
SEWB+/REPOSITORY 辞書設計ガイド

手引書

3020-3-N82

マニュアルの購入方法

このマニュアル，および関連するマニュアルをご購入の際は，
巻末の「ソフトウェアマニュアルのサービス ご案内」をご参
照ください。

HITACHI

対象製品

P-2451-1434 SEWB+ 基本開発環境セット 03-00 (適用 OS : Windows 2000 , Windows XP , Windows Server 2003 , Windows Server 2003 x64 , Windows Vista)

輸出時の注意

本製品を輸出される場合には、外国為替および外国貿易法ならびに米国の輸出管理関連法規などの規制をご確認の上、必要な手続きをお取りください。

なお、ご不明な場合は、弊社担当営業にお問い合わせください。

商標類

CORBA は、Object Management Group が提唱する分散処理環境アーキテクチャの名称です。

ERwin は、Computer Associates International,Inc. の登録商標です。

Java 及びすべての Java 関連の商標及びロゴは、米国及びその他の国における米国 Sun Microsystems,Inc. の商標または登録商標です。

Microsoft は、米国およびその他の国における米国 Microsoft Corp. の登録商標です。

Microsoft Excel は、米国 Microsoft Corp. の商品名称です。

Microsoft Word は、米国 Microsoft Corp. の商品名称です。

ODBC は、米国 Microsoft Corp. が提唱するデータベースアクセス機構です。

ORACLE は、米国 Oracle Corporation の登録商標です。

Oracle は、米国 Oracle Corporation 及びその子会社、関連会社の登録商標です。

Sun は、米国 Sun Microsystems,Inc. の米国およびその他の国における商標または登録商標です。

UNIFY は、米国 Unify Corp. の登録商標です。

UNIFY2000 は、米国 Unify Corp. の商品名称です。

Windows は、米国およびその他の国における米国 Microsoft Corp. の登録商標です。

Windows Server は、米国およびその他の国における米国 Microsoft Corp. の商標です。

Windows Vista は、米国 Microsoft Corporation の米国及びその他の国における登録商標または商標です。

Xupper は、ケン・システム開発 (株) の商品名称です。

発行

2007 年 8 月 (第 1 版) 3020-3-N82

著作権

All Rights Reserved. Copyright (C) 2007, Hitachi, Ltd.

はじめに

このマニュアルは次に示すプログラムプロダクトのリポジトリ管理機能およびリポジトリブラウザ機能を使って、辞書を設計および管理する方法について説明したものです。

- P-2451-1434 SEWB+ 基本開発環境セット

SEWB+ 基本開発環境セットは、SEWB+ 基本開発環境と Groupmax ObjectServer から構成されます。

また、SEWB+ 基本開発環境は、次に示す機能から構成されます。

- リポジトリ管理機能
- リポジトリブラウザ機能
- プログラム構築支援機能
- レコード設計支援機能

なお、このマニュアルでは、リポジトリ管理機能 SEWB+/REPOSITORY、リポジトリブラウザ機能を SEWB+/REPOSITORY-BROWSER またはリポジトリブラウザと表記しています。また、特に区別する必要がない場合は、SEWB+/REPOSITORY と SEWB+/REPOSITORY-BROWSER (またはリポジトリブラウザ) を併せて、SEWB+/REPOSITORY と表記しています。

対象読者

- SEWB+/REPOSITORY を使って、システム開発プロジェクトで共有する辞書を設計および管理する方
- マニュアル「SEWB+/REPOSITORY 運用ガイド」で説明されている、リポジトリのシステム開発資源の構成、および SEWB+/REPOSITORY の機能の概要を理解されている方
- 使用しているサーバ OS とクライアント OS の機能および操作を理解されている方

マニュアルの構成

このマニュアルは、次に示す章と付録から構成されています。

第 1 章 概要

システム開発での辞書の位置付け、辞書が管理する情報や、辞書と SEWB+ ツール (分散開発環境でシステム開発を支援する SEWB+ シリーズのツールの総称) がどのように関係するのかについて説明しています。また、辞書の設計から運用までの、辞書に関する作業の流れについて説明しています。

第 2 章 辞書の設計

SEWB+/REPOSITORY の辞書の仕組みを有効に利用して、システム開発に効果的に役立たせるための辞書の設計について説明しています。

第 3 章 辞書の環境設定と運用

サーバの環境構築ユティリティを使った辞書の準備について説明しています。また、辞書を管理

する場合に参考となる資源管理の目安、アクセス権の設定や、複数サーバ間での辞書の運用について説明しています。

第 4 章 辞書情報の一括登録と出力

SEWB+/REPOSITORY では、既存資産を有効利用し、辞書情報を一括して扱うための機能をサーバに用意しています。ここでは、CSV 形式ファイルによる辞書情報の入出力、データベースからのデータ項目の生成、および SEWB3 データ項目辞書の移行について説明しています。

第 5 章 EUR Professional Edition のレポート設計

帳票設計ツールに EUR Professional Edition を使用すれば、辞書に登録した情報を利用して効率良くレポートを設計したり、レポート出力アプリケーションを作成したりできます。ここでは、辞書の情報がどのように利用されるかや、利用の手順などを説明しています。

第 6 章 APPGALLERY Enterprise のクライアントアプリケーション作成

SEWB+/REPOSITORY では、アプリケーション開発環境に APPGALLERY Enterprise を使用したクライアントアプリケーションの作成を支援しています。ここでは、連携の概要、辞書側での準備や利用の手順などを説明しています。

