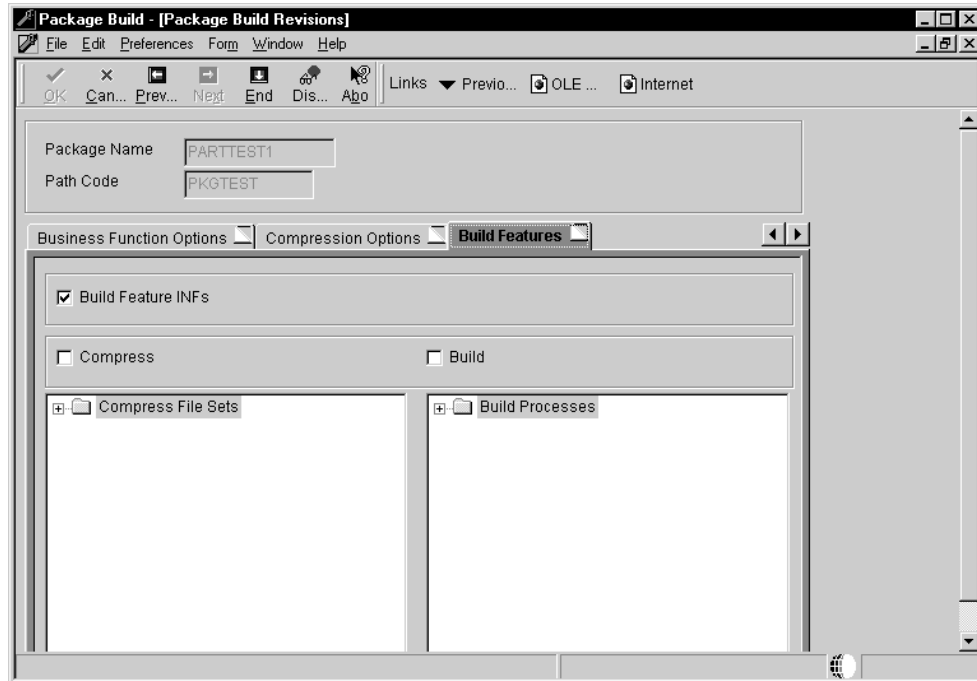
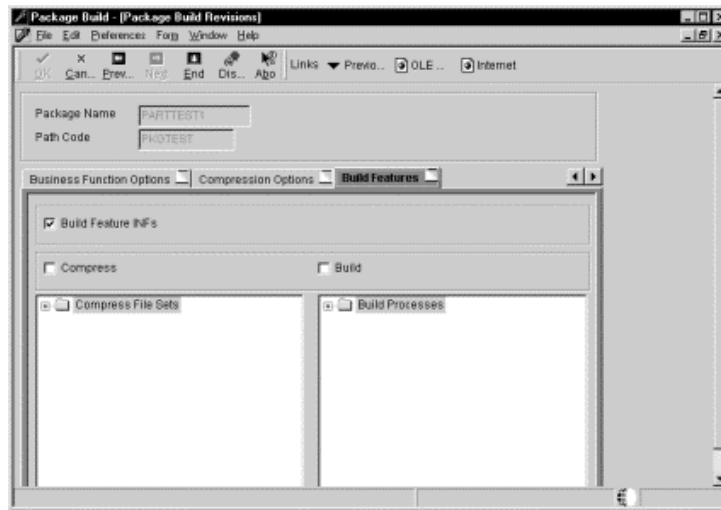
このステップを完了しないと、プロセスが機能に含まれている場合でも、プロセスはいずれも実行されません。

5. [Next]をクリックし、残りのフォームに入力してパッケージビルドの定義を完了します。

▶ 既存のパッケージ・ビルド定義に対して機能を構成するには

〈Package and Deployment Tools〉メニュー(GH9083)から〈Package Build〉プログラム(P9621)を選択します。

1. 機能を含んでいるパッケージを検索して選択します。
2. [Row]メニューの[Build Revisions]を選択します。
3. 〈Package Build Revisions〉フォームで、[Build Features(ビルド機能)]タブをクリックします。



4. 次に示す既存のビルド機能設定のいずれかを修正または追加します。
 - Build Feature INFs(ビルド機能 INF)
 - Compress
 - Build
5. [Compress]を選択した場合は、[Form]メニューから[Revise File Sets(ファイルセットの改訂)]を選択して、ファイルセットを修正します。
 <File Set Selection>フォームが表示されます。
6. ファイルセットを修正し終わったら、[Close]をクリックして<Package Build Revisions>フォームに戻ります。

7. [Build]を選んだ場合は、[Revise Processes(プロセスの改訂)]をクリックしてプロセスを修正します。

〈Build Processes Selection〉フォームが表示されます。

8. プロセスを修正し終わったら、[Close]をクリックして〈Package Build Revisions〉フォームに戻ります。
9. [Build]を選択した場合は、[Form]メニューから[Build Processes]を選び、パッケージ・ビルド時に実行する各プロセスを手動で選択します。

このステップを完了しないと、プロセスが機能に含まれている場合でも、プロセスはいずれも実行されません。

10. [OK]をクリックして、パッケージ・ビルド定義を終了します。

機能を含んだパッケージのインストール

機能を含むパッケージをワークステーションやサーバーにインストールする場合も、他のパッケージをインストールする場合と同様に、〈Workstation Installation〉アプリケーションや〈Deployment Server Installation〉アプリケーションを使用します。

これらのインストールを実行するときは、いずれも[Custom(カスタム)]オプションをオンにすることで、インストールする機能を選択できます。

参照

- ワークステーションとデプロイメント・サーバーのインストールについては『インストール・ガイド』

Package.INF ファイル内の機能項目の理解

パッケージに機能が含まれている場合は、その機能名と、機能ごとに作成される feature.INF ファイルのロケーションが Package.INF ファイルの [Features] セクションに記述されます。feature.INF ファイルには、ショートカット情報、レジストリ設定、初期化ファイル設定、環境情報といった、機能に関連する情報が記述されます。

参照

- 「パッケージ INF ファイルの理解」
- 「機能 INF ファイルの理解」

パッケージ・ビルドの履歴およびログの表示

〈Package Build History〉プログラム(P9622)を使用すると、ビルド定義作成時に指定したオプションおよびオブジェクトを含む、ビルド・プロセスに関する情報を表示できます。このプログラムでは、次のビルド情報が提供されます。

- パッケージ名
- パス・コード
- ビルドされた日時
- パッケージがビルドされたサーバー名
- 現行のビルド状況および状況記述
- 選択したスペック・テーブルの現行状況
- 書き込まれたスペックの数
- 読み書きされたパッケージ・レコード

[Form]メニューの[View Logs(ログの表示)]オプションを使用すると、ビルド・プロセスに関する追加情報を含む 4 種類のログを表示できます。これらのログは、ビルドが正常に終了せず、ビルドの過程で発生したエラーについて検討する必要がある場合に参照してください。

ビルドが正常に終了しなかった場合、[Resubmit Build]オプションを使用して、プロセスが中断したところからビルドを再開することができます。この場合、正常にビルドできなかったビジネス関数およびオブジェクトのみビルドされ、パッケージ全体はリビルドされません。

ビルドが中断または完了できない場合、ビルド状況を[Build Started]から[Build Definition Complete]にリセットする必要がある場合があります。状況をリセットすると、失敗したところからビルドを再開する[Resume Build(ビルドの再開)]機能とは違って、ビルド・プロセスが最初から開始されます。

パッケージ・ビルドの履歴表示

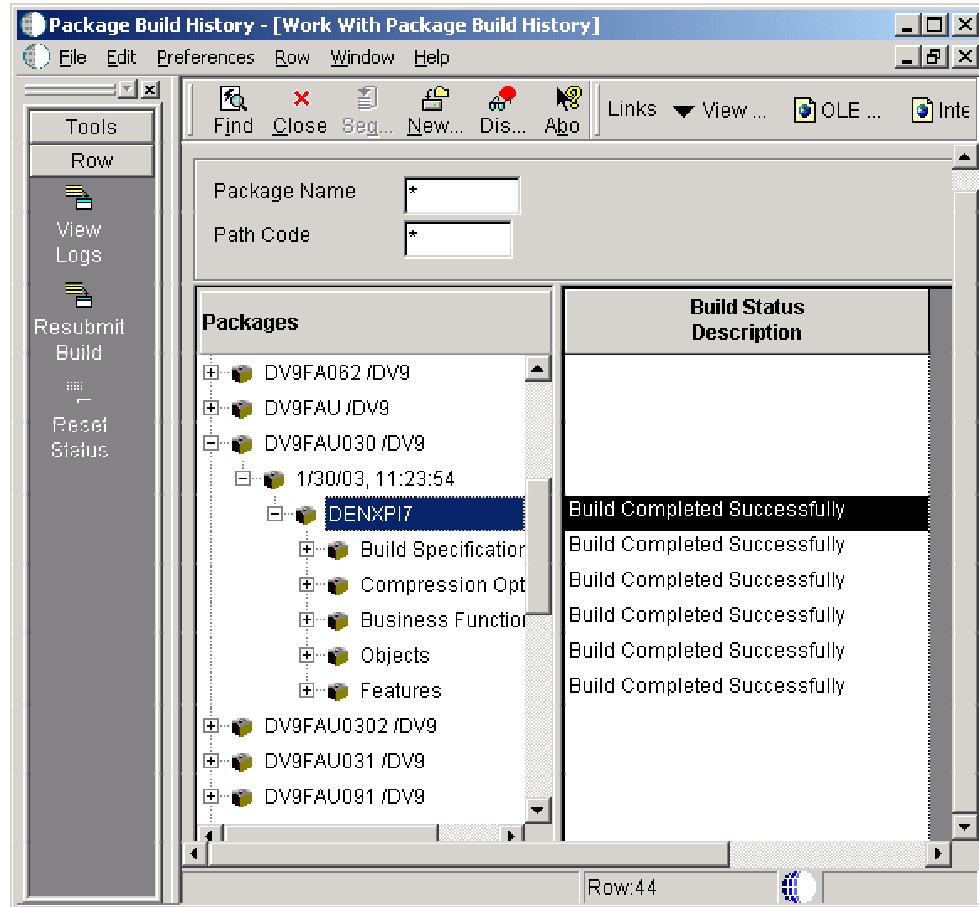
パッケージ・ビルドの履歴は、ソフトウェア・パッケージ・ビルド詳細 - 履歴テーブル(F96225)に保管されます。このテーブルには、ビルドが正常に終了しなかったパッケージ・コンポーネントに関する詳細が含まれています。

ビルド・プロセス実行中にエラーが発生し、パッケージ・ビルドが正常に終了しなかった場合、パッケージを再投入し、ビルドが失敗した時点からのビルドを続行することができます。この場合、F96225 テーブルを確認し、状況が[Not Built(ビルドされていない)]または[Error(エラー)]のビジネス関数とその他のパッケージ・コンポーネントのみ再度ビルドされます。パッケージ全体ではなく、ビルドが正常に終了しなかったパッケージ・コンポーネントのみビルドされます。この機能を使用すると、ビルドが正常に終了しなかったパッケージ・コンポーネントがわずかな場合に時間を大幅に節約できます。

パッケージの圧縮を指定していた場合に、ビルドを再開するためにパッケージを再投入すると、パッケージのビルド完了後にディレクトリが自動的に圧縮されます。

▶ パッケージ・ビルドの履歴を表示するには

〈Package and Deployment Tools〉メニュー(GH9083)から〈Package Build History〉プログラム(P9622)を選択します。



1. 〈Work with Package Build History (パッケージ・ビルド履歴の処理)〉フォームで、[CLIENT (クライアント)] またはサーバー名を選択し、それらのコンピュータでの現行のビルド状況に関する情報を表示します。ツリーを拡張して次の情報を表示することもできます。

- ビルド・スペック・オプション
- 圧縮オプション
- ビジネス関数オプション
- オブジェクト

これらのオプションとオブジェクトは、パッケージ・ビルド定義の作成時に指定したものです。たとえば、選択したスペックのみをビルドすることを選んだ場合、各スペックの状況およびその他の関連情報を確定できます。

2. ビルド履歴情報を閉じるには、[Close] をクリックします。

ログの表示

パッケージ・ビルド終了後、ビルド・プロセス時に発生したエラーを一覧するためのログを表示できます。たとえば、次のようなログを表示できます。

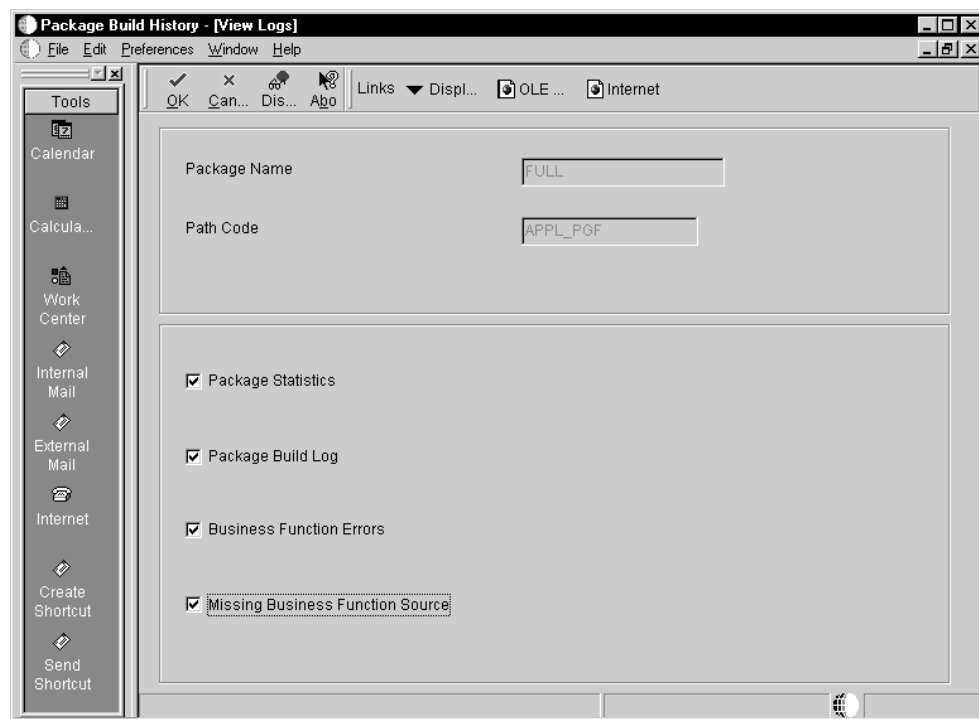
- Package Statistics Log(パッケージ統計ログ)
- Package Build Log(パッケージ・ビルド・ログ)
- Business Function Errors Log(ビジネス関数エラー・ログ)
- Missing Business Function Source Errors Log(ビジネス関数ソース検出エラー・ログ)

各ログには、パッケージ名、日付、ビルド・マシン、およびパス・コードを含む見出しが含まれています。

▶ ログを表示するには

〈Package and Deployment Tools〉メニュー(GH9083)から〈Package Build History〉プログラム(P9622)を選択します。

1. 〈Work With Package Build History〉フォームで、[Form]メニューから[View Logs]を選択します。



2. 〈View Logs〉フォームで、次のログ・オプションのいずれかをオンにして[OK]をクリックします。

- Package Statistics(パッケージ統計)
- Package Build Log(パッケージ・ビルド・ログ)
- Business Function Errors(ビジネス関数エラー)
- Missing Business Function Source(ビジネス関数ソースが検索できない)

選択したログは、それぞれ固有のウィンドウに表示されます。

3. ログの表示が完了したら、各ログ・ウィンドウを閉じます。
4. 〈View Logs〉フォームで、[Cancel]をクリックして〈Work with Package Build History〉フォームに戻ります。

フィールド記述

記述	用語解説
パッケージ統計	このオプションによって、ビルドされたパッケージ・ディレクトリのカウントおよびサイズ統計を参照することができます。
パッケージ・ビルド・ログ	パッケージのビルド時に起こったエラーを表示できます。エラーは、パッケージのスペックファイルやオブジェクトの作成時に起こることがあります。
ビジネス関数エラー	このパッケージのビジネス関数の結果を表示します。レポートにはエラーと警告数が表示されます。レポートの終わりには要約が表示され、各 DLL に対して発生したエラーと警告が表示されます。この情報を参考にして、リビルドが必要かどうか決めてください。
不足しているビジネス関数ソース	このオプションはビジネス関数が作成された時には有効でなかったすべてのソース・メンバをリストします。各メンバーは、オブジェクト・ライブラリアン・テーブル(F9860)にレコードを持っているため、プログラムによりそのメンバーの検索が試みられます。ただし、ソース・ディレクトリには合致するソースは検索されません。これらのエラーを解決するには、オブジェクト・ライブラリアン・レコードを削除するか、あるいはソース・メンバを用意します。

エラー・ログの検索場所

〈Package Build History〉プログラム(P9622)を使わずにエラー・ログを検討するには、現在のディレクトリで適切なログを検索します。エラー・ログは、デプロイメント・サーバー上のパッケージ自体のディレクトリの下位ディレクトリに保管されます。パッケージ・ビルド・ログは、パッケージ・ディレクトリに保管されます。また、Package Statistics ログ、Business Function Source Errors ログおよび Missing Business Function Source Errors ログは、パッケージの作業ディレクトリに保管されます。

エラー・ログを表示するには、適切なディレクトリにアクセスし、テキスト・ファイルを表示できる、Microsoft のメモ帳などのアプリケーションを使用します。

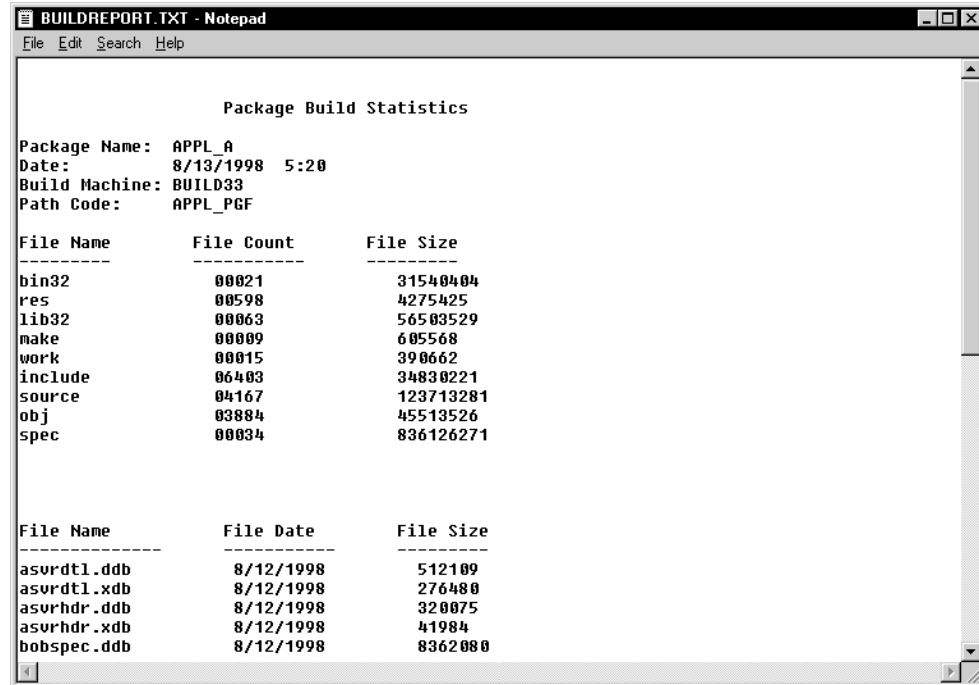
次の例では、パッケージ名として PD9FA が使用されています。実際のディレクトリを確定するには、PD9FA を実際のパッケージ名に置き換えてください。

- Package Statistics (パッケージ統計) : ¥PD9FA¥work¥buildreport.log
- Package build log (パッケージ・ビルド・ログ) : ¥PD9FA¥builderror.log
- Business function errors log (ビジネス関数エラー・ログ) ¥PD9FA¥work¥buildlog.txt
- Missing business function source log (ビジネス関数ソース検出ログ) : ¥PD9FA¥work¥NoSource.txt

Package Statistics ログ

Package Statistic ログは、パッケージ・ビルドの結果を要約し、各ディレクトリのサイズおよびファイル数を含む、パッケージ内のディレクトリに関する統計を表示します。このログは、ビルド・ディレクトリの確認に使用できる完全なビルドを表示します。このレポートには、spec ディレクトリ内のファイルの内訳、各ファイルのサイズ、およびファイルの総数と合計サイズが表示されます。このログを使用すると、パッケージが正しくビルドされたかどうかを確認できます。

Package Statistics ログの一例を次に示します。



The screenshot shows a Notepad window with the title 'BUILDREPORT.TXT - Notepad'. The text inside is as follows:

```
Package Build Statistics

Package Name:  APPL_A
Date:          8/13/1998  5:20
Build Machine: BUILD33
Path Code:     APPL_PGF

File Name      File Count      File Size
-----
bin32          00021           31540404
res            00598           4275425
lib32          00063           56503529
make           00009           605568
work           00015           390662
include        06403           34830221
source         04167           123713281
obj            03884           45513526
spec           00034           836126271

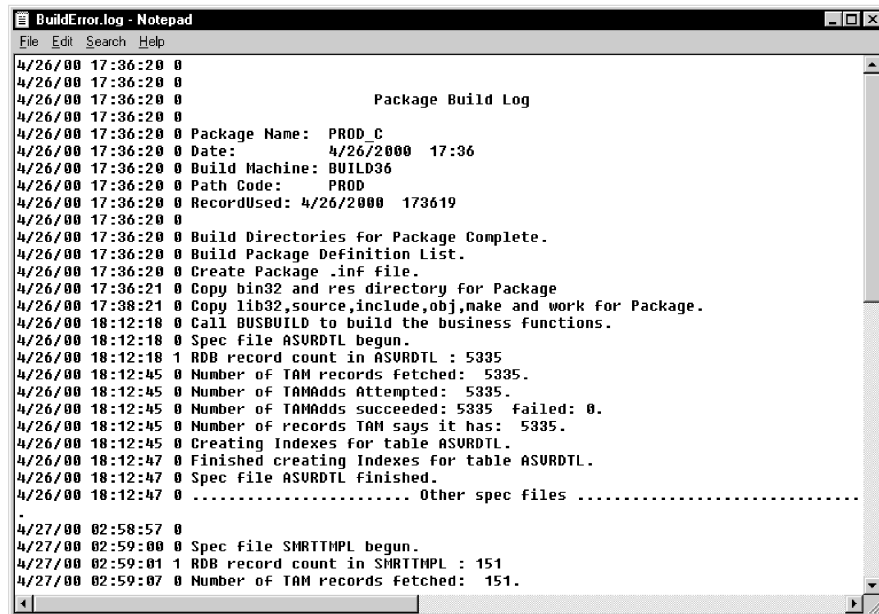
File Name      File Date       File Size
-----
asvrdtl.ddb    8/12/1998       512109
asvrdtl.xdb    8/12/1998       276480
asvrhdr.ddb    8/12/1998       320075
asvrhdr.xdb    8/12/1998       41984
bobspec.ddb    8/12/1998       8362080
```

Package Build ログ

Package Build ログは、パッケージ名ディレクトリに作成されます。このログは、パッケージ・ビルド中に完了したステップ、およびパッケージ・ビルド・プロセスで発生したエラーを一覧表示します。また、パッケージ・ビルドに関連したステップ也表示します。このログの最終ページでは、パッケージが正しくビルドされたかどうかが表示されます。

各種パッケージ・ビルド・プロセス(フル、部分、アップデート)の Package Build ログの例を次に示します。

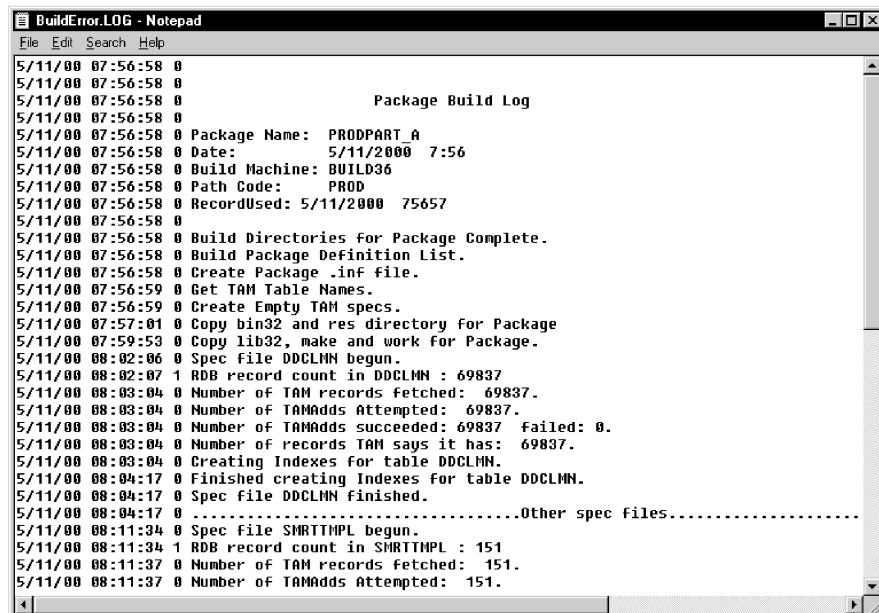
フル・パッケージ



```
BuildError.log - Notepad
File Edit Search Help

4/26/00 17:36:20 0
4/26/00 17:36:20 0
4/26/00 17:36:20 0
4/26/00 17:36:20 0
4/26/00 17:36:20 0 Package Build Log
4/26/00 17:36:20 0
4/26/00 17:36:20 0 Package Name: PROD_C
4/26/00 17:36:20 0 Date: 4/26/2000 17:36
4/26/00 17:36:20 0 Build Machine: BUILD036
4/26/00 17:36:20 0 Path Code: PROD
4/26/00 17:36:20 0 RecordUsed: 4/26/2000 173619
4/26/00 17:36:20 0
4/26/00 17:36:20 0 Build Directories for Package Complete.
4/26/00 17:36:20 0 Build Package Definition List.
4/26/00 17:36:20 0 Create Package .inf file.
4/26/00 17:36:21 0 Copy bin32 and res directory for Package
4/26/00 17:38:21 0 Copy lib32,source,include,obj,make and work for Package.
4/26/00 18:12:18 0 Call BUSBUILD to build the business functions.
4/26/00 18:12:18 0 Spec File ASURDTL begun.
4/26/00 18:12:18 1 RDB record count in ASURDTL : 5335
4/26/00 18:12:45 0 Number of TAM records fetched: 5335.
4/26/00 18:12:45 0 Number of TAMadds Attempted: 5335.
4/26/00 18:12:45 0 Number of TAMadds succeeded: 5335 failed: 0.
4/26/00 18:12:45 0 Number of records TAM says it has: 5335.
4/26/00 18:12:45 0 Creating Indexes for table ASURDTL.
4/26/00 18:12:47 0 Finished creating Indexes for table ASURDTL.
4/26/00 18:12:47 0 Spec file ASURDTL finished.
4/26/00 18:12:47 0 ..... Other spec files .....
.
4/27/00 02:58:57 0
4/27/00 02:59:00 0 Spec File SMRTTHPL begun.
4/27/00 02:59:01 1 RDB record count in SMRTTHPL : 151
4/27/00 02:59:07 0 Number of TAM records fetched: 151.
```

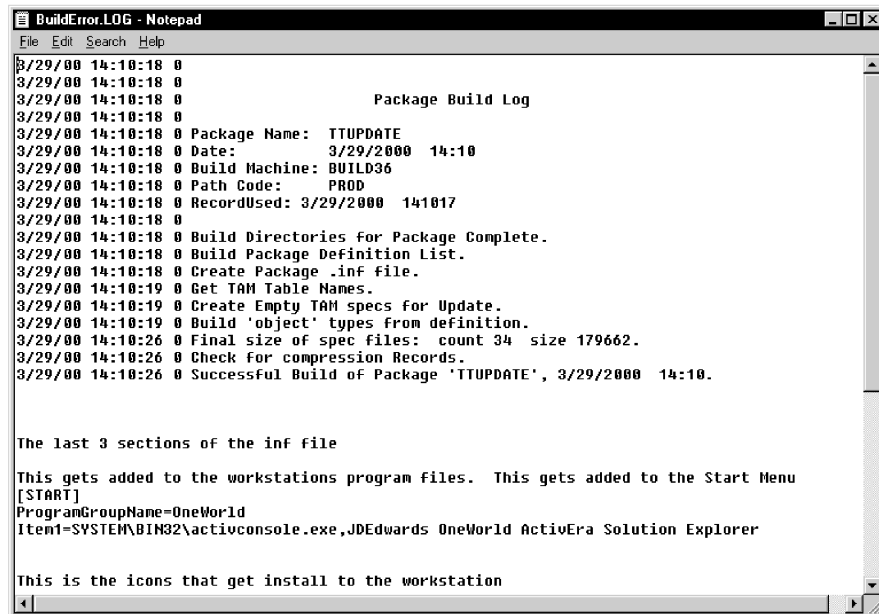
部分パッケージ



```
BuildError.LOG - Notepad
File Edit Search Help

5/11/00 07:56:58 0
5/11/00 07:56:58 0
5/11/00 07:56:58 0
5/11/00 07:56:58 0
5/11/00 07:56:58 0 Package Build Log
5/11/00 07:56:58 0
5/11/00 07:56:58 0 Package Name: PRODPART_A
5/11/00 07:56:58 0 Date: 5/11/2000 7:56
5/11/00 07:56:58 0 Build Machine: BUILD036
5/11/00 07:56:58 0 Path Code: PROD
5/11/00 07:56:58 0 RecordUsed: 5/11/2000 75657
5/11/00 07:56:58 0
5/11/00 07:56:58 0 Build Directories for Package Complete.
5/11/00 07:56:58 0 Build Package Definition List.
5/11/00 07:56:58 0 Create Package .inf file.
5/11/00 07:56:59 0 Get TAM Table Names.
5/11/00 07:56:59 0 Create Empty TAM specs.
5/11/00 07:57:01 0 Copy bin32 and res directory for Package
5/11/00 07:59:53 0 Copy lib32, make and work for Package.
5/11/00 08:02:06 0 Spec File DDCLMN begun.
5/11/00 08:02:07 1 RDB record count in DDCLMN : 69837
5/11/00 08:03:04 0 Number of TAM records fetched: 69837.
5/11/00 08:03:04 0 Number of TAMadds Attempted: 69837.
5/11/00 08:03:04 0 Number of TAMadds succeeded: 69837 failed: 0.
5/11/00 08:03:04 0 Number of records TAM says it has: 69837.
5/11/00 08:03:04 0 Creating Indexes for table DDCLMN.
5/11/00 08:04:17 0 Finished creating Indexes for table DDCLMN.
5/11/00 08:04:17 0 Spec file DDCLMN finished.
5/11/00 08:04:17 0 .....Other spec files.....
5/11/00 08:11:34 0 Spec File SMRTTHPL begun.
5/11/00 08:11:34 1 RDB record count in SMRTTHPL : 151
5/11/00 08:11:37 0 Number of TAM records fetched: 151.
5/11/00 08:11:37 0 Number of TAMadds Attempted: 151.
```

アップデート・パッケージ



```
BuildError.LOG - Notepad
File Edit Search Help

3/29/00 14:10:18 0
3/29/00 14:10:18 0
3/29/00 14:10:18 0
3/29/00 14:10:18 0
3/29/00 14:10:18 0 Package Name: TTUPDATE
3/29/00 14:10:18 0 Date: 3/29/2000 14:10
3/29/00 14:10:18 0 Build Machine: BUILD36
3/29/00 14:10:18 0 Path Code: PROD
3/29/00 14:10:18 0 RecordUsed: 3/29/2000 141017
3/29/00 14:10:18 0
3/29/00 14:10:18 0 Build Directories for Package Complete.
3/29/00 14:10:18 0 Build Package Definition List.
3/29/00 14:10:18 0 Create Package .inf file.
3/29/00 14:10:19 0 Get TAM Table Names.
3/29/00 14:10:19 0 Create Empty TAM specs for Update.
3/29/00 14:10:19 0 Build 'object' types from definition.
3/29/00 14:10:26 0 Final size of spec files: count 34 size 179662.
3/29/00 14:10:26 0 Check for compression Records.
3/29/00 14:10:26 0 Successful Build of Package 'TTUPDATE', 3/29/2000 14:10.

The last 3 sections of the inf file

This gets added to the workstations program files. This gets added to the Start Menu
[START]
ProgramGroupName=OneWorld
Item1=SYSTEM\BIN32\activconsole.exe,JDEdwards OneWorld ActivEra Solution Explorer

This is the icons that get install to the workstation
```

エラー・メッセージと警告メッセージは、それぞれ ERROR および WARNING という語の後に表示されます。これら以外のメッセージは、いずれもビルド・プロセスのステップを記述する情報メッセージです。ビルド・プロセスに失敗した場合、失敗直前に完了していたステップを確定するために、このログをトラブルシューティング・ツールとして使用します。

たとえば、ログには次のような内容が記されます。ステップの実行中または実行後にビルドが失敗した場合には、それぞれのステップに見合ったトラブルシューティングの推奨手順も合わせて示されます。

- Build directories for package complete (パッケージ用ディレクトリ作成が終了)
このステップでは、パッケージのディレクトリを作成します。ビルドでディレクトリを作成できなかった場合は、サーバーが正常に動作していることを確認してください。また、パッケージ・ディレクトリの下位ディレクトリを作成する権限が与えられていることも確認してください。
- Build package definition list (パッケージ定義リストのビルド)
このステップでは、パッケージに定義されたすべての項目のリストがビルドされます。このステップでビルドが失敗した場合、ソフトウェア・パッケージ詳細テーブル(F9631)でオブジェクトが定義されていることを確認してください。
- Create empty TAM specs (空の TAM スペックの作成)
このステップでは、specs ディレクトリの下位パッケージ・ディレクトリにローカル・ワークステーション・スペック (TAM テーブル) を作成します。このステップでビルドが失敗した場合、サーバーが正常に動作していることを確認してください。また、spec ディレクトリの下位ディレクトリを作成する許可が与えられていることも確認してください。
- Copy lib32, source, include, obj, make and work for [Full] package (フル・パッケージ用 lib32、source、include、obj、make および work のコピー)
このステップでは、lib32、source、include、obj、make および work ディレクトリがチェックイン・ロケーションからパッケージ・ディレクトリにコピーされます。このステップでビルドが失敗し

た場合、これらのディレクトリがそれぞれチェックイン・ロケーションに存在し、パッケージ・ディレクトリの下位にもあることを確認してください。

- Copy lib32, make and work for Partial package (部分パッケージ用 lib32、make および work のコピー)

このステップでは、lib32、make および work ディレクトリがチェックイン・ロケーションからパッケージ・ディレクトリにコピーされます。このステップでビルドが失敗した場合、これらのディレクトリがそれぞれチェックイン・ロケーションに存在し、パッケージ・ディレクトリの下位にもあることを確認してください。

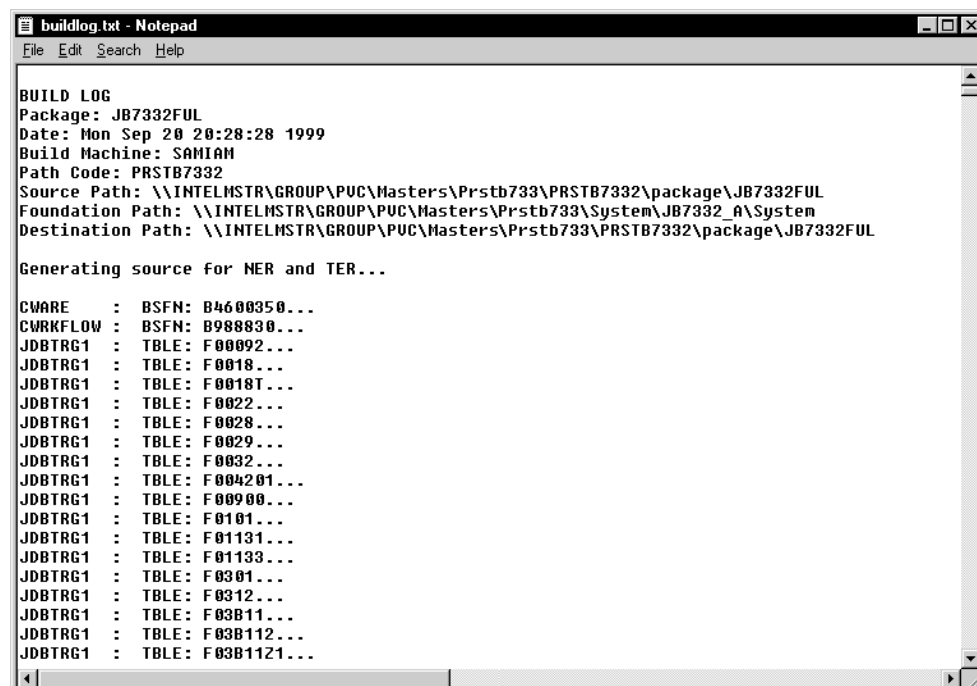
- Spec file DDCLMN begun (スペック・ファイル DDCLMN の開始)
- Spec file DDCLMN finished (スペック・ファイル DDCLMN の終了)

