



[プライマネージ]

PRIMANAGE

CSS Open Package Series for Enterprise Resource Planning

準備システム上でのシステム構築

1. 準備システム

開発を終えた ソフトウェア部品システム を、エンドユーザー向けに出荷する際には、準備システム^{【注1】} という、中間システムを経由させます。

【注1】 準備システムについては、弊社ホームページの「技術情報」の項目の「道具箱」の中にある、「ソフトウェア部品配布システム」というドキュメントで説明を行っていますので、ここでは、簡単な説明としています。

【準備システムの役割】

準備システム の役割は、エンドユーザー向けのソフトウェア部品システムを、確定することにあります。

準備システムの上で行う作業は、具体的には次の 2 項になります。

- (1) 開発を行ったシステム (PRIMANAGE MEISTER) では、エンドユーザーに出荷するモジュールと、開発環境だけに必要なモジュールが混在しているので、エンドユーザー向けのモジュールを分離して、準備システムに移す。
- (2) エンドユーザーに出荷するモジュールだけで、ソフトウェア部品システムを構成して動作の確認を行い、出荷できることを確認する。

【準備システムの特徴】

準備システムは、次のような特徴を持ちます。

- 準備システムは、開発システム (PRIMANAGE MEISTER) とは、独立したシステムです。
- 準備システムは、基本的に実行環境です。
- 準備システムの DB は、エンドユーザーのシステムの DB と同じものになります。

以上が基本ですが、エンドユーザーのシステムの DB と、開発システムの DB が異なる場合には、追加の作業が必要になります。

【エンドユーザーのシステムの DB と、開発システムの DB が異なる場合】

準備システムの DB が、開発システムの DB と異なる場合は、開発システムで使用している、[DB へのアクセスを行うモジュール] は、使えません。(= このモジュールでは DB にアクセスできません。)

このため、準備システム上で、(準備システム上の)DB に アクセス できるように、[DB へのアクセスを行うモジュール] を作成する必要があります。

このとき、第 1 相 (GUI)、第 2 相 (業務処理プログラム) の変更は必要がなく、[DB へのアクセスを行う モジュール] だけを作成します。^[注 2]

^[注 2] ソフトウェア部品システム では、DB が異なっているということは、第 3 相 が異なっているということですが、各相間 は互いに独立していますので、第 3 相 が変わっても、第 1 相 および 第 2 相 には影響がありません。

そこで、準備システムの DB が、開発システムの DB と異なる場合は、【準備システムの役割】に、次の 1 項目が追加されます。

(3) 準備システムの上で、[DB へのアクセスを行うモジュール] を作成する。

2. [DB へのアクセスを行うモジュール]の詳細

- 1) [DB へのアクセスを行う モジュール] は、具体的には、
 - ① 埋め込み型の SQL 文を処理する部分、
 - ② 第 1 相、あるいは第 2 相 からの、DB へのアクセスのリクエストを受け付け、① の “埋め込み型の SQL 文を処理する部分” とやりとりを行う部分(この中には通信の部分も含まれます)、

から構成されます。

- 2) ① は、ソフトウェア部品システム で、“DB アクセス関数” と呼ばれているものです。
 - ② には、具体的には、DxxxSvr.exe や、tsundll.dll 等の、サポート・モジュール が対応しています。

- 3) DB アクセス関数は、準備システム の DB に対応したものであることが必要です。つまり、DB アクセス関数のソース・ファイル は、準備システム の DB に対応した記述(シンタックス) となっていることが必要です。

準備システムの DB が、MS SQL Server(7.0 あるいは 2000)である場合、標準部品については、MS SQL Server のソース・ファイルを、開発システム (DB=Oracle)の DB アクセス関数から作成することができます。^[注 3]

^[注 3] Oracle で動作している DB アクセス関数のソース・ファイル と、MS SQL Server で動作している DB アクセス関数のソース・ファイル を比較してみると、多くの共通点があるので、Oracle の DB アクセス関数のソース・ファイル を変換して、MS SQL Server の DB アクセス関数のソース・ファイル を生成します。