付録 A 標準データ項目辞書の紹介

標準データ項目辞書について説明しています。

付録 B ERwin/ERX からの辞書登録

ERwin/ERX からの辞書登録について説明しています。

付録 C 命名ルールインタフェース

命名ルールインタフェースについて説明しています。

付録 D 検査項目インタフェース

検査項目インタフェースについて説明しています。

付録 E 標準提供の同一項目用業務ルール

標準提供の同一項目用業務ルールについて説明しています。

付録 F 用語解説

マニュアル中で使用している用語について説明しています。

関連マニュアル

このマニュアルの関連マニュアルを次に示します。必要に応じてお読みください。

SEWB+/REPOSITORY 運用ガイド (3020-3-N81)

SEWB+ クライアントサーバシステム開発ガイド (3020-3-N83)

SEWB+/CONSTRUCTION アプリケーション開発ガイド (3020-3-N84)

SEWB+/RECORD DEFINER ユーザーズガイド (3020-3-N85)

Groupmax Object Server Version 6 システム管理者ガイド (3020-3-B56)

SEWB+/CS - DESIGN ユーザーズガイド (3020-3-770)

SEWB+/STANDARD-DICTIONARY 標準データ項目辞書 使用の手引 (3020-3-719)

SEWB+/REPORT MANAGER ドキュメント作成支援 使用の手引 (3020-3-720)

SEWB+/STANDARD-SUBROUTINE 標準サブルーチン 使用の手引 (3020-3-725)

APPGALLERY プログラマーズガイド (3020-7-432)

EUR 導入 (3020-7-053) ¹

EUR 帳票設計 (3020-7-054) ¹

EUR 帳票出力 (3020-7-055) ¹

EUR 導入 (3020-7-471) ²

EUR 帳票設計 (3020-7-472) ²

EUR 帳票出力 (3020-7-473) ²

SEWB3 データ項目設計支援 使用の手引 (3000-7-455)

SEWB3 運用の手引 (3000-7-451)

注 1

EUR Version 4 の場合にお読みください。

注 2

EUR Version 5 の場合にお読みください。

マニュアル体系

SEWB+ シリーズのマニュアル体系を次に示します。

SEWB+ クライアントサーバシステム開発ガイド (概) (3020-3-N83)

《リポジトリ管理》

SEWB+/REPOSITORY 運用ガイド (解)(手) (3020-3-N81)

SEWB+/REPOSITORY 辞書設計ガイド (手) (3020-3-N82)

Groupmax Object Server Version 6 システム管理者ガイド (解)(手) (3020-3-B56)

SEWB+/STANDARD-DICTIONARY 標準データ項目辞書 使用の手引 (手) (3020-3-719)

《オブジェクト指向分析・設計》

SEWB+ オブジェクト指向分析・設計支援 使用の手引 (手)(操) (3020-3-581)

《分散オブジェクト設計支援》

SEWB+/CS - DESIGN ユーザーズガイド (手)(操) (3020-3-770)

《クライアントサーバシステム設計支援》

SEWB+/OLTP DEFINER ユーザーズガイド (手)(操) (3020-3-909)

《アプリケーション開発》

SEWB+/CONSTRUCTION アプリケーション開発ガイド (手)(文) (3020-3-N84)

SEWB+/RECORD DEFINER ユーザーズガイド (手)(操) (3020-3-N85)

SEWB+/REPORT MANAGER ドキュメント作成支援 使用の手引 (手) (3020-3-720)

SEWB+/CODE ANALYZER ユーザーズガイド (手)(操) (3020-3-820)

SEWB+/CODE-DESIGN コード設計支援 使用の手引 (手) (3020-3-721)

SEWB+ バッチシステム向けアプリケーションフレームワーク・部品 使用の手引 (手) (3020-3-711)

SEWB+/STANDARD-SUBROUTINE 標準サブルーチン 使用の手引 (手) (3020-3-725)

SEWB+ COBOL 構造図エディタ 使用の手引 (手)(操) (3020-3-811)

〈記号〉

(概) : 概説書

(解) : 解説書

(手) : 手引書

(文) : 文法書

(操) : 操作書

読書手順

このマニュアルは、利用目的に合わせて次の個所をお読みいただくことをお勧めします。

マニュアルを読む目的	記述個所
SEWB+/REPOSITORY の辞書の概要について知りたい	1 章

マニュアルを読む目的	記述箇所	
辞書の設計について知りたい	設計手順	2.1 節
	ドメインの分析	2.2 節
	名称の標準化	2.3 節
	データ項目の設計	2.4 節
	レコード構造の設計	2.5 節
	業務ルールの設計	2.6 節
	辞書登録の確認	2.7 節
辞書の環境設定と、管理・運用の作業について知りたい	辞書の準備	3.1 節
	辞書情報のセキュリティ	3.2 節
	辞書の配布、複製	3.3 節
辞書情報の一括登録と出力について知りたい	CSV 形式ファイルを使って辞書登録と出力をする	4.1 節
	データベースから辞書のレコード定義（最上位結合項目）を生成する	4.2 節
	SEWB3 資産を移行する	4.3 節
EUR Professional Edition を使ったレポートの設計について知りたい	5 章	
APPGALLERY Enterprise と連携したクライアントアプリケーションの作成について知りたい	6 章	
標準データ項目辞書について知りたい	付録 A	
ERwin/ERX から辞書登録について知りたい	付録 B	
命名ルールインタフェースについて知りたい	付録 C	
検査項目インタフェースについて知りたい	付録 D	
標準提供の同一項目用業務ルールについて知りたい	付録 E	
このマニュアルで使用する用語について知りたい	付録 F	

このマニュアルでの表記

このマニュアルでは、製品名および機能名を次のように表記しています。

正式名称	表記	
Groupmax Object Server Version 6	Groupmax Object Server	Object Server
Groupmax High-end Object Server Version 6	Groupmax High-end Object Server	
JP1/NETM/Audit - Manager	JP1/NETM/Audit	
Microsoft(R) Visual C++(R)	VC++	