この 2 つのメッセージは関連しており、TAM スペック・ファイルが作成されたことを示します。begun メッセージが表示されても、finished メッセージが表示されない場合、スペック・ファイルはビルドされませんでした。パッケージが完成した後に、失敗した 1 つの TAM スペック・ファイルのみをビルドする必要があります。ログには、その TAM スペック・ファイルのビルド中に発生したエラーを示すエントリが設定される可能性があります。

Business Functions Errors ログ

Business Functions Errors ログでは、ビジネス関数のビルド中に発生したエラーを表示できます。このログの最終ページでは、ビジネス関数が正しくビルドされたか、またはビルド時にエラーが発生したかが示されます。このレポートに表示されるビジネス関数はまだ開発中であり、チェックインされていない場合があります。ビジネス関数がチェックインされていない場合、そのビジネス関数はソースをもたないので、Missing Business Function Source Errors ログに一覧表示されます。

Business Functions Errors ログの一例を次に示します。



```
buildlog.txt - Notepad
File Edit Search Help

BUILD LOG
Package: JB7332FUL
Date: Mon Sep 20 20:28:28 1999
Build Machine: SAMIAM
Path Code: PRSTB7332
Source Path: \\INTELMSTR\GROUP\PUC\Masters\Prstb733\PRSTB7332\package\JB7332FUL
Foundation Path: \\INTELMSTR\GROUP\PUC\Masters\Prstb733\System\JB7332_A\System
Destination Path: \\INTELMSTR\GROUP\PUC\Masters\Prstb733\PRSTB7332\package\JB7332FUL

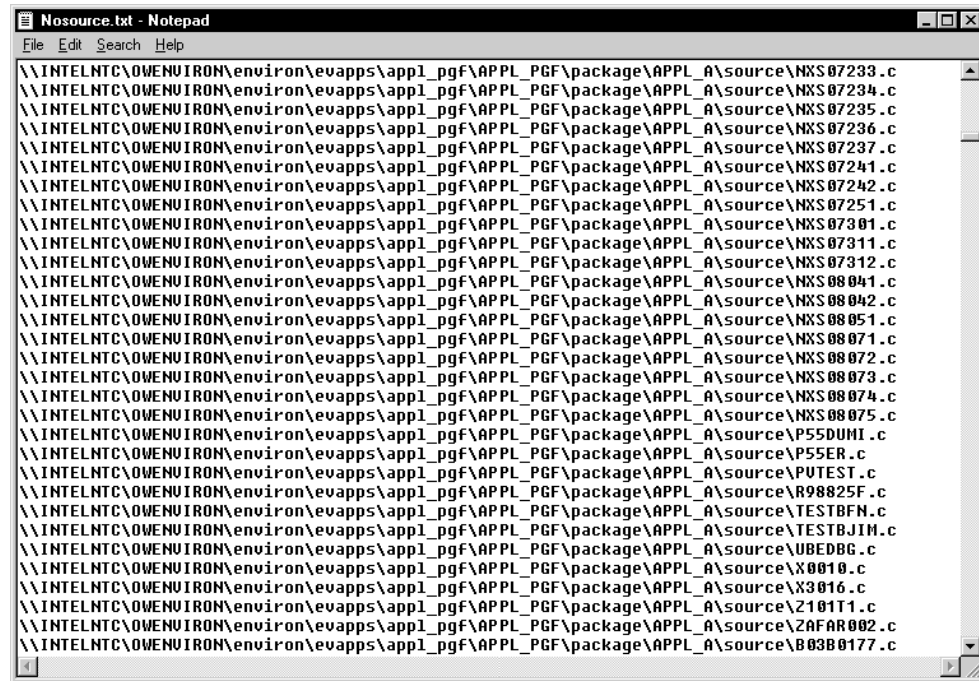
Generating source for NER and TER...

CWARE      : BSFN: B4600350...
CWRKFLOW   : BSFN: B988830...
JDBTRG1    : TBLE: F00092...
JDBTRG1    : TBLE: F0018...
JDBTRG1    : TBLE: F0018T...
JDBTRG1    : TBLE: F0022...
JDBTRG1    : TBLE: F0028...
JDBTRG1    : TBLE: F0029...
JDBTRG1    : TBLE: F0032...
JDBTRG1    : TBLE: F004201...
JDBTRG1    : TBLE: F00900...
JDBTRG1    : TBLE: F0101...
JDBTRG1    : TBLE: F01131...
JDBTRG1    : TBLE: F01133...
JDBTRG1    : TBLE: F0301...
JDBTRG1    : TBLE: F0312...
JDBTRG1    : TBLE: F03B11...
JDBTRG1    : TBLE: F03B112...
JDBTRG1    : TBLE: F03B1121...
```

Missing Business Function Source Errors ログ

Missing Business Function Source Errors ログには、パッケージに含まれるビジネス関数のうち、〈Object Librarian〉で定義され、レコードが存在するものの、ソースがなかったためにビルドできなかったものがすべて記述されます。

Missing Business Functions Source Errors ログの一例を次に示します。



```
Nosource.txt - Notepad
File Edit Search Help

\\INTELNTC\\OWENVIRON\\environ\\evapps\\app1_pg\\APPL_PGF\\package\\APPL_A\\source\\NXS07233.c
\\INTELNTC\\OWENVIRON\\environ\\evapps\\app1_pg\\APPL_PGF\\package\\APPL_A\\source\\NXS07234.c
\\INTELNTC\\OWENVIRON\\environ\\evapps\\app1_pg\\APPL_PGF\\package\\APPL_A\\source\\NXS07235.c
\\INTELNTC\\OWENVIRON\\environ\\evapps\\app1_pg\\APPL_PGF\\package\\APPL_A\\source\\NXS07236.c
\\INTELNTC\\OWENVIRON\\environ\\evapps\\app1_pg\\APPL_PGF\\package\\APPL_A\\source\\NXS07237.c
\\INTELNTC\\OWENVIRON\\environ\\evapps\\app1_pg\\APPL_PGF\\package\\APPL_A\\source\\NXS07241.c
\\INTELNTC\\OWENVIRON\\environ\\evapps\\app1_pg\\APPL_PGF\\package\\APPL_A\\source\\NXS07242.c
\\INTELNTC\\OWENVIRON\\environ\\evapps\\app1_pg\\APPL_PGF\\package\\APPL_A\\source\\NXS07251.c
\\INTELNTC\\OWENVIRON\\environ\\evapps\\app1_pg\\APPL_PGF\\package\\APPL_A\\source\\NXS07301.c
\\INTELNTC\\OWENVIRON\\environ\\evapps\\app1_pg\\APPL_PGF\\package\\APPL_A\\source\\NXS07311.c
\\INTELNTC\\OWENVIRON\\environ\\evapps\\app1_pg\\APPL_PGF\\package\\APPL_A\\source\\NXS07312.c
\\INTELNTC\\OWENVIRON\\environ\\evapps\\app1_pg\\APPL_PGF\\package\\APPL_A\\source\\NXS08041.c
\\INTELNTC\\OWENVIRON\\environ\\evapps\\app1_pg\\APPL_PGF\\package\\APPL_A\\source\\NXS08042.c
\\INTELNTC\\OWENVIRON\\environ\\evapps\\app1_pg\\APPL_PGF\\package\\APPL_A\\source\\NXS08051.c
\\INTELNTC\\OWENVIRON\\environ\\evapps\\app1_pg\\APPL_PGF\\package\\APPL_A\\source\\NXS08071.c
\\INTELNTC\\OWENVIRON\\environ\\evapps\\app1_pg\\APPL_PGF\\package\\APPL_A\\source\\NXS08072.c
\\INTELNTC\\OWENVIRON\\environ\\evapps\\app1_pg\\APPL_PGF\\package\\APPL_A\\source\\NXS08073.c
\\INTELNTC\\OWENVIRON\\environ\\evapps\\app1_pg\\APPL_PGF\\package\\APPL_A\\source\\NXS08074.c
\\INTELNTC\\OWENVIRON\\environ\\evapps\\app1_pg\\APPL_PGF\\package\\APPL_A\\source\\NXS08075.c
\\INTELNTC\\OWENVIRON\\environ\\evapps\\app1_pg\\APPL_PGF\\package\\APPL_A\\source\\P55DUMI.c
\\INTELNTC\\OWENVIRON\\environ\\evapps\\app1_pg\\APPL_PGF\\package\\APPL_A\\source\\P55ER.c
\\INTELNTC\\OWENVIRON\\environ\\evapps\\app1_pg\\APPL_PGF\\package\\APPL_A\\source\\PUTEST.c
\\INTELNTC\\OWENVIRON\\environ\\evapps\\app1_pg\\APPL_PGF\\package\\APPL_A\\source\\R98825F.c
\\INTELNTC\\OWENVIRON\\environ\\evapps\\app1_pg\\APPL_PGF\\package\\APPL_A\\source\\TESTBFN.c
\\INTELNTC\\OWENVIRON\\environ\\evapps\\app1_pg\\APPL_PGF\\package\\APPL_A\\source\\TESTBJIM.c
\\INTELNTC\\OWENVIRON\\environ\\evapps\\app1_pg\\APPL_PGF\\package\\APPL_A\\source\\UBEDBG.c
\\INTELNTC\\OWENVIRON\\environ\\evapps\\app1_pg\\APPL_PGF\\package\\APPL_A\\source\\X0010.c
\\INTELNTC\\OWENVIRON\\environ\\evapps\\app1_pg\\APPL_PGF\\package\\APPL_A\\source\\X3016.c
\\INTELNTC\\OWENVIRON\\environ\\evapps\\app1_pg\\APPL_PGF\\package\\APPL_A\\source\\Z10111.c
\\INTELNTC\\OWENVIRON\\environ\\evapps\\app1_pg\\APPL_PGF\\package\\APPL_A\\source\\ZAFAR002.c
\\INTELNTC\\OWENVIRON\\environ\\evapps\\app1_pg\\APPL_PGF\\package\\APPL_A\\source\\B03B0177.c
```

サーバー・ログ

エンタープライズ・サーバーのコンパイル・ログは、いずれもそのサーバー自体の、オブジェクトが属する DLL のソース・ディレクトリに置かれます。たとえば、BuildArea が /u02/jdedwardsoneworld/packages であるパッケージ PACKAGE1 が HP9000 上に存在し、そのパッケージに含まれている〈Sales Order Entry(受注オーダー入力)〉マスター・ビジネス関数(B4200310)のログを表示する場合を考えてみます。この場合、B4200310 は CDIST.DLL に含まれているので、/u02/jdedwardsoneworld/packages/PACKAGE1/source/CDIST/b4200310.log というファイルが生成されます。

HP9000 上の CDIST.DLL(共有ライブラリ)をリンクできない場合は、/u02/jdedwardsoneworld/packages/PACKAGE1/obj/CDIST/CDIST.log というファイルが生成されます。

AS/400 の場合、コンパイルに失敗したビジネス関数のログは、パッケージ・ライブラリ内の FAILED というファイルのメンバです。前の例では、ライブラリ PACKAGE1 内の FAILED ファイルのメンバ B4200310 を表示できます。

パッケージ・ビルドの再投入

すべてのオブジェクトまたはビジネス関数が正常にビルドされなかった場合は、パッケージ・ビルドを再投入できます。

▶ パッケージ・ビルドを再投入するには

〈Package and Deployment Tools〉メニュー(GH9083)から〈Package Build History〉プログラム(P9622)を選択します。

1. 〈Work with Package Build History〉フォームが表示されます。次のオプションのいずれかを使用して、再投入するパッケージを検索し、選択します。
 - 特定サーバーのビルドだけを再投入するために、そのサーバーを選択します。
 - ワークステーション・ビルドだけを再投入するために、[CLIENT]ヘッダーを選択します。
2. [Row]メニューから[Resubmit Build]を選択します。

ビルドの初回投入時に NER を生成した場合、NER を再生成するかどうかを確認するウィンドウが表示されます。

3. NER を再生成する場合は[OK]をクリックし、このプロセスをスキップする場合は[Cancel]をクリックします。

注:

NER を再生成しない場合は、〈Package Build History〉プログラムの[Generate NER(NER の生成)]処理オプションに"2"を入力すると、このウィンドウが表示されなくなります。

4. ビルド・レポートの出力先として次のいずれかを選択して、[OK]をクリックします。

- On Screen
- To Printer

フォームが自動的に閉じて、パッケージのビルドが開始されます。ビルドの時間は、パッケージに含まれる項目数とサイズによって異なります。ビルドが完了すると、指定した出力先に応じて、レポートが画面に表示されるか、印刷されます。

5. レポートを検討し、パッケージ内のコンポーネントがすべて正常に生成されたことを確認します。レポートにエラーがある場合、エラー・ログで詳細を検討してください。

参照

- パッケージの配布スケジュールについては「配布」

ビルド状況の変更

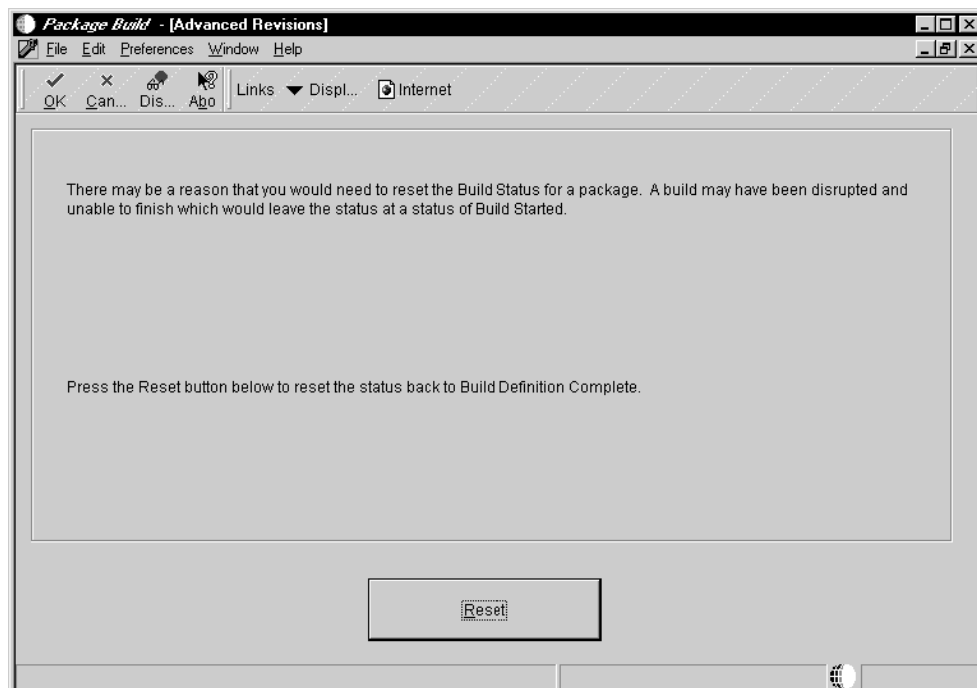
ビルドが失敗した時点からビルドを再開するのではなく、パッケージをリビルドしなければならない場合があります。リビルドする場合は、事前にパッケージのビルド状況を「Build Started」から「Build Definition Complete」に変更する必要があります。

パッケージのビルド状況をリセットするときは、パッケージをビルドするサーバーのみに対して、またはすべてのサーバーとクライアント・ワークステーションに対して状況をリセットできます。

▶ ビルド状況を変更するには

〈Package and Deployment Tools〉メニュー(GH9083)から〈Package Build〉プログラム(P9621)を選択します。

1. 〈Work with Package Build Definition〉フォームが表示されます。状況をリセットするパッケージを検索します。パッケージ名の下で、パッケージをビルドするサーバーのみ、またはサーバーとクライアント・ワークステーションを選択します。
2. [Row]メニューから[Advanced]を選択します。



3. 〈Advanced Revisions (上級改訂)〉フォームで、[Reset (リセット)]をクリックし、パッケージのビルド状況を「Build Started」から「Build Definition Complete」に変更します。
4. [OK]をクリックして〈Work with Package Build Definition〉フォームに戻ります。
5. 必要に応じて、パッケージ名を選択し、[Row]メニューから[Submit Build]を選択します。
6. 現在のビルドを削除するか、削除せずに処理を継続するかを確認するメッセージが表示されます。

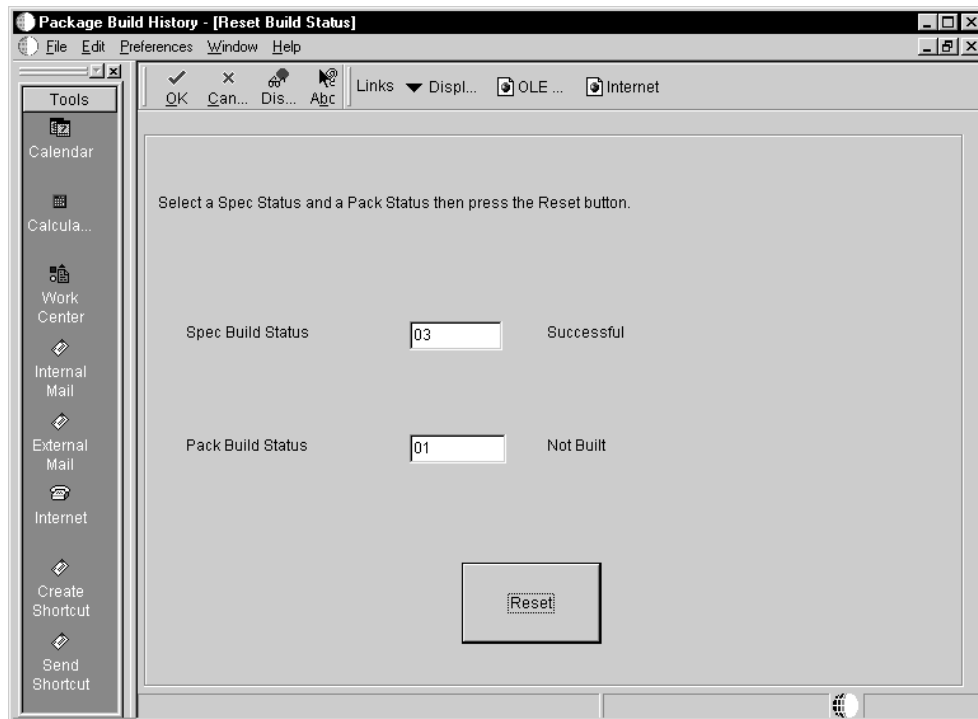
スペック・ビルド状況とパック・ビルド状況の変更

また、パッケージのスペック状況とパック状況を、指定した状況にリセットすることもできます。このオプションは、〈Package Build History〉プログラム(P9622)から使用できます。

▶ スペック状況とパック状況を変更するには

〈Package and Deployment Tools〉メニュー(GH9083)から〈Package Build History〉プログラム(P9622)を選択します。

1. 〈Work with Package Build History〉フォームで、状況をリセットするパッケージを検索し、パッケージを展開して、個々の項目を選択します。
2. [Row]メニューから[Reset Status(リセット状況)]を選択します。



3. 〈Reset Build Status(ビルド状況のリセット)〉フォームで、[Spec Build Status(スペック・ビルド状況)]フィールドと[Pack Build Status(パック・ビルド状況)]フィールドに、適切な状況を入力します。

この2つのフィールドは、使用可能な状況の決定を支援するビジュアル・アシスト機能を備えています。

4. [Reset]をクリックします。
5. [OK]をクリックして〈Work with Package Build History〉フォームに戻ります。

パッケージ INF ファイルの理解

パッケージ INF ファイルは、パッケージ・ビルドと〈Workstation Installation〉プログラムとの間の重要なインターフェイスです。パッケージ INF ファイルは、パッケージに含まれるコンポーネント、これらのコンポーネントのソースおよび指定先パス、およびパッケージのインストールに必要な属性を定義します。

INF ファイルはパッケージ・ビルドのプロセス実行時に作成され、リリース・マスター・ディレクトリの下独自の package_inf ディレクトリに保管されます。〈Workstation Installation〉は、ワークステーションにロードするコンポーネントおよびそれらのロケーションを確定するために、インストールするパッケージのパッケージ INF ファイルを読み取ります。

INF ファイルの例

パッケージ PD9FA(PD9 パス・コードのフル・パッケージ A)の標準的な INF ファイルを次に示します。

[Attributes]

このセクションには、ERP 9.0 の現行リリース、グローバル・テーブル、スペック・ファイル、ヘルプ・ファイル、および JDE.INI ファイルに関する情報が含まれます。

項目	目的
PackageName=STEST	パッケージ名を示します。
PathCode=	パッケージがビルドされるパス・コードを示します。
Built=	パッケージ状況を示します。状況が「50」または「70」の場合は、パッケージをインストールする準備が整っていることを意味します。
PackageType=Full	パッケージ・タイプを示します(フル、部分、アップデート)。
Release=B9	ERP 9.0 の現行リリースを示します。これにより、ワークステーション用の jde.ini ファイルをビルドするときに使用する setup.inf ファイルが確定します。このリリースは、system および helps のパスを確定する場合にも使用されます。
SystemBuildType	ビルド・タイプ(DEBUG または RELEASE)を示します。このオプションは owver.dll から取り込まれます。
MFCVersion	MFC コンパイラのバージョンを示します。
SpecFilesAvailable=Y	スペック・ファイルがマージまたはコピーできることを示します。このオプションは、フル・パッケージおよび部分パッケージの場合、常に「Y」に設定されます。アップデート・パッケージの場合、パッケージがオブジェクトに含まれている場合にのみ、「Y」に設定されます。
DelBtblTbl=Y	インストール時にグローバル・テーブルを削除するかどうかを示します。このオプションは、フル・パッケージおよび部分パッケージの場合、「Y」に設定されます。アップデート・パッケージの場合は、オブジェクトにテーブル・オブジェクトが含まれる場合のみ「Y」に設定されます。

項目	目的
ReplaceIni=Y	インストール時に既存の jde.ini ファイルを削除して新規のファイルを作成するかどうかを示します。このオプションは、フル・パッケージおよび部分パッケージの場合、「Y」に設定されます。アップデート・パッケージの場合、パッケージ・ビルド定義時に jde.ini ファイルを置換するかどうかを指示する必要があります。
AppBuildDate=Mon Jul 20 11:22:22 2005	パッケージがビルドされた日時を示します。フル・パッケージおよび部分パッケージでは、常にこの日時が設定されます。パッケージにオブジェクトが含まれていない場合はブランクになります。
FoundationBuildDate=Wed Jun 03 15:08:34 2005	ファンデーションがビルドされた日時を示します。ERP 9.0 以降のリリースでは、この日時は owver.dll から取り込まれます。フル・パッケージおよび部分パッケージでは、常にこの日時が設定されます。パッケージにファンデーション・ロケーションが指定されていない場合は、ブランクになります。
DataBuildDate=Wed Jun 03 15:08:34 2005	データベース・ファイル(jdeb7.mdb)がビルドされた日時を示します。フル・パッケージおよび部分パッケージでは、常にこの日時が設定されます。パッケージでデータベース・ロケーションが指定されていない場合は、ブランクになります。
HelpBuildDate=Wed Jun 03 15:08:34 2005	ヘルプ(help.jz、圧縮ヘルプ・ファイル)がビルドされた日時を示します。フル・パッケージおよび部分パッケージでは、常にこの日時が設定されます。パッケージでヘルプ・ロケーションが指定されていない場合は、ブランクになります。
DeploymentServerName	デプロイメント・サーバーのコンピュータ名を示します。
Location	コンピュータ(デプロイメント・サーバー)のロケーションを示します。
DeploymentStatus=	パッケージ・デプロイメント状況を示します。
PackageDescription=	パッケージ記述を示します。
Icon Description	デスクトップ・アイコンを示します。
Default Environment	デフォルト環境を示します。
Spec	ファイルセットで指定されているスペック・ファイル名を示します。
ServerHelpPath	ヘルプ・ファイルへのパスを示します。

[SrcDirs]

〈Workstation Installation〉プログラムは情報のコピー元のソースパスを確定するために、これらの設定を使用します。パッケージ・ビルドはこれらのディレクトリを圧縮します。〈Workstation Installation〉は、圧縮済みディレクトリをワークステーションにコピーします。

項目	目的
SPathcode=¥¥MachineName¥ ONEWORLD¥B9¥PACKAGE¥ PD9FA	パッケージ・ビルドで〈Workstation Installation〉用に圧縮するパッケージのロケーションを示します。このパスのデフォルト値は、パッケージをビルドしたパス・コード・ディレクトリです。別のパッケージを使用する場合は、この設定を変更できます。
SSYS=¥¥MachineName¥ ONEWORLD¥B9¥SYSTEM	パッケージ・ビルドで〈Workstation Installation〉用に圧縮するシステム・ディレクトリのロケーションを示します。このパスのデフォルト値は、パッケージをビルドしたパス・コードに関連付けられているシステム・ディレクトリです。通常、このディレクトリは、リリース共有名(B9)の下位ディレクトリです。この項目は、パッケージに含まれている場合のみ表示されます。
SPathcodeDATA=¥¥MachineName¥ ONEWORLD¥B9¥ PD9¥PACKAGE¥DATA	パッケージ・ビルドで〈Workstation Installation〉用に圧縮するJ.D. Edwards サポート対象ローカル・データベースのロケーションを示します。このパスのデフォルト値は、パッケージをビルドしたパス・コードの下位データ・ディレクトリです。この項目は、パッケージに含まれている場合のみ表示されます。
SHELP=¥¥MachineName¥ ONEWORLD¥B9¥HELPS	パッケージ・ビルドで〈Workstation Installation〉用に圧縮するヘルプ・ファイルのロケーションを示します。ユーザーは、ワークステーションのディスク領域をヘルプ・ファイルに割り当てない場合があります。パッケージをビルドするとき、この設定のデフォルト値は、リリース共有パス(B9)の下のヘルプ・ディレクトリになります。この設定は、ヘルプが存在するロケーションを変更する場合に変更できます。この項目は、パッケージに含まれている場合のみ表示されます。

[DestDirs]

〈Workstation Installation〉プログラムはワークステーション上の指定先パスを確定するために、これらの設定を使用します。「%INSTALL」は、ユーザー表示設定テーブル(F00921)とユーザー定義コード言語状況テーブル(F00051)で設定されているユーザーのコンピュータ構成に置き換えられます。

項目	目的
DPathcode=%INSTALL¥path code	パッケージのインストール先ディレクトリを示します。
DSYS=%INSTALL¥system	システム・ファイルのインストール先ディレクトリを示します。この項目は、パッケージに含まれている場合のみ表示されます。
DPathcodeDATA=%INSTALL¥path code¥data	データベースのインストール先ディレクトリを示します。この項目は、パッケージに含まれている場合のみ表示されます。
DHELP=%INSTALL¥helps	ヘルプ・ファイルのインストール先ディレクトリを示します。この項目は、パッケージに含まれている場合のみ表示されます。

[Filesets]

このセクションの設定は、パッケージのパス・コードの下位にあるさまざまなソース・ディレクトリとインストール先ディレクトリを示します。「Y」は圧縮されていることを、また「N」は圧縮されていないことを示します。ソース・ディレクトリ名とインストール先ディレクトリ名は、それぞれ S および D で始まります。

項目	目的
Pathcode1=Y, \$Spathcode¥bin32, \$Dpathcode¥bin32	パス・コードの下位の bin32 ディレクトリを示します。
Pathcode2=Y, \$Spathcode¥res, \$Dpathcode¥res	パス・コードの下位の res ディレクトリを示します。
Pathcode3=Y, \$Spathcode¥spec, \$Dpathcode¥spec	パス・コードの下位の spec ディレクトリを示します。
Pathcode4=Y, \$Spathcode¥include, \$Dpathcode¥include	パス・コードの下位の include ディレクトリを示します。
Pathcode5=Y, \$Spathcode¥lib, \$Dpathcode¥lib	パス・コードの下位の lib ディレクトリを示します。
Pathcode6=Y, \$Spathcode¥obj, \$Dpathcode¥obj	パス・コードの下位の obj ディレクトリを示します。
Pathcode7=Y, \$Spathcode¥source, \$Dpathcode¥source	パス・コードの下位の source ディレクトリを示します。
Pathcode8=Y, \$Spathcode¥work, \$Dpathcode¥work	パス・コードの下位の work ディレクトリを示します。
Pathcode9=Y, \$Spathcode¥make, \$Dpathcode¥make	パス・コードの下位の make ディレクトリを示します。
PathcodeDATA=Y, \$SpathcodeDATA¥data.CAB, \$DpathcodeDATA	パッケージ・ビルドが生成する圧縮データベースを示します。
HELP=Y, \$SHELP¥Helps.CAB, \$DHELP	圧縮ヘルプ・ファイルを示します。

[Components]

このセクションの設定は、ファンデーション、本稼働用および開発オブジェクト、データベース、およびヘルプ・ファイルのロケーションを示します。

項目	目的
ProdObj=APPL_PKG1, APPL_PKG2, APPL_PKG3	本稼働用オブジェクトのロケーションを示します。
DevObj=APPL_PKG4, APPL_PKG5, APPL_PKG6, APPL_PKG7, APPL_PKG8, APPL_PKG9	開発オブジェクトのロケーションを示します。
Foundation=SYS	ファンデーション・ロケーションを示します。
Data=<pathcode> DATA	データベース・ロケーションを示します。
Help=HELP	ヘルプ・ファイルのロケーションを示します。

[Typical]

このセクションでは、標準的な開発ユーザー設定について説明します。

項目	目的
Name=Development	パッケージが開発ユーザー用であることを示します。
Description=Install the development objects	パッケージ記述を示します。
Components=ProdObj, DevObj, Foundation, Data	パッケージに本稼働用オブジェクト、開発オブジェクト、データベースおよびファンデーションが含まれている必要があることを示します。これらのコンポーネントのロケーションを確定するには、「Components の設定」を参照してください。

[Compact]

このセクションでは、標準的な本稼働用ユーザー設定について説明します。

項目	目的
Name=Production	パッケージが本稼働用ユーザー用であることを示します。
Description=Install the production objects	パッケージ記述を示します。
Components=ProdObj, Foundation, Data	パッケージに本稼働用オブジェクト、データベースおよびファンデーションが含まれている必要があることを示します。これらのコンポーネントのロケーションを確定するには、「Components の設定」を参照してください。

[MSDE]

このセクションには、ローカル・データ・ソースに関する情報が含まれています。このセクションの名前は、[ODBC Data Sources(ODBC データ・ソース)]セクションのローカル・データ・ソース名に対応します。

項目	目的
Driver=jdboledb.dll	ドライバ名を示します。
DefaultDir=C:\ACCESS	デフォルト・ディレクトリを示します。
UID=sa	サポート対象ローカル・データベースの必須属性設定を示します。
Driver32=jdboledb.dll	サポート対象ローカル・データベース用の ODBC ドライバを示します。
DBQ=\$DAPPL_PGFDATA¥JDELocal.mdf	サポート対象ローカル・データベース用のパス・コードを示します。
MaxBufferSize=2048	バッファの最大サイズを示します。
Threads=1028	スレッドの数を指定します。

[Features]

このセクションでは、パッケージに含まれている機能について説明します。機能とは、ワークステーションやサーバーへ配布するためにパッケージ内に包含されるファイルセットや構成オプションのことです。機能の詳細については、「パッケージへの機能の組み込み」を参照してください。

項目	目的
FeatureName=SFEAT	パッケージ内の機能の名前を示します。
Feature.inf location=	機能の feature.inf ファイルの位置を示します。

機能 INF ファイルの理解

パッケージに機能が含まれている場合は、パッケージ INF ファイルの [Features] というセクションに、機能名と、パッケージ内の機能ごとに作成される機能 INF ファイルへのポインタが記述されます。これらの機能 INF ファイルは、インストール中に実行するアクションをインストール・プログラムに指示する仕様を提供します。

機能 INF ファイルには次のセクションが含まれます。

- [Header]
- [Registry]
- [INI]
- [FileSets]
- [Shortcut]
- [ThirdPartyApps]
- [ODBCDataSources]

機能 INF ファイルの例

次に、各機能コンポーネントの仕様が各セクションに含まれている標準的な機能 INF ファイルを示します。

[Header]

[Header(見出し)]セクションには、機能に関する一般情報が含まれており、機能のインストール・オプションを指定します。

項目	目的
Feature=	Feature –機能の名前
FeatureType=	FeatureType –機能のタイプ
Description=	Description –機能のテキスト記述
Required=	Required –機能のインストールが必須かどうかを示す設定
InitialChoice=	InitialChoice –ユーザーがインストールする機能に対するデフォルトの選択を指定する設定

[Required(必須)]エン트리と[InitialChoice]エン트리は、〈Feature Information〉フォーム上の[Feature Installation(機能インストール)]オプションの設定(Required(必須)、Selected(選択済み)、Deselected(選択解除))によって設定されます。これら3つのオプションのいずれかを選択すると、ERP 9.0 は、次の値を機能 inf ファイルの[Required]エン트리と[InitialChoice]エン通りに自動的に書き込みます。

機能インストール・オプション	必須	InitialChoice
Required(必須)	Y	Both
Selected(選択済み)	N	Both
Deselected(選択解除)	N	Custom

[Registry]

このセクションには、機能が Windows レジストリに与える影響に関する情報が記述されます。

項 目

Registry[no.]=Root[value],Key,[prefix]Name,[prefix]Value

目 的

機能の Windows レジストリ設定に関する次の仕様を示します。

Root -レジストリ内のルートを決める値で記述します。

- 「0」はルートを意味します。
- 「1」は現在のユーザーを意味します。
- 「2」はローカル・マシン・ロケーションを意味します。
- 「3」はユーザーを意味します。

Key -レジストリ値のキーを示します。

Name -レジストリ値の名前。名前のプレフィックスの意味は次のとおりです。

- 「+」は、機能のインストール時に名前が(まだ存在していなければ)作成されることを意味します。
- 「-」は、機能のアンインストール時にすべてのサブ・キーと共に名前が削除されることを意味します。
- 「*」は、機能のインストール時に名前が(まだ存在していなければ)作成され、機能のアンインストール時にすべてのサブ・キーと共に削除されることを意味します。

Value -レジストリ値の名前。値のプレフィックスの意味は次のとおりです。

- 「#x」は、値が16進数で記憶されることを意味します。
- 「#%」は、値が拡張可能な文字列として記憶されることを意味します。
- 「#」は、値が整数として記憶されることを意味します。
- 「#\$」は、値が文字列として記憶されることを意味します。

[INI]

このセクションには、機能が JDE.INI ファイルに与える影響に関する情報が記述されます。

項目	目的
Ini[no.]=FileName,Directory,Section,Key,Value,Action[value]	宛先 INI ファイルに関する次の情報を示します。 FileName -宛先 INI ファイルの名前 Directory -宛先 INI ファイルのロケーション Section -宛先ファイル内のセクションの名前 Key - 宛先ファイルのセクション内のキーの名前 Value -宛先ファイルのキーに書き込まれる値 Action -INI エントリに関して行う操作 <ul style="list-style-type: none">「0」は、INI エントリの作成を意味します。「1」は、まだ存在していない場合のみ、INI エントリを作成することを意味します。「3」は、INI エントリを作成するか、または既存エントリに追加することを意味します。

[FileSets]

このセクションには、機能が正しく動作するためにインストールする必要のあるその他のファイルに関する情報が記述されます。

項目	目的
Fileset[no.]=Compression, SourceDirectory, FileName, TargetDirectory	ファイルセットに関する次のような仕様を示します。 <ul style="list-style-type: none">Compression -ファイルセットを圧縮するかどうかを示すオプションSource Directory -ファイルセットのソース・ロケーションFileName - ファイルセットの CAB ファイルの名前Target Directory -ファイル・セットを格納するターゲット・ロケーション