この変換は、PRIMANAGE MEISTER に付属するツール(NT2NT 変換ツール) を使用して行います。

なお、DB アクセス関数 が、新しく(PRIMANAGE MEISTER 上で)開発されたものである場合には、正しく変換できるかどうかは保証されません。

4) [DB へのアクセスを行う モジュール] を作成する上で、開発環境としては、C の コンパイル ができる環境が必要となります。(Cobol および GUI の開発環境は要りません。) また、この環境では、

- ① 準備システムの DB に対応している、埋め込み型の SQL 文が記述されている ファイル を C のソース・ファイルに変換する プリコンパイラ、
- ② その C のソース・ファイル をコンパイルするときに参照する Include ファイル、および リンク のときに使用する ライブラリ、
- ③ 準備システム の DB に対応している、BSS の LIB、

が利用できるようになっている、ことが必要です。

3. 準備システム上での ソフトウェア部品システム の構築

準備システム の上で行う作業を、(実行の順番に) 挙げていくと、次のようになります。

下表(表 1)で “○” は、その項目を実施することを表し、“—” は、その項目を飛ばす(=実施しない)ことを表します。 また、(D) の欄は、準備システムの DB と開発システムの DB が異なる 場合の実施項目、(S) の欄は、準備システムの DB と開発システムの DB が同じ 場合に実施する項目を表しています。

表 1 準備システム の上で行う作業

No	(D)	(S)	実施項目	備考
1	○	○	コンピュータ・システムを 1 台用意します。(このコンピュータ・システムは、OS がインストールされているだけの空の状態であるものとします。) なお、システムとしての準備 (パーティションの設定や、開発システムとの接続、等) は終わっているものとします。	
2	○	○	ターゲットの DB (エンドユーザーのシステムで使用する DB) をインストールします。	
3	○	--	DB の プリコンパイラ をインストールします。	
4	○	--	C コンパイラ をインストールします。	
5	○	○	COBOL のランタイム・モジュール (osx) をインストールします。(osx は実行環境では必要となります。)	
6	○	○	開発システム上にある、エンドユーザー向けの ソフトウェア部品システム を、コピー します。(=実行環境で必要となる分ですが、まず第 1 段階分をコピーします。)	[*1] [*2]
7	○	--	エンドユーザー向けの ソフトウェア部品システムの 第 2 段階分をコピーします。(具体的には、Include, Lib、をコピーします。)	[*3]
8	○	--	開発システム上にある、Oracle 用の DB アクセス関数を、ツールを使って、準備システムの DB 用の DB アクセス関数に変換します。	[*4]
9	○	--	前項 8. で生成した、(準備システムの DB 用の)DB アクセス関数を、準備システムの src ディレクトリの下にコピーします。	[*5]
10	○	--	(準備システムの)DB に対応する、make, shell を準備システムに用意します。	[*6]
11	○	--	残りの ディレクトリ を作成します。(準備システムの DB が、MS SQL Server(7.0 あるいは 2000) の場合を除いて、これらのディレクトリは空です。)	[*7]
12	--	○	開発システム上にある、エンドユーザー向けの ソフトウェア部品システム から、shell を コピー します。	[*8]

13	○	○	準備システム の環境変数を セット します。	
14	○	--	準備システム の DB へのアクセスに関する モジュールのコンパイル をします。	【*9】
15	○	○	準備システム の DB に、テーブルを作成します。	【*10】
16	○	○	hosts や services の設定が、正しく行われていることを確認します。	
17	○	○	DB を起動し、SDaemon を起動して、動作のチェックを行います。	

<備考欄の説明>

- 【*1】開発システムには、エンドユーザー向けの モジュールや ファイル以外にも、開発環境用のモジュールなど、多くの モジュールや ファイルがありますし、他のエンドユーザー向けの部品も存在していることがありますから、コピー を行うときには、当該エンドユーザーのシステムに必要な モジュールや ファイルだけを コピー する必要があります。
- 【*2】ステップ(No)6 でインストールする分は、準備システムの DB と、開発システムの DB が、異同に関わらず、共通して インストールする分です。ディレクトリ としては、bat, bin, Bsssvr, client, dat, help, Lnktbl, Manual, Menu, oit (oit¥dat も含みます), spec, sys、が該当します。(sys の中の、bss_env.dat は、当該エンドユーザー用のものであることが必要です。)
- 【*3】ここでの目的は、ソフトウェア部品システムの Include ファイルと、Lib を コピー することです。
- 【*4】このツールは、NT2NT 変換ツール と呼ばれています。現時点では、Oracle 用の DB アクセス関数から、MS SQL Server (7.0 あるいは 2000) に変換することが可能です。