はじめに

正式名称	表記		
Microsoft(R) Windows(R) 2000 Professional Operating System 日本語版	Windows 2000 Professional	Windows 2000	Windows
Microsoft(R) Windows(R) 2000 Server Operating System 日本語版	Windows 2000 Server		
Microsoft(R) Windows(R) 2000 Advanced Server Operating System 日本語版			
Microsoft(R) Windows(R) 2000 Datacenter Server Operating System 日本語版			
Microsoft(R) Windows(R) XP Home Edition Operating System 日本語版	Windows XP		
Microsoft(R) Windows(R) XP Professional Operating System 日本語版			
Microsoft(R) Windows Server(TM) 2003, Enterprise Edition x86 日本語版	Windows Server 2003		
Microsoft(R) Windows Server(TM) 2003, Standard Edition x86 日本語版			
Microsoft(R) Windows Server(TM) 2003 R2, Enterprise Edition x86 日本語版			
Microsoft(R) Windows Server(TM) 2003 R2, Standard Edition x86 日本語版			
Microsoft(R) Windows Server(TM) 2003, Enterprise x64 Edition 日本語版	Windows Server 2003 x64		
Microsoft(R) Windows Server(TM) 2003, Standard x64 Edition 日本語版			

正式名称	表記
Microsoft(R) Windows Server(TM) 2003 R2, Enterprise x64 Edition 日本語版	Windows Vista
Microsoft(R) Windows Server(TM) 2003 R2, Standard x64 Edition 日本語版	
Microsoft(R) Windows Vista(TM) Business 日本語版	
Microsoft(R) Windows Vista(TM) Enterprise 日本語版	
Microsoft(R) Windows Vista(TM) Ultimate 日本語版	
Microsoft(R) Word	Microsoft Word
SEWB+/CS - DESIGN	SEWB+/CS-DESIGN
SEWB+ 基本開発環境のプログラム構築支援機能	SEWB+/CONSTRUCTION
SEWB+ 基本開発環境のレコード設計支援機能	SEWB+/RECORD DEFINER

このマニュアルでのバージョン表記

このマニュアルでは、SEWB+ 基本開発環境セットと SEWB+ リポジトリ セットのバージョンを次のように表記しています。

バージョン	このマニュアルでのバージョン表記	
SEWB+ 基本開発環境セット	03-00 SEWB+ 基本開発環境 03-00	
SEWB+ リポジトリ セット	02-57 02-55	SEWB+/REPOSITORY 02-08
	02-53	SEWB+/REPOSITORY 02-07
	02-51	SEWB+/REPOSITORY 02-06
	02-49	SEWB+/REPOSITORY 02-05
	02-47	SEWB+/REPOSITORY 02-04
	02-45	SEWB+/REPOSITORY 02-03
	02-43	SEWB+/REPOSITORY 02-02
	02-41	SEWB+/REPOSITORY 02-01
	02-00	SEWB+/REPOSITORY 02-00

このマニュアルで使用する略語

略語	正式名称
API	<u>A</u> pplication <u>P</u> rogram <u>I</u> nterface
C/S システム	<u>C</u> lient and <u>S</u> erver <u>S</u> ystem
CORBA	<u>C</u> ommon <u>O</u> bject <u>R</u> equest <u>B</u> roker <u>A</u> rchitecture
CSV	<u>C</u> omma <u>S</u> eparated <u>V</u> alues
DAM	<u>D</u> irect <u>A</u> ccess <u>M</u> ethod
DB	<u>D</u> ata <u>B</u> ase
DBMS	<u>D</u> atabase <u>M</u> anagement <u>S</u> ystem
DLL	<u>D</u> ynamic <u>L</u> inkage <u>L</u> ibrary
EUC	<u>E</u> nd <u>U</u> ser <u>C</u> omputing
GUI	<u>G</u> raphical <u>U</u> ser <u>I</u> nterface
IDL	<u>I</u> nterface <u>D</u> efinition <u>L</u> anguage
ODBC	<u>O</u> pen <u>D</u> atabase <u>C</u> onnectivity
OMG	<u>O</u> bject <u>M</u> anagement <u>G</u> roup
ORB	<u>O</u> bject <u>R</u> equest <u>B</u> roker
OS	<u>O</u> perating <u>S</u> ystem
PC	<u>P</u> ersonal <u>C</u> omputer
RDB	<u>R</u> elational <u>D</u> ata <u>B</u> ase
RPC	<u>R</u> emote <u>P</u> rocedure <u>C</u> all
SQL	<u>S</u> tructured <u>Q</u> uery <u>L</u> anguage
TAM	<u>T</u> able <u>A</u> ccess <u>M</u> ethod

このマニュアルで使用する記号

このマニュアルで使用する記号を次のように定義しています。

記号	意味
[]メニュー []ボタン []ダイアログ など	メニュータイトル, メニュー項目, ダイアログボックスのボタン, 操作キー, ウィンドウ, ダイアログボックスなどを示します。
[A] - [B]	- の前に示した [A] メニューから [B] を選択することを示します。

構文の説明で使用する記号

このマニュアルのコマンドの構文の説明では, 次の記号を使用します。

記号	意味
{ }	この記号で囲まれている項目は、省略してもよいことを示します。 複数の項目が縦または横に並べて記述されている場合には、すべてを省略できます。
...	記述が省略されていることを示します。 この記号の直前に示された項目を繰り返して複数個を指定できます。

図中で使用する記号

このマニュアルの図中で使用する記号を、次のように定義します。

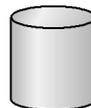
- パーソナル
コンピュータ (PC)