[Shortcut]

このセクションには、機能インストールの一環として、Windows デスクトップに表示されるショートカットに関する情報が記述されます。

項目	目的
Shortcut[no.]=Directory,Name,Target,Arguments,Description,HotKey,Icon,IconIndex,ShowCmd[value],WkDir	ショートカットに関する次のような仕様を示します。 Directory -ショートカットが作成されるディレクトリ Name -ショートカットのリンク・ファイルの名前

Target -ショートカットの実行可能ファイルの名前

Arguments -ショートカットのコマンド行引数

Description -ショートカットの説明

HotKey -ショートカットを起動するホット・キー

Icon -ショートカット・アイコンおよびロケーション

IconIndex -アイコンがイメージ・リスト内にある場合のアイコンのインデックス

ShowCmd -アプリケーション・ウィンドウ用のコマンド。次の値をオプションとして使用できます。

- 0 は、標準サイズのウィンドウを表示することを意味します。
- 3 は、最大化したウィンドウを表示することを意味します。
- 7 は、最小化した、非アクティブなウィンドウを表示することを意味します。

WkDir -ショートカットの作業ディレクトリ

[ThirdPartyApps]

このセクションには、機能と共にインストールするサードパーティ製品に関する情報が記述されます。

項目

ThirdPartyApp[no.] = Source Directory, Description, Synchronous/Asynchronous, FileName

目的

サードパーティ・アプリケーションに関する次のような仕様を示します。

Source Directory -サードパーティ・アプリケーション実行用の実行可能ファイルのソース・ロケーション

Description -サードパーティ・アプリケーションの説明

Synchronous/Asynchronous -サードパーティ・アプリケーションのインストールを並行して行えるか(同期)、または順次行う必要があるか(非同期)を示すオプション

FileName -サードパーティ・アプリケーションを起動するファイルの名前

[ODBCDataSources]

このセクションには、機能と共にインストールする ODBC データ・ソースに関する情報が記述されます。

機能 INF ファイルには、ODBC データ・ソースに関するセクションが 2 つあります。一方のセクションにはヘッダー情報が記述され、もう一方のセクションには詳細情報が記述されます。feature.inf には 1 つのヘッダー・セクションがあり、機能に含まれているすべてのデータ・ソース・コンポーネントが一覧表示されます。ヘッダーに一覧表示されたデータ・ソースには、それぞれに対応する詳細セクションがあります。次の表には、ヘッダー・セクションのみを示します。詳細セクションについては、選択した ODBC ドライバのドキュメンテーションを参照してください。

項目

DataSourceName=DataSourceDriver

目的

ODBC データ・ソースに関する次の仕様を示します。

DataSource Name -ODBC データ・ソースの名前

DataSource Driver -データ・ソースで使用されているドライバ

配布

パッケージを組立て、定義し、ビルドを完了した後、パッケージは、いくつかの方法でワークステーションやサーバーに配布できます。

パッケージ・デプロイメントの理解

パッケージをビルドした後、いくつかの方法から選択して、パッケージを社内全体のワークステーションとサーバーに配布できます。ワークステーションに配布する場合、ERP 9.0 がワークステーションにインストールされているかどうかによって、選択方法が異なります。

ERP 9.0 がインストールされていないワークステーションへの配布

ERP 9.0 がワークステーションにまだインストールされていない場合、〈Workstation Installation (ワークステーション・インストール)〉プログラムを使用してパッケージを配布できます。〈Workstation Installation〉プログラムは、フル・パッケージおよび部分パッケージを配布する場合にのみ使用します。このプログラムは、ERP 9.0 がインストールされていないワークステーションへのアップデート・パッケージの配布には使用できません。

〈Workstation Installation〉は、パッケージに指定された項目を取り込みます。パッケージは、指示書付きの部品表のようなもので、〈Workstation Installation〉プログラムがローカル・ワークステーションに配布する必要のあるすべてのコンポーネントを取り込む場所を示します。このプログラムは対話型モードで(ワークステーションのユーザーによって起動されます)、または[Silent (自動)]モードで実行できます。

プッシュ・インストール機能を使用する場合、〈Package Deployment (パッケージ・デプロイメント)〉を使用して、パッケージを配布できます。プッシュ・インストールを使用すると、ユーザーの操作なしでシステム管理者がデプロイメント・サーバーからワークステーションへのパッケージのインストールを実行できます。この機能を使用するには、プッシュ・インストール用〈Listener (リスナー)〉アプリケーションをワークステーションにインストールし、〈Machine Identification (マシン ID)〉プログラム(P9654A)を使用して、マシンを定義する必要があります。

参照

- 〈Workstation Installation〉アプリケーションについては『インストール・ガイド』

ERP 9.0 がインストールされているワークステーションへの配布

ERP 9.0 がインストールされているワークステーションに新規パッケージを再ロードするには、次の 2 つの方法のいずれかを使用します。

- 〈Workstation Installation〉(フルおよび部分パッケージ)
- 〈Package Deployment〉(フル、部分、アップデート・パッケージ用プログラム(P9631))

パッケージを定義およびビルドした後、個々のワークステーションまたは選択グループに配布するパッケージをスケジュールするには、〈Package Deployment〉を使用します。パッケージの受取人としてスケジュールされているユーザーは、指定の配布日に、ERP 9.0 にサインオンすると、パッケージをロードできます。

プッシュ・インストレーション機能を使用しない場合、〈Package Deployment〉では、ERP 9.0 がワークステーションにロードされていることが必要です。新規のフル・パッケージまたは部分パッケージでは、既存パッケージを置換するようにスケジュールでき、アップデート・パッケージでは、ワークステーション上の既存パッケージとマージするようにスケジュールできます。

これらの配布方法にはそれぞれ利点があります。〈Workstation Installation〉は、パッケージをスケジュールすることなく、パッケージをビルド直後にまたはビルドしてからしばらくしてインストールする場合に使用すると便利です。一方、〈Package Deployment〉は、パッケージを使用可能にする日時をコントロールする必要がある場合、パッケージのインストールを必須にする場合、またはパッケージをサーバーだけでなく、ワークステーションにも配布する場合に便利です。

サーバーへの配布

サーバーは、ワークステーション用にビルドしたパッケージを異なるフォーマットで受け取ります。パッケージを組み立てて、パッケージ・ビルド定義を作成する場合、パッケージを配布するサーバーを指定できます。パッケージを配布するには、〈Package Deployment〉アプリケーション(P9631)を使用し、パッケージをワークステーションに配布する場合と同じスケジュール方式を使用します。スケジュールは、同一のフォームで、クライアント・ワークステーションおよびサーバーへの配布を簡単に設定できます。サーバーに配布する際には、プッシュ・インストレーション機能を使用できません。

階層デプロイメント・ロケーションへの配布

複数階層デプロイメントを使用すると、複数のデプロイメント・ロケーションおよび複数のデプロイメント・マシンからソフトウェアをワークステーションにインストールできます。1 日に 50 台以上のワークステーションでインストールを行う場合、または WAN を使用したワークステーションへのインストールに非常に時間がかかる場合、この配布方法を使用します。

CD からワークステーションへの配布

システムに CD ライターがインストールされている場合、CD ライターをデプロイメント・ロケーションとして定義できます。これにより、CD ライターは、擬似デプロイメント・サーバーとして定義され、ERP 9.0 パッケージを空の CD にコピーできます。その後、この CD に含まれている〈Workstation Installation〉プログラムを使用して、ERP 9.0 をワークステーションにインストールできます。

マシン、ロケーション、およびデプロイメント・グループの定義

パッケージを配布する前に、配布先のワークステーション、サーバー、グループまたはロケーションを特定する必要があります。これらを特定すると、〈Package Deployment Director (パッケージ・デプロイメント・ディレクタ)〉を使用してパッケージをスケジュールする準備が整った時に、パッケージを受け取るマシン、グループまたはロケーションをパッケージ受信者として使用可能になります。

デプロイメント・グループは、ジョブ機能、チーム、またはその他の指定グループなどの基準によって分類されたワークステーションのグループです。たとえば、ソフトウェア開発グループ、テスト・グループ、本稼働用グループなどがあります。〈Machine Group Identification (マシン・グループ ID)〉プログラム(P9652A)を使用すると、複数のワークステーションを含むグループを定義または改訂することができます。

ロケーションは、物理ロケーションに対応するワークステーションおよびサーバーのグループです。たとえば、法人および事業所のロケーション、または建物 5 および建物 7 などのロケーションを指定

できます。ロケーションは、複数階層デプロイメントを使用する場合、または広域ネットワーク(WAN)を使用して配布する場合にも便利です。そのような場合には、地理的ロケーションのそれぞれに対してロケーションを定義することもできます。〈Machine Identification〉プログラム(P9654A)を使用すると、社内のマシンとロケーションを定義、改訂できます。

これらのアプリケーションを使用すると、パッケージを複数のユーザーに配布する必要がある場合のデプロイメント・プロセスを簡易化できます。また、各ワークステーションまたはサーバーへの配布をスケジュールするのではなく、ロケーションに応じた配布をスケジュールすることができます。

マシン定義を入力する時は、実際には、ERP 9.0 構成におけるマシンの用途を定義します。たとえば、デプロイメント・サーバーをデータ・サーバーとして使用することができます。マシン定義を入力する際は、次の推奨事項を考慮してください。

- JAS (Java アプリケーション・サーバー) は、データ・サーバーやエンタープライズ・サーバーなどではなく、Java アプリケーション・サーバーとしてのみ定義できます。
- デプロイメント・サーバーをワークステーションとして使用できません。
- デプロイメント・サーバーはデータ・サーバーとして使用できます。
- 調整およびパフォーマンスの理由で、デプロイメント・サーバーをエンタープライズ・サーバーとして使用できません。

マシンの定義

〈Package Deployment Director〉を使用して各クライアント・ワークステーションにパッケージを配布する前に、パッケージを受け取る各マシン上のマシン・マスター・テーブル(F9650)にレコードがあることを確認してください。

この F9650 テーブルにレコードを入力するには、次のいずれかの方法を選択します。

- 手入力。ERP 9.0 にサインオンするために使用されたことがないマシンでは、〈Machine Identification〉アプリケーション(P9654A)を使用して、F9650 テーブルにレコードを手入力します。
- 自動。ユーザーが新規マシンで ERP 9.0 にはじめてサインオンすると、ERP 9.0 は、F9650 テーブルでレコードを自動的に作成します (また、ユーザーがワークステーションにサインオンするたびに、F9650 テーブルに存在するレコードは自動的に更新されます)。

F9650 テーブルにマシンを登録する最も簡単な方法は、新規マシンのユーザー全員に ERP 9.0 にサインオンさせることです。パッケージを配布してからユーザーがログオンできるようにする必要があります。〈Machine Identification〉プログラムを使用すると、このタスクを実行できます。

〈Machine Identification〉プログラムを使用すると、ワークステーションを定義だけでなく、次のマシンの定義を入力または改訂することもできます。

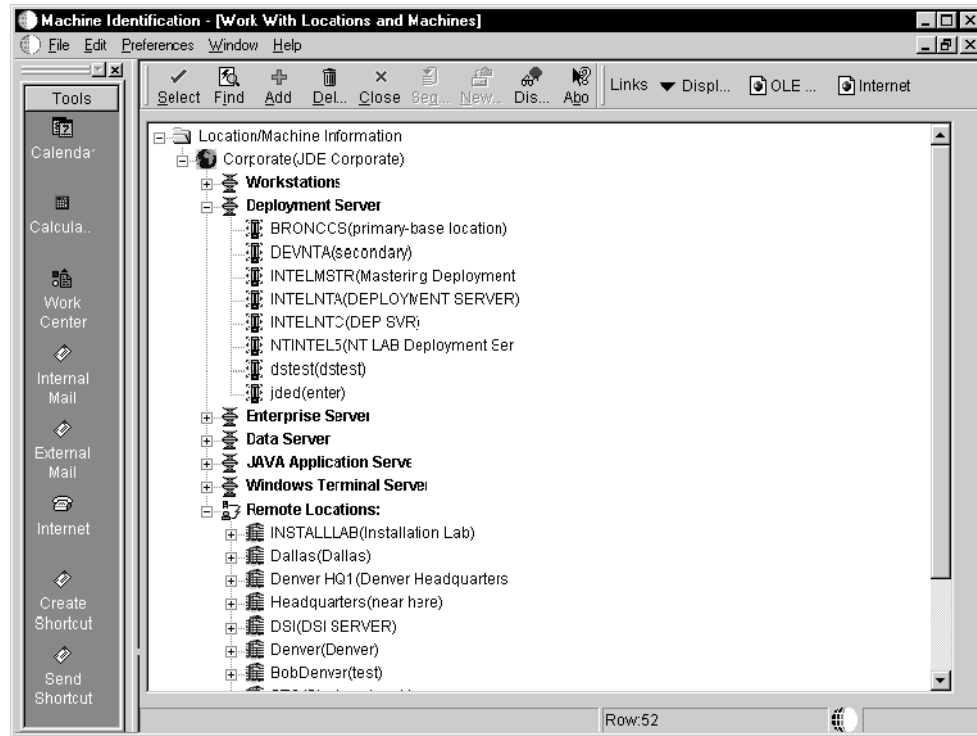
- デプロイメント・サーバー
- エンタープライズ・サーバー
- データ・サーバー
- Java アプリケーション・サーバー
- Windows ターミナル・サーバー

マシンは、ワークステーション兼データ・サーバーとして、またはデータ・サーバーのみとして定義できますが、これら以外の組合せは使用できません。たとえば、マシンをワークステーションおよびデプロイメント・サーバーとして定義することはできません。

遠隔ロケーションを含む複数のロケーションで、これらのマシンの定義を入力または改訂できます。

▶ マシンを定義するには

〈Package and Deployment Tools〉メニュー(GH9083)から〈Machine Identification〉を選択します。



1. 〈Work With Locations and Machines(ロケーション/マシンの処理)〉で[Find]をクリックします。
2. 該当するロケーションの下で、追加するマシンのタイプとして次のいずれかを選びます。
 - ワークステーション
 - デプロイメント・サーバー
 - エンタープライズ・サーバー
 - データ・サーバー
 - Java アプリケーション・サーバー
 - Windows ターミナル・サーバー
3. 新規マシンを追加するには、[Add]をクリックします。

〈Machine Revisions(マシンの改訂)〉フォームが表示されます。このフォームには、追加するマシン・タイプによって異なるフィールドが表示されます。

The screenshot shows a software window titled "Machine Identification - [Enterprise Server Revisions]". It has a menu bar (File, Edit, Preferences, Form, Window, Help) and a toolbar with buttons like OK, Cancel, Dismiss, and Abort. A left sidebar contains icons for Tools, Form, Environ..., Generate Scripts, Generate Svr Map, OCM, Path Code, Data, Foundat..., and Help. The main area has tabs for Workstation, Deployment, Enterprise (selected), Data, JAS, and WTS. The Enterprise tab contains several input fields: Machine Usage (value: 20), Enterprise Server, Location (Corporate), Machine Name, Primary User (JDE), Description, Release (B7333), Host Type, Port Number (6005), Logical Machine Name, Database Type, Server Map Data Source, Installation Path, and Deployment Server Name (BRONCOS).

4. 〈Machine Revisions〉フォームで、次のフィールドに値を入力します。

- マシ名

マシン名では大文字小文字が区別されます。矛盾した命名法によって問題が生じないように、マシン名はすべて大文字で定義してください。

- 記述

- リリース No.

- ホスト・タイプ

たとえば、AS/400、HP9000、Intel NT など。

- 基本ユーザー

マシンの基本ユーザーの名前。

定義するマシン・タイプに対応するタブが自動的に表示されます。

5. [Workstation(ワークステーション)]タブで、定義するマシンの次のフィールドに値を入力します。

- デプロイメント・サーバー名

6. [Deployment [Server](デプロイメント[サーバー])]タブで、次のフィールドに値を入力します。

- 1 次デプロイメント・サーバー

基本デプロイメント・サーバーをすでにセットアップしている場合、新規デプロイメント・サーバーを定義する時に、[Primary Deployment Server(基本デプロイメント・サーバー)]フィールドにアクセスできません。このフィールドの値は、基本デプロイメント・サーバー定義を改訂する時、または基本デプロイメント・サーバーを補助サーバーに変更する時にしか変更できません。このような場合、異なるサーバーを基本デプロイメント・サーバーとして指定できます。

- サーバー共有パス

補助デプロイメント・サーバーを定義する時は、[Form(フォーム)]メニューのオプションを使用して、パス・コード、データ項目、ファンデーション・モジュールおよびヘルプ項目を選択できます(これらのオプションは基本デプロイメント・サーバー用には使用できません)。

7. [Enterprise [Server](エンタープライズ[サーバー])]タブで、次のフィールドに値を入力します。

- ポート No.
- 論理マシン名
- データベース・タイプ
- サーバー・マップ・データ・ソース
- インストール・パス
- デプロイメント・サーバー名

エンタープライズ・サーバーを定義する時は、[Form]メニューのオプションを使用して、インストール・スクリプトを生成し、サーバー・マップ・テーブルを生成して自動入力し、さらにOCM(オブジェクト構成マネージャ)を使用して、ワークステーション・サーバー・マップを更新することができます。エンタープライズ・サーバーでは、マシン環境および記述を表示することもできます。

フィールドによっては、すでに定義されており、変更できない場合があります。

8. [Data [Server](データ[サーバー])]タブで、次のフィールドに値を入力します。

- データ・ソース・タイプ

9. [JAS [Server](JAS[サーバー])]タブで、次のフィールドに値を入力します。

- インストール・パス

10. [WTS(Windows ターミナル・サーバー)]タブで、次のフィールドに値を入力します。

- インストール・パス

11. [Form]メニューから、[Environment(環境)]を選択します。

12. <Machine Environment Revisions(マシン環境の改訂)>で、[Environment]カラムをクリックし、新しいマシンに追加する1つまたは複数の環境を選択します。

13. [OK]をクリックします。

14. 〈Machine Revisions〉で、もう一度[OK]をクリックし、〈Work with Locations and Machines〉フォームへ戻ります。

15. 〈Work with Locations and Machines〉で、[Close]をクリックして終了します。

既存マシン定義の改訂プロセスは、新規定義の追加プロセスに似ています。〈Work With Locations and Machines〉フォームで[Add]をクリックする代わりに、改訂する定義が含まれたマシンを検索して選択し、[Select(選択)]をクリックします。同じ改訂フォームが表示され、マシン定義の変更が可能になります。

フィールド記述

記述	用語解説
マシン用途	このマシンの使用目的。データベースサーバー、ロジックサーバー、TSE サーバーなどがあります。
マシン名	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。
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。
ホスト・タイプ	ホストマシン・タイプ
ロケーション	デプロイメント・ロケーションの名前
基本ユーザー	リスト済みマシンの基本ユーザー
サーバー・マップ・データ・ソース	データ・ソースの名前。
デプロイメント・サーバー名	デプロイメントに使用する特定のサーバー名
1 次デプロイメント・サーバー	デプロイメントサーバーが特定のディレクトリに対してプライマリであるかどうかを指定します。
データベース・タイプ	データベースのタイプ
論理マシン名	このマシンまたはポートに割り当てられた論理マシン名。マシンはワークステーションを表すこともあります。一台のマシンで ONEWORLD のセッションが複数実行されることがあるので、固有のマシン名およびこのセッションを実行しているポートを識別する論理マシン名を割り当てる必要があります。J.D. EDWARDS では、論理マシン名をリリースおよびマシンの目的(財務データ・サーバー-B73.3、流通ロジックサーバー-B73.3 など)を示すように設定することをお勧めします。
パッケージ日付/時刻	フル・パッケージが特定のワークステーションに展開された日付と時刻

パッケージ名	<p>ワークステーションに展開するコンポーネントがサーバーのどこにあるかを説明します。パッケージには 3 つのタイプがあります。</p> <p>フル・パッケージ: システム・アプリケーション (すべてのスペック) を含みます。</p> <p>部分パッケージ: システムの最小構成。すべてのアプリケーションを初期インストールするのではなく、選択したアプリケーションのみをロードすることができます。</p> <p>更新: このパッケージ・タイプに含まれるシステム・オブジェクトは、ワークステーションがパッケージを受け取り、ユーザーがシステムにサインオンした後にロードされます。該当するスペックのないオブジェクトが更新パッケージに含まれる場合、アプリケーションの古いバージョンはワークステーションから削除され、ユーザーが次にそのアプリケーションにアクセスする時に現バージョンに置き換えられます。更新パッケージは、常にシステム管理者が指定した日時に展開されます。</p>
システム更新日付/時刻	指定されたワークステーションに現行システムが展開された日付と時刻
パッケージ更新日付/時刻	指定されたワークステーションに最新の更新パッケージがインストールされた日付と時刻
パッケージ更新名	指定されたワークステーションに展開された最新の更新パッケージ名
ポート No.	与えられた ONEWORLD のポートを識別します。ワークステーションが接続するポートは JDE.INI ファイルにより制御されるため、ワークステーションではこのポート番号は参照用にのみ使用されます。
サーバー共有パス	<p>World の場合: サーバー共有パス・フィールドは、環境が現サーバーのロケーションを決定するのに使用されます。</p> <p>J.D. Edwards ERP の場合: このフィールドはこのパス・コードへの共有ディレクトリを示します。</p> <p>ファイルサーバー上に保管されているオブジェクトは、このパス上にあります。</p>

ロケーションの定義

1 つの会社の所在地が複数の建物、都市、国に分散している場合がしばしばあります。このような場合、パッケージをワークステーションとサーバーに個別に配布するのではなく、1 つのロケーションに配布できます。続いて、各ロケーションにあるセカンダリ・デプロイメント・サーバーがパッケージをそのロケーションのワークステーションおよびサーバーに配布します。

企業規模が大きいほど、ロケーションへの配布のメリットが大きくなります。複数階層デプロイメントを使用してパッケージを遠隔ロケーションに配布する場合、ロケーションの概念が重要になります。

「ロケーション」とは、基本的にマシン、データベース、および環境についてのユーザー定義グループを意味しますが、本店から地理的に離れている遠隔の事業所が存在する場合などは、WAN (広域ネットワーク) で接続された実際の物理ロケーションを示すこともあります。たとえば、事業所の 1 フロア、企業構内の個々の建物、各都市の支店、または別の都市にある施設をロケーションという場合もあります。

新規ロケーションを作成した場合、そのロケーションに関連付けられたマシン名を定義することで、そのロケーションのワークステーションとサーバーを追加できます。

〈Machine Identification〉プログラム(P9654A)の起動時に表示される最上位のロケーションが、基本ロケーションです。この基本ロケーションを変更したり、削除することはできませんが、その下位にあるロケーションを作成/改訂することはできます。

あるロケーションの下位ロケーションを作成すると、元のロケーションは親ロケーションになり、下位ロケーションは子ロケーションになります。たとえば、Tokyo というロケーションを作成し、その下に Osaka というロケーションを作成した場合、Tokyo は親ロケーション、Osaka は子ロケーションになります。

参照

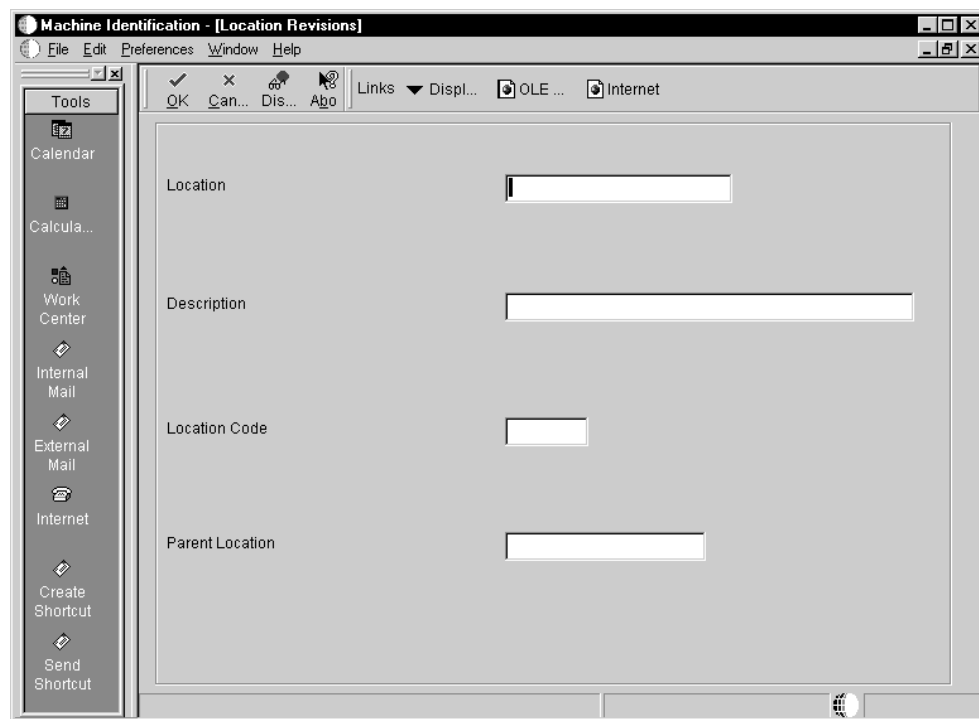
- 遠隔地へのパッケージの配布については、「複数階層デプロイメント」

▶ ロケーションを定義するには

〈Package and Deployment Tools〉メニュー(GH9083)から〈Machine Identification〉を選択します。

〈Work With Locations and Machines〉フォームが表示されます。新規のロケーションを下位に配置する現行ロケーションを選択します。たとえば、新しい遠隔ロケーションを追加する場合、[Remote Locations(遠隔ロケーション)]アイコンをクリックします。

1. 新規ロケーションを追加するには、[Add]をクリックします。



2. 〈Location Revisions(ロケーションの改訂)〉フォームで、次のフィールドに値を入力します。

- ロケーション

- 記述
- ロケーション・コード
- 親ロケーション

3. ロケーションを選択したら、[Close]をクリックします。
4. 〈Location Revisions〉で[OK]をクリックします。
5. 〈Work with Locations and Machines〉上で[Close]をクリックします。

既存ロケーションの改訂プロセスは、新規ロケーションの追加プロセスに似ています。〈Work with Locations and Machines〉フォームで[Add]をクリックする代わりに、改訂するロケーションを検索して選び、[Select]をクリックします。すると、同じ改訂フォームが表示され、ロケーションを変更することができます。

フィールド記述

記述	用語解説
ロケーション	デプロイメント・ロケーションの名前
ロケーション・コード	システムを展開する現在のディレクトリを表すフィールド。
親ロケーション	親ロケーション名

パッケージ・デプロイメント・グループの定義

部署、チームまたは機能に基づくデプロイメント・グループを作成できます。たとえば、管理グループ、テスト・グループ、本稼働用グループなどがあります。

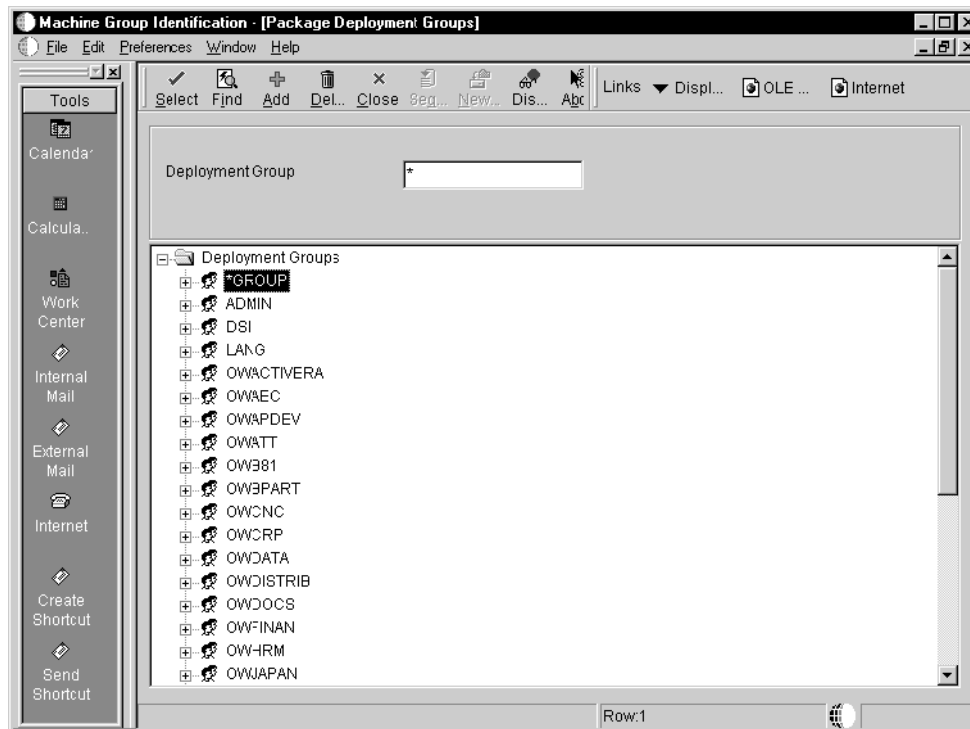
規模が大きな企業で、ワークステーションへのパッケージ・デプロイメントを個別にスケジュールすると時間がかかる場合、パッケージ・デプロイメント・グループを使用すると便利です。このような環境では、デプロイメント・グループを使用すると、よりスピーディにパッケージを配布することができます。

グループには、サブグループ(グループ内のグループ)を含めることができます。たとえば、大きな Development(開発)グループに Quality Assurance(品質保証)というグループを含めることもできます。

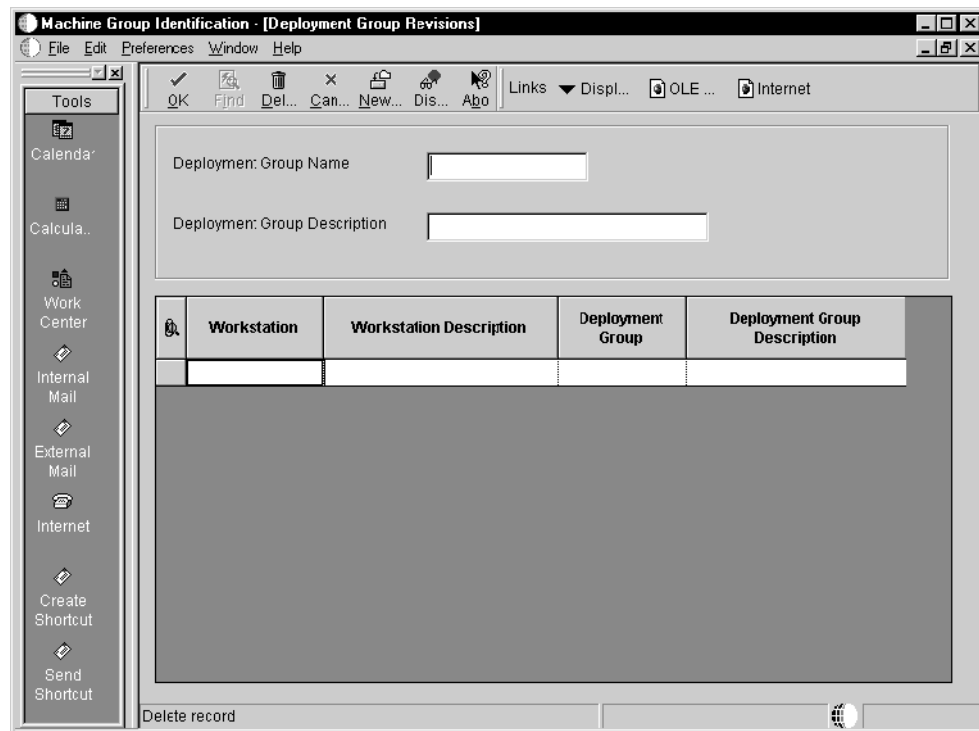
開発グループにわかりやすい名前を設定すると、パッケージをビルドし、スケジュールするのに便利です。たとえば、テストを担当する品質保証スタッフを含むグループには「Testing(テスト)」という名前を付けます。

▶ パッケージ・デプロイメント・グループを定義するには

〈Package and Deployment Tools〉メニュー(GH9083)から〈Machine Group Identification〉を選択します。



1. 〈Package Deployment Groups (パッケージ・デプロイメント・グループ)〉フォームが表示されます。新規デプロイメント・グループを追加するには、[Add]をクリックします。



2. デプロイメント・グループ名および記述を入力します。
3. グループに組み込むワークステーションまたはデプロイメント・グループを入力します。
ビジュアル・アシスト機能を使用すると、有効なワークステーションとデプロイメント・グループの名前を検索できます。
4. ワークステーションまたはデプロイメント・グループを追加したら、[OK]をクリックします。
〈Package Deployment Director〉フォームが表示されます。
5. [Close]をクリックして、〈Package Deployment Groups〉フォームを終了します。

フィールド記述

記述

デプロイメント・グループ名

用語解説

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。

デプロイメントグループ記述

選択したデプロイメントグループ

ワークステーション	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。
ワークステーション記述	ユーザー定義名称または備考。
デプロイメントグループ	親グループの一部として定義されたグループ

デプロイメント・グループの改訂

次に、既存パッケージ・デプロイメント・グループを改訂する方法について説明します。

▶ パッケージ・デプロイメント・グループを改訂するには

〈Package and Deployment Tools〉メニュー(GH9083)から〈Machine Group Identification〉を選択します。

1. 〈Package Deployment Groups〉フォームが表示されます。必要なグループを選択し、[Select]をクリックします。

既存グループを改訂する時は、グループ名は変更できませんが、記述は変更できます。

2. 〈Deployment Group Revisions(デプロイメント・グループの改訂)〉フォームで、グループを追加するには、最後のロー(空のロー)を選択し、メンバを追加するワークステーションまたはデプロイメント・グループの名前を入力します。

この作業は、[Workstation]または[Deployment Group(デプロイメント・グループ)]フィールドに名前を手入力するか、またはこれらのフィールドで[Search(検索)]ボタンを使用して行えます。

[Workstation]フィールドで[Search]ボタンを使用すると、〈Machine Select(マシンの選択)〉フォームが表示されます。[Deployment Group]フィールドで[Search]ボタンを使用すると、〈Deployment Group Search(デプロイメント・グループの検索)〉フォームが表示されます。

3. ワークステーションまたはデプロイメント・グループを追加または改訂したら、[OK]をクリックします。

〈Package Deployment Groups〉フォームが再表示されます。

4. [Close]をクリックして、〈Package Deployment Groups〉フォームを終了します。

Package Deployment Director の処理

パッケージを定義してビルドした後は、〈Package Deployment〉プログラム(P9631)を使用して、個々のワークステーション、デプロイメント・サーバーまたはエンタープライズ・サーバーへのパッケージ・デプロイメントをスケジュールします。パッケージの受取人としてスケジュールされているユーザーは、指定の配布日に ERP 9.0 にサインオンすると、パッケージをロードできます。

また、特定マシンの代わりに、デプロイメント・グループまたはロケーションに対してパッケージをスケジュールすることができます。通常、多くのワークステーションおよびサーバーにパッケージを配布する大企業にとって、デプロイメント・グループは便利です。

Package Deployment Director の理解

〈Package Deployment〉プログラム(P9631)は、ビルドされたパッケージをワークステーションとサーバーへの配布をスケジュールし、実際に配布するプロセスを簡素化し、迅速に処理します。このディレクタは一連のフォームを表示し、それらによって、配布するパッケージ、配布先、および配布時刻を指定できます。

配布するパッケージを指定した後に、次のいずれかの配布先を指定します。

- クライアント・ワークステーション
- エンタープライズ・サーバー
- デプロイメント・サーバーまたはデプロイメント・グループ
- ロケーション

パッケージを特定のワークステーションおよびサーバーに配布したり、あるいは、デプロイメント・グループまたはロケーションに基づいてデプロイメントをスケジュールすることができます。これら 2 つの方法は併用できず、いずれか一方を選択する必要があります。

パッケージは、それをインストールしないと、ERP 9.0 にアクセスできないという必須のパッケージにすることができます。パッケージが任意である場合、パッケージをインストールするか、または拒否するまで、ERP 9.0 にサインオンするたびにパッケージのインストール・オプションが表示されます。

また、「プッシュ・インストール」を指定することもできます。この場合、ユーザーとの対話を求めずに、パッケージをデプロイメント・サーバーから、指定したワークステーションに配布できます。