【*5】src ディレクトリ は、マニュアルで作成します。また、このディレクトリの下には、各サブシステム毎のフォルダも作成します。(このディレクトリ構造は、開発システムの src や obj ディレクトリの構造と同じです。)
なお、PRIMANAGE MEISTER の中にある SQL7Tgt サブディレクトリの下には、MS SQLServer (7.0 あるいは 2000) 用の DB アクセス関数のソース・ファイル が用意されていますので、準備システムの DB が MS SQLServer (7.0 あるいは 2000) である場合は、この PRIMANAGE MEISTER の中のソース・ファイルを、準備システムにコピーして使うことができます。(ディレクトリ構造ごと コピー します。)
ただ、この PRIMANAGE MEISTER の中にある ソース・ファイルは、標準のソフトウェア部品システム で使用されている DB アクセス関数についての分だけですので、開発システム上で、新たに開発された DB アクセス関数については、別途、そのソース・ファイルを変換してから、準備システム に持ち込む必要があります。

【*6】make は、準備システム での、DB アクセス関数と関連モジュール を コンパイル するためのものです。また、shell は、準備システム の DB に テーブルを作成するためのものです。
この make と shell は、基本的には、準備システム の DB に合わせて、マニュアルで作成します。
準備システムの DB が、MS SQL Server (7.0 あるいは 2000) である場合は、make, shell の両方とも、PRIMANAGE MEISTER の中に用意されていますので、コピーして使うことができます。(これも、PRIMANAGE MEISTER の SQL7Tgt フォルダの下にあります。)

【*7】作成が必要なのは、Errlog, obj, work, wrk です。
obj の下には、サブシステム毎のディレクトリ(フォルダ) を作成しておきます。(空のフォルダです。) obj のこれらのフォルダは、make を行うときに、コンパイラ によって使用されます。
準備システムの DB が、MS SQL Server (7.0 あるいは 2000) である場合は、PRIMANAGE MEISTER の SQL7Tgt フォルダの下にある obj をコ

ピーして使って下さい。(obj フォルダ には、少数ですが ファイル が収められていて、完全な空ではありません。)

【*8】 Oracle の DB テーブルを作成するスクリプトがあります。

【*9】 ここで、DB アクセス関数と、関連モジュールのコンパイルを行って、DBxxx.dll, tsundll.dll, DxxxSvr.exe を生成します。この項の詳細については、4.項で説明をいたします。

【*10】 shell の下にある スクリプト を使用します。スクリプトは既に、準備システムの DB に対応したものになっています。この段階で、DB の データの インポート ができます。

4. DB へのアクセスに関する コンパイル作業

ここでは、準備システム は、次のような状態になっているものとします。

- 1) C のコンパイラ が インストールされている。(具体的には、VC++)
- 2) 準備システム の DB に合わせた、埋め込み型 SQL 文を処理する環境が インストール されている。

(準備システムの DB が、MS SQL Server7.0 の場合は、“MSSql7” フォルダが存在していれば、この環境は インストールされています。)

3) 準備システム の DB に合わせた、ソフトウェア部品システム用の ヘッダ・ファイルと ライブラリ(=BSS の Include と LIB)が インストールされている。

4) 準備システムの DB に合わせた、make ファイル(群) が インストールされている。

5) 環境変数がセットされている。

(nmake.exe や cl.exe を起動するとき、また make を行うとき、環境変数のいくつか [例えば、BSS_HOME] が参照されます。)

コンパイルは、次のように行います。

① 準備システム の BSS_HOME の、make ディレクトリ で、DOS 窓を開きます。

(あるいは、適当な場所で DOS 窓を開き、BSS_HOME の下にある make ディレクトリ に移動します。)

② DOS プロンプト から、Alldbmak.bat を起動(実行)します。

- このバッチ・ファイルは、サブシステム(例:ACS, BHN, ...)毎に、そのサブシステムに属するテーブルへの アクセス関数を生成します。^[注4]