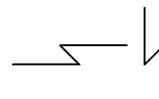
- 入出力の動作



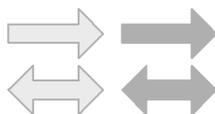
- ファイル



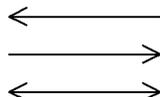
- ネットワーク



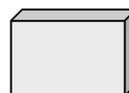
- データの流れ



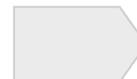
- 制御の流れ



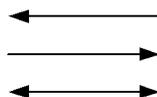
- プログラム



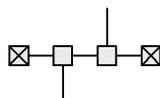
- 工程、作業項目の
流れ



- その他の流れ



- バス形の LAN



マニュアルで使用する文字種の表現

このマニュアルでは、文字種は次の「文字セット」で表現します。

文字セット	定義	備考
英字	英大文字 (A ~ Z) および英小文字 (a ~ z)	1 バイトを 1 文字とする半角文字
英大文字	A ~ Z, ¥, @, #	
カナ文字	ア ~ ン, °, °, -, およびカナ小文字	
数字	0 ~ 9	
数値	数字, ., +, -	
英数字	英字および数字	
日本語文字	シフト JIS 標準漢字コードで定められた漢字	2 バイトを 1 文字とする全角文字

このマニュアルで使用する画面図と操作説明で使用する OS のメニュー項目について

このマニュアルでは、特に断りのないかぎり、次に示す OS の場合に表示される画面、メニュー名およびアイコンを使用して説明しています。これらの OS 以外を使用している場合、画面、メニュー名およびアイコンが異なることがあります。詳細は各 OS のマニュアルを参照してください。

機能名	OS
SEWB+/REPOSITORY	Windows Server 2003
SEWB+/REPOSITORY-BROWSER	Windows Vista
SEWB+/CONSTRUCTION	
SEWB+/RECORD DEFINER	

マニュアルとヘルプを効果的にお使いいただくために

SEWB+/REPOSITORY ではオンラインヘルプ（以降、ヘルプと呼びます）を提供しています。

利用の目的に応じて、マニュアルとヘルプを使い分けることをお勧めします。

マニュアルは、SEWB+/REPOSITORY 全体の機能や使い方を理解したいときや、プロジェクトに適した運用を検討するときにお使いください。

ヘルプは、SEWB+/REPOSITORY を操作中に機能の詳細を知りたいときにお使いください。ダイアログの操作方法、メニューバーの利用方法の詳細、または注意事項などを参照できます。

ヘルプでは、次の内容を説明しています。

- リポジトリブラウザの操作
- ユティリティの操作
- コマンドの操作
- 各定義項目の説明
- 用語の説明

常用漢字以外の漢字の使用について

このマニュアルでは、常用漢字を使用することを基本としていますが、次に示す用語については常用漢字以外の漢字を使用しています。

個所（かしょ） 汎用（はんよう）

KB（キロバイト）などの単位表記について

1KB（キロバイト）、1MB（メガバイト）、1GB（ギガバイト）、1TB（テラバイト）はそれぞれ 1,024 バイト、1,024² バイト、1,024³ バイト、1,024⁴ バイトです。

目次

1	概要	1
1.1	辞書を共有資源とするデータ中心のシステム開発	2
1.2	辞書の構成	5
1.2.1	辞書の考え方	5
1.2.2	辞書フォルダによるマルチ辞書の構築	6
1.2.3	データ項目とは	7
1.2.4	業務ルールとは	8
1.3	辞書を利用したシステム開発	10
1.3.1	辞書を利用した C/S システム設計	10
1.3.2	辞書を利用したアプリケーション開発	12
1.4	SEWB+/REPOSITORY の機能	17
1.5	辞書に関する作業の流れ	19
2	辞書の設計	25
2.1	設計手順	26
2.2	ドメインの分析	30
2.2.1	ドメインの分析とは	30
2.2.2	ドメインの値域と制約条件を基に、辞書を設計する	31
2.3	名称の標準化	33
2.3.1	データ項目の名称基準	33
2.3.2	名称を標準化する	34
2.4	データ項目の設計	35
2.4.1	データ項目に定義する情報	35
2.4.2	データ項目の継承	51
2.4.3	データ項目の結合	53
2.4.4	ドメインを登録する	56
2.4.5	ドメインにグルーピングされたデータ項目を登録する	57
2.4.6	データ項目を削除する	58
2.5	レコード構造の設計	61
2.5.1	辞書のレコード定義（最上位結合項目）とプログラムの関係	61
2.5.2	辞書のレコード定義（最上位結合項目）を登録する	64
2.6	業務ルールの設計	67
2.6.1	業務ルールに定義する情報	67

2.6.2	業務ルールとプログラムの関係	75
2.6.3	ルールスクリプトの記述	79
2.6.4	適用条件の指定	95
2.6.5	システムで共用する業務ルールを登録する	99
2.6.6	特定業務で扱う業務ルールを登録する	105
2.6.7	同一項目用業務ルールを登録する	112
2.7	辞書登録の確認	114
2.7.1	定義内容を検査する	114
2.7.2	目的のデータ項目や業務ルールを検索する	116
2.7.3	関連を基に、資源の相互関係を調査する	117
2.7.4	データ項目、業務ルールを印刷する	118
3	辞書の環境設定と運用	119
3.1	辞書の準備	120
3.1.1	環境を設定する	120
3.1.2	辞書フォルダを作成する	132
3.1.3	同一項目用業務ルールを辞書に準備する	134
3.1.4	タイプマッピングの規則を定義する	135
3.2	辞書情報のセキュリティ	136
3.2.1	アクセス権の設定	136
3.2.2	排他制御	138
3.3	辞書の配布、複製	139
3.3.1	複数のリポジトリで辞書を運用する	139
3.3.2	辞書のエクスポート	143
3.3.3	辞書のインポート	148
4	辞書情報の一括登録と出力	153
4.1	CSV形式ファイルを使って辞書登録と出力をする	154
4.1.1	アプリケーション間の辞書情報の受け渡し	154
4.1.2	CSV形式ファイルの入力	157
4.1.3	CSV形式ファイル入力コマンド (RPCsvin) の使い方	165
4.1.4	CSV形式ファイルの出力	172
4.1.5	CSV形式ファイル出力コマンド (RPCsvout) の使い方	175
4.1.6	CSV形式の記述規則	179
4.1.7	CSV形式ファイルフォーマット	183