〈Package Deployment Director〉では、プッシュ・インストールを使用する場合を除き、ERP 9.0 をワークステーションにロードしておく必要があります。新規のフル・パッケージまたは部分パッケージでは、既存パッケージを置換するようにスケジュールでき、アップデート・パッケージでは、ワークステーション上の既存パッケージとマージするようにスケジュールできます。

〈Package Deployment Director〉は次のテーブルを使用します。

- マシン・マスター(F9650)
- マシン詳細(F9651)
- デプロイメント・グループ・ヘッダー(F9652)
- デプロイメント・グループ詳細定義(F9653)
- デプロイメント・ロケーション定義(F9654)
- パッケージ・デプロイメント・スケジュール(F98825)
- サーバーへのパッケージ・デプロイメント情報(F98826)
- ソフトウェア・パッケージ・ヘッダー(F9603)

次の表に、〈Package Deployment Director〉の各フォームの機能を示します。

〈Package Deployment Director(パッケージ・デプロイメント・ディレクタ)〉フォーム	〈Package Deployment Director〉の記述を表示します。
〈Package Selection(パッケージの選択)〉フォーム	配布するパッケージを検索および選択するために使用します。
〈Package Deployment Targets(パッケージ・デプロイメント・ターゲット)〉フォーム	パッケージの配布先を指定するために、このフォームを使用します。 各クライアント・ワークステーション、デプロイメント・サーバーおよびエンタープライズ・サーバーを選択できます。また、パッケージをデプロイメント・グループまたはロケーションに配布できます。
〈Package Deployment Attributes(パッケージ・デプロイメント属性)〉フォーム	このフォームでは、パッケージを配布する日時を入力します。また、パッケージを必須にする(つまり、すべてのパッケージ受信者によってインストールされる必要がある)かどうか、およびパッケージを配布するために〈Push Installation(プッシュ・インストール)〉を使用するかどうかを指定します。
〈Deployment Client Workstation Selection(デプロイメント・クライアント・ワークステーションの選択)〉フォーム	パッケージを受け取る各クライアント・ワークステーションを選択するために使用します。
〈Deployment Server Selection(デプロイメント・サーバーの選択)〉フォーム	パッケージを受け取る各デプロイメント・サーバーを選択するために使用します。
〈Enterprise Server Selection(エンタープライズ・サーバーの選択)〉フォーム	パッケージを受け取る各エンタープライズ・サーバーを選択するために使用します。
〈Deployment Location Selection(デプロイメント・ロケーションの選択)〉フォーム	パッケージを受け取るデプロイメント・ロケーションを指定するために使用します。
〈Deployment Groups Selection(デプロイメント・グループの選択)〉フォーム	各メンバがパッケージを受け取るデプロイメント・グループを指定するために使用します。
〈Build Selection(ビルドの選択)〉フォーム	複数階層デプロイメントの場合に、宛先デプロイメント・サーバーに配布するサーバー・パッケージまたはクライアント・パッケージを指定するために使用します。
〈Work with Package Deployment(パッケージ・デプロイメントの処理)〉フォーム	〈Package Deployment Director〉で入力したロケーションとパッケージ受信者を検討し、改訂するために使用します。

参照

- プッシュ・インストールについては「プッシュ・インストールの使用」

Package Deployment Director の使用

パッケージを組み立ててビルドし、すべてのマシンを定義し、デプロイメント・グループを確認した後は、〈Package Deployment Director〉を使用してパッケージ受信者を指定し、パッケージ・デプロイメントをスケジュールする準備が整います。

デプロイメント・プロセスでは、常に前後のフォームに移動することができます。また、随時、プロセスをキャンセルすることもできます。

デプロイメント・グループまたはロケーションではなく、マシンへのパッケージ・デプロイメントをスケジュールする場合、クライアント・ワークステーション、デプロイメント・サーバー、エンタープライズ・サーバー、またはこれらの組合せに対してパッケージを配布するようにスケジュールすることができます。表示されるフォームは、選択した配布方法によって異なります。たとえば、クライアント・ワークステーションおよびデプロイメント・サーバーへのパッケージ・デプロイメントをスケジュールすることを示した場合、特定のワークステーションおよびデプロイメント・サーバーを選択するためのフォームが表示されます。クライアント・ワークステーションのみにパッケージ・デプロイメントをスケジュールする場合、サーバー選択フォームは表示されません。

〈Package Deployment Director〉にアクセスする際、〈Work with Package Deployment〉フォームでは、配布パッケージの情報をマシン、デプロイメント・グループ、ロケーションまたはパッケージ別に表示することができます。

ツリーを拡張すると、選択した表示項目に応じた情報がツリーに表示されます。次に、ツリーを拡張した場合に、各レベルで表示される情報について説明します。

- マシン

- レベル 1: クライアント・ワークステーション、デプロイメント・サーバーおよびエンタープライズ・サーバーのヘッダー

- レベル 2: 上記(レベル 1)の 3 つのヘッダーの下の特定のマシン

- レベル 3: マシンに配布される特定のパッケージ(存在する場合)

- デプロイメント・グループ

- レベル 1: 特定のグループ

- レベル 2: これらのグループのメンバ

- レベル 3: グループ・メンバに配布される特定のパッケージ

- ロケーション

- レベル 1: 特定ロケーション

- レベル 2: クライアント・ワークステーション、デプロイメント・サーバー、エンタープライズ・サーバーおよび遠隔地のヘッダー

- レベル 3: クライアント・ワークステーション、デプロイメント・サーバーおよびエンタープライズ・サーバーのヘッダーの下の特定のマシン

- 遠隔地の下のレベル 3 のみ: 定義済みの遠隔地

- レベル 4: 各マシンに配布される特定のパッケージ(存在する場合)

- 遠隔地の下のレベル 4 のみ: クライアント・ワークステーション、デプロイメント・サーバーおよびエンタープライズ・サーバーのヘッダー

- 遠隔地の下のレベル 5 のみ: クライアント・ワークステーション、デプロイメント・サーバーおよびエンタープライズ・サーバーのヘッダーの下の特定のマシン

- 遠隔地の下のレベル 6 のみ: 各マシンに配布される特定のパッケージ(存在する場合)

- パッケージ

レベル 1: パッケージ名

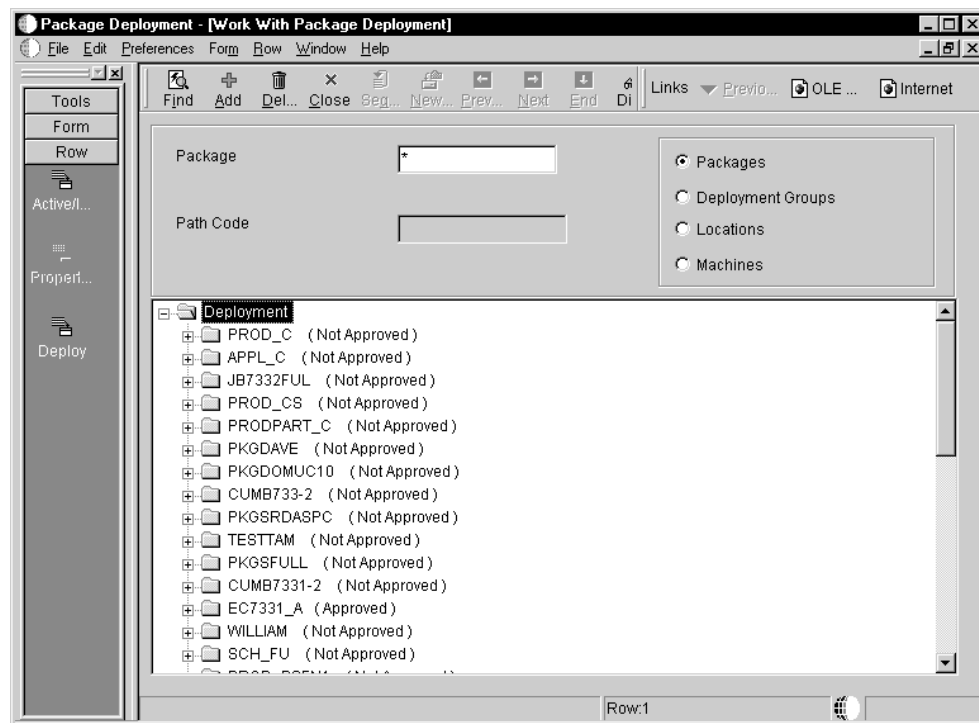
レベル 2: クライアント・ワークステーション、デプロイメント・サーバーおよびエンタープライズ・サーバーのヘッダー

レベル 3: 各ヘッダーのパッケージの配布日付

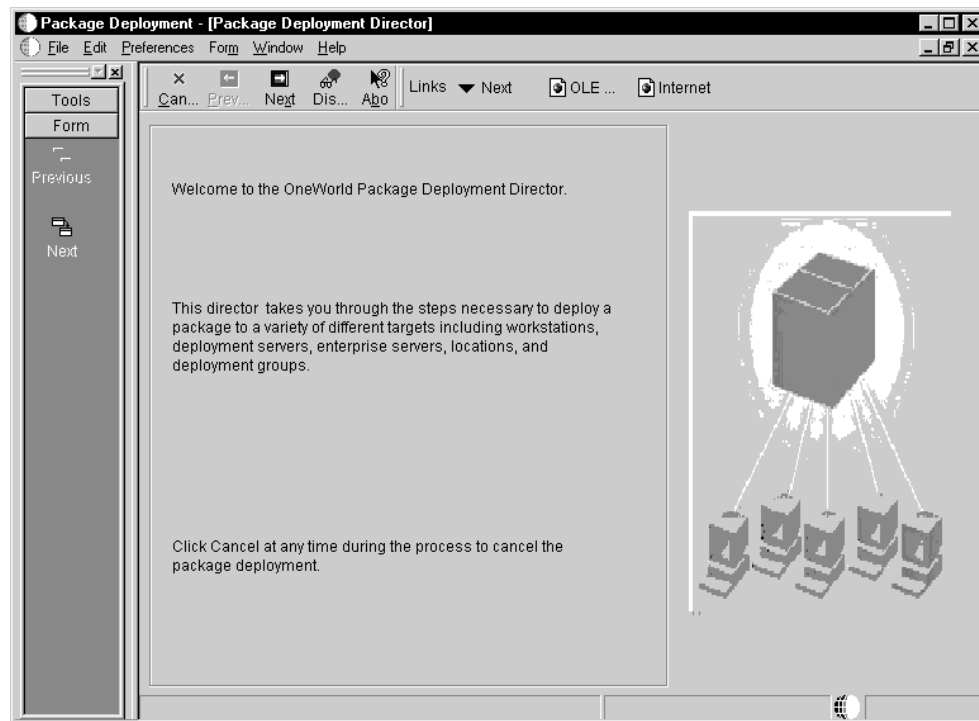
レベル 4: その日時にそのパッケージが配布された特定のマシン

▶ クライアント・ワークステーションまたはサーバーへのパッケージ・デプロイメントをスケジュールするには

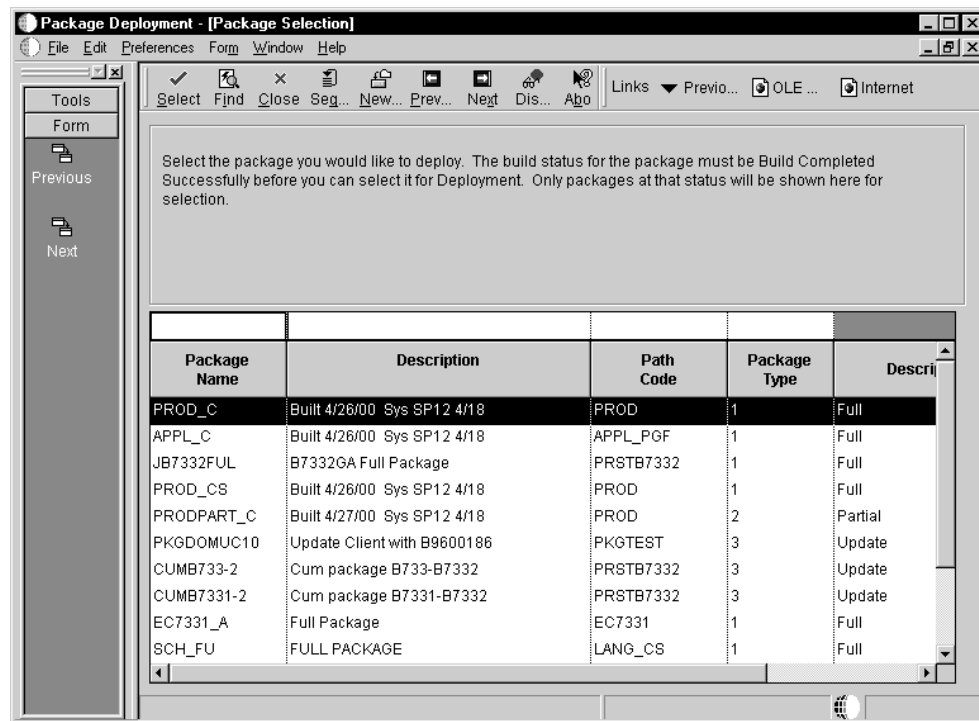
〈Package and Deployment Tools〉メニュー(GH9083)から〈Package Deployment〉を選択します。



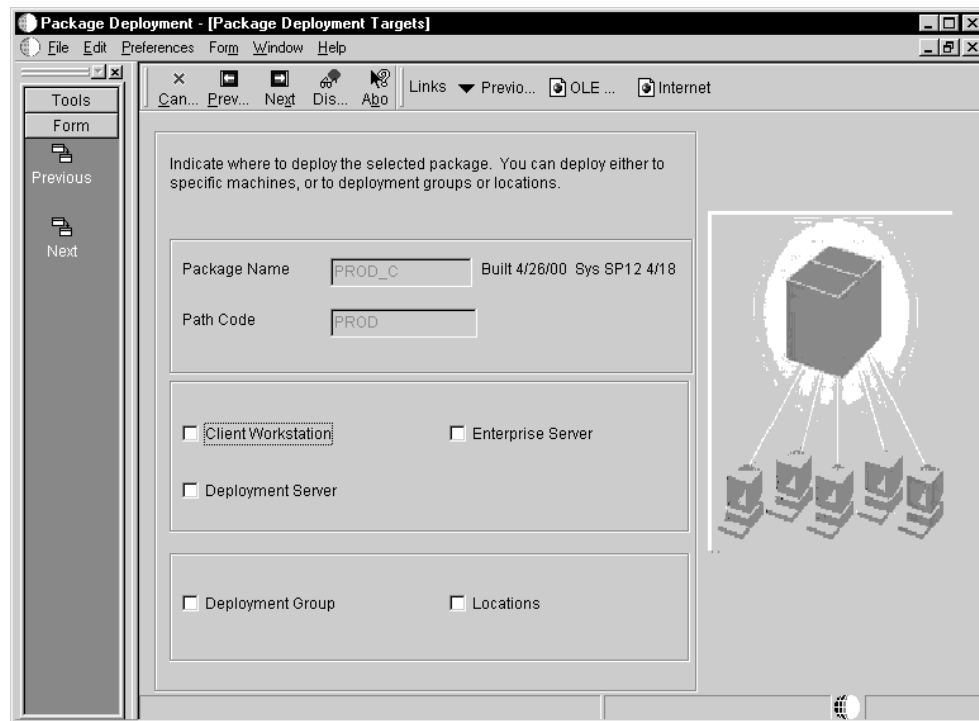
1. 〈Work with Package Deployment〉フォームが表示されます。〈Package Deployment Director〉を起動するには、[Add]をクリックします。



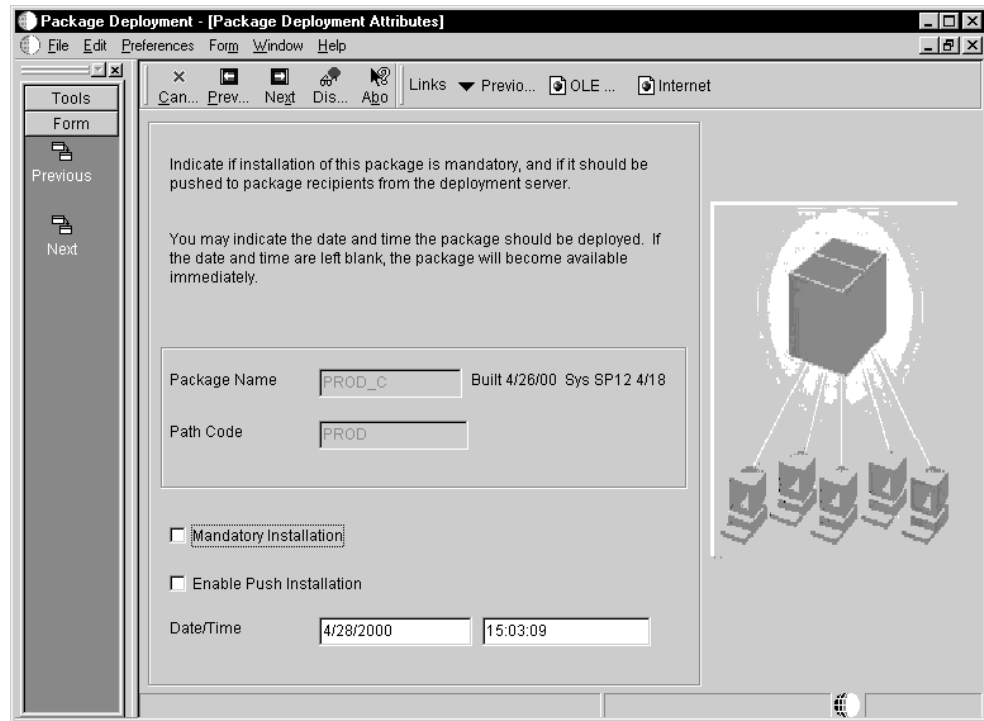
2. 〈Package Deployment Director〉フォームが表示されます。[Next]をクリックします。



3. 〈Package Selection〉フォームが表示されます。配布するパッケージを選択し、[Next]をクリックします。



4. 〈Package Deployment Targets〉フォームが表示されます。次のいずれかのオプションをクリックして、パッケージを配布するマシンのタイプを指定し、[Next]をクリックします。
- クライアント・ワークステーション
 - デプロイメント・サーバー
 - エンタープライズ・サーバー

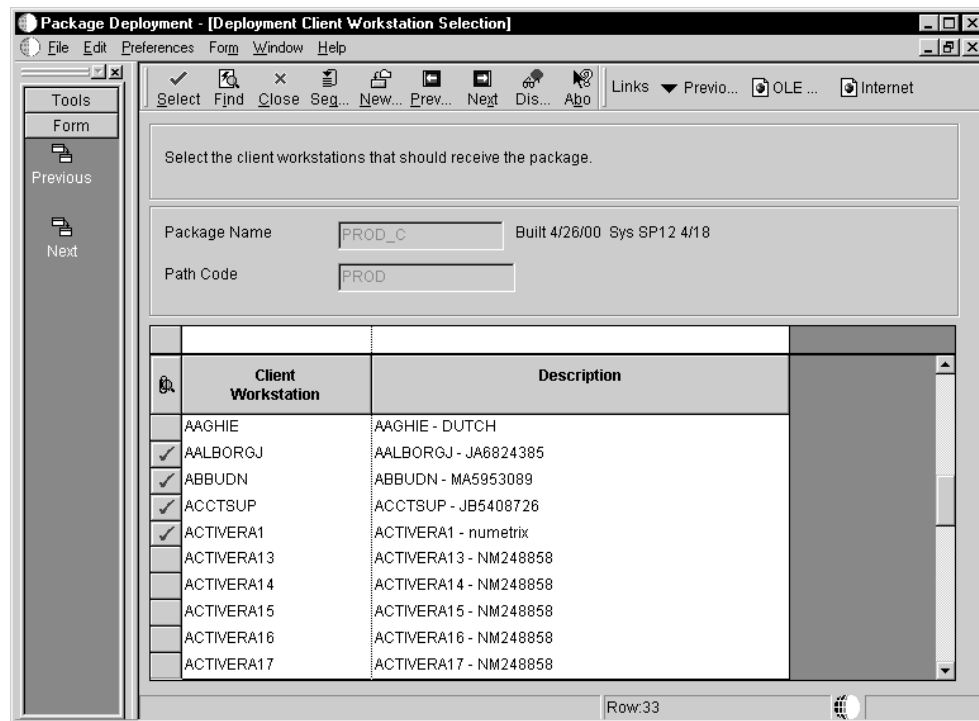


5. 〈Package Deployment Attributes〉フォームが表示されます。次のフィールドに値を入力します。
- 必須インストール
 - プッシュ・インストール
 - 日付/時刻
6. パッケージをデプロイメント・サーバーからワークステーションにプッシュするプッシュ・インストールを使用してパッケージを配布する場合、[Enable Push Installation(プッシュ・インストールの有効化)]オプションをクリックして、[Next]をクリックします。

注:

プッシュ・インストールの詳細については、「プッシュ・インストールの使用」を参照してください。

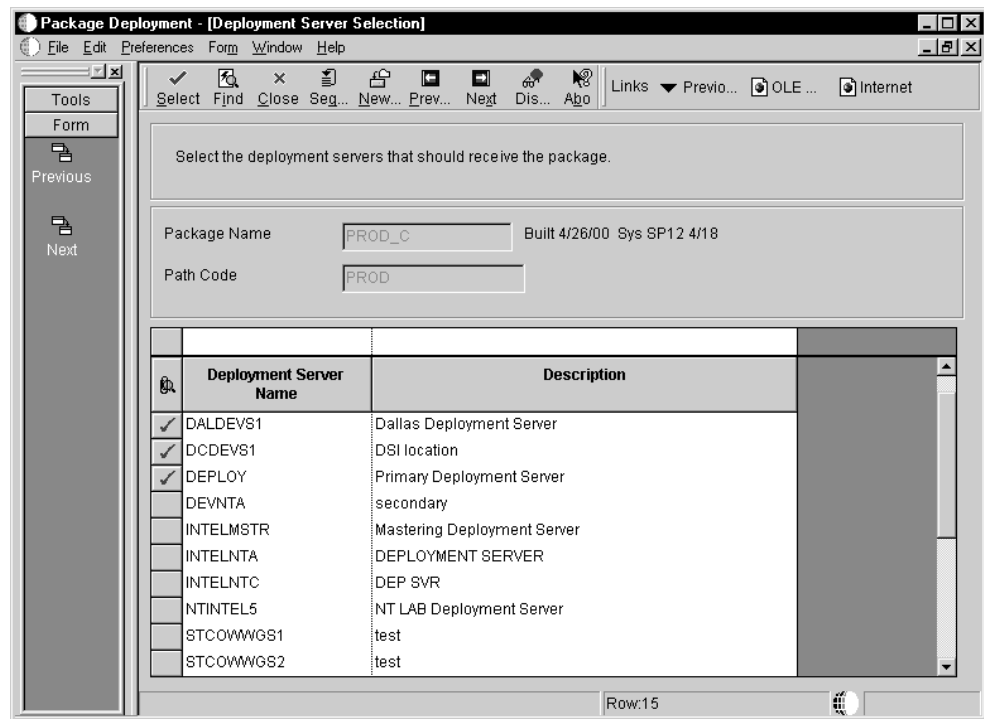
ワークステーションに配布する場合、〈Deployment Client Workstation Selection〉フォームが表示されます。パッケージをワークステーションに配布しない場合、このステップはとばしてください。



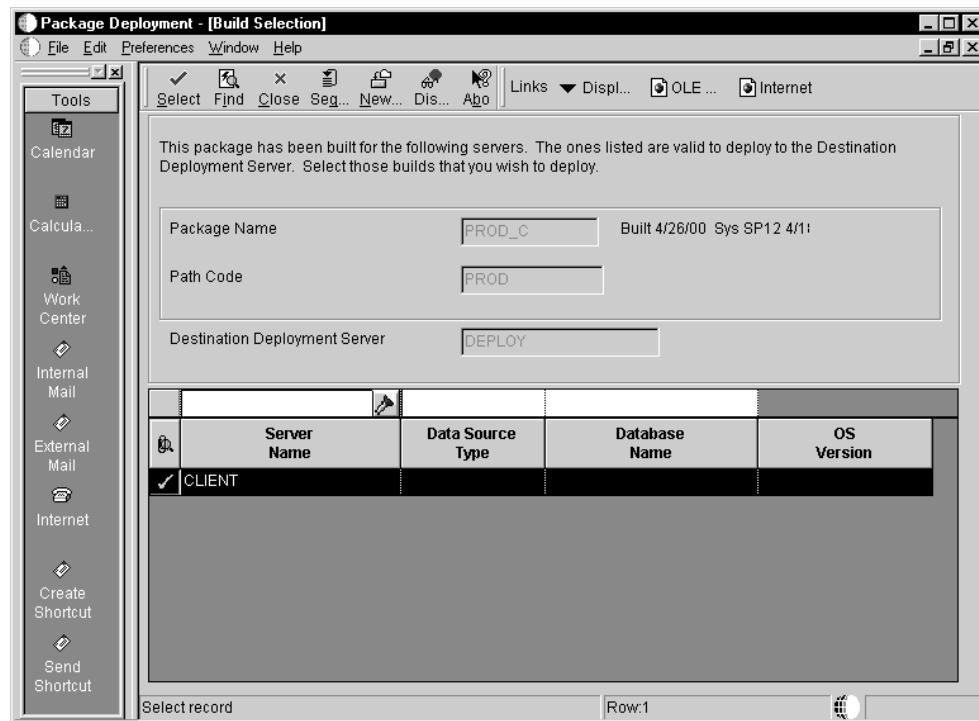
7. パッケージを配布するワークステーションを検索して選択し、[Next]をクリックします。

ワークステーションを選択するには、そのロー・ヘッダーをダブルクリックします。選んだ各ワークステーションのロー・ヘッダーにチェック・マークが表示されます。

デプロイメント・サーバーに配布する場合は、〈Deployment Server Selection〉フォームが表示されます。デプロイメント・サーバーに配布しない場合、次のステップはとばしてください。

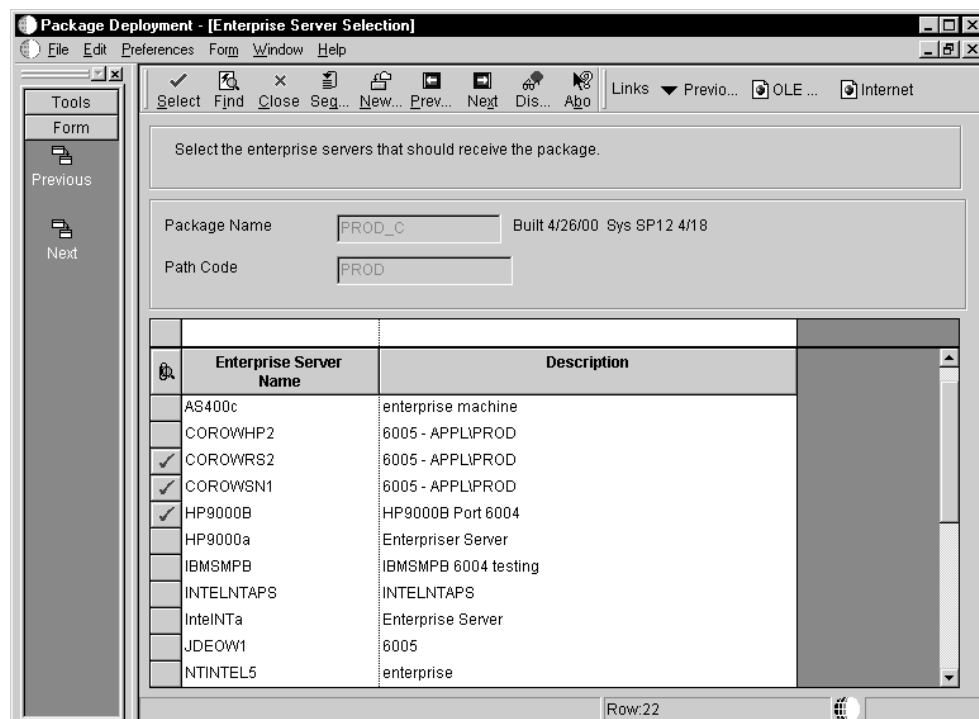


8. パッケージを配布するデプロイメント・サーバーを検索し、[Next]をクリックします。
サーバーを選択するには、そのロー・ヘッダーをダブルクリックします。選んだ各サーバーの横にチェック・マークが表示されます。
9. 〈Build Selection〉フォームで、宛先デプロイメント・サーバーに配布するサーバー・パッケージ・ビルドを選択し、[Close]をクリックします。

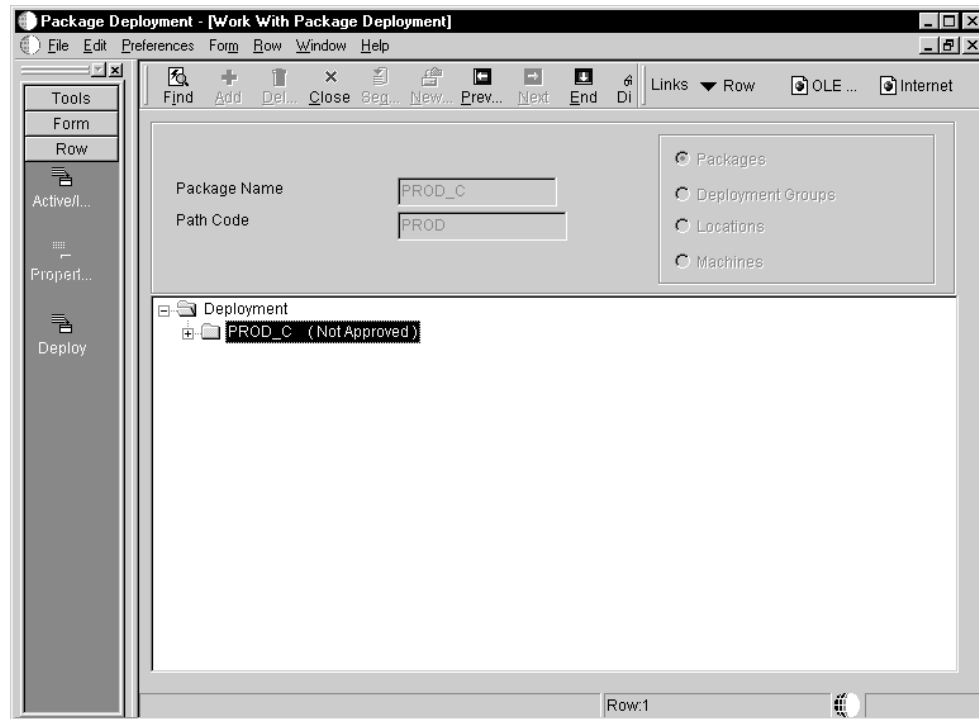


10. [Next]をクリックします。

エンタープライズ・サーバーに配布する場合は、〈Enterprise Server Selection〉フォームが表示されます。エンタープライズ・サーバーに配布しない場合、次のステップはとばしてください。



11. パッケージを配布するエンタープライズ・サーバーを検索して選択し、[Next]をクリックします。
サーバーを選択するには、そのロー・ヘッダーをダブルクリックします。
12. 〈Work with Package Deployment〉フォームが表示されます。ここで、デプロイメントの選択内容を検討します。
13. 選択内容を変更するには、[Prev]をクリックして前の適切なフォームに戻ります。



14. デプロイメントの選択内容を検討し、変更を終了したら、[End(終了)]をクリックします。
15. サーバー・パッケージを配布する場合は、〈Work with Package Deployment〉フォームでサーバー・パッケージを検索して選択し、[Row]メニューから[Deploy]を選択します。

注:

このステップの詳細については、「プッシュ・インストール用パッケージのスケジュール」を参照してください。

パッケージ・デプロイメントをスケジュールすると、指定した日時にパッケージがワークステーションに配布されます。ユーザーが ERP 9.0 にサインオンすると、このパッケージを使用可能にできます。

プッシュ・インストールを使用すると、パッケージは、〈Schedule Jobs〉プログラム(P91300)で指定した時刻に自動的にインストールされます。

注:

詳細については、「プッシュ・インストールの使用」を参照してください。

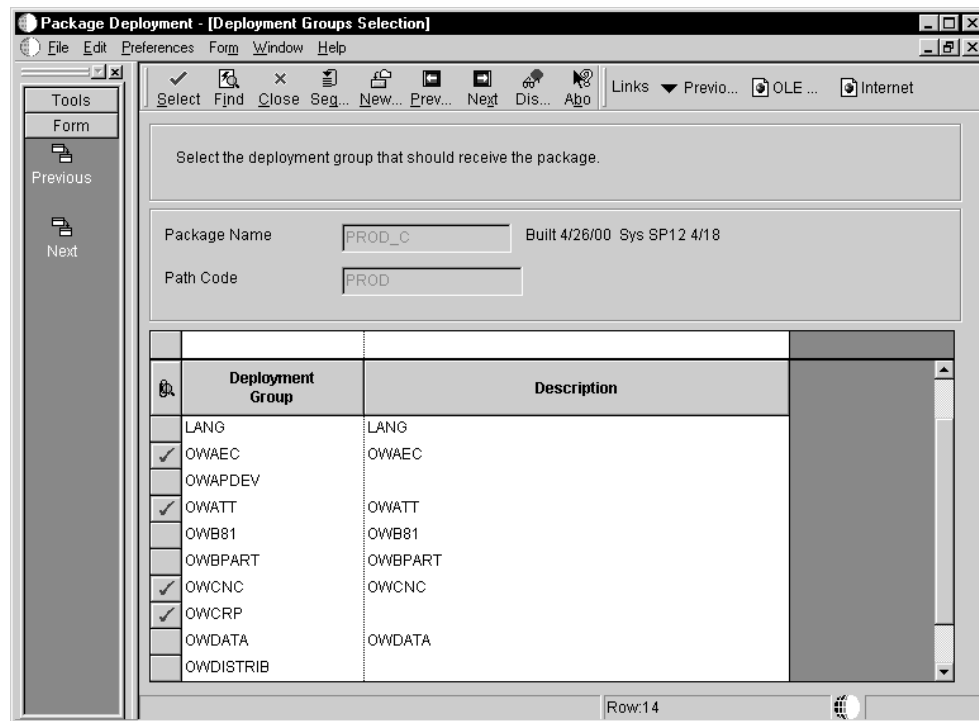
参照

- プッシュ・インストールについては「プッシュ・インストールの使用」
- プッシュ・インストールを使用してパッケージを配布する方法については、「プッシュ・インストール用パッケージのスケジュール」

▶ デプロイメント・グループまたはロケーションへのパッケージ・デプロイメントをスケジュールするには

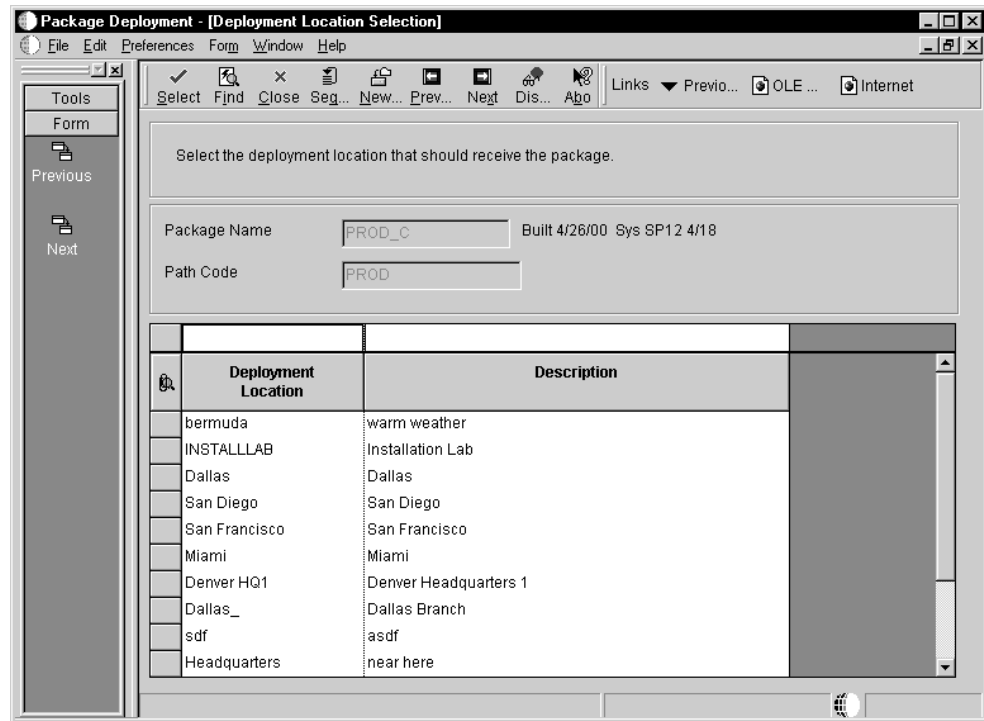
〈Package and Deployment Tools〉メニュー(GH9083)から、[Package Deployment]を選択します。