【注4】 アクセス関数の実装形態は、DLL です。サブシステム毎に 1 つの DLL が生成され、その DLL は、そのサブシステムに属するテーブル への全アクセス関数を、エクスポート します。

- Alldbmak.bat は、BSS が用意した(コンパイルの)バッチファイル なので、ソフトウェア部品システム の標準のサブシステム についての アクセス関数(の DLL)を生成できますが、お客様にて開発された DB アクセス関数については、生成できません。(次項で生成します。)

③ お客様にて開発された DB アクセス関数の DLL を生成します。

- **PRIMANAGE MEISTER** の dbtool をご使用の場合は、dbtool が、(専用の)バッチファイルを生成していますので、このバッチファイルを用います。
- なおこの場合、いくつかの ヘッダ・ファイル が変更されていますので、その変更が反映されるようにしておく必要があります。

この点につきましては、「PRIMANAGE MEISTER ユーザーズ・マニュアル(C 版)」の、「第 5 章 DBTool の使用方法」をご参照下さい。お客様にて作成された DB アクセス関数の コンパイル方法が説明されています。

④ ② の DOS プロンプトから、Alltsuma.bat を実行します。

- このバッチファイルは、次の(a)～(c) の、3つの make を順次実行するものです。

Alltsuma.bat を起動せずに、この3つの make を((a)→(b)→(c) の順番で)実行しても、全く差し支えありません。

- (a) nmake -f tsundll.mak
- (b) nmake -f tsun1.mak
- (c) Tmake.bat

- (a)では、tsundll.dll を生成します。
(b)と(c)を実行すると、各サブシステム毎の、DxxxSvr.exe が生成されます。

前の make の結果を後で使うので、順序は大切です。

- 上の3つの make で得られる結果は、(これも)BSS が用意したもので、ソフトウェア部品システムの 標準の サブシステム についての DxxxSvr.exe は生成しますが、お客様にて作成されたサブシステムの DxxxSvr.exe は生成できません。

また、tsundll.dll も、ソフトウェア部品システムの標準の DB のテーブルについて アクセス する機能を持ちますが、この dll を使った場合、お客様にて作成されたテーブルについての アクセスができません。(これらも、次項で生成します。)

- ⑤ お客様にて開発されたサブシステム の情報を tsundll.dll に組み込み、お客様の作成された サブシステムの DxxxSvr.exe を生成します。

- この場合の make ファイルは、Alltsusr.bat を使用します。
このバッチファイルを走らせると、次の 3 つの make を実行します。

- (a) nmake -f tsundll.mak
- (b) nmake -f tsunUSR1.mak
- (c) TmakeUsr.bat

- これらの3つの make は、(tsundll.mak を除いて名称は少し異なっていますが、)実行される事柄は、上記の ④項 と同じです。ただ、対象がお客様の サブシステム である点が ④項 とは異なっています。(つまり、(a) で、お客様の作成された サブシステムの情報が、tsundll.dll に付加され、(b) と (c) で、お客様のサブシステムについての、DxxxSvr.exe が作成されます。)
- この場合についても、「PRIMANAGE MEISTER ユーザーズ・マニュアル(C版)」の、「第5章 DBTool の使用方法」をご参照下さい。ここで実行されている事柄の内容と、操作の詳細を説明いたしております。

5. 動作試験

「3. 準備システム上での ソフトウェア部品システム の構築」の 17 項で、動作チェックに入りますが、最初の試験は、「システム管理」の「部門テーブル一覧」(SYS9860)と、「部門検索」(SYS9880)を試されることをお勧めします。「部門テーブル一覧」は、クライアント側から直接 DB にアクセスしています。また、「部門検索」は、Cobol を経由して DB に アクセス しています。このため、いずれか(あるいは両方)で正常な動作が行われなかった場合は、原因の切り分けに有効な情報が得られます。

6. 出荷の際の注意

開発システムの DB と、準備システムの DB が同じではない場合は、準備システム上に、C の コンパイル環境がありますので、出荷のときには、コンパイラのモジュール、Include、Lib の各フォルダ の内容を外して出荷するように注意します。