4.2	データベースから辞書のレコード定義（最上位結合項目）を生成する	192
4.2.1	データベースの表および列情報の入力	192
4.2.2	結合項目と構成項目の生成規則	193
4.2.3	ODBC 入力コマンド（RPOdbcin）の使い方	208
4.3	SEWB3 資産を移行する	212
4.3.1	SEWB3 データ項目辞書の移行作業	212
4.3.2	基本項目の変換規則	214
4.3.3	詳細項目の変換規則	218
4.3.4	SEWB3 辞書移行コマンド（RPTrans）の使い方	223

5

EUR Professional Edition のレポート設計	227
5.1 辞書のレコード定義（最上位結合項目）を基にレポートを設計する	228
5.1.1 辞書のレコード定義（最上位結合項目）とレポート出力アプリケーションの関係	228
5.1.2 レポート設計の手順	231
5.1.3 フィールドの生成規則	232

6

APPGALLERY Enterprise のクライアントアプリケーション作成	237
6.1 辞書のレコード定義（最上位結合項目）を基に OpenTP1 クライアントアプリケーションを作成する	238
6.2 データ項目と辞書の環境設定を基に GUI を作成する	240
6.2.1 SEWB+REPOSITORY と APPGALLERY Enterprise で作成する GUI の関係	240
6.2.2 データ項目とマッピング情報を利用した GUI の統一	240
6.2.3 プロパティセットを利用した GUI の統一	242

付録

付録 A 標準データ項目辞書の紹介	245
付録 A.1 SEWB+STANDARD-DICTIONARY（標準データ項目辞書）	246
付録 A.2 SEWB+STANDARD-SUBROUTINE（標準データ項目辞書サブルーチン）	246
付録 B ERwin/ERX からの辞書登録	247
付録 B.1 ERwin/ERX からデータ分析情報を連携する	247
付録 B.2 エンティティレポートの連携	248
付録 B.3 属性レポートの連携	266
付録 B.4 テーブルレポートの連携	273
付録 B.5 辞書登録の例	278

付録 C 命名ルールインタフェース	283
付録 C.1 条件 DLL の作成	283
付録 C.2 ヘッダファイル (Nrl_interface.h)	284
付録 C.3 関数の記述形式	285
付録 D 検査項目インタフェース	287
付録 D.1 検査 DLL の作成	287
付録 D.2 ヘッダファイル (Dsc_interface.h)	288
付録 D.3 関数の記述形式	290
付録 D.4 エラーメッセージ	297
付録 E 標準提供の同一項目用業務ルール	298
付録 E.1 同一項目用業務ルールの概要	298
付録 E.2 ルールスクリプト	298
付録 F 用語解説	302

索引

309

目次

図 1-1	システム開発での辞書の位置づけ	3
図 1-2	マルチ辞書の例	6
図 1-3	データ項目の継承の例	7
図 1-4	データ項目の結合の例	8
図 1-5	データ項目に関連する業務ルールの例	8
図 1-6	辞書の利用の流れ (SEWB+/CS-DESIGN)	11
図 1-7	辞書の利用の流れ (SEWB+/CONSTRUCTION)	12
図 1-8	辞書の利用の流れ (EUR Professional Edition)	14
図 1-9	辞書の利用の流れ (APPGALLERY Enterprise)	15
図 1-10	新規システム開発での作業例	19
図 2-1	辞書の設計手順	27
図 2-2	ドメインの値域と制約条件の継承	32
図 2-3	データ項目を作成するダイアログ ([基本属性] タブ)	36
図 2-4	データ項目を作成するダイアログ ([ソース生成属性] タブ)	38
図 2-5	取りうる値を定義するダイアログ	50
図 2-6	定義情報の継承と変更波及の例	52
図 2-7	結合項目の構成の例	54
図 2-8	結合関係の継承の例	56
図 2-9	辞書のレコード定義 (最上位結合項目) がプログラムに反映されるまで	62
図 2-10	業務ルールを作成するダイアログ ([属性] タブ)	68
図 2-11	業務ルールを作成するダイアログ ([データ項目] タブ)	70
図 2-12	同一項目用業務ルールを作成するダイアログ ([データ項目] タブ)	73
図 2-13	業務ルールを作成するダイアログ ([ルールスクリプト] タブ)	75
図 2-14	業務ルールがプログラムに反映されるまで	76
図 2-15	検査の仕組み	114
図 3-1	エクスポート・インポートの流れ	140
図 3-2	下位データ項目を指定した場合のエクスポート範囲の例	144
図 3-3	下位データ項目を指定しない場合のエクスポート範囲の例	145
図 3-4	辞書フォルダ間に相互関連がある場合	146
図 3-5	関連情報のエクスポートの例	147
図 3-6	相互に関連する辞書フォルダのインポート	150
図 4-1	アプリケーション間での情報の受け渡しの例	155
図 4-2	CSV 形式ファイルのサンプル	156

図 4-3 ユーザ作業の例と CSV 形式ファイルの入出力の関係	159
図 4-4 データベースからのデータの流れ	193
図 4-5 生成されるデータ項目間の関係	194
図 4-6 データソースから生成されるデータ項目の関係	195
図 4-7 移行作業の流れ	213
図 4-8 内部項目の移行の例	223
図 5-1 辞書のレコード定義（最上位結合項目）がレポート出力アプリケーションに反映されるまで	229
図 6-1 辞書のレコード定義（最上位結合項目）の利用の概要	238
図 6-2 データ項目とマッピング情報の利用の概要	241
図 6-3 プロパティセットの利用の概要	243
図 B-1 ERwin/ERX からの辞書登録	247
図 B-2 エンティティレポート形式の辞書登録の例	250
図 B-3 エンティティレポート形式の辞書登録の例（同一カラム名をマージする場合）	252
図 B-4 属性レポート形式の辞書登録の例	268
図 B-5 属性レポート形式の辞書登録の例（同一カラム名をマージする場合）	269