1. 〈Work with Package Deployment〉フォームが表示されます。〈Package Deployment Director〉を起動するには、[Add]をクリックします。
2. 〈Package Deployment Director〉フォームが表示されます。[Next]をクリックします。
3. 〈Package Selection〉フォームが表示されます。配布するパッケージを選択し、[Next]をクリックします。
4. 〈Package Deployment Targets〉フォームが表示されます。[Deployment Group]または[Locations]を選択して、[Next]をクリックします。
5. 〈Package Deployment Attributes〉フォームが表示されます。次のフィールドに値を入力します。
 - 必須インストール
 - プッシュ・インストール
 - 日付/時刻
6. パッケージをデプロイメント・サーバーからワークステーションにプッシュするプッシュ・インストールを使用してパッケージを配布する場合、[Enable Push Installation]オプションをクリックします。
7. [Next]をクリックします。



デプロイメント・グループに配布する場合は〈Deployments Group Selection (デプロイメント・グループの選択)〉フォームが表示されます。ロケーションに配布する場合、次のステップはとばしてください。

8. パッケージを受け取るデプロイメント・グループを検索して選択し、[Next]をクリックします。
グループを選択するには、そのロー・ヘッダーをダブルクリックします。
9. ロケーションに配布する場合は、〈Deployment Location Selection〉フォームが表示されます。デプロイメント・グループに配布する場合、次のステップはとばしてください。



10. パッケージを受け取るデプロイメント・ロケーションを検索して選択し、[Next]をクリックします。
ロケーションを選ぶには、そのロー・ヘッダーをダブルクリックします。
11. 〈Work with Package Deployment〉フォームが表示されます。ここで、デプロイメントの選択内容を検討します。
12. 選択内容を変更するには、[Prev]をクリックして前の適切なフォームに戻ります。
13. デプロイメントの選択内容を検討し、変更を終了したら、[End]をクリックします。
14. サーバー・パッケージを配布する場合は、〈Work with Package Deployment〉フォームでサーバー・パッケージを検索して選択し、[Row]メニューから[Deploy]を選択します。

パッケージ・デプロイメントをスケジュールすると、指定した日時にパッケージがワークステーションに配布されます。ユーザーが ERP 9.0 にサインオンすると、このパッケージを使用可能にできます。

プッシュ・インストールを使用すると、パッケージは、〈Schedule Jobs〉プログラム(P91300)で指定した時刻に自動的にインストールされます。

参照

- サーバー・パッケージを配布する方法については、「Package Deployment Director の処理」

▶ デプロイメント・オプションを改訂するには

〈Package and Deployment Tools〉メニュー(GH9083)から〈Package Deployment〉を選択します。

1. マシン名に応じた情報を表示するには、[Machines(マシン)]を選択してから[Find(検索)]をクリックします。
2. オプションを修正するパッケージを検索して選択し、[Row]メニューから[Properties(プロパティ)]を選択します。〈Deployment Properties Revisions(デプロイメント・プロパティの改訂)〉フォームが表示されます。

Package Deployment - [Deployment Properties Revisions]

File Edit Preferences Window Help

OK Cancel Dismiss Apply Links Display OLE Internet

Tools

Calendar

Calculator

Work Center

Internal Mail

External Mail

Internet

Create Shortcut

Send Shortcut

Package Name: SP1220PT2

Path Code: MSTB7323

Machine Key: APOLLO

Install Status: 20 Approved

Date: 7/7/1999 17:00:00

☐ Mandatory Installation ☐ Enable Push Installation

3. 次のいずれかのフィールドの情報を改訂します。
 - インストール状況
 - 日付
 - 必須インストール
 - プッシュ・インストールを可能にする
4. 改訂が完了したら、[OK]をクリックして〈Work with Package Deployment〉フォームに戻ります。

フィールド記述

記述	用語解説
パッケージ名	<p>World では、インストール・パッケージの識別名。</p> <p>J.D. Edwards ERP では、ワークステーションに展開するコンポーネントがサーバーのどこにあるかがパッケージに記述されています。パッケージには 3 つのタイプがあります。</p> <p>フル・パッケージ: J.D. Edwards ERP アプリケーション(すべてのスペック)を含みます。</p> <p>部分パッケージ: J.D. Edwards ERP の最小構成。すべてのアプリケーションを初期インストールするのではなく、選択したアプリケーションのみをロードすることができます。</p> <p>更新パッケージ: このパッケージ・タイプに含まれる J.D. Edwards ERP オブジェクトは、ワークステーションがパッケージを受け取り、ユーザーが J.D. Edwards にサインオンした後にロードされます。更新パッケージが〈ジャスト・イン・タイム〉アプリケーションを含む場合、アプリケーションの古いバージョンはワークステーションから削除され、ユーザーが最初にそのアプリケーションにアクセスする時に現バージョンに置き換えられます。更新パッケージは、常にシステム管理者が指定した日時に展開されます。</p> <p>アプリケーションを除き、パッケージはすべて、特定のパス・コード向けセントラル・オブジェクトのその時点での「スナップ・ショット」です。〈ジャスト・イン・タイム〉アプリケーションは動的であり、ビルドはされません。</p>
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。
必須インストール	<p>パッケージが必須か任意かを示します。</p> <p>有効な値は次のとおりです。</p> <p>Y デプロイメントは必須です。パッケージをインストールしてください。 N デプロイメントは任意です。</p>
プッシュ・インストールを可能にする	フィールドに“1”を入力すると、プッシュ・インストールでパッケージがインストールされます。
日付	アップデートされたオブジェクトをリストされたマシンに展開する日付

パッケージのアクティブ化

パッケージ・デプロイメントの定義を正しく完了したら、パッケージをアクティブにし、〈Workstation Installation〉プログラムを使用してインストールを可能にする必要があります。〈Workstation Installation〉の起動時に、アクティブにしていないパッケージは、使用可能パッケージ・リストには含まれません。

場合によっては、どのパッケージをインストール可能にするかを制御する必要がある場合もあります。たとえば、パッケージがテスト・グループ専用である場合、〈Workstation Installation〉プログラムを使用してインストールできないように、そのパッケージを非アクティブにする必要があります。その場合は、代わりに〈Package Deployment〉プログラム(P9631)を使用して、そのパッケージをテスト・グループのメンバに配布するようにスケジュールできます。

▶ パッケージをアクティブにするには

〈Package and Deployment Tools〉メニュー(GH9083)から〈Package Deployment〉プログラム(P9631)を選択します。

1. 〈Work With Package Deployment〉フォームで、[Packages(パッケージ)]ボタンをクリックし、[Find]をクリックします。
2. アクティブまたは非アクティブにするパッケージをリストから選択します。
または、[Package]フィールドにパッケージ名を入力します。
3. [Row]メニューから[Active/Inactive(アクティブ/非アクティブ)]を選択します。

スケジュール済みパッケージのインストール

ユーザーがパッケージを受け取る場合、または配布予定日以後に ERP 9.0 にサインオンした時点で、ユーザーはパッケージをインストールできます。

パッケージが必須の場合、ユーザーはパッケージをロードするまで ERP 9.0 にアクセスできません。

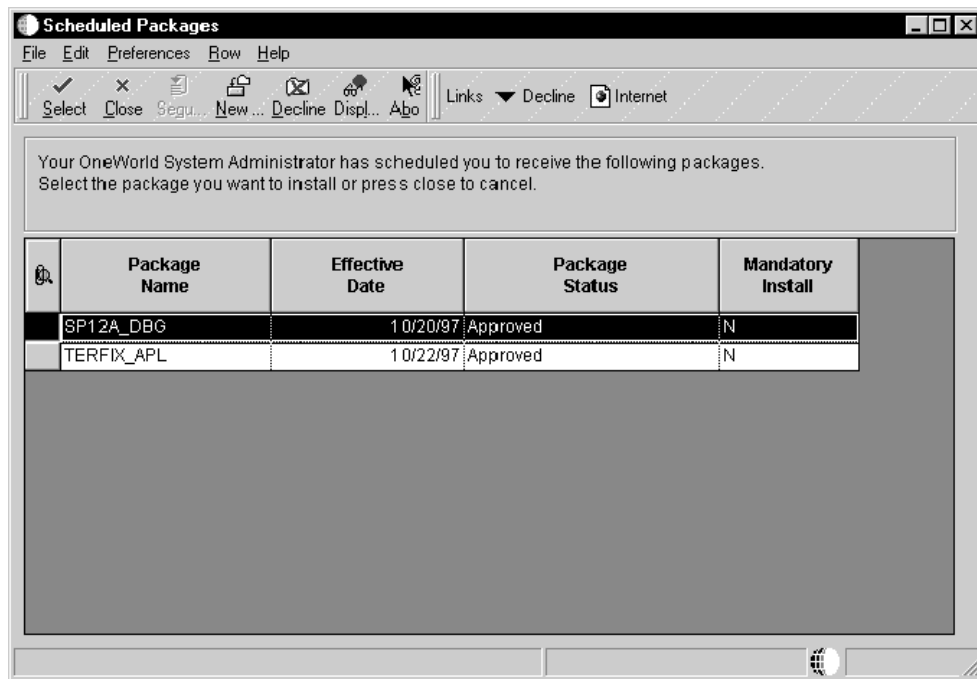
パッケージが任意の場合、パッケージを拒否することもできれば、インストールを後日に延期することもできます。パッケージのインストールを延期した場合、ERP 9.0 エクスプローラが起動し、次回サインオンした時にパッケージをインストールできるようになります。

パッケージが予定されていた時間帯にワークステーションの電源がオフにされていた場合など、何らかの理由でプッシュ・インストールがスケジュールされていたパッケージをロードできなかった場合、そのパッケージは、ユーザーが ERP 9.0 にサインオンした時に使用可能パッケージ・リストに含まれます。

▶ スケジュール済みパッケージをインストールするには

1. ERP 9.0 にサインオンします。

パッケージを受け取るようにスケジュールすると、〈Just-In-Time Installation (ジャストインタイム・インストール)〉プログラムが起動して、〈Scheduled Packages (スケジュール済みパッケージ)〉フォームが表示されます。



2. 次のいずれかの処理を実行します。

- パッケージを直ちにロードする場合は、次のステップにスキップします。
- パッケージを永続的に拒否する場合は、[Row]メニューから[Decline (拒否)]を選びます。
- パッケージ内のすべての項目を一覧表示するには、[Row]メニューから[Package Detail (パッケージ詳細)]を選択します。
- 別の時にパッケージをロードする場合は、[Close]をクリックする。パッケージが必須の場合は、パッケージをロードするまで ERP 9.0 エクスプローラにアクセスできません。

3. パッケージをロードするには、インストールするパッケージを選択してから、[Select]をクリックします。

〈Workstation Installation〉プログラムはパッケージをロードします。複数のパッケージを選択してあった場合、プログラムはそれらを順次インストールします。インストールが完了すると、ERP 9.0 エクスプローラが起動します。

サーバー・パッケージの配布

サーバー・パッケージのデプロイメント・プロセスは、ワークステーションへのパッケージ・デプロイメント・プロセスとほとんど同じです。どちらの場合も、〈Package Assembly (パッケージ・アセンブリ)〉プログラム(P9601)、〈Package Build (パッケージ・ビルド)〉プログラム(P9621)、および〈Package Deployment〉プログラム(P9631)を使用して、配布するパッケージの組立て、定義、ビルド、およびスケジュールを行う必要があります。

サーバー・パッケージ・デプロイメントをスケジュールした後は、サーバーに配布するためのバッチ・プログラムを起動する追加のステップを実行する必要があります。エンタープライズ・サーバーまたはデプロイメント・サーバーにパッケージを配布する場合は、このタスクを実行する必要があります。

注意:

エンタープライズ・サーバーは、インストール・プロセス時にビジネス・アプリケーションとバッチ・プロセスを処理できないため、サーバー・パッケージは、必要な時にのみ配布してください。ただし、エンタープライズ・サーバーはパッケージ・インストール時にシャットダウンされません。エンタープライズ・サーバーに投入されたジョブは待ち行列に入り、インストール終了直後に実行されます。したがって、ユーザーへの影響を最小限にするために、数時間後にエンタープライズ・サーバー・パッケージを配布するようにスケジュールすることをお勧めします (パッケージをエンタープライズ・サーバーに配布する前に、ERP 9.0 サービスが配布時に動作することを確認してください)。

開発環境と本稼動用環境が同じエンタープライズ・サーバー上で共存している場合、ネットワークとユーザーへの影響を最小限に抑えるために、開発者が各自のオブジェクトをサーバー・パッケージに移動できないようにしてください。代わりに、管理者がこの機能を実行する必要があります。

サーバー・パッケージを配布するには、〈Work with Package Deployment〉フォームの [Row] メニューから [Deploy] 機能を選択します。これは、複数階層デプロイメント時にパッケージをデプロイメント・サーバーに配布する際に使用する機能と同じです。

[Row] メニューから [Deploy] 機能を選択すると、〈Work with Package Deployment〉フォームで現在選択されている項目に基づいて、以下のバッチ・プログラムが呼び出されます。

- 特定のデプロイメント・サーバーが選択されている場合、〈Multitier Deployment〉バッチ・プログラム(R98825C)が起動します。
- [Deployment Server (デプロイメント・サーバー)] フォルダが選択されている場合、スケジュール済みパッケージが含まれているすべてのデプロイメント・サーバーに対して〈Multitier Deployment〉バッチ・プログラムが起動します。
- 特定のエンタープライズ・サーバーが選択されている場合、〈Enterprise Server Deployment (エンタープライズ・サーバー・デプロイメント)〉バッチ・プログラム(R98825D)が起動します。
- [Enterprise Server (エンタープライズ・サーバー)] フォルダが選択されている場合、スケジュール済みパッケージが含まれているエンタープライズ・サーバーに対して〈Enterprise Server Deployment〉バッチ・プログラムが起動します。
- 特定のパッケージが選択されている場合、選択されたパッケージに対し、〈Multitier Deployment〉バッチ・プログラムと〈Enterprise Server Deployment〉バッチ・プログラムが順に起動します。

- パッケージでソートし、[Deployment (デプロイメント)] フォルダが選択されている場合、すべてのパッケージに対して、〈Multitier Deployment〉バッチ・プログラムと〈Enterprise Server Deployment〉バッチ・プログラムが共に起動します。

特定ワークステーションまたは[Workstations]フォルダが選択されている場合、[Deploy]オプションは使用できません。

全サーバーまたは全パッケージに対してバッチ・プログラムが起動した時に、特定サーバーのパッケージがスケジュールされていないと、デプロイメントは実行されません。

はじめる前に

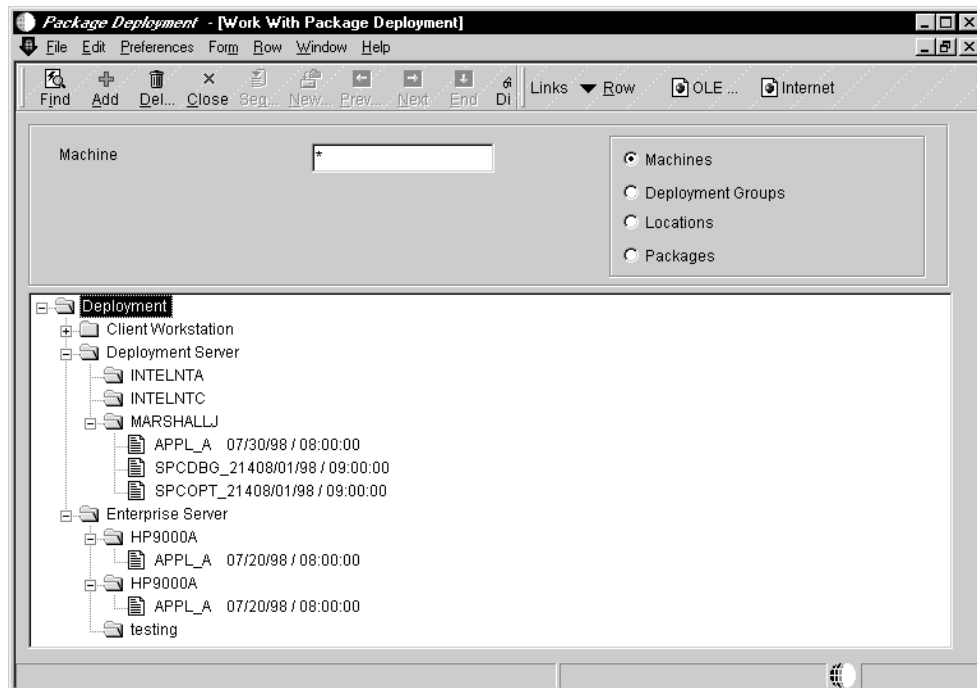
- サーバー・パッケージを組み立てます。
- サーバー・パッケージを定義します。
- サーバー・パッケージをビルドします。
- 該当するサーバーへのパッケージ・デプロイメントをスケジュールします。

参照

- パッケージをサーバーに配布するバッチ・アプリケーションの起動方法については、「プッシュ・インストール・バッチ・アプリケーションのスケジュール」

▶ サーバー・パッケージを配布するには

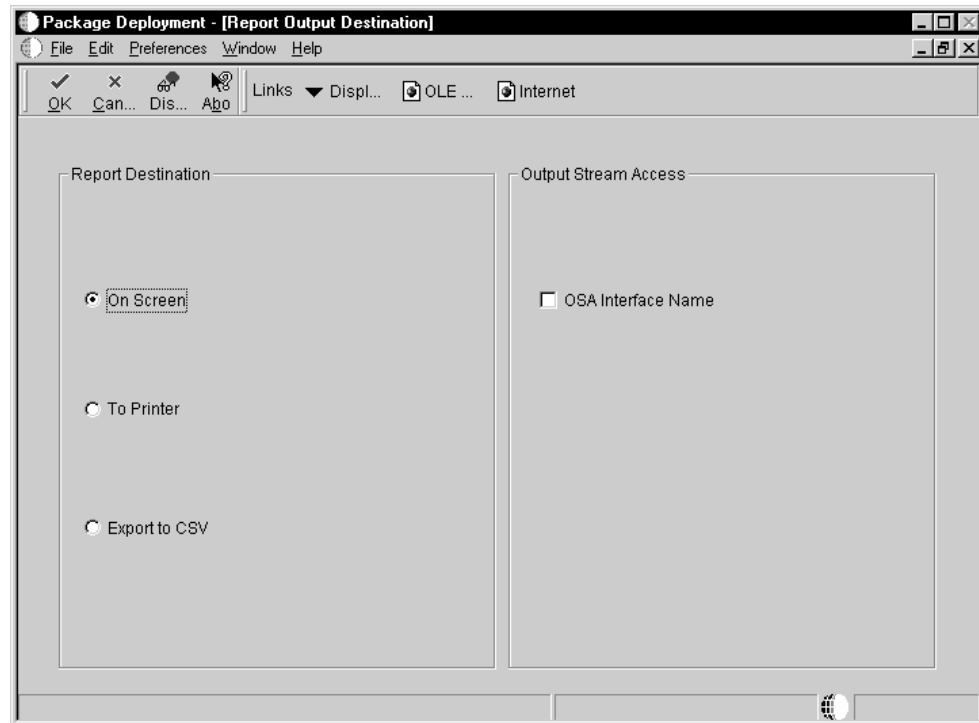
〈Package and Deployment Tools〉メニュー(GH9083)から〈Package Deployment〉を選択します。



1. 〈Work With Package Deployment〉フォームで、配布するサーバー・パッケージを検索します。

または、エンタープライズ・サーバーを選択するか、パッケージを複数台のサーバーに配布するようにスケジュールされている場合は、[Enterprise Servers]フォルダを選択します。

2. [Row]メニューから、[Deploy]を選択します。



3. 〈Report Output Destination(レポート出力先)〉フォームが表示されます。ここで、[On Screen(画面)]を選択します。
4. [OK]をクリックして続行します。

プッシュ・インストールの使用

プッシュ・インストールは、自動的に非在席でパッケージ・デプロイメントを提供する唯一の配布方法です。これは、ワークステーション・ユーザーの操作を求めずに、システム管理者がワークステーションまたはグループに1つのパッケージ(または複数のパッケージ)を配布できることを意味します。

たとえば、管理者は就業時間後にパッケージを特定のグループに配布するようにスケジュールできます。そのグループのメンバーから、翌朝作業するという報告を受けていれば、このパッケージはすぐに使用可能になります。

プッシュ・インストールは、パッケージを迅速に配布し、通常の本稼働および開発ルーチンに対する影響を最小限に抑える必要がある場合に、特に有効です。管理者はパッケージ・デプロイメントを賢明に計画およびスケジュールすることにより、非常に多くのパッケージ・デプロイメントを伴うネットワーク・パフォーマンスの負荷を最小にできます。また、プッシュ・インストールは、管理者がERP 9.0をワークステーションにはじめてインストールする時にも使用できます。この機能により、ダウンタイムを大幅に短縮できます。また、デプロイメントの柔軟性を最大にすることができます。

プッシュ・インストールの理解

プッシュ・インストールでは、パッケージの内容がデプロイメント・サーバーからワークステーションにプッシュされます。プッシュ・インストールに対して、〈Workstation Installation〉プログラムはデプロイメント・サーバーからワークステーションにパッケージの内容をプルします。また、プッシュ対応をしていないスケジュール済みパッケージで〈Package Deployment〉プログラム(P9631)を使用するように設定されているインストールでも、パッケージをプルできます。

パッケージの内容がプッシュされるか、プルされるかにかかわらず、デプロイメントの最終結果は同じです。ただし、プッシュ・インストールには、パッケージ・デプロイメントの予定時間にワークステーションをオンの状態にしておくこと以外、ワークステーション・ユーザーの操作が一切求められないという利点があります。

プログラム・スペックを含んでいる部分パッケージやアップデート・パッケージの場合、「パッケージの内容」という用語はスペックを意味します。また、アプリケーション・スペックを含まないフル・パッケージやアップデート・パッケージの場合、「パッケージの内容」という用語はオブジェクトを意味します。

次のリストにプッシュ・インストール・プロセスのステップを要約します。

1. プッシュされるパッケージを受け取る各ワークステーションに、プッシュ・インストール用〈Listener〉プログラムをインストールします。このプログラムは、サーバーで動作し、インストール状況のモニタなどの機能を実行する〈Push Package Installation (プッシュ・パッケージ・インストール)〉バッチ・プログラム(R98825)の進行状況をモニタします。〈Listener〉はローカル・サービスまたはネットワーク・サービスとして実行できます。
2. 〈Package Deployment〉プログラム(P9631)を使用して、パッケージをスケジュールします。〈Push Package Installation〉バッチ・プログラムはスケジュール・テーブルから読み取り、パッケージ・デプロイメントがスケジュールされているすべての関連ワークステーション上の〈Listener〉にメッセージを送信します。
3. 〈Schedule Jobs〉プログラム(P91300)を使用して、エンタープライズ・サーバー上でバッチ・プログラムを起動します。このプログラムでは、ジョブ名、バージョン、開始日時、および定期実行を指定できます。
4. 指定の開始時刻に、〈Schedule Jobs〉プログラムは、〈Push Package Installation〉バッチ・プログラム(R98825)を起動し、パッケージ・インストール・プロセスをデプロイメント・サーバーから開始します。このプロセスの間、〈Listener〉とバッチ・プログラムは、インストールが完了するまで連絡します。インストール状況(失敗、成功、進行中など)を示すコードが、〈Listener〉からバッチ・プログラムに送られます。
5. インストールが完了すると、ワークステーションの基本ユーザーに電子メールでメッセージが送信されます。このメッセージは、インストールが成功したかどうかを示します。電子メールによる通知は、パッケージ受信者がマシン・マスター・テーブル(F9650)に一覧表示され、電子メール・アドレスがプロファイルに記述にされている場合のみ実行されます。
6. パッケージ受信者がワークステーションをオフにしていた場合など、何らかの理由でプッシュ・インストールが失敗した場合、インストール状況は「Failed」に変更されます。インストールを再びスケジュールする場合、失敗したジョブが含まれているローをまず削除してから、ジョブを再度スケジュールする必要があります。

プッシュ・インストールが失敗した場合、ユーザーが ERP 9.0 にサインオンした時に、標準スケジュール画面が表示されます。ここで、ユーザーは必須パッケージを受け入れるか、または ERP 9.0 を終了できます。

プッシュ・インストール用エンタープライズ・サーバーの準備

サーバーでプッシュ・インストールをセットアップするには、Microsoft Windows NT Server 4.0 に組み込まれている DNS(Microsoft Domain Name Service)を最初にインストールして構成する必要があります。ドメイン・ネーム・サービスをセットアップしていない場合、Microsoft DNS をインストールするには、[Control Panel(コントロール・パネル)]の[Network(ネットワーク)]アイコンを選択し、[Service(サービス)]タブを選択して、Microsoft DNS Server を追加します。

Microsoft DNS を追加したら、ドメイン名およびサーバーを指定して、DNS を構成する必要があります。

UNIX および AS/400 の考慮事項

Windows NT サーバーは、ワークステーション・アドレスを動的に割り当てるため、DHCP(動的ホスト構成プロトコル)用に構成された環境では、ワークステーション・アドレスを検索するために、Windows NT Server 4.0 をサーバーで実行する必要があります。

名前で識別させるには、Windows NT DNS サーバーで IP アドレスを検索してアドレスを識別するようにサーバーを構成し、また、DHCP がネットワーク・ドメインで有効な時に WINS データベースを参照するように Windows NT DNS サーバーを構成する必要があります。

このようにサーバーを構成すると、プッシュ・インストール・プロセス時に、次のプロセス・フローが発生します。

1. 〈Push Installation〉バッチ・プログラム(R98825)を実行するホスト・サーバーから、ビジネス関数は DNS サーバーを参照して、マシン・ホスト・アドレスの取り込みを試みます。
2. DNS サーバーは IP アドレスを含まないので、WINS サーバーから IP アドレスを取り込みます。
3. WINS サーバーは DNS サーバーにアドレスを戻します。
4. DNS サーバーはホスト・サーバーにアドレスを戻します。
5. ホスト・サーバーはクライアント・ワークステーション上の〈Listener〉を検索し、ワークステーション・インストール情報を送信します。
6. ワークステーション・インストール・プロセスが開始されます。

プッシュ・インストール用ワークステーションの準備

インストールをワークステーションにプッシュする前に、サーバー上で動作するビジネス関数と関係する〈Listener〉をワークステーションにインストールする必要があります。パッケージ配布先のマシンに ERP 9.0 がすでにインストールされているか、または ERP 9.0 をはじめてインストールするかにかかわらず、この〈Listener〉は、プッシュ・インストールに使用可能なすべてのワークステーションにインストールする必要があります。

〈Listener〉は、Windows NT ワークステーション上でサービスとして(Windows 95 マシン上では擬似サービスとして)継続的に実行され、Task Manager を使用してモニタできます。〈Listener〉は、インストール・プロセス時にメッセージを送受信して、サーバー上のバッチ・アプリケーションと通信します。また、〈Listener〉はインストールの進捗状況をモニタし、インストール完了コードを保存します。

Listener のインストール

〈Listener〉を社内のワークステーションにインストールする時は、ローカル・サービスとして実行するか、またはネットワーク・サービスとして実行するかを指定します。ワークステーション上でサービスをローカルに実行する場合、プッシュ・インストールのスケジュール済みのパッケージを受け取る際に、ユーザーはログオンしている必要があります。サービスをネットワーク上で実行する場合、〈Listener〉はネットワーク・アカウントとして動作し、ユーザーは、プッシュ・インストールによってパッケージを受け取る際にログオンしている必要ありません。ネットワーク・サービスでは、アドミニストレータ・ユーザーID が必要です。

ネットワーク上でサービスを実行する欠点は、企業内のすべてのユーザーを管理するのが困難なことです。たとえば、〈Listener〉のパラメータはネットワーク上のユーザー全員に適用されるので、プッシュ・インストール・ユーザーは、同じロケーションから同じロケーションに ERP 9.0 をインストールする必要があります。1 人のユーザーが、ERP 9.0 をドライブ C にインストールして、別のユーザーが同じリリースをドライブ D にインストールすることはできません。また、ユーザーがサインオン・パスワードを変更するたびに、システム管理者は、〈Listener〉サービスを更新して新しいパスワードを有効にし、これらのユーザーがサービスを利用できるようにする必要があります。これらの理由により、J.D. Edwards では、〈Listener〉を各ワークステーションにローカルにインストールすることをお勧めします。

〈Listener〉をローカル・サービスとして実行する場合も、ネットワーク・サービスとして実行する場合も、スケジュール済みのパッケージを受け取るには、ワークステーションの電源をオンにしておく必要があります。

〈Listener〉をワークステーションにインストールする時は、次のいずれかの方法を選択できます。

- Tivoli Management Environment (TME10) Software Distribution System、または SMS(Microsoft System Management Server)ソフトウェアなどのサードパーティ・ソフトウェア配布システムを使用する。
- イン트라ネット Web サイトまたは WWW 経由で、実行可能なインストール・プログラム (setup.exe ファイル) と関連補助ファイルを配布する。
- C プログラムを呼び出すために、Windows NT ログオン・スクリプト (.bat ファイル) を使用する。
- WWW からインストールする。

注意:

〈Listener〉がワークステーションにインストールされていないか、動作していない場合 (または、ワークステーションの電源がオフになっている場合) は、プッシュ・インストールを実行できません。パッケージをスケジュールした後に、勤務時間外でもワークステーションの電源を切らずに〈Listener〉サービスを実行させておくように、パッケージ受信者に必ず通知してください。また、〈Listener〉をローカル・サービスとして実行するように設定している場合も、ログオン状態にするようにユーザーに通知してください。

はじめる前に

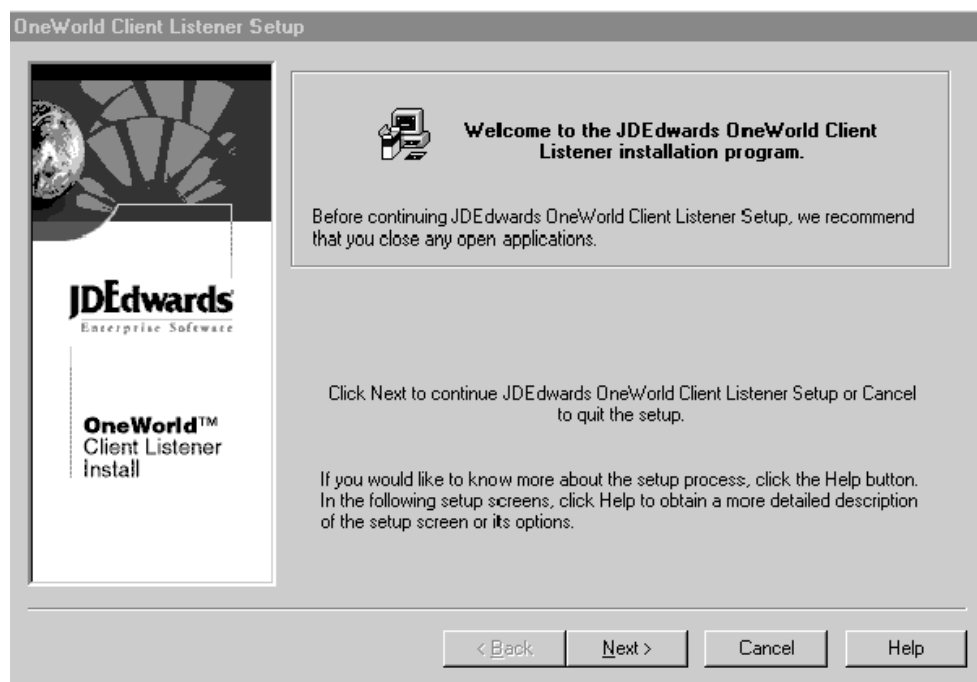
- 現在実行しているアプリケーションをすべて終了します。
- 〈Listener〉をインストールする先のディレクトリに十分なディスクの空き領域が存在することを確認します。〈Listener〉のファイルとコンポーネントをすべてインストールするには、約 2MB の空きディスク領域が必要です。

▶ Listener をワークステーションにインストールする方法

ここでは、インストール・プログラム (setup.exe プログラム) を起動して、〈Listener〉をインストールする方法を説明します。このプログラムを社内のユーザーに配布するには、プログラム自体またはショートカットを電子メールで送信するか、プログラムのサーバー上の格納場所を通知します。

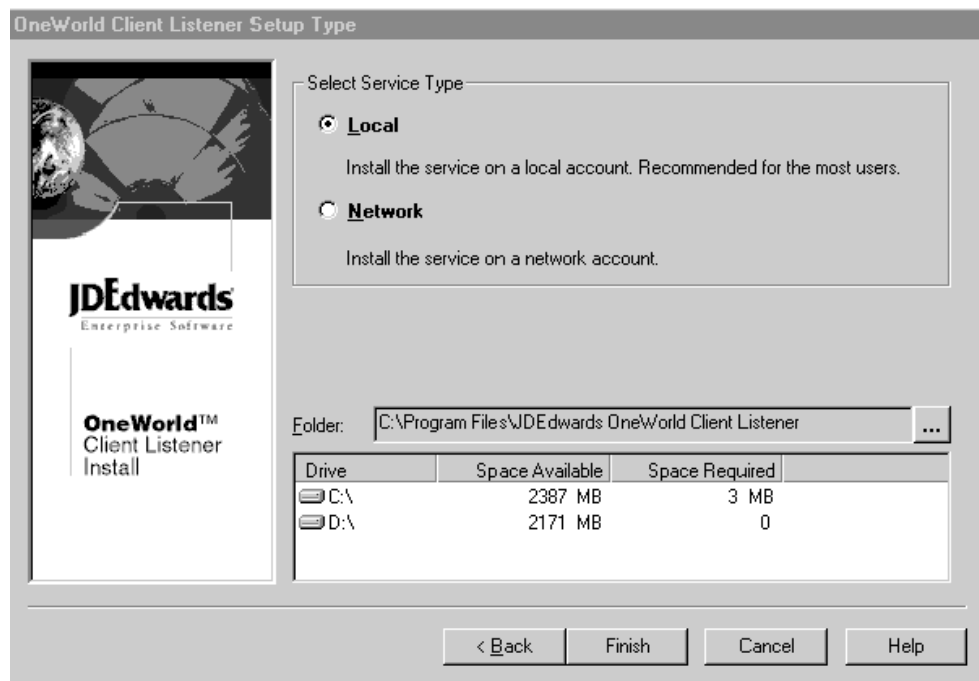
〈Listener〉の旧バージョンをインストール済みの場合、インストール・プログラムは新規バージョンをワークステーションにコピーする前に、旧バージョンを削除します。

1. [setup.exe] アイコンをダブルクリックすると、インストール・プログラムが起動します。



2. 最初の〈ERP 9.0 Client Listener Setup (ERP 8.0 クライアント・リスナー・セットアップ)〉フォームで、[Next]をクリックします。

3. 2 番目の〈ERP 9.0 Client Listener Setup〉フォームで、次のフィールドに値を入力します。
 - ERP 9.0 リリース
プッシュ・インストールによってインストールする ERP 9.0 リリースを選択します。
 - Path name (パス名)
選択した ERP 9.0 リリースで、ERP 9.0 のインストールを開始するデプロイメント・サーバー上の絶対パス名を入力します。
 - ERP 8.0 installation drive (ERP 9.0 インストール・ドライブ)
指定した ERP 9.0 のリリースをインストールするドライブを指定します。
 - Uninstall previous release (前回リリースのアンインストール)
新しいフル・パッケージをインストールする前に既存バージョンの ERP 9.0 をアンインストールするには、このアンインストール・オプションをオンにします。
 - Start the Listener automatically on startup (〈Listener〉サービスの自動起動)
ワークステーションの起動時に、常に〈Listener〉サービスを自動的に起動するには、この自動起動オプションをオンにします。
4. 次のインストール・フォームに進むには、[Next]をクリックします。



5. 次のいずれかのオプションをクリックします。

- Local
- Network(ネットワーク)

システム管理者がネットワーク上に<Listener>をインストールしていない場合、<Listener>をローカル・ワークステーションにインストールするには、[Local(ローカル)]をクリックします。

6. 次のフィールドに値を入力します。

- Folder(フォルダ)

[Listener(リスナー)]ファイルを常駐させたいドライブおよびフォルダを指定します。

7. インストールを終了するには、[Finish(終了)]をクリックします。

<Listener>がワークステーションに正しくインストールされると、耳の形をした小さなアイコンがWindows タスクバーに表示され、<Listener>がロードされたことを示します。このアイコンをダブルクリックすると、<Listener>を起動または停止したり、デフォルトのパラメータ設定を変更できます。

参照

- <Listener>の再構成については「デフォルト・パラメータ設定の変更」

Silent インストールを使用した<Listener>のインストール

<Listener>をワークステーションにインストールする場合、Silent インストールを使用するほうがより便利な場合があります。この方法を使用する時は、システム管理者がすべてのワークステーションの構成設定を入力し、ワークステーションにユーザーが次回サインオンした時に自動的に<Listener>のインストールを開始するバッチ・ファイルを配布します。

〈Listener〉のインストールで Silent インストールを使用すると、インストール・プロセスがワークステーション・ユーザーに対して透過的になり、ユーザーが構成情報を入力したり、インストール・プロセスを実行する必要がなくなります。

次のタスクでは、Silent インストールを使用して〈Listener〉をインストールする方法を説明しています。通常、このタスクは、システム管理者が行います。

▶ Silent インストールによって〈Listener〉をインストールするには

1. ソフトウェア CD に収められている listen_silent_setup.inf ファイルで、次の設定を編集します。

- ServiceType (サービス・タイプ)
〈Listener〉サービスを実行する場所に応じて、“Local”または“Network”を入力します。
- WorkstationDirPath (ワークステーション・ディレクトリ・パス)
〈Listener〉プログラムと関連ファイルをインストールする、ワークステーション上のロケーションを入力します (たとえば、「C:¥Program Files¥OneWorld Client Listener」など)。
- Release (リリース)
ソフトウェアの基本リリースを入力します。たとえば B9 などです。累積アップデート・リリース (B9.1 など) は入力しないでください。
- InstallPath (インストール・パス)
ERP 9.0 をインストールする、ワークステーション上のロケーションを入力します (D:¥b9 など)。
- LaunchPath (起動パス)
デプロイメント・サーバーの名前と〈Client Workstation Installation (クライアント・ワークステーション・インストール)〉プログラムを実行するロケーションを入力します (¥¥server name¥b9¥OneWorld Client Install¥setup.exe など)。
- AutoStart (自動起動)
ワークステーションの起動時に〈Listener〉サービスを自動的に起動するには、“1”を入力します。自動起動を無効にする場合は、“0”を入力します。
- UninstallPackage (パッケージのアンインストール)
新しいフル・パッケージをインストールする前に ERP 9.0 の前のバージョンを自動的にアンインストールする場合は、“1”を入力します。自動アンインストールを無効にする場合は、“0”を入力します。

2. 〈ListenSetup.exe〉プログラム用の Silent インストール・パラメータ/s を組み込むには、バッチ・ファイルを作成または修正します。バッチ・ファイルは〈ListenSetup.exe〉プログラムと同じロケーションに存在する必要があります。

たとえば、バッチ・ファイルには次のような行が含まれることがあります。

```
start ¥¥servername¥b9¥client¥misc¥ListenSetup.exe /s listen_silent_setup.inf
```

3. ワークステーション・ユーザーに inf ファイルおよびバッチ・ファイルを配布します。電子メール経由でこれらのファイルを配布、またはワークステーション・ユーザーが各自のワークステーションにファイルをコピーできるネットワーク・サーバーにこれらを置くことができます。
4. ユーザーのワークステーションを再起動し、バッチ・ファイルを実行して、〈Listener〉を Silent インストールによってロードするよう指示します。

ワークステーション・ユーザーが〈Listener〉を正しくインストールすると、Windows タスクバーに [Listener] アイコンが表示されます。このアイコンをダブルクリックすると、〈Listener〉の起動、停止、設定変更を行うことができます。

注意:

ソフトウェア付属の [Listener Silent] インストール・ファイルの名前は変更しないでください。ファイル名は、「listen_silent_setup.inf」にしておく必要があります。

Listener の停止

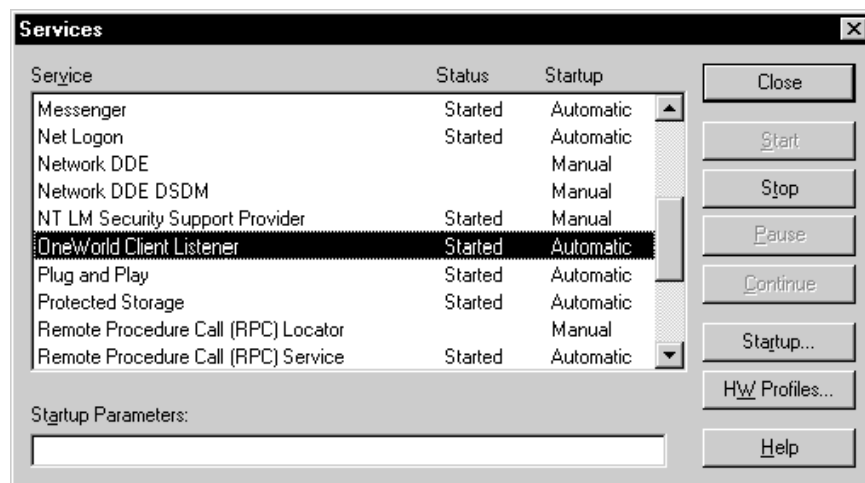
ERP 9.0 パッケージをインストールする時にプッシュ・インストールを使用しないことが明かな場合は、〈Listener〉を停止できます。その後、インストールを使用することになった場合は、〈Listener〉を再起動できます。

▶ Listener を停止するには

〈Listener〉を停止する最も簡単な方法は、Windows タスクバー上の [Listener] アイコンを右クリックしてから、〈Stop Listener (Listener の停止)〉を選択することです。

その他の方法として、次のステップを実行することにより、〈Listener〉を停止できます。

1. [Control Panel]を開きます。
2. [Services]を選択します。



3. [OneWorld Client Listener]を選択します。
4. [Stop (停止)]をクリックします。

▶ Listener をアンインストールするには

1. [Control Panel]を開きます。
2. [Add/Remove Programs(プログラムの追加/削除)]を選択します。
3. [OneWorld Client Listener]を選択します。
4. [Remove All Components(全コンポーネントの削除)]をクリックします。

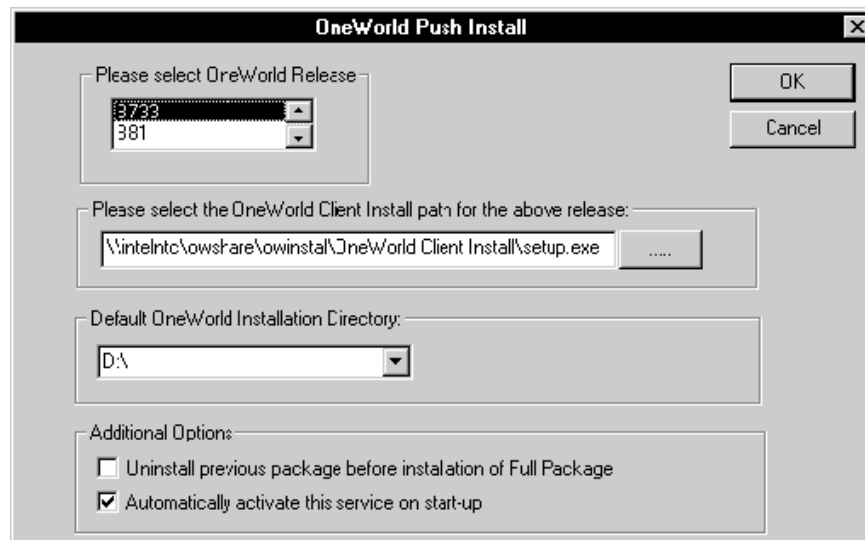
デフォルト・パラメータ設定の変更

Windows タスク バーに表示される[Listener]のアイコンを使用すると、〈Listener〉を最初にインストールしたときに入力したデフォルト・パラメータ設定を変更できます。次の項目を指定できます。

- ERP 9.0 リリースおよびそのリリースのクライアント・インストール・パス
- デフォルトの ERP 9.0 インストール・ディレクトリ
- 新規フル・パッケージをインストールする前に、前回のパッケージをアンインストールするかどうか

▶ デフォルト・パラメータ設定を変更するには

1. Windows タスク バー上の[Listener]アイコンを右クリックします。
2. [Change Default Parameters(デフォルト・パラメータの変更)]を選びます。



3. 次のフィールドに値を入力します。
 - ERP 9.0 リリース
プッシュ・インストールによってインストールする ERP 9.0 リリースを選択します。
 - Path name(パス名)
選択した ERP 9.0 リリースで、ERP 9.0 のインストールを開始するデプロイメント・サーバー上の絶対パス名を入力します。必要に応じて、フィールドの右のボタンをクリックして、使用可能なパスを参照します。

- Default ERP 9.0 installation directory (デフォルトの ERP 9.0 インストール・ディレクトリ)
指定した ERP 9.0 のリリースをインストールするドライブを指定します。
- Uninstall previous package before installation of Full Package (フル・パッケージをインストールする前に、前回のパッケージをアンインストールする)
フル・パッケージをインストールする前に、前回のパッケージをアンインストールするかどうかを指定します。このオプションをオンにすると、ERP 9.0 は、新しいフル・パッケージをインストールする前に、既存パッケージを完全にアンインストールします。
- Automatically activate this service on start-up (起動時にこのサービスを自動的に起動する)
ワークステーションにサインオンした時に〈Listener〉サービスを自動的に起動する場合、このオプションをオンにします。

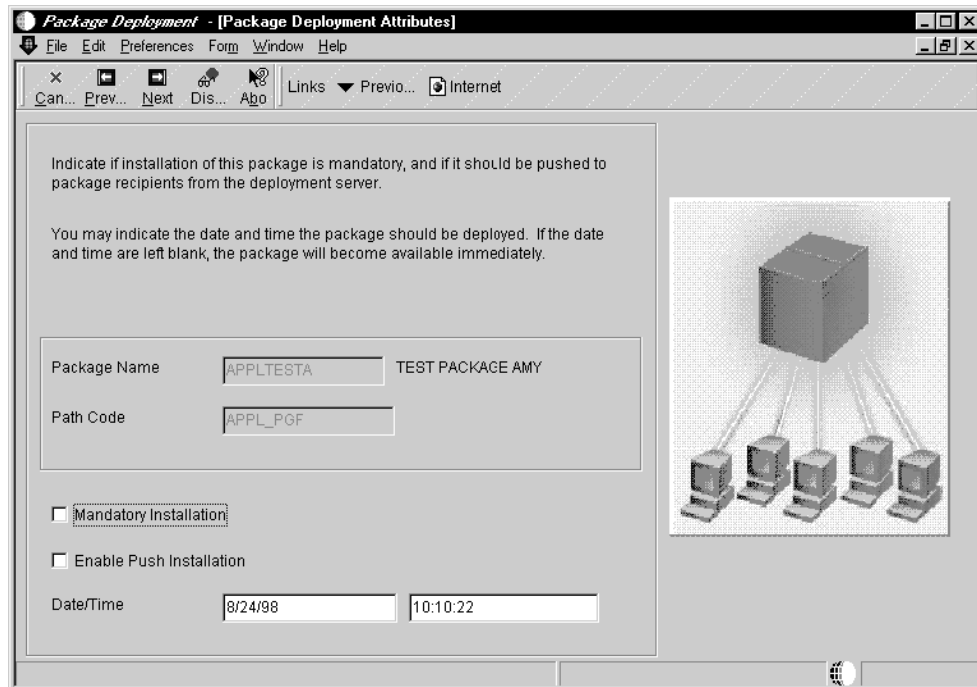
4. 変更が終了したら、[OK]をクリックします。

プッシュ・インストール用パッケージのスケジュール

〈Listener〉をワークステーションにインストールした後、パッケージ・デプロイメントをスケジュールできます。

プッシュ・インストール用パッケージのスケジュールのプロセスは、〈Package Deployment〉プログラム(P9631)を使用してパッケージをスケジュールするプロセスと同じです。このプログラムを使用してパッケージをスケジュールする時は、〈Package Deployment Attributes〉フォームで[Enable Push Install]オプションをオンにします。このオプションをオンにしないと、パッケージはデフォルトのスケジュールに従って配布されます。

〈Schedule Jobs〉プログラム(P91300)を使用すると、期間を設定して、ジョブが正常に完了するまでの実行頻度を確定できます。「期間」を設定しない場合、ジョブは一度だけ実行されます。〈Push Installation〉の場合、「期間」はインストール試行の時間間隔を確定します。パッケージが正常に配布されると、ジョブは終了します。



他のパッケージ・デプロイメントをスケジュールするには、Machine Master(F9650)テーブルにすべてのマシン名(つまり、パッケージ受信者)を定義する必要があります。このテーブルは、ユーザーがERP 9.0 にサインオンすると自動入力されます。また、〈Machine Identification〉プログラム(P9654A)を使用して、マシン名を手入力することもできます。

プッシュ・インストール・バッチ・アプリケーションのスケジュール

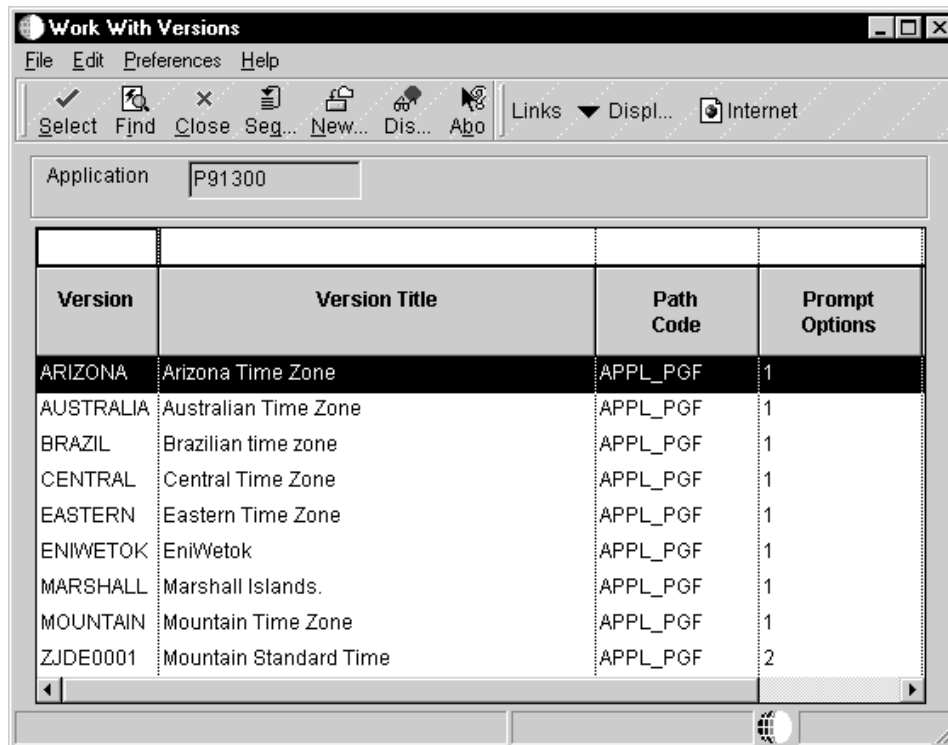
関連するすべてのワークステーションに〈Listener〉をインストールし、〈Package Deployment〉プログラム(P9631)を使用してパッケージをスケジュールした後は、〈Schedule Jobs〉プログラム(P91300)を使用して、〈Push Package Installation〉バッチ・プログラム(R98825)をサーバーで実行する必要があります。

はじめる前に

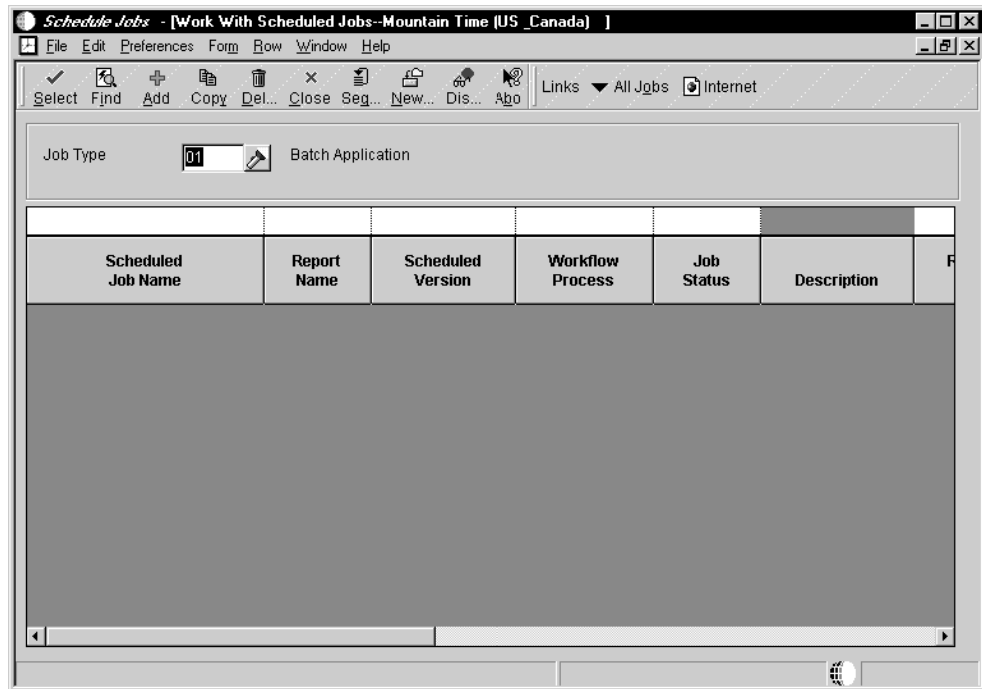
- 就業時間後もワークステーションをオフにしないように、パッケージ受信者に通知する。
- ログオンしている必要があるローカル・サービスのユーザーに通知する。
- パッケージ受信者に〈Listener〉が実行されていることを確認するように通知する。

▶ プッシュ・インストール・バッチ・アプリケーションをスケジュールするには

〈Job Scheduler〉メニュー(GH9015)から〈Schedule Jobs〉(P91300)を選びます。



1. 〈Work With Versions(バージョンの処理)〉フォームが表示されます。セットアップに適用する標準時間帯を選択します。



2. 〈Work with Scheduled Jobs(スケジュール済みジョブの処理)〉フォームが表示されます。新規ジョブを入力するには、[Add]をクリックします。

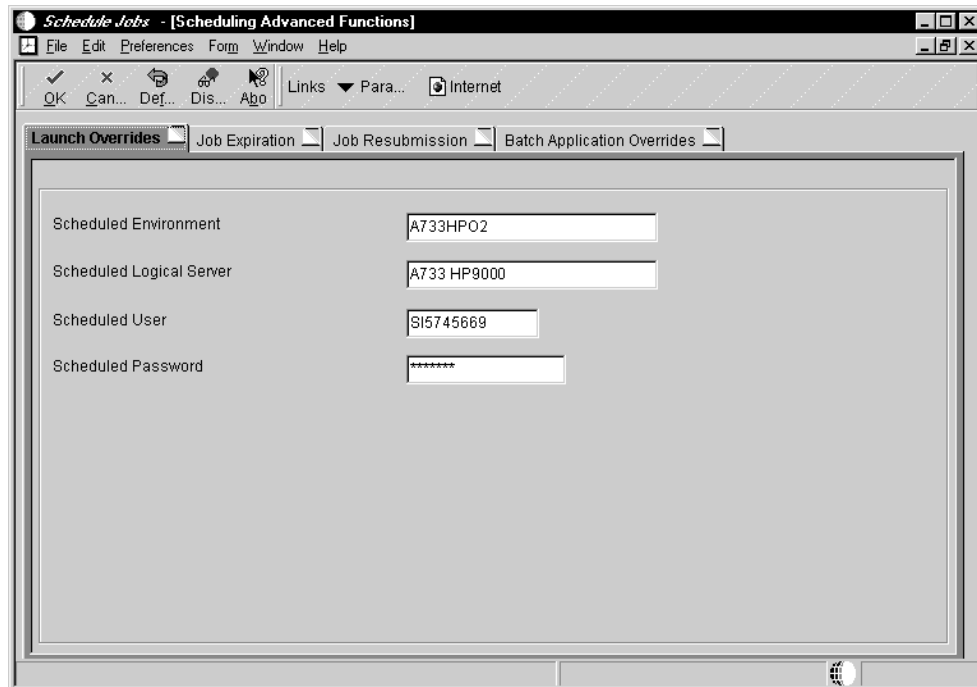
3. 〈Scheduling Information(スケジュール情報)〉フォームで、次のオプションをクリックします。
 - Batch(バッチ)
4. 次のフィールドに値を入力します。

- スケジュール済みジョブ名
- スケジュール済みジョブ状況
- スケジュール済みバッチ・アプリケーション
このフィールドに“R98825”を入力します。
- スケジュール済みバージョン
たとえば「XJDE0001」など。
- スケジュール済み開始日付/時刻

5. ジョブの「期間」(ジョブを実行する頻度を指定するには)を設定するには、[Form]メニューから[Recurrence(期間)]を選択します。

このフォームのフィールドに値を入力して期間を指定しない場合、ジョブは一度だけ実行されます。プッシュ・インストール・バッチ・プログラムの場合は、30 分間隔で実行するように期間を設定することをお勧めします。

6. <Recurring Scheduling Information Revisions(定期スケジュール情報の改訂)>フォームで、[OK]をクリックします。
7. <Scheduling Information>フォームで、一時変更、再投入、満了の各オプションを入力するには、[Form]メニューから[Advanced Options(上級オプション)]を選びます。



8. 〈Scheduling Advanced Functions(上級オプションのスケジュール)〉フォームが表示されます。
入力または改訂する情報に対応するタブをクリックします。

- Launch Overrides(起動一時変更)
- Job Expiration(ジョブの有効期限)
- Job Resubmission(ジョブの再投入)
- Batch Application Overrides(バッチ・アプリケーション一時変更)

9. [OK]をクリックします。

ジョブをスケジュールをした後は、〈Object Configuration Manager〉プログラム(P986110)を使用して、〈Push Installation〉バッチ・プログラムを実行しているサーバーが、〈Package Deployment〉プログラムが使用するのと同じパッケージ・デプロイメント・スケジュール・テーブル(F98825)およびマシン・マスター・テーブル(F9650)をポイントしていることを確認できます。

参照

- 〈Schedule Jobs〉プログラムの使用方法については、『System Administration Guide(システム・アドミニストレーション)』ガイドの「Scheduling Jobs(ジョブのスケジュール)」を参照してください。

フィールド記述

記述	用語解説
スケジュール済みジョブ名	スケジュール済みのジョブをシステムとユーザーに固有に識別します。月次締めや夜間のバックアップなどの機能を指示するのに使用できます。
スケジュール済みジョブ状況	スケジュールされたジョブの現在の状況。状況がアクティブであるかぎり、スケジューラは、ジョブをサーバーに投入するかどうか指定します。スケジュールされた終了日付になると、状況が無効に変更されます。スケジューラによるジョブの投入を停止するには、終了日付になる前に状況を無効(または一時停止)に変更します。ジョブを再度含める場合は、ジョブを再度有効にすることができます。この場合、有効にできるジョブは、終了日付が将来になっている場合のみです。
スケジュール済みバッチ・アプリケーション	スケジューラがサーバーに投入するレポートのオブジェクト名
スケジュール済みバージョン	実行するようスケジュールされたレポートのバージョン。特定のデータ選択やバッチジョブで使用される順序を識別します。
スケジュール済み開始日付/時刻	スケジュールされたジョブをサーバーに投入した次の日付

Push Package Installation Results レポートの実行

〈Package and Deployment Tools〉メニュー(GH9083)から〈Push Package Installation Results (プッシュ・パッケージ・インストール結果)〉レポート(R98825B)を選択します。

このレポートには、〈Push Package Installation〉バッチ・プログラム(R98825)の実行時に得られる情報と同じ情報が提供されます。

このレポートには、次の情報が含まれています。

- マシン・キー
- パッケージ名およびパス・コード
- ユーザー・クラスまたはグループ
- パッケージ状況および状況記述
- インストール状況
- パッケージ・インストール記述
- 必須インストール(「はい」または「いいえ」)

プッシュ・インストール状況コード

次の表に、〈Push Package Installation〉プログラム(R98825)が使用する状況コードと記述を示します。アスタリスク(*)が付いたコードは、〈Push Package Installation〉プログラムが、その次回実行時にインストールの試行を継続する状況を示します。

状況コード	説明
200*	Scheduled(スケジュール済み)
210*	In Progress(進行中)
220	Successful Install(インストール完了)
230	Install Failed(インストール失敗)
240*	Install Running(インストール実行中)
250*	ERP 9.0 Running(ERP 8.0 実行中)
260*	Listener Not Started/Installed(Listener が起動/インストールされていない)
270	General Error(一般エラー)
280	Already Installed(インストール済み)
290	Invalid Package(無効パッケージ)
300	Install Attempted(インストール試行中)
310	Machine Down(マシンがダウン中)

CD からの ERP 9.0 ワークステーションのインストール

システムに CD ライターがインストールされている場合、パッケージをビルドして、CD ライター・ロケーションに配布できます。パッケージを CD にコピーすると、その CD は、ワークステーション・インストールを実行するための携帯型デプロイメント階層として使用できます。つまり、ERP 9.0 の〈Workstation Installation〉プログラムを起動する〈setup.exe〉プログラムを CD から実行できるようになります。

パッケージを CD ライターに配布し、ERP 9.0 を CD からインストールできるように社内の環境を設定できます。

CD ライター・ロケーションの定義

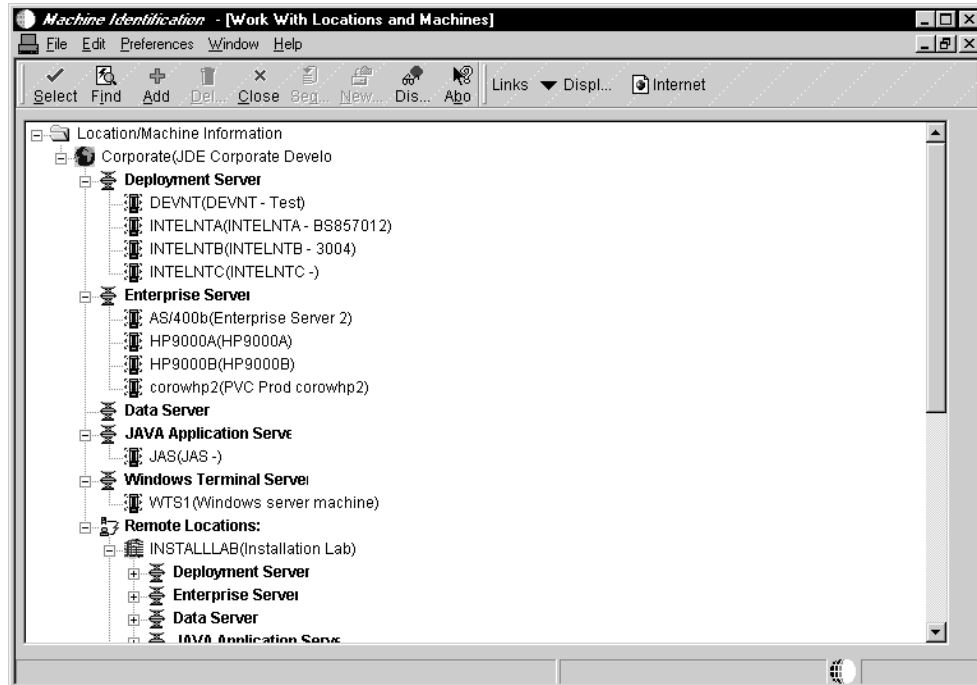
CD からのデプロイメントを実行するためのシステム構成プロセスの最初のステップは、CD ライター・ロケーションが未定義の場合に、これを定義することです。このステップでは、CD ライターのソフトウェアを使用して、後で CD にパッケージ・データをコピーする擬似デプロイメント・サーバーを作成します。

〈Machine Identification〉アプリケーションで CD ライター・ロケーションを定義する時は、[Environments] エグジットへの正確なパスコードを追加する必要があります。

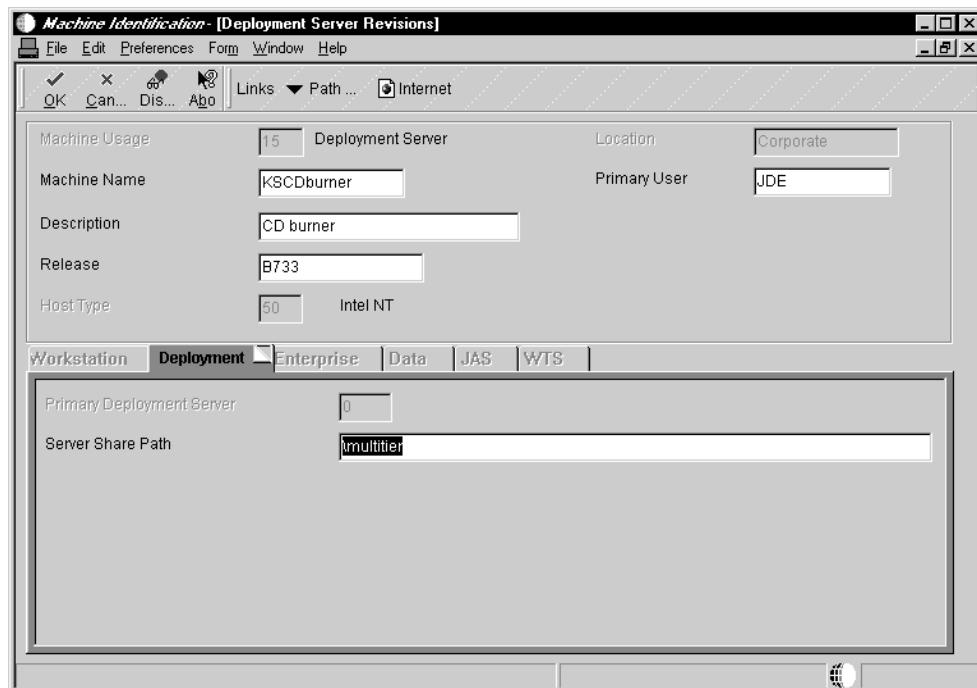
このロケーションを定義するプロセスは、他の新規デプロイメント・サーバーを定義するプロセスと同じです。

▶ CDライター・ロケーションを定義するには

〈Package and Deployment Tools〉メニュー(GH9083)から〈Machine Identification〉を選択します。



1. 適切なロケーションの下で、[Deployment]サーバーを選択します。
2. 新規マシンを追加するには、[Add]をクリックします。



3. 〈Deployment Server Revisions〉フォームで、次のフィールドに値を入力します。

- マシン名
 - 記述
 - リリース No.
 - 基本ユーザー
 - サーバー共有パス
4. データ、ファンデーションまたはヘルプ・ファイルのロケーションを指定する場合は、[Form]メニューから[Data(データ)]、[Foundation(ファンデーション)]または[Helps(ヘルプ)]を選択します。
 データ、ファンデーション、ヘルプの各ロケーションを指定しない場合は、デフォルト・ロケーションが使用されます。
 5. [OK]をクリックして、〈Work With Locations and Machines〉フォームに戻ります。
 6. [Close]をクリックして、〈Work With Locations and Machines〉フォームを終了します。
 7. Windows エクスプローラで、[ERP 9.0 Client Install(ERP 8.0 クライアント・インストール)]という名前のフォルダを検索します。
 8. このフォルダを CD ライター・ロケーションにドラッグしてコピーします。
 このロケーションは、〈Deployment Server Revisions〉フォームで入力したサーバー共有パスです。

フィールド記述

記述	用語解説
マシン名	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。
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。
基本ユーザー	リスト済みマシンの基本ユーザー
サーバー共有パス	World の場合: サーバー共有パス・フィールドは、環境が現サーバーのロケーションを決定するのに使用されます。 J.D. Edwards ERP の場合: このフィールドはこのパス・コードへの共有ディレクトリを示します。 ファイルサーバー上に保管されているオブジェクトは、このパス上にあります。

CDライター・ロケーションへのパッケージの配布

CDライターをデプロイメント・サーバーとして定義すると、指定した CDライター・ロケーションにパッケージを配布する準備が整います。このタスクには、次の2つの手順が伴います。

- CDに書き込むパッケージをCDライター・ロケーションに配布する。
- パッケージの内容をCDに書き込むには、Install.inf および Package.inf ファイルを修正します。

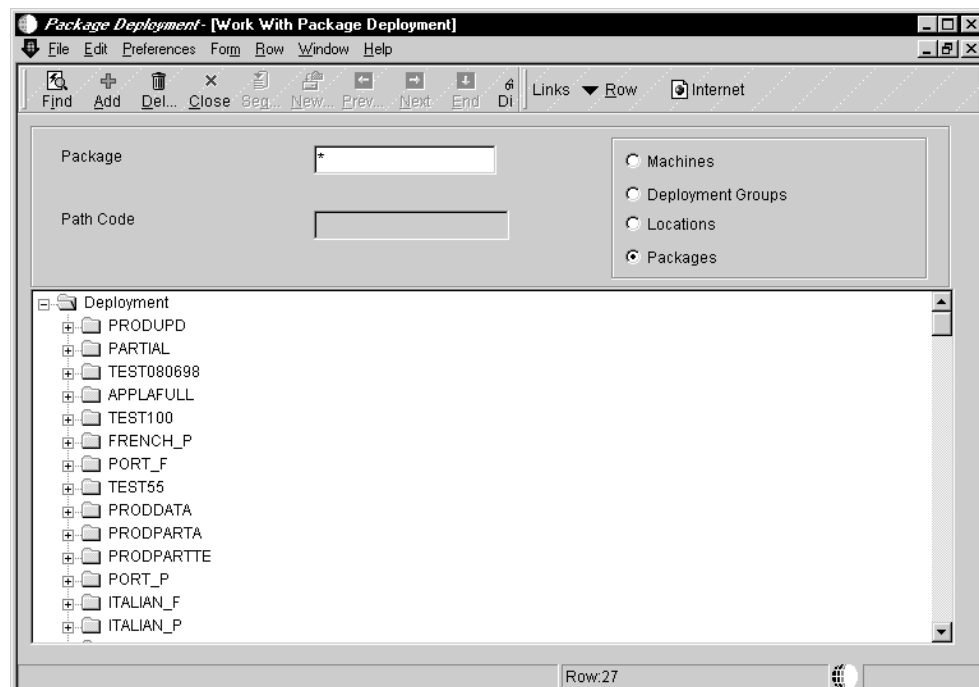
はじめる前に

- CDに書き込むパッケージを組み立て、定義し、ビルドします。

▶ パッケージをCDライター・ロケーションに配布するには

〈Package and Deployment Tools〉メニュー(GH9083)から〈Package Deployment〉プログラム(P9631)を選択します。

〈Work With Package Deployment〉フォームが表示されます。



1. 〈Work with Package Deployment〉フォームで、[Add]をクリックします。〈Package Deployment Director〉が起動します。
2. 他のパッケージの場合と同様に、〈Deployment Director(デプロイメント・ディレクタ)〉の各フォームで値を入力します。
3. 〈Work with Package Deployment〉フォームから、デプロイメントをスケジュールしたばかりのパッケージを検索して選択し、[Row]メニューから[Deploy]を選択して、パッケージを配布します。

パッケージをCDライター・ロケーションに配布した後の、そのロケーションのディレクトリ構造の一例を次に示します。

```
Multitier¥package_inf¥Appl_B.inf
Multitier¥systemcomp¥system.cab
Multitier¥datacomp¥data.cab
Multitier¥helpscomp¥helps.cab
Multitier¥Appl_pg¥Package¥Appl_B
Multitier¥package_inf¥Appl_B.inf
```

この例で、「Multitier」はサーバー共有パス名、「Appl_B」はパッケージ名です。

注:

フォルダを CD にコピーした場合、サーバー共有パス名は含まれません。先の例で、CD にコピーされた項目は、¥package_inf¥Appl_B.inf、¥systemcomp¥system.cab などです。

► **Install.inf および Package.inf ファイルを修正するには**

1. Windows エクスプローラで、CD ライター・ロケーションを検索し、配布したパッケージを含むフォルダを開きます。

このフォルダには、CD ライター・ロケーションを定義した時に、〈Deployment Server Revisions〉フォームの[Server Share Path]フィールドに入力した名前が設定されています。先の例では、サーバー共有パス名は「Multitier」です。
2. [ERP 9.0 Client Installation]フォルダを開き、Install.inf ファイルを開きます。

つまり、そのファイルのアイコンをダブルクリックして、〈Microsoft Notepad (Microsoft メモ帳)〉アプリケーションを起動します。
3. [FileLocations]セクションの行を修正します。2 つのピリオドとバックスラッシュ(¥)を「package_inf」エントリの前に配置します。修正後の行は、次のようになります。

```
[FileLocations]
PackageInfs=..\package_inf
```

[FileLocations]

PackageInfs=..¥package_inf

4. 同様に、[Package_inf]フォルダを開き、package name.inf ファイルを開きます。

先の例では、このファイル名は「Appl_b.inf」です。
5. [SrcDirs]セクションで各行を修正します。2 つのピリオドとバックスラッシュ(¥)が各エントリの前に配置します。

修正後の[SrcDirs]セクションの一例を次に示します。

```
[SrcDirs]
SAPPL_PGFS=..¥APPL_PGFS¥package¥APPL_B
SSYS=..¥systemcomp
SAPPL_PGFDATA=..¥datacomp
SHELP=..¥helpscomp
```

ERP 9.0 インストール CD の作成

パッケージを CD ライター・ロケーションに配布し、Install.inf および Package.inf ファイルを修正した後は、パッケージの内容を CD にコピーする準備が整います。このプロセスは、CD ライターに添付されるソフトウェアを使用して実行し、通常は、パッケージの内容を CD にコピーすることも含まれます。このプロセスの詳細については、CD ライターに添付されるドキュメンテーションを参照してください。

パッケージを CD にコピーするには、サーバー共有パス・ディレクトリの下位にあるサブディレクトリをピーします。サーバー共有パス・ディレクトリは CD には作成されません（先の例では、サーバー共有パス・ディレクトリは「Multitier」という名前で、これは、〈Deployment Server Revisions〉フォームの [Server Share Path] フィールドに入力した名前と同じです）。

ディレクトリを CD にコピーした後、CD には次のディレクトリが含まれます。

- Appl.pgf (パッケージ情報が含まれます)
- datacomp (データベース・キャビネット・ファイルが含まれます)
- helpscomp (ヘルプ・キャビネット・ファイルが含まれます)
- systemcomp (ファンデーション・キャビネット・ファイルが含まれます)
- package_inf (Package.inf ファイルが含まれます)
- ERP 9.0 Client Install (〈Workstation Installation〉プログラムが含まれます)

注:

システムによっては、実際の名前が上記の名前と異なる場合があります。

部分パッケージ・テンプレートの配布

ラップトップ・コンピュータを使用している ERP 9.0 ユーザーの場合、フル・パッケージでは、インストールに必要なディスク領域が大きすぎることがあります。また、これらのユーザーの多くが、フル・パッケージに含まれている ERP 9.0 のすべてのアプリケーションを使用するわけではありません。このようなユーザーに対応するため、システム管理者がラップトップ・コンピュータ・ユーザー専用の部分パッケージ・テンプレートを作成して配布できるようになっています。部分パッケージ・テンプレートは、標準の ERP 9.0 部分パッケージに基づいて作成されており、ラップトップ・コンピュータ・ユーザーが特定のプログラムを実行するのに必要なオブジェクトも含んでいます。

各企業によって状況もニーズも異なるため、管理者が部分パッケージ・テンプレートに収める実際のコンポーネントは様々ではありません。一例として、ここでは、〈Sales Force Automation (セールス・フォース・オートメーション)〉プログラムの部分パッケージ・テンプレートを作成します。

部分パッケージ・テンプレートの作成と配布には、他の部分パッケージの作成に使用するのと同じ〈Package Assembly Director〉プログラム(P9601)、〈Package Build Director〉プログラム(P9621)、および〈Deployment Director〉プログラム(P9631)を使用できます。

部分パッケージ・テンプレートの組立て

部分パッケージ・テンプレートを作成するには、まず〈Package Assembly〉プログラム(P9601)を使用して部分パッケージを組み立てます。プロセスは、他の部分パッケージを組み立てる場合と同じです。パッケージを組み立てる時は、ユーザーが部分パッケージ・テンプレートをロードした後で、プログラムを実行するのに必要なオブジェクト・コンポーネントを必ず追加してください。

▶ 部分パッケージ・テンプレートを組み立てるには

〈Package and Deployment Tools〉メニュー(GH9083)から〈Package Assembly〉プログラム(P9601)を選択します。

1. 新規パッケージを組み立てるには[Add]をクリックします。

〈Package Assembly Director〉が開きます。

2. 〈Welcome〉フォームで[Next]をクリックします。

〈Package Information (パッケージ情報)〉フォームが表示されます。

3. 次のフィールドに値を入力して、[Next]をクリックします。

- パッケージ名
- 記述
- パス・コード

〈Package Type Selection (パッケージ・タイプの選択)〉フォームが表示されます。

4. [Partial] ボタンをクリックすることにより、部分パッケージを組み立てることを指示します。

5. [Next] をクリックします。

〈Foundation Component (ファンデーション・コンポーネント)〉フォームが表示されます。

6. 〈Foundation Component〉フォームと、それに続く2つのフォーム(〈Help Component (ヘルプ・コンポーネント)〉フォームと〈Database Component (データベース・コンポーネント)〉フォーム)で、デフォルト・ロケーションを受け入れる場合は、[Next]をクリックし、使用するファンデーション、ヘルプ・ファイル、またはデータベース用の別のロケーションを指定する場合は、[Browse (ブラウズ)]をクリックします。

7. [Next] をクリックします。〈Object Component (オブジェクト・コンポーネント)〉フォームが表示されます。このフォームでは、パッケージに含めるオブジェクトを個別に指定できます。

8. [Browse] をクリックして、〈Object Component Selection (オブジェクト・コンポーネントの選択)〉フォームを表示します。

このフォームは、パッケージに含めるオブジェクトを検索および選択するために使用します。これらは、作成しているパッケージ・テンプレートのプログラムをユーザーが実行するのに必要なコンポーネントです。

指定したオブジェクトは、すべての部分パッケージに自動的に含まれるオブジェクトに追加されます。部分パッケージには、次のオブジェクトを含める必要があります。

- 対話型およびバッチ・アプリケーション (APPL および UBE オブジェクト)

- データ構造体(DSTR オブジェクト)
- ビジネス・ビュー(BSVW オブジェクト)

たとえば、〈Sales Force Automation〉アプリケーション用のパッケージ・テンプレートを作成する場合、APPL(対話型アプリケーション)、DSTR(データ構造体)、BSVW(ビジネス・ビュー)の各オブジェクト・タイプを使用して、次のシステム・コードを含める必要があります。

- 00 基本環境
- 01 住所録
- 17 CSM
- 32 コンフィギュレーション管理
- 40 在庫/OP 基本
- 41 在庫管理
- 42 受注管理
- 45 上級価格設定

9. オブジェクトの追加を終了したら、[Close]をクリックして、〈Object Component〉フォームに戻ります。

10. [Next]をクリックして、〈Language Component(言語コンポーネント)〉フォームを表示します。

このフォームでは、英語以外の言語をパッケージ言語スペックに追加できます。

11. [Next]をクリックして、〈Package Component Revisions(パッケージコンポーネントの改訂)〉フォームを表示します。

このフォームでは、現在のファンデーション、ヘルプ、データベースの各ロケーションと共に、パッケージ内のオブジェクトと言語選択を一覧できます。

12. これらのパッケージ・コンポーネントを変更するには、変更するコンポーネントのアイコンをクリックします。

該当するパッケージ・コンポーネントのフォームが表示されます。

13. パッケージの組立てが完了したら、[End]をクリックして、〈Package Assembly Director〉を終了します。

14. 〈Work with Packages〉フォームから作成したパッケージを選択し、[Row]メニューから[Active/Inactive]を選択してアクティブ化します。

参照

- データベース・ロケーションの入力については「データベース・ロケーションの入力」
- ヘルプ・ロケーションの入力については「ヘルプ・ファイル・ロケーションの入力」を参照してください。
- ファンデーションの入力の詳細については、「ファンデーション・ロケーションの入力」を参照してください。

部分パッケージ・テンプレートのビルド

部分パッケージの組立てが完了したら、〈Package Build Definition Director〉を使用してパッケージ・ビルドを定義できます。

▶ 部分パッケージ・テンプレートをビルドするには

〈Package and Deployment Tools〉メニュー(GH9083)から〈Package Assembly〉を選択します。

1. 〈Work With Packages〉フォームで、組み立てたばかりの部分パッケージを選択し、[Row]メニューから[Build Director(ディレクトリの作成)]を選択します。

〈Package Build Definition Director〉フォームが表示されます。

パッケージ定義が「In Definition(定義中)」の場合、パッケージをビルドする前にその状況を「Definition Complete(定義完了)」に変更する必要があります。

パッケージの状況を変更するには、パッケージをハイライトして[Row]メニューから[Activate]を選択します。

2. [Next]をクリックします。

〈Package Build Location(パッケージ・ビルドのロケーション)〉フォームが表示されます。部分パッケージをビルドしているので、サーバー用のパッケージはビルドできません。

3. ワークステーション用のパッケージをビルドするには、[Client(クライアント)]オプションをクリックして、[Next]をクリックします。

〈Build Specification Options(ビルド指定オプション)〉フォームが表示されます。

4. [Build Specification Options(ビルド・スペック・オプション)]オプションをクリックしてオンにし、[Next]をクリックします。

〈Business Function Options(ビジネス関数オプション)〉フォームが表示されます。最近、フル・パッケージをビルドした場合、[Build Functions Options(ビルド機能オプション)]をオンにする必要はありません。

5. 〈Business Function Options〉フォームで、[Next]をクリックします。

〈Compression Options(圧縮オプション)〉フォームが表示されます。

6. [Compress Options(圧縮オプション)]オプションをクリックしてオンにし、[Next]をクリックします。

〈Package Build Revisions(パッケージ・ビルドの改訂)〉フォームが表示されます。このフォームでは、パッケージ用に指定した現在のビルド・オプション、ビジネス関数オプション、および圧縮オプションを一覧できます。

7. これらのオプションを変更するには、変更するオプションのタイプに対応するタブをクリックします。

8. ビルド・オプションの検討および変更が完了したら、[End]をクリックして、〈Package Build Definition Director〉を終了します。

〈Work With Package Build Definition(パッケージ・ビルド定義の処理)〉フォームが表示されます。

9. 部分パッケージを選択し、[Row]メニューから[Active/Inactive]を選択します。
10. パッケージ・ビルドを開始する準備が整ったら、[Row]メニューから[Submit Build(ビルドの投入)]を選択します。
11. ビルド・レポートの出力先として次のいずれかを選択して、[OK]をクリックします。
 - On Screen
 - To Printer

フォームが自動的に閉じて、パッケージのビルドが行われます。ビルドの時間は、パッケージに含まれる項目数とサイズによって異なります。ビルドが終了すると、指定した出力先に応じて、レポートが画面に表示されるか、印刷されます。

部分パッケージ・テンプレートの配布

部分パッケージ・テンプレートを作成した後は、ラップトップ・ユーザー全員がパッケージを使用可能にする前に、パッケージをワークステーションに配布してテストします。部分パッケージ・テンプレートを配布する手順は、他のパッケージを配布する場合と同じです。

▶ 部分パッケージ・テンプレートを配布するには

〈Package and Deployment Tools〉メニュー(GH9083)から〈Package Deployment〉を選択します。

1. [Packages]オプションをクリックし、[Find]をクリックして、部分パッケージ・テンプレートを検索します。
2. 配布するパッケージを選択し、[Row]メニューから[Active/Inactive]を選択して、パッケージをアクティブ化します。
3. [Add]をクリックして〈Package Deployment Director〉を起動します。

〈Package Deployment Director〉フォームが表示されます。

4. [Next]をクリックします。

〈Package Selection〉フォームが表示されます。

5. 配布するパッケージを選択し、〈Package Deployment Director〉の残りのフォームに値を入力します。

これらのフォームでは次のことができます。

- 配布するパッケージを指定する。
- 必須インストール、プッシュ・インストール、配布日時などの属性を入力する。
- パッケージを受け取るワークステーションを指定する。

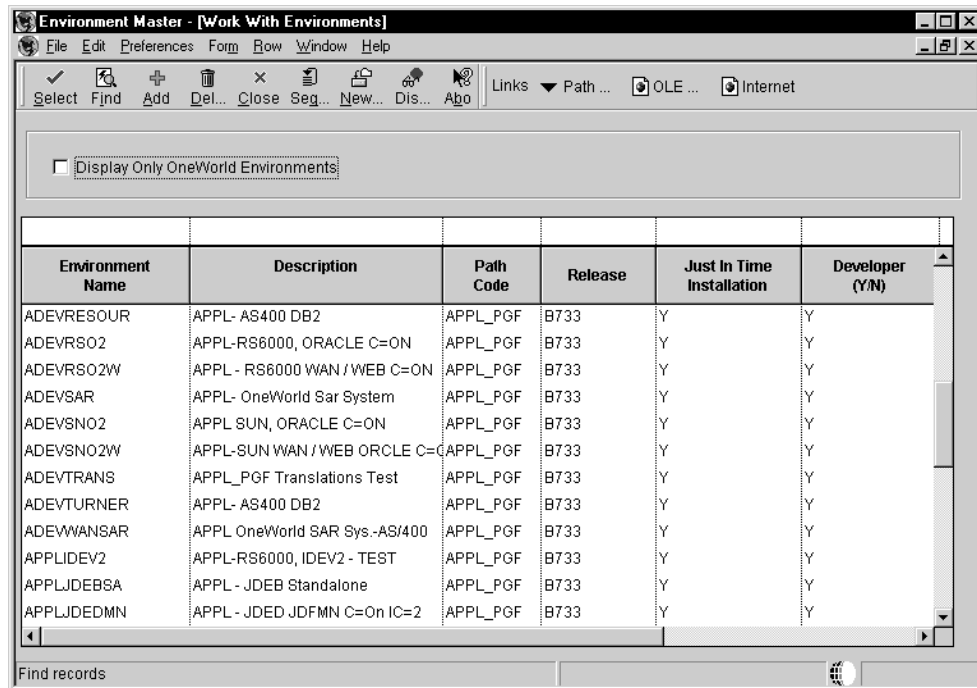
パッケージ・デプロイメントをスケジュールすると、指定した日時にパッケージがワークステーションが配布されます。ユーザーが ERP 9.0 にサインオンすると、このパッケージが使用可能になります。

部分パッケージ・テンプレートのテスト

部分パッケージ・テンプレートをワークステーション(ラップトップ)にインストールした後は、パッケージに含めたプログラムを実行できるかどうかテストします。このテストでは、ラップトップをネットワークから切断している時にプログラムを実行でき、ジャストインタイム・インストールによってオブジェクトをラップトップにインストールする必要がないことを確認します。

▶ 部分パッケージ・テンプレートをテストするには

1. ラップトップがネットワークに接続されていることを確認します。
2. ERP 9.0 を起動してサインオンします。
3. 〈Environments(環境)〉メニュー(GH9053)から〈Environment Master(環境設定マスター)〉を選択します。



The screenshot shows the 'Environment Master - [Work With Environments]' window. It has a menu bar (File, Edit, Preferences, Form, Row, Window, Help) and a toolbar with icons for Select, Find, Add, Del..., Close, Seg..., New..., Dis..., and Ago. Below the toolbar is a checkbox labeled 'Display Only OneWorld Environments'. The main area contains a table with the following data:

Environment Name	Description	Path Code	Release	Just In Time Installation	Developer (Y/N)
ADEVRESOUR	APPL- AS400 DB2	APPL_PGF	B733	Y	Y
ADEVRSO2	APPL-RS6000, ORACLE C=ON	APPL_PGF	B733	Y	Y
ADEVRSO2W	APPL - RS6000 WAN / WEB C=ON	APPL_PGF	B733	Y	Y
ADEV SAR	APPL- OneWorld Sar System	APPL_PGF	B733	Y	Y
ADEV SNO2	APPL SUN, ORACLE C=ON	APPL_PGF	B733	Y	Y
ADEV SNO2W	APPL-SUN WAN / WEB ORACLE C=ON	APPL_PGF	B733	Y	Y
ADEVTRANS	APPL_PGF Translations Test	APPL_PGF	B733	Y	Y
ADEVTURNER	APPL- AS400 DB2	APPL_PGF	B733	Y	Y
ADEWANSAR	APPL OneWorld SAR Sys.-AS/400	APPL_PGF	B733	Y	Y
APPLDEV2	APPL-RS6000, IDEV2 - TEST	APPL_PGF	B733	Y	Y
APPLJDEBSA	APPL - JDEB Standalone	APPL_PGF	B733	Y	Y
APPLJDEDMN	APPL - JDED JDFMN C=On IC=2	APPL_PGF	B733	Y	Y

At the bottom of the window, there is a 'Find records' field and a search icon.

4. 部分パッケージ・テンプレートを作成した時の環境を検索して選択します。

Environment Master - [Environment Revisions]

File Edit Preferences Form Window Help

OK Cancel Dismiss Abort Links Copy ... OLE ... Internet

Environment Name: P7332SNO2

Description: P7332 SUN Oracle C=On

Path Code: P7332

Release: B733

Just In Time Installation: N

Developer (Y/N): Y

WAN Configured Environment: ☐

B73.3 OneWorld

Save record

5. [Just In Time Installation (ジャストインタイム・インストール)] フィールドが「N」に設定されていることを確認します。

このフィールドを「N」に設定すると、ジャストインタイム・インストールがオフになります。ジャストインタイム・インストールは、ラップトップがネットワークに接続されていない時にオフになる必要があります。

6. ラップトップをネットワークから切断し、アプリケーションを実行します。

プログラムは、ラップトップがサーバーから切断されている時に実行できなければなりません。プログラムに必要なオブジェクトをパッケージに組み込んでいない場合、使用している環境ではジャストインタイム・インストールが使用できないという主旨のエラー・メッセージが表示されます。

このメッセージが出力された場合、組み込まれていないオブジェクトを確認して、パッケージを新たにビルドする必要があります。そうしないと、サーバーから切断されている時にラップトップで ERP 9.0 を使用できなくなります。

複数階層デプロイメントの用語

次の表では、複数階層デプロイメントの用語を説明します。

基本デプロイメント・ロケーション(第 1 階層)	基本デプロイメント・ロケーションは、補助または遠隔ロケーションに配布されるパッケージがビルドされるロケーションです。ソフトウェアのインストールおよび保守管理用に最低 1 台のデプロイメント・サーバーが必要です。基本デプロイメント・ロケーションにあるサーバーは、ERP 9.0 の配布および運用のための専用サーバーとし、それ以外の目的では使用しないでください。このマシンに必要なハードウェアとソフトウェアの詳細については、『インストール・ガイド』を参照してください。
階層デプロイメント・ロケーション(第 2 階層)	階層デプロイメント・ロケーション(遠隔地または補助ロケーションとも呼ばれます)には、1 台以上のデプロイメント・サーバーがあります。これにより、ERP 9.0 ソフトウェアをそのロケーションにあるワークステーションにインストールできます。これらのサーバーは、基本ロケーションにあるデプロイメント・サーバーからパッケージを受け取ります。階層デプロイメント・ロケーションにあるマシンは、オブジェクトのチェックインやチェックアウトなどのオブジェクト管理ワークベンチの機能を使用できません。これらのマシンはデプロイメント専用で設計されており、遠隔開発用には設計されていません。設定できる階層ロケーションの数は、ネットワークとサーバーのキャパシティによって異なります。
階層ワークステーション	階層ワークステーションは、ソフトウェアをインストールするために階層デプロイメント・ロケーションに接続するワークステーションです。各デプロイメント・ロケーションのワークステーションの数は、ネットワークおよびマシンの負荷によって異なります。すべての階層ワークステーションは、Windows NT ドメインに接続されている必要があります。ドメインに接続すると、階層ワークステーションは、各デプロイメント・ロケーションの共有ドライブからファイルの接続、読取りおよびコピーをすることが可能になります。

参照

- これらのマシンに必要なハードウェアとソフトウェアについては、『インストール・ガイド』

パッケージ・ビルド時間の比較

一般的なフル・パッケージ、部分パッケージ、少数のオブジェクトを持つ部分パッケージ、例として使用した Sales Force Automation 部分パッケージのそれぞれのパッケージ・ビルド時間と圧縮時間の比較対照表を以下に示します。

パッケージ・タイプ	パッケージ・サイズ	ビルド時間	圧縮時間	合計ビルド時間
フル・パッケージ	1.25 GB	8～9 時間	NA	8～9 時間
部分パッケージ	137 MB	25 分	13 分	38 分
部分、1447 個の A/R オブジェクト	183 MB	30 分	15 分	45 分
部分、1307 個の A/P オブジェクト	190 MB	33 分	16 分	49 分
部分、2676 個の SFA オブジェクト	293 MB	58 分	20 分	78 分

複数階層デプロイメント

ERP 9.0 ソフトウェアは、通常、中央にあるロケーションからワークステーションに配布されます。多くの場合、ソフトウェア・インストールでは計画的なスケジュールを使用することにより、単一のデプロイメント・サーバー上のパフォーマンスの影響を最小限にすることができます。たとえば、サイトでパッケージ・インストールを必要とするワークステーションが 50 以上あり、ソフトウェア・リリースが年 4 回のみの場合は、インストールを効率的に行うために、夜間や週末、オフピークの時間などにスケジュールすることができます。

この配布方法がソフトウェア・デプロイメントの最も単純なアプローチであるのに対して、ネットワーク能力は複数リモート・サイトまたは単一サイトでの大量のユーザーの構成に制約を与えます。たとえば、56KB の回線によって中央のデプロイメント・ロケーションに接続しているワークステーションへのソフトウェア・インストールの場合、所要時間は 4～6 時間となります。

複数階層デプロイメントは、各サイトで複数のデプロイメント・サーバーや複数のデプロイメント・ロケーションから、パッケージを個々のワークステーションやサーバー上に柔軟にインストールできるようにします。この追加のデプロイメント・ロケーションやサーバーは、デプロイメント階層と呼ばれています。特に、複数階層デプロイメントにより、WAN 回線で複数のワークステーションにインストールする代わりに、圧縮したパッケージを中央のロケーションから第 2 のデプロイメント階層とされるリモート・ワークグループ・サーバーに移行できるようにおこないます。そのため、複数階層デプロイメントは、複数のデプロイメント階層からデプロイメントすることを意味します。

たとえば、主なロケーションで 1 つのデプロイメント・サーバー、さらに遠隔地用に第 2 のデプロイメント・サーバーを持つことを想定してください。遠隔地のサーバーがそのロケーションにあるワークステーションやサーバーにパッケージを配布することになるので、単一階層のデプロイメント構成のように、メインのデプロイメント・サーバーから WAN でパッケージを配布する必要がなくなります。

ワークグループ・サーバーは、大量のワークステーションにソフトウェアを同時にインストールする必要のある LAN 環境で、第 2 階層のデプロイメント・ロケーションとして機能させることもできます。サイトで 1 日あたり 50 以上のワークステーションが ERP 9.0 ソフトウェアのインストールを行う場合には、この複数階層デプロイメントを導入することをお勧めします。

複数階層デプロイメントの理解

複数階層デプロイメントの主な機能は、複数の地理的ロケーションをもつ企業が、遠隔サイトにある第 2 デプロイメント・サーバーから配布できるようにすることにより、ネットワーク・トラフィック（および、トラフィックが多いことにより生じる遅延）を低減することです。つまり、WAN を使用してデプロイメント・サーバーから遠隔ロケーションにあるワークステーションにパッケージをインストールする代わりに、基本ロケーションにあるデプロイメント・サーバーから、遠隔ロケーションにあるデプロイメント・サーバーに、パッケージおよび package.inf ファイルをコピーできます。これで、遠隔ロケーションのサーバーは、そのロケーションにあるワークステーションおよびサーバーにパッケージを配布できます。

次のいずれかが当てはまる場合は、複数階層デプロイメント構成のインプリメントを検討してください。

- 同じロケーションからパッケージをインストールするワークステーションの数が多すぎて、サーバーとネットワークのパフォーマンスが低下する。
- ワークステーションが、WAN 上のデプロイメント・サーバーに遠隔接続されているため、インストールに非常に時間がかかる。

通常は、CNC インプリメンテーションの際に複数階層デプロイメントをインプリメントするかどうかを判断します。この機能はいつでも有効にできますが、通常は、ERP 9.0 をインストールして本稼働用サイトの準備が整った後にセットアップします。

複数階層デプロイメントをセットアップするには、デプロイメントに使用されるマシン(関連パス・コードも)を定義する必要があります。さらに、ソフトウェアが階層デプロイメント・ロケーションにプッシュされる日時を定義するには、スケジューラ機能を使用します。

複数階層デプロイメント用に個々のユーザーの特性を定義する必要もあります。通常、これは、ユーザー・プロファイルでそのユーザーがパッケージをプルするデプロイメント・ロケーションを修正して行います。

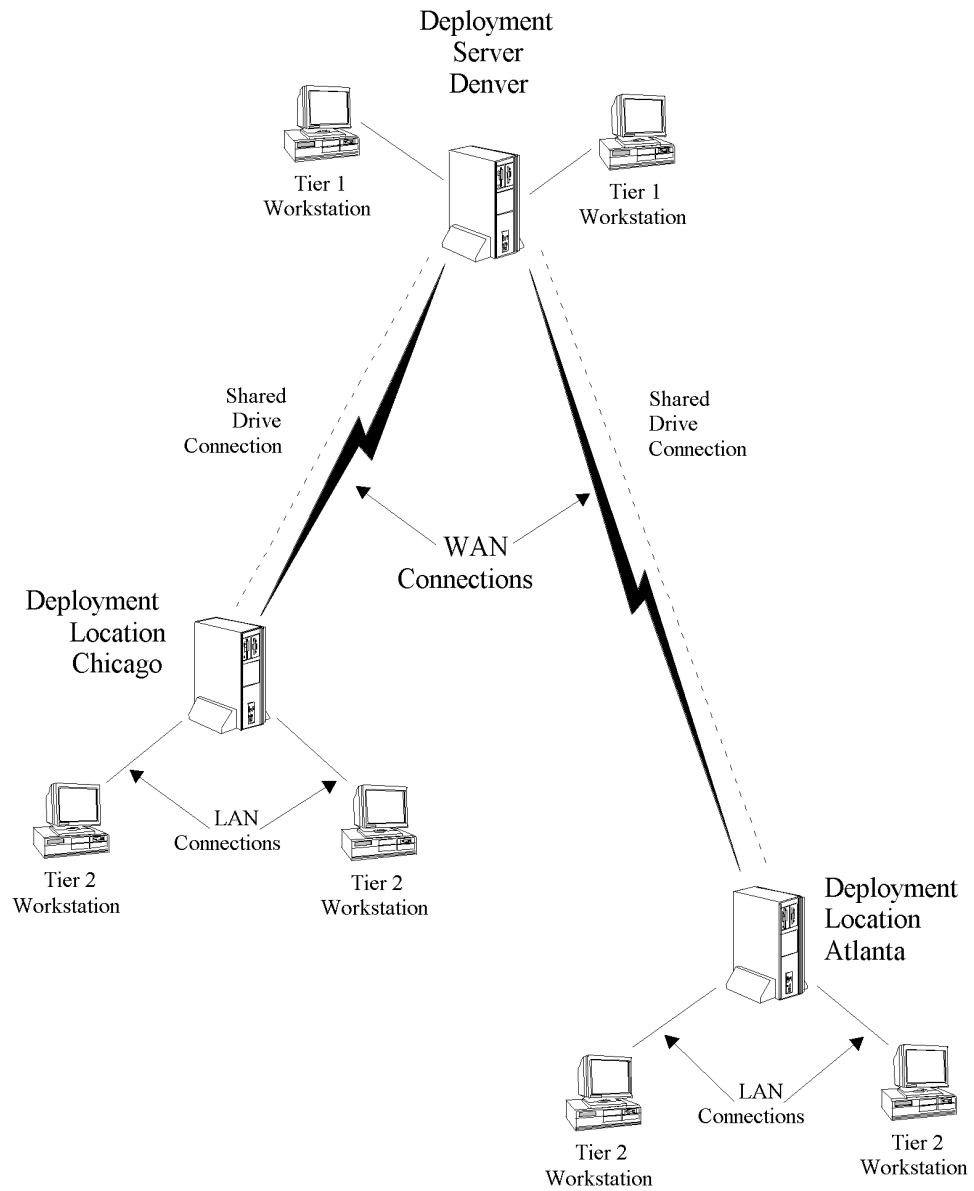
複数階層デプロイメントの機能

複数階層デプロイメントは次の機能を提供します。

- ワークステーションに何台のデプロイメント・サーバーからでも配布できます。
- デプロイメント・ロケーションを簡単に追加できる。デプロイメント・マシンはサーバーである必要はありません。Windows 95 または Windows NT ワークステーションなどを使用することができます。
- セットアップおよび管理が簡単に行えます。
- 遠隔デプロイメント・ロケーションに転送されるファイルおよびデータを集中管理できます。
- 遠隔サイトに配布するソフトウェアを簡単にスケジュールできます。
- 複数階層デプロイメントは、〈Package Deployment Director(パッケージ・デプロイメント・ディレクタ)〉に統合される。そのため、デプロイメント・プロセスは、単一階層セットアップでのデプロイメント・プロセスと本質的に同じです。

例:2 階層デプロイメント構成

次に一般的な2階層デプロイメント構成を示します。



複数階層のインプリメンテーション

パッケージは常に基本ロケーションにあるデプロイメント・サーバー上でビルドされます。遠隔ロケーションに配布するパッケージをビルドした後は、次のステップを実行して複数階層デプロイメントをインプリメントする必要があります。

▶ デプロイメント・ロケーションの定義

配布先の各物理ロケーションを定義する必要があります。たとえば、デンバーにある本店から、シアトルにある支店に配布する場合は、シアトルのデプロイメント・ロケーションを定義する必要があります。

▶ デプロイメント・サーバー定義の作成

各遠隔ロケーションでデプロイメント・サーバーを定義する必要もあります。

注:

ERP 9.0 のインストール時に、〈Remote Location Workbench(リモート・ロケーション・ワークベンチ)〉を使用してデプロイメント・サーバー定義を作成した場合は、このステップは必要ありません。

▶ パッケージのスケジュール設定

このプロセスの次のステップでは、〈Package Deployment(パッケージ・デプロイメント)〉プログラム(P9631)を使用して、パッケージ・デプロイメントをスケジュールします。複数階層デプロイメント用パッケージをスケジュールするプロセスは、その他のパッケージをスケジュールするプロセスと同じです。

▶ ワークステーションへのパッケージの配布

遠隔ロケーションにあるデプロイメント・サーバーにパッケージを配布した後、そのサーバーは、そのロケーションにあるワークステーションに配布できます。

はじめる前に

- パッケージ、ワークステーション・インストール、パス・コード、セントラル・オブジェクト、レプリケートされるオブジェクトおよびユーザー・プロファイルを定義しインプリメントできるように、CNC の概念を十分に理解しておいてください。
- 階層デプロイメント・ロケーションとして使用するマシン用のディスク・ドライブに十分なディスク領域があることを確認してください。なお、フル・パッケージを配布して追加するごとに約 1.4GB、部分パッケージを追加するごとに約 165MB のディスク領域が必要になります。J.D. Edwards サポート対象ローカル・データベースにレプリケートするデータ量によって、必要なディスク領域の大きさは異なります。

参照

- ロケーションの設定については「ロケーションの定義」
- デプロイメント・サーバーの設定については「デプロイメント・サーバーの定義」

- リモート・サーバーからワークステーションへの配布については、「デプロイメント・ロケーションへのソフトウェアの配布」
- デプロイメント・サーバーのディスク容量の最低要件については、『インストール・ガイド』の「ディスク容量の最低要件」
- 〈Remote Location Workbench〉については「インストール・ガイド」の「Installation Workbenchの実行」

デプロイメント・サーバーの定義

〈Machine Identification (マシン ID)〉アプリケーション(P9654A)を使用すると、新規のデプロイメント・サーバー定義を追加したり、既存の定義を修正することができます。新規のデプロイメント・サーバー定義を追加すると、ERP 9.0 は、デプロイメント・ロケーション定義テーブル(F9654)に各デプロイメント・ロケーションのレコードを作成します。このため、各デプロイメント・ロケーションにあるサーバーを定義する必要があります。

このテーブルには、通常、基本デプロイメント・サーバー用の 1 つのレコードと、各リリースのデプロイメント・ロケーションごとに 1 つのレコードがあります。ERP 9.0 の複数のリリースを使用している場合は、各デプロイメント・ロケーションにあるサーバー用の複数のレコードを作成する必要があります。

ERP 9.0 のインストール時に〈Remote Location Workbench〉を使用してデプロイメント・サーバー定義を作成した場合は、デプロイメント・サーバーを再度定義する必要はありません。

参照

- 〈Remote Location Workbench〉については「インストール・ガイド」の「Installation Workbenchの実行」

新規デプロイメント・サーバーの定義

次に、新規のデプロイメント・サーバーを定義する方法について説明します。たとえば、支店を別の都市に開設したことによって、そのロケーションにあるワークステーションへのデプロイメントを処理できる補助デプロイメント・サーバーを定義する必要がある場合は、新規デプロイメント・サーバーを定義する必要があります。

はじめる前に

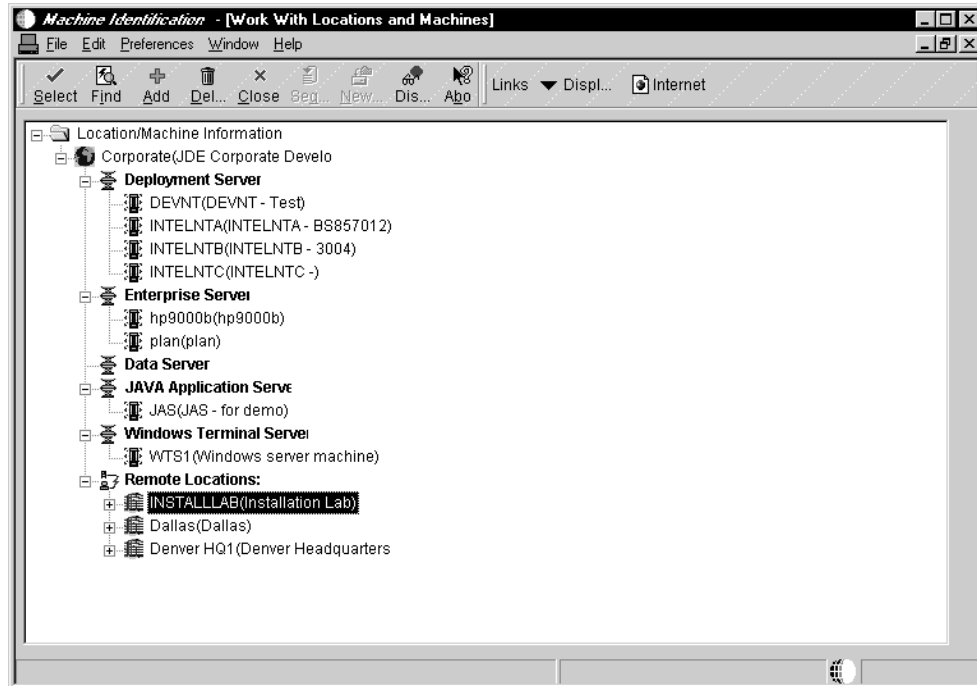
- 遠隔ロケーションにあるデプロイメント・サーバーの新規定義を追加する場合、最初にそのロケーションを定義する必要があります。ロケーションの定義については、「ロケーションの定義」を参照してください。

注:

デプロイメント・ロケーションは、DEV パス・コードを使って定義しないことをお勧めします。DEV パス・コードは、通常、リモート・ロケーションに配属されていない開発者に関連付けられるためです。