表目次

表 2-1	データ項目の定義項目	35
表 2-2	分類区分とタイプの対応 (COBOL 又は OOCOBOL)	38
表 2-3	分類区分とタイプの対応 (C 又は C++)	39
表 2-4	分類区分とタイプの対応 (SQL)	40
表 2-5	分類区分とタイプの対応 (IDL(CORBA))	40
表 2-6	分類区分とタイプの対応 (帳票定義)	41
表 2-7	分類区分とタイプの対応 (Java)	42
表 2-8	分類区分とタイプの対応のカスタマイズ例 (言語区分 SQL のタイプを HiRDB として使用する場合)	43
表 2-9	タイプに対応するけた数 (COBOL 又は OOCOBOL)	44
表 2-10	タイプに対応するけた数 (C 又は C++)	44
表 2-11	タイプに対応するけた数 (SQL)	45
表 2-12	タイプに対応するけた数 (IDL(CORBA))	45
表 2-13	タイプに対応するけた数 (帳票定義)	46
表 2-14	タイプに対応するけた数 (Java)	47
表 2-15	タイプ「フリー定義」の記述方法	49
表 2-16	結合項目の定義項目	54
表 2-17	プログラム生成時に参照されるデータ項目の定義情報	64
表 2-18	業務ルールの定義項目	67
表 2-19	タイプと @type の値の対応 (COBOL 又は OOCOBOL)	88
表 2-20	タイプと @type の値の対応 (C 又は C++)	88
表 2-21	タイプと @type の値の対応 (SQL)	89
表 2-22	タイプと @type の値の対応 (IDL (CORBA))	89
表 2-23	タイプと @type の値の対応 (帳票定義)	90
表 2-24	タイプと @type の値の対応 (Java)	90
表 2-25	@@rule 文の記述例と抽出される業務ルール	98
表 3-1	レコード生成キーワードの内容 (COBOL 又は OOCOBOL)	123
表 3-2	レコード生成キーワードの内容 (C 又は C++)	124
表 3-3	可変記号の生成規則	125
表 3-4	指定できるプロパティのデータ型	130
表 3-5	プロパティセットの設定例	131
表 4-1	データ項目の初期値	181
表 4-2	業務ルールの初期値	182

表 4-3	データ項目のヘッダの記述内容	184
表 4-4	データ項目の定義情報の記述内容	185
表 4-5	業務ルールのヘッダの記述内容	189
表 4-6	業務ルールの定義情報の記述内容	189
表 4-7	データソースの表から求められる列情報の例	194
表 4-8	生成されるデータ項目の定義内容の例	194
表 4-9	データソースから生成されるデータ項目の定義内容	196
表 4-10	ODBC タイプと分類, けた数, 小数部けた数の対応	198
表 4-11	ODBC タイプと言語別タイプの対応 (COBOL 又は OOCOBOL, C 又は C++)	200
表 4-12	ODBC タイプと言語別タイプの対応 (SQL)	202
表 4-13	ODBC タイプと言語別タイプの対応 (IDL(CORBA), 帳票定義, Java)	203
表 4-14	基本項目の変換	214
表 4-15	タイプから分類区分への変換	215
表 4-16	タイプから言語別タイプへの変換	217
表 4-17	詳細項目からデータ項目および業務ルールへの変換	218
表 5-1	フィールド定義情報の生成規則	233
表 5-2	データ種別とサンプルデータの生成規則	233
表 5-3	データ種別記号とデータ種別	235
表 B-1	エンティティレポートの出力項目と指定順序	249
表 B-2	マージされる属性間の条件	253
表 B-3	データ項目辞書とエンティティレポートの対応	254
表 B-4	ERwin/ERX のタイプの変換規則 A-1 (ORACLE)	256
表 B-5	ERwin/ERX のタイプの変換規則 A-2 (ORACLE)	258
表 B-6	ERwin/ERX のタイプの変換規則 B-1 (ORACLE)	259
表 B-7	ERwin/ERX のタイプの変換規則 B-2 (ORACLE)	260
表 B-8	ERwin/ERX のタイプの変換規則 C-1 (HiRDB)	261
表 B-9	ERwin/ERX のタイプの変換規則 C-2 (HiRDB)	263
表 B-10	データ項目の定義情報の記述内容 (エンティティレポート形式)	265
表 B-11	属性レポートの出力項目と指定順序	266
表 B-12	マージされる属性間の条件	270
表 B-13	データ項目辞書と属性レポート形式の対応	270
表 B-14	データ項目の定義情報の記述内容 (属性レポート形式)	272
表 B-15	データ項目辞書とテーブルレポート形式の対応	275
表 B-16	データ項目の定義情報の記述内容 (テーブルレポート形式)	278
表 B-17	[CSV 入力] ダイアログ	279

表 B-18	[CSV 入力オプション] ダイアログ	280
表 B-19	[CSV 入力] ダイアログ	281
表 B-20	[CSV 入力オプション] ダイアログ	282

1

概要

SEWB+/REPOSITORY は、システム開発に使用するデータの名称とその情報を「辞書」に格納して、データ中心アプローチによる開発を支援します。辞書はシステム開発で扱う情報の基本単位となる「データ項目」と、データ項目特有の処理を部品化した「業務ルール」を管理します。