▶ 新規デプロイメント・サーバー定義を追加するには

〈Package and Deployment Tools〉メニュー(GH9083)から〈Machine Identification〉アプリケーション(P9654A)を選択します。



1. デプロイメント・サーバーが存在するロケーションを検索します。必要に応じて、異なるマシン・タイプを表示するには、ツリーを拡張します。
2. [Deployment Server]見出しを選んで[Add]をクリックします。
〈Deployment Server Revisions〉フォームが表示されます。

3. 〈Deployment Server Revisions〉フォームで、次のフィールドに値を入力します。

- マシン名
- 記述
- リリース No.
- 基本ユーザー
- サーバー共有パス

基本デプロイメント・サーバー以外のサーバーの定義を入力しているため、[Primary Deployment Server(基本デプロイメント・サーバー)]フィールドは使用できません。現行の基本デプロイメント・サーバー定義を最初に削除する場合のみ、このフィールドに値を入力できます。

4. [Form]メニューから[Path Code]を選択します。

〈Machine Path Code Revisions〉フォームが表示されます。

5. 〈Machine Path Code Revisions〉フォームで、補助ロケーションがパッケージを受け取る基本ロケーションのパス・コードを入力します。パス・コードを入力すると、サーバー共有パスは、そのマシンの基本サーバー共有パスにデフォルト設定されます。

重要

すべての補助ロケーションにあるデプロイメント・サーバーにパス・コードを指定する必要があります。パス・コードを指定しないと、複数階層デプロイメントが機能しないことがあります。

6. ユーザー定義ファンデーション、データまたはヘルプ・ファイルがパッケージに含まれている場合は、[Form]メニューの[Foundation]、[Data]または[Helps]を選択して、これらの項目を指定します。
7. 〈Deployment Server Revisions〉フォームで[OK]をクリックして、〈Work With Locations and Machines〉フォームに戻ります。

フィールド記述

記述	用語解説
マシン名	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。
記述	ユーザー定義名称または備考。
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。
基本ユーザー	リスト済みマシンの基本ユーザー

サーバー共有パス

World の場合:サーバー共有パス・フィールドは、環境が現サーバーのロケーションを決定するのに使用されます。

J.D. Edwards ERP の場合:このフィールドはこのパス・コードへの共有ディレクトリを示します。

ファイルサーバー上に保管されているオブジェクトは、このパス上にあります。

既存デプロイメント・サーバー定義の改訂

定義済みデプロイメント・サーバーの定義は、修正が必要になることがよくあります。たとえば、サーバー共有パスや ERP 9.0 リリースが変更された場合や、基本デプロイメント・サーバーとして別のサーバーを指定する場合は、定義を変更する必要があります。

既存のデプロイメント・サーバー定義の改訂手順は、新規定義の追加手順とよく似ています。

▶ 既存デプロイメント・サーバー定義を改訂するには

〈Package and Deployment Tools〉メニュー(GH9083)から〈Machine Identification〉アプリケーション(P9654A)を選択します。

1. 定義を修正するデプロイメント・サーバーを検索し、[Select(選択)]をクリックします。

〈Deployment Server Revisions〉フォームが表示されます。

2. 〈Deployment Server Revisions〉フォームでは、次のフィールドを修正できます。

- 記述
- リリース No.
- 基本ユーザー
- サーバー共有パス

基本デプロイメント・サーバーを修正する場合は、基本デプロイメント・サーバー・フィールドを変更することもできます。その他のサーバーでは、このフィールドの値は変更できません。このフィールドに"1"が入力されている場合は、デプロイメント・サーバーが基本デプロイメント・サーバーです。基本デプロイメント・サーバーは 1 台だけです。

3. [Form]メニューから[Path Code]を選択します。

〈Machine Path Code Revisions〉フォームが表示されます。

Path Code	Description	Server Share Path
APPL_PG	Applications Development	multitier
PROD	Production Environment	multitier

4. 〈Machine Path Code Revisions〉フォームで、補助ロケーションがパッケージを受け取る基本ロケーションのパス・コードを入力します。パス・コードを入力すると、デフォルトのサーバー共有パスが表示されます。これは、そのマシンの基本サーバー共有パスです。

重要

すべての補助ロケーションにあるデプロイメント・サーバーにパス・コードを指定する必要があります。パス・コードを指定しないと、複数階層デプロイメントが機能しない場合があります。

5. ユーザー定義ファンデーション、データまたはヘルプ・ファイルがパッケージに含まれている場合は、[Form]メニューの[Foundation]、[Data]または[Helps]を選択して、これらの項目を指定します。
6. 〈Deployment Server Revisions〉フォームで、[OK]をクリックします。

参照

- マシン、ロケーション、およびデプロイメント・グループの定義については、「マシン、ロケーション、およびデプロイメント・グループの定義」

デプロイメント・ロケーション用パッケージのスケジュール設定

基本デプロイメント・サーバーがデプロイメント・ロケーションに配布するパッケージを定義できます。また、パッケージを配布する日時および場所を定義することもできます。

階層デプロイメント・ロケーションからインストールするワークステーションは、ヘルプ・ファイルをそのワークステーションに配布しない場合は、ヘルプ・ファイルを使用する際に階層デプロイメント・サーバーにアクセスします。

デプロイメント・ロケーションを作成または修正した後、〈Package Deployment〉プログラム(P9631)を使用して、パッケージ・デプロイメントをスケジュールできます。補助デプロイメント・サーバーに対するパッケージをスケジュールするプロセスは、基本サーバーに対するパッケージをスケジュールするプロセスと同じです。パッケージをスケジュールする際、補助サーバー・ロケーションを必ず指定してください。

注:

配布スケジュールを作成するときは、複数のパッケージを同時に配布するようにスケジュールすることはできません。

はじめる前に

- 既存デプロイメント・ロケーションおよびデプロイメント・サーバー定義を作成または修正します。詳細については、「デプロイメント・サーバーの定義」および「マシンの定義」を参照してください。

デプロイメント・ロケーションへのソフトウェアの配布

ソフトウェアをデプロイメント・ロケーションに配布するには、〈Package Deployment〉プログラム(P9631)または〈Multitier Deployment (複数階層デプロイメント)〉バッチ・プログラム(R98825C)を使用します。スケジュール・パラメータを定義するために、またはパッケージをすぐに配布するために、〈Package Deployment〉プログラムを使用します。一方、デプロイメント・ロケーションにソフトウェアを配布するには、〈Multitier Deployment〉バッチ・アプリケーション・バージョンを使用します。

ソフトウェアをプッシュするか、プルするかは、スケジュール設定プログラムのデプロイメント機能またはバッチ・プログラムを実行するマシンによって決まります。基本デプロイメント・サーバーまたはワークステーションからプログラムやレポートを実行する場合は、ソフトウェアをプッシュします。逆に、階層デプロイメント・ロケーションにあるワークステーションからプログラムを実行する場合は、ソフトウェアをプルします。どちらにしても、プッシュ・インストールでは基本デプロイメント・サーバー・マシンで、プル・インストールでは配布先のロケーションでアプリケーションを実行します。

注意:

ソフトウェアをプッシュする場合は、バッチ・アプリケーションを実行しない場合でも、基本デプロイメント・サーバーと階層デプロイメント・サーバーへの完全な読取りおよび書込み権限が必要です。すべてのサーバーにおいて読取りおよび書込み権限をもたない場合、デプロイメントは失敗します。

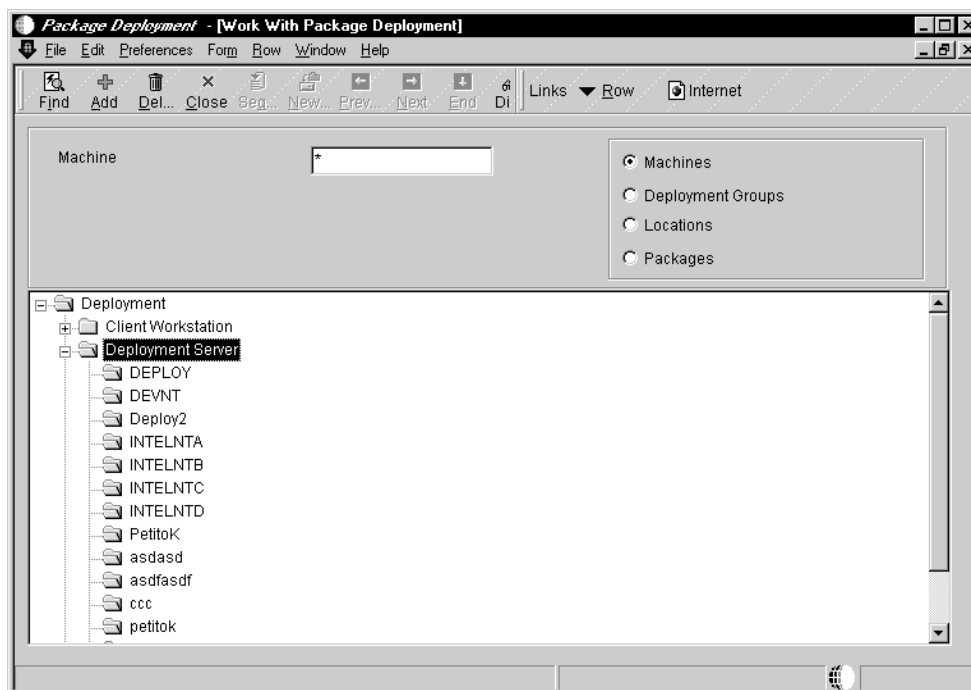
〈Package Deployment〉プログラムまたは〈Multitier Deployment〉バッチ・プログラムによってパッケージ・ソフトウェアが自動的に配布された後、基本デプロイメント・サーバーから階層デプロイメント・ロケーションにワークステーション・インストール・プログラムを手動でコピーする必要があります。これらのプログラムは、base installation ディレクトリの client 部分に置かれます。

Package Deployment を使用したソフトウェアの配布

この方法を使用するのは、デプロイメント・スケジュールを定義した直後にソフトウェアを配布する必要がある場合です。このオプションを選ぶと、スケジューリング・フィールドで指定したタイミング・パラメータに関係なく、ソフトウェアは即時に配布されます。

▶ 〈Package Deployment〉を使用してソフトウェアを配布するには

〈Package and Deployment Tools〉メニュー(GH9083)から〈Package Deployment〉を選択します。



1. 〈Work with Package Deployment〉フォームで、[Machines (マシン)] オプションをクリックし、[Find (検索)] をクリックします。
2. 配布するデプロイメント・サーバーを選択します。
複数のデプロイメント・サーバーに対してパッケージの配布をスケジュールしている場合は、[Deployment Server] フォルダを選択すると、各サーバーに個別に配布するのではなく、該当するすべてのサーバーに配布することができます。
3. サーバー名の下に表示されているすべてのパッケージを配布するには、デプロイメント・サーバー名を選択します。
または、配布するパッケージだけを選択します。
4. [Row (ロー)] メニューから [Deploy] を選択します。

このオプションを選択すると、〈Multitier Deployment〉バッチ・プログラム(R98825C)が起動し、デプロイメント・サーバーに配布できるようになります。

この[Deploy]オプションでは、エンタープライズ・サーバー・パッケージを配布するバッチ・アプリケーションを起動することもできます。そのため、〈Work With Package Deployment〉フォームでエンタープライズ・サーバー名か、[Enterprise Server(エンタープライズ・サーバー)]フォルダを選択した場合は、〈Multitier Deployment〉バッチ・プログラムではなく、〈Enterprise Server Deployment(エンタープライズ・サーバー・デプロイメント)〉バッチ・プログラム(R98825D)が起動します。ワークステーションを選択した場合、[Deploy]オプションは使用できません。

複数階層デプロイメントでのサーバー・パッケージの配布

複数階層デプロイメントによって配布するサーバー・パッケージを作成した後は、〈Package Deployment〉プログラム(P9631)を使用して、そのパッケージを同一ロケーションまたは遠隔地のサーバー宛てにスケジュールできます。サーバー・パッケージが圧縮されていることを確認します。クライアント・パッケージと違って、サーバー・パッケージは圧縮されていない限り複数階層デプロイメントによって配布できません。

また、サーバー複数階層デプロイメントでサーバー・パッケージを配布するためには、パッケージ状況が 50(ビルドが正常に終了している)または 70(ビルドがエラー付きで終了している)でなければなりません。パッケージの状況が 50 または 70 ではない場合は、配布をスケジュールする際にそのパッケージを使用できません。

複数階層デプロイメント用サーバー・パッケージをスケジュールするプロセスは、同一ロケーション内パッケージをスケジュールするプロセスと本質的に同じです。

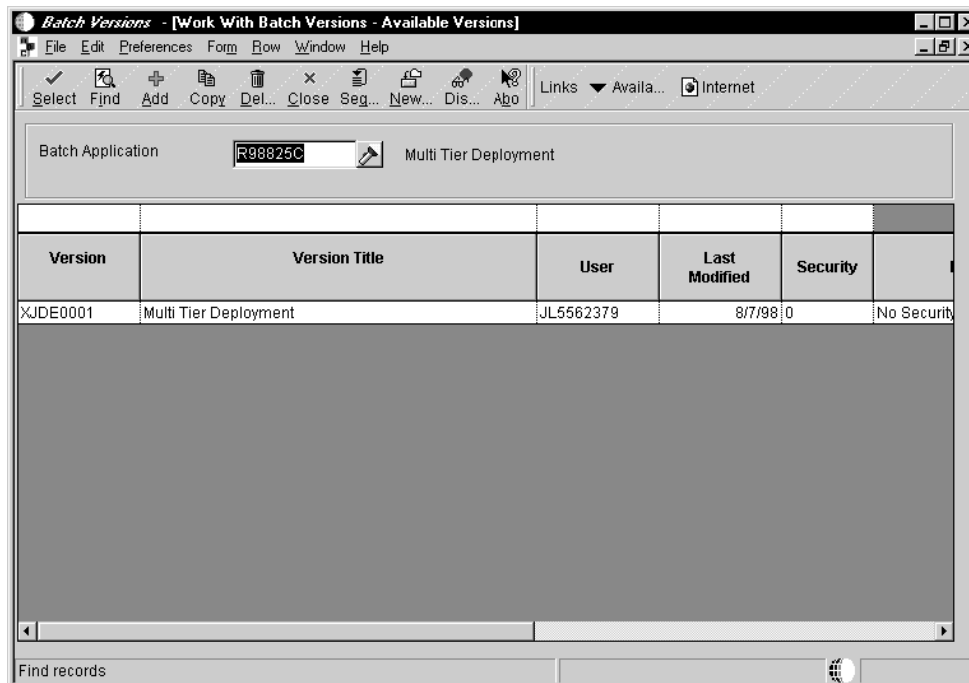
〈Multitier Deployment〉バッチ・プロセスを使用したソフトウェアの配布

前回〈Package Deployment〉プログラム(P9631)を使用して定義およびスケジュールしたソフトウェアを配布する場合、この方法を使用します。スケジュールされた日時をすでに過ぎている場合、バッチ・プログラムを実行することにより、ソフトウェアが配布されます。

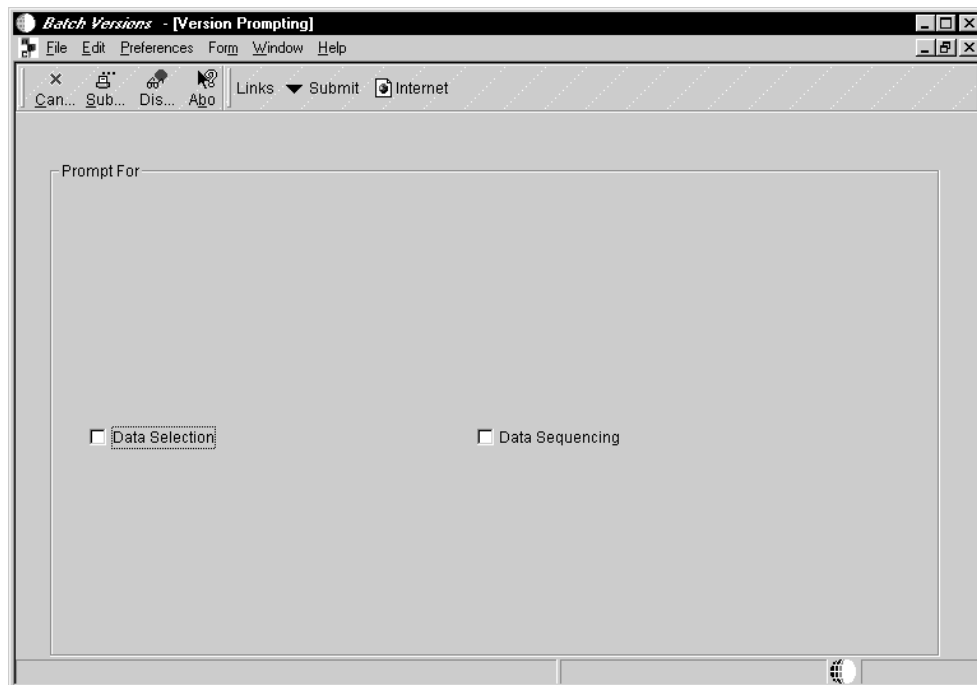
たとえば、レポートを 2005 年 6 月 1 日の 13:00 に実行するようにスケジュールした場合、2005 年 5 月 31 日の 13:00 にレポートを実行しても影響はありません。この例では、ソフトウェアを配布するために、2005 年 6 月 1 日の 13:00 以降にバッチ・プロセスを実行する必要があります。

▶ <Multitier Deployment>パッチ・アプリケーションを使用してソフトウェアを配布するには

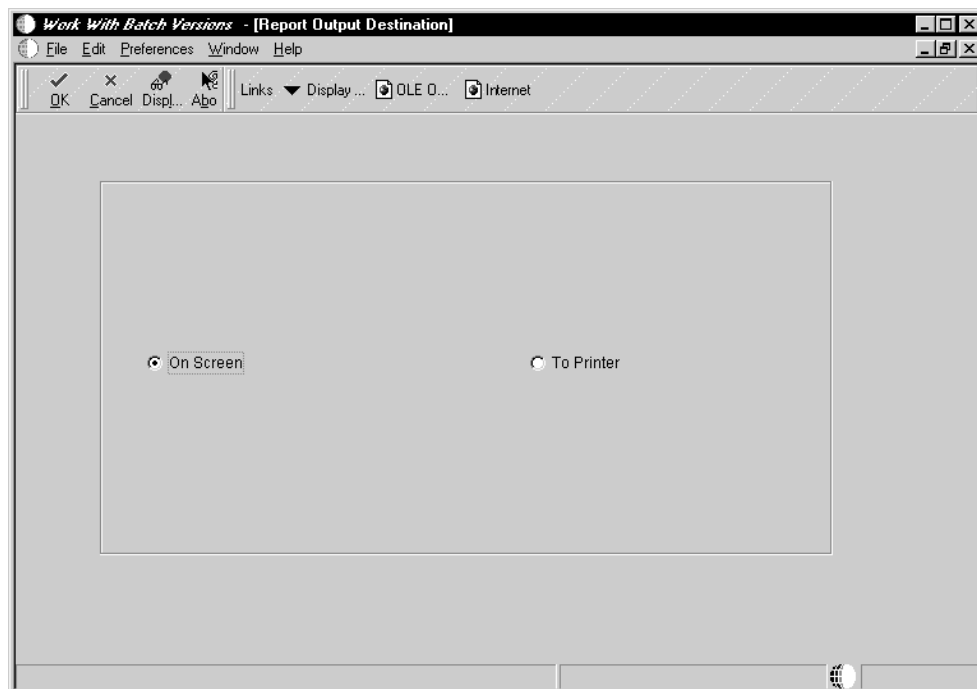
<System Administration Tools>メニュー(GH9011)から<Batch Versions (パッチ・バージョン)>を選びます。



1. 次のフィールドに入力します。
 - Batch Application (パッチ・アプリケーション)
R98825C と入力してください。
2. 使用する[Multi-tier Deployment (複数階層デプロイメント)]バージョンを見つけて選択します。
<Version Prompting>フォームが表示されます。



3. [Submit (投入)]をクリックします。
 〈Report Output Destination〉フォームが表示されます。



4. 〈Report Output Destinations〉で、次のいずれかのオプションをクリックします。
 - On Screen
 - Printer

出力は Adobe Acrobat Reader に送られます。この出力は、レポートが正常に終了したことを示します。

デプロイメント・ロケーションへの〈Workstation Installation〉プログラムのコピー

パッケージを配布するためにどの方法を使用するかにかかわらず、基本デプロイメント・サーバーから第 2 階層デプロイメント・ロケーションに〈Client Workstation Installation (クライアント・ワークステーション・インストール)〉プログラムを手作業でコピーする必要があります。遠隔デプロイメント・ロケーションのマシン名および環境を反映させるために、package.inf ファイルを変更する必要もあります。

これらのステップを実行すると、ERP 9.0 は、〈Client Workstation Installation〉プログラムをデプロイメント・ロケーションでローカルに実行します。これらの 2 つのステップを実行しない場合、ERP 9.0 は、基本ロケーションで〈Client Workstation Installation〉プログラムと関連ファイルを検索し、WAN を通じて、これらのファイルをデプロイメント・ロケーションにコピーします。

▶ デプロイメント・ロケーションに〈Workstation Installation〉プログラムをコピーするには

1. 各デプロイメント・ロケーションに接続します。続いて、次のクライアント・ディレクトリを第 2 階層ワークステーションにドラッグするために、Windows エクスプローラを使用します。

¥¥Tier1DeploymentServerName¥B9¥OneWorld Client Install

2. package.inf ファイルを開いて、[FileLocations] セクションを検索します。PackageInfs 行を変更し、デプロイメント・ロケーションのデプロイメント・サーバーのマシン名と環境を反映させます。次に例を示します。

PackageInfs=¥¥machine name¥environment¥ENVIRON¥Evapps¥appl_pg¥package_inf

3. [File (ファイル)] メニューから [Save (保存)] を選択して、変更内容を保存します。
4. [File] メニューから、[Exit (終了)] を選択します。

デプロイメント・ロケーションからのワークステーション・パッケージのインストール

デプロイメント・ロケーションからワークステーションにパッケージをインストールする手順は、通常のパッケージ・インストール手順とよく似ています。このプロセスは、基本的に setup.exe プログラムの実行で構成されます。

複数階層デプロイメントの場合、このプログラムは、階層デプロイメント・ロケーション上の base installation ディレクトリの下に client サブディレクトリに存在します。デプロイメント・ロケーションに接続されたワークステーションは、デプロイメント・ロケーション上のこのディレクトリへの読取りアクセス権が必要です。

サーバー・パッケージの複数階層デプロイメント

サーバー複数階層デプロイメントでは、1 台のデプロイメント・サーバーから別のデプロイメント・サーバーへ自動的にサーバー・パッケージを配布できます。パッケージ配布先のターゲット・デプロイメント・サーバーは、パッケージ送信元のソース・デプロイメント・サーバーと同じ場所に位置していても、1 つまたは複数のリモート・ロケーションに位置していてもかまいません。

パッケージは、一般に基本ロケーションのプライマリ・デプロイメント・サーバーでビルドしますが、基本ロケーションからのインストールでは、別のサーバーを使用するほうが有利な場合があります。また、リモート・ユーザーに WAN を介してサーバー・パッケージにアクセスさせるよりも、リモート・ロケーションにあるサーバーへサーバー・パッケージを配布した方が良い場合があります。

サーバー・パッケージ複数階層デプロイメントでは、配布するパッケージ・ビルドをユーザーが選択できます。たとえば、複数サーバー・プラットフォーム用の Prod パッケージをビルドしている場合に、ビルドに成功したプラットフォーム用のビルドを選択できます。ビルドを選択できる利点は、ユーザーが新しいクライアント・パッケージのインストールを待つ必要がないことです。

サーバーの複数階層デプロイメントでは、〈Package Deployment〉プログラム(P98631)を使用します。サーバー・パッケージを配布すると、次のコンポーネントがコピーされます。

- ファンデーション、データ、およびヘルプ
- パッケージ名ディレクトリの下にある次のサブディレクトリ
 - bin32
 - include
 - lib32
 - obj
 - source
 - spec
- サーバー・ビルド・ディレクトリの下にある次のサブディレクトリ
 - bin32
 - spec
 - obj
 - lib32
- 次のサーバー・ビルド・ディレクトリ内の ServerPackage.inf ファイル
- package.inf ファイル
- パッケージ名ディレクトリの下のパック・ディレクトリ

クライアント・ビルドなしでサーバー・ビルドだけを配布する場合は、パッケージ名ディレクトリの下の子・ディレクトリは、パック・サブディレクトリ以外コピーされません。

次の表に、デプロイメント・プロセスにおける主なパッケージ・コンポーネントのコピー元とコピー先を示します。サーバー共有パスは、マシン詳細テーブル(F9651)から得られます。

パッケージ・コンポーネント	コピー元	コピー先
パッケージ	package.inf ファイルの SrcDirs セクションに示されたパスからコピーされます。	宛先デプロイメント・サーバー上の<サーバー共有パス>%<パス・コード>%<パッケージ名>にコピーされます。
ファンデーション、データ、およびヘルプ	package.inf ファイルの SrcDirs セクションに示されたパスからコピーされます。	デフォルトのロケーションが選択されている場合は、宛先デプロイメント・サーバー上の<サーバー共有パス>%systemcomp にコピーされます。
Package.inf ファイル	ソース・デプロイメント・サーバーが基本ロケーションのプライマリ・デプロイメント・サーバーである場合、このファイルは、オブジェクト・パス・マスター・ファイル・テーブル(F00942)に示されているパスからコピーされます。それ以外の場合、このファイルは、そのパスからマシン詳細テーブル(F9651)内の package.inf ファイルにコピーされます。	宛先デプロイメント・サーバー上の<サーバー共有パス>%package_inf にコピーされます。

スマート・デプロイメント

サーバー複数階層デプロイメントにはスマート・デプロイメント機能があり、パッケージ宛先の使用可能サーバーに適合するサーバー・パッケージだけを使用可能にします。たとえば、基本ロケーションに HP9000 と AS/400 の両方のサーバー・パッケージがあっても、宛先ロケーションに HP9000 しか存在しない場合、宛先ロケーションに配布できるのは HP9000 パッケージだけになります。つまりサーバー・パッケージは、パッケージの宛先がそのサーバー・プラットフォームをサポートしていない限り配布できません。

ロケーションが複数ある場合でも、スマート・デプロイメントにより、宛先のプラットフォームに適合するサーバー・パッケージのみが使用可能になります。

package.inf ファイルの自動更新

サーバー・パッケージをリモート・ロケーションに配布する際には、package.inf ファイル内の次のセクションが更新されます。

- SrcDirs
- Attributes
- DeploymentServerName
- Location

基本ロケーション内のデプロイメント・サーバーに配布するときは、package.inf ファイルはコピーされません。

はじめる前に

- 「パッケージ・ビルド」の説明に従って、サーバー・パッケージをビルドします。

▶ 複数階層デプロイメント用にサーバー・パッケージをスケジュールするには

〈Package and Deployment Tools〉メニュー(GH9083)から〈Package Deployment〉プログラム(P9631)を選択します。

1. [Add]をクリックして〈Package Deployment Director〉を開きます。
特に指示がない限り、〈Package Deployment Director〉を開いた後は[Next]をクリックして次のフォームに進んでください。
2. 配布するサーバー・パッケージを選択します。
3. 〈Package Deployment Targets (パッケージ・デプロイメント先)〉で、[Deployment Server]オプションをクリックします。
4. 〈Package Deployment Attributes〉で、配布の日付と時刻を入力し、デプロイメント・オプションをオンにします。
5. 〈Deployment Server Selection〉で、サーバー・パッケージ配布先のマシン名を入力します。
ロー・ヘッダーをダブルクリックして、マシンを選択します。
6. 〈Build Selection〉で、ロー・ヘッダーをダブルクリックして、配布するパッケージのビルド(またはバージョン)を選択します。
スマート・デプロイメント機能により、宛先ロケーションの構成に適合するパッケージ・ビルドだけを選択できます。たとえば、AS/400 用のパッケージ・ビルドと HP 9000 用のパッケージ・ビルドがあっても、宛先ロケーションに HP9000 しか存在しない場合は、AS/400 ビルドを選択できません。
7. [Close]をクリックして〈Build Selection〉フォームを終了します。
8. 〈Work With Package Deployment〉で、[End]をクリックしてサーバー・パッケージのスケジューリング・プロセスを終了します。

参照

- 〈Package Deployment〉プログラム(P9631)または〈Multitier Deployment〉バッチ・プログラム(R98825C)を使用したパッケージの配布については、「デプロイメント・ロケーションへのソフトウェアの配布」

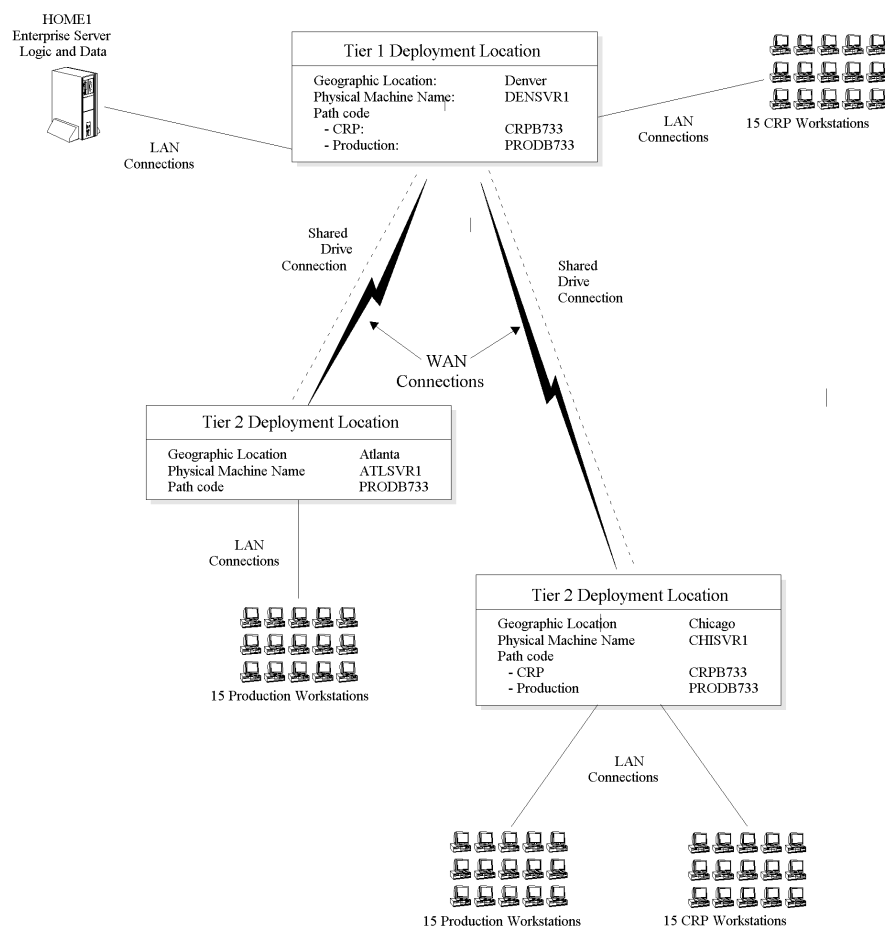
複数階層デプロイメント・ケーススタディの理解

ここでは、2 階層デプロイメント環境のケーススタディを紹介します。このケーススタディが示すように、WAN 接続を使用してパッケージをワークステーションに配布するのは、概して非効率です。代わりに、基本デプロイメント・サーバーから階層デプロイメント・ロケーションに配布する必要があります。その後、LAN 接続されたワークステーションに、各ローカル・デプロイメント・ロケーションからパッケージをインストールできます。

パッケージのインストールでは、遠隔デプロイメント・ロケーションがファイル・サーバーとして機能します。パッケージを遠隔デプロイメント・ロケーションでビルドすることはできません。パッケージは基本デプロイメント・ロケーションでビルドする必要があります。

ローカルに接続されたワークステーションは、階層デプロイメント・ロケーションからパッケージをプルできますが、これらのワークステーションは依然としてエンタープライズ・サーバーおよびデータベース・サーバーと接続する必要があります。

次の図はこのケーススタディの実例を示しています。



次の表に、このケーススタディで第 1 階層ロケーションが使用する仮サーバーを示します。

デンバー(本店、第 1 階層デプロイメント・ロケーション)

特性	設定	
エンタープライズ・サーバー名	HOME1	
デプロイメント・サーバー名	デンバー	DENSVR1
	アトランタ	ATLSVR1
	シカゴ	CHISVR1
プロトタイプ・ワークステーション	デンバー	15
	アトランタ	0
	シカゴ	15
本稼働用ワークステーション	デンバー	0
	アトランタ	15
	シカゴ	15
リリース	B9	
デプロイメントの階層	デンバー	1
	アトランタ	2
	シカゴ	2
パス・コード	デンバー	PD9
		PY9
	アトランタ	PD9
		PD9
	シカゴ	PD9
		PY9

▶ 複数階層デプロイメント用にシステムを構成するには

次に、複数階層デプロイメント・システムを構成するために必要なステップの概要を示します。

1. デプロイメント・ロケーションを定義します。

デプロイメント・サーバーでデプロイメント・ロケーションを定義します(この例では、DENSVR1)。すべてのデプロイメント・ロケーションを定義するには、〈Machine Identification〉アプリケーション(P9654A)を使用します。

このケース・スタディでは、次のフィールドに値を入力して、デプロイメント・ロケーションごとに 1 つずつ 3 つのロケーション、つまりデンバー、アトランタ、およびシカゴを定義します。

- Location

デプロイメント・ロケーション名を入力します。

このケーススタディでは、物理デプロイメント・ロケーションにそれぞれ次の名前を割り当てます。

デンバー

アトランタ

シカゴ

- Description

各デプロイメント・ロケーションの記述を入力します(最長 30 文字の値)。

次に例を示します。

デンバー:デンバー – 第 1 階層

アトランタ:アトランタ – 第 2 階層

シカゴ:シカゴ – 第 2 階層

- Location Code(ロケーション・コード)

ERP 9.0 デプロイメントの現行ロケーションを入力します。

たとえば、DEN などです。

- Parent Location(親ロケーション)

追加するロケーションの親ロケーションの名前を入力します。

たとえば、Corporate などです。

2. デプロイメント・サーバー定義を作成します。

作成したデプロイメント・ロケーションの各デプロイメント・サーバーの定義を作成するには、〈Machine Identification〉プログラム(P9654A)を使用します。このケーススタディでは、アトランタおよびシカゴのデプロイメント・サーバーを定義する必要があります。デンバーのデプロイメント・サーバーは基本(第 1 階層)サーバーなので、すでに定義されています。

このケーススタディでは、次の表に示すように、〈Deployment Server Revisions〉フォームのフィールドに値を入力します。

フィールド	値						
Machine Name	<p>物理マシン名を入力します。</p> <p>このケーススタディでは、次の値を入力します。</p> <table> <tr> <td>デンバー</td><td>DENSVR1</td></tr> <tr> <td>アトランタ</td><td>ATLSVR1</td></tr> <tr> <td>シカゴ</td><td>CHISVR1</td></tr> </table>	デンバー	DENSVR1	アトランタ	ATLSVR1	シカゴ	CHISVR1
デンバー	DENSVR1						
アトランタ	ATLSVR1						
シカゴ	CHISVR1						
Description	<p>記述を入力します(最長 30 文字の値)。</p> <p>たとえば、Multitier Deployment – Denver などです。</p>						
Release	<p>ERP 9.0 リリースを入力します。</p> <p>たとえば、B9 などです。</p>						
Primary User(基本ユーザー)	入力したマシンの基本ユーザーを入力します。						
Server Share Path (サーバー共有パス)	<p>システム・ファイルとその他のファイルが存在するパス・コードの共有ディレクトリ名を入力します。</p> <p>たとえば、¥B9 などです。</p>						

3. パッケージをスケジュールします。

〈Package Deployment〉プログラム(P9631)または〈Multitier Deployment〉バッチ・プログラム(R98825C)を使用して、第 1 階層デプロイメント・サーバーから第 2 階層デプロイメント・サーバーにパッケージの配布をスケジュールします。

参照

- デプロイメント・サーバーの設定については「デプロイメント・サーバーの定義」
- パッケージの配布スケジュールについては「デプロイメント・ロケーションへのソフトウェアの配布」
- デプロイメント・ロケーションの設定については「ロケーションの定義」