この章では、SEWB+/REPOSITORY の辞書によってシステム開発で使用するデータを一元管理する概要について説明します。

1.1 辞書を共有資源とするデータ中心のシステム開発

1.2 辞書の構成

1.3 辞書を利用したシステム開発

1.4 SEWB+/REPOSITORY の機能

1.5 辞書に関する作業の流れ

1.1 辞書を共有資源とするデータ中心のシステム開発

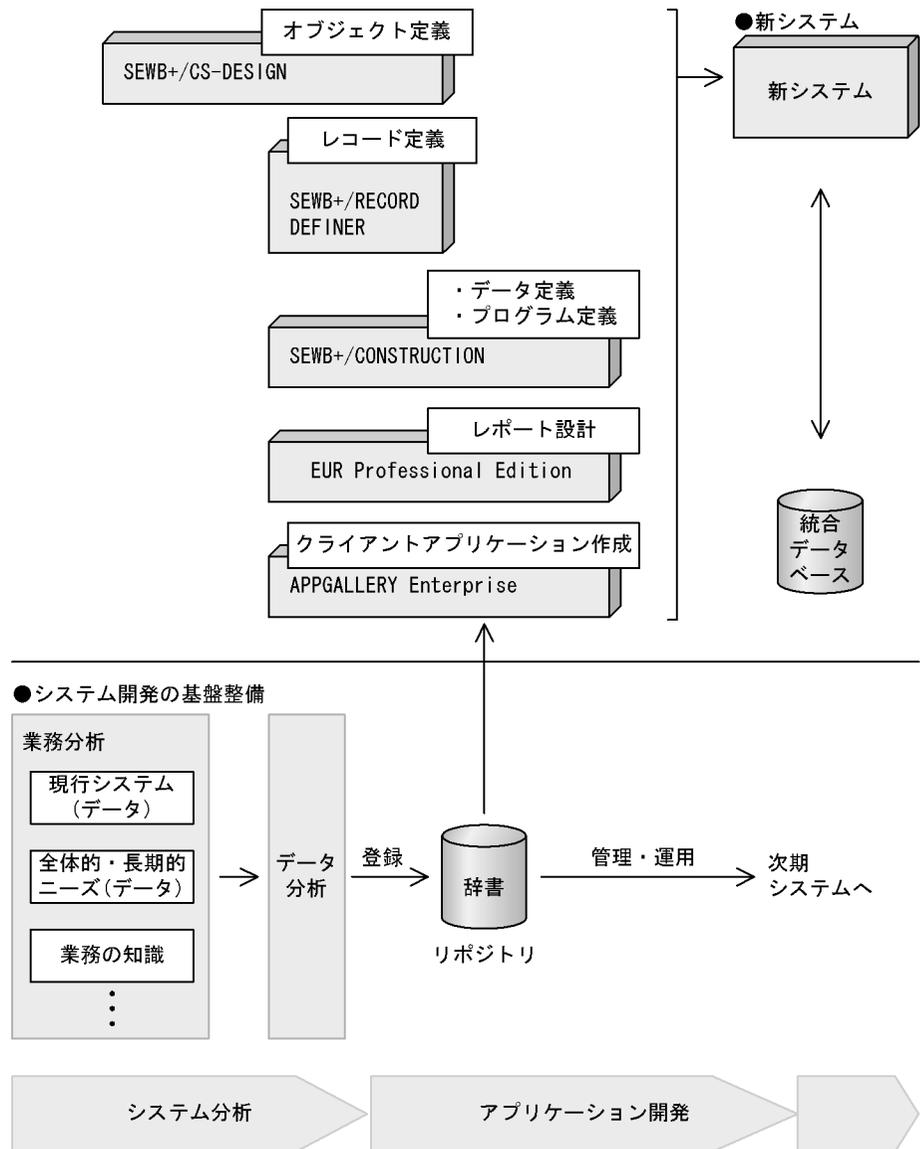
SEWB+/REPOSITORY は、サーバでリポジトリを管理し、クライアントの SEWB+ ツールやほかのアプリケーションと連携してシステム開発資源を蓄積したり、供給したりする役割を担います。リポジトリが管理するシステム開発資源は、システムの各工程で作成される情報である「ドキュメント」、データ項目と業務ルールを蓄積する「辞書」に大別されます。

データ中心アプローチの基盤となる辞書

データ中心アプローチとは、データを企業全体の重要な財産であると考え、共有資源として扱うシステム開発の方法論です。企業情報システムの基盤整備のために必要な方法論として提唱されてきました。データ中心アプローチを採用したシステム開発では、先にデータを共有資源として分析し、標準化します。データの基盤を整えたあと、システムやプログラムを設計、開発します。SEWB+/REPOSITORY の辞書は、このデータ中心アプローチの考え方に基づいて、データ中心のシステム開発を支援する機能を備えています。

システム開発での辞書の位置づけを、図 1-1 に示します。

図 1-1 システム開発での辞書の位置づけ



システム開発の概要

システム開発の概要を次に説明します。

まず、システム開発に先立って、現行システム、長期的・全体的ニーズを基にした目標とするシステムなどを対象に、データ分析をします。このデータ分析で抽出、標準化したデータを辞書に登録して、システム開発の基盤を整備します。そのあと、辞書をシステム開発の共有資源として管理・運用していきます。

SEWB+/CS-DESIGN を使った C/S (クライアントサーバ) システム設計、SEWB+/RECORD DEFINER を使ったレコード設計、および SEWB+/

1. 概要

CONSTRUCTION , EUR Professional Edition または APPGALLERY Enterprise を使ったアプリケーション開発時に、辞書は共有資源として利用されます。開発を終えたあとも、辞書は次のシステム開発に継続して利用されていきます。

1.2 辞書の構成

1.2.1 辞書の考え方

辞書は、データ中心アプローチの考え方に基づいて、システム開発の基本単位であるデータ項目と、データ項目特有の処理を部品化した業務ルールから構成されます。

辞書の基本構成

辞書は、データ項目を管理する「データ項目辞書」と、業務ルールを管理する「業務ルール辞書」から構成されています。この2種類の辞書は「辞書フォルダ」という単位で管理されます。辞書フォルダは、リポジトリで辞書を管理する場合の最上位の単位となります。

- データ項目辞書とは

データ項目は、システム開発で使用する情報の基本単位となるものです。データ中心アプローチによるシステム開発では、データ項目を標準化し、そのデータ項目を使ってシステムやプログラムを設計します。データ項目辞書は、この標準化されたデータ項目の保管庫の役割を果たします。

- 業務ルール辞書とは

業務ルールは、データ項目辞書のデータ項目に対応して、個々のデータ項目の性質、データ項目、データ項目間の制約条件などを表す処理です。業務ルール辞書は、業務ルールの保管庫の役割を果たします。

異なる各国語、プログラミング言語の情報を一元管理する

データ項目や業務ルールの定義内容をアプリケーション開発に使用するプログラミング言語間、国語間で共通化して一元管理できます。このため、同じ辞書を異なる国語圏で相互に使用するというような運用形態に対応できます。また、同じ辞書から C、COBOL、Java などの複数のプログラミング言語に対応できます。

独自の辞書にカスタマイズする

システムで使用する国語や開発言語、登録時の制限、データ項目や業務ルールの名称基準など、開発内容に合わせて辞書をカスタマイズできます。

辞書の構築を支援する検査機能

辞書登録やメンテナンス時などに、定義内容の妥当性を診断するための検査機能が提供されています。名称をリポジトリ内で重複して付けていないか、プロジェクトの名称基準（命名ルール）に従って名称が付けられているかどうかなど、辞書の構築を支援する検査項目が用意されています。また、ユーザ独自の検査項目を作成し、利用することもできます。

ほかのアプリケーションと辞書情報を連携する

ほかのアプリケーションから生成した CSV (Comma Separated Values) 形式ファイルを辞書に登録できます。また、辞書情報を、CSV 形式ファイルに出力できます。このほか、ODBC (Open Database Connectivity) インタフェースをサポートしている DBMS (Database Management System) 上のデータベースから、データ

1. 概要

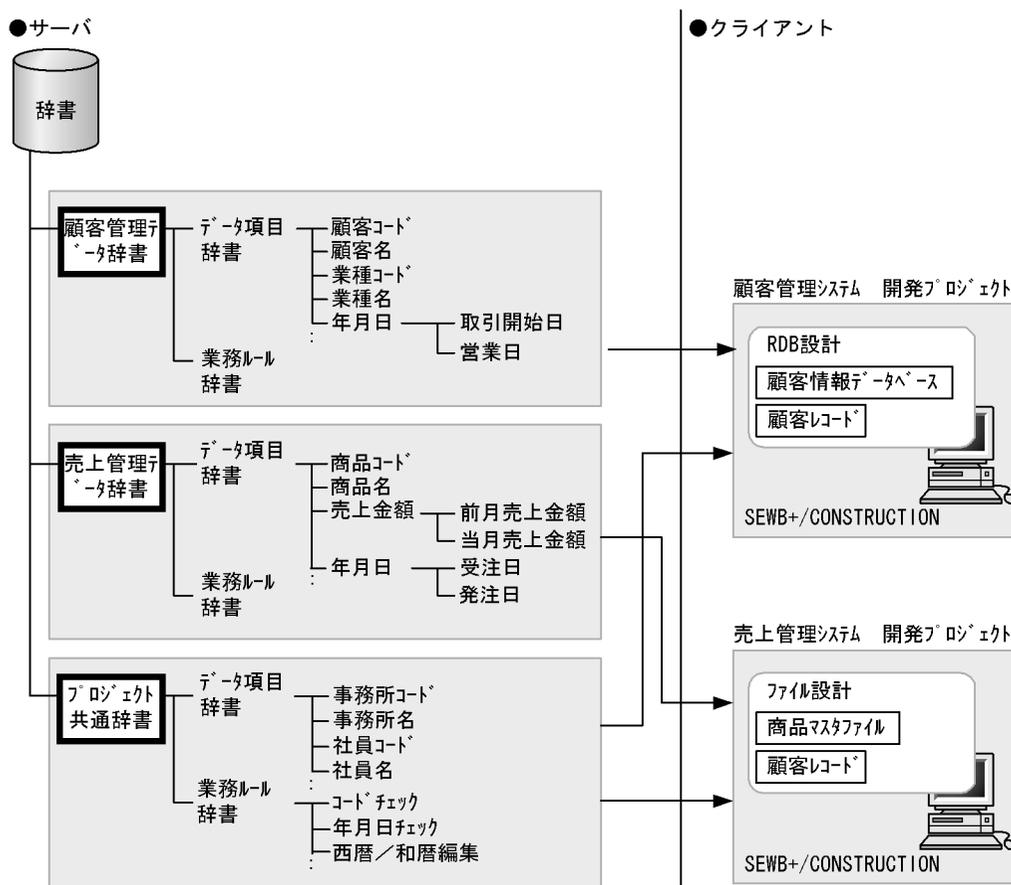
項目辞書の情報を生成できます。

1.2.2 辞書フォルダによるマルチ辞書の構築

一つのリポジトリで、複数の辞書フォルダを管理できます。これを、マルチ辞書といいます。マルチ辞書の構成は、システム開発形態、プロジェクトの体制といった独自の環境に合わせて構築できます。また、辞書フォルダには、独自の名称を付けられます。

マルチ辞書の構成の例を、図 1-2 に示します。

図 1-2 マルチ辞書の例



(凡例)

辞書フォルダ

この例では、顧客管理システムの開発で使用する「顧客管理データ辞書」、売上管理システムの開発で使用する「売上管理データ辞書」、プロジェクトで共通に使用する「プロジェクト共通辞書」というように、辞書をプロジェクトの用途に応じて分類しています。

この例のように用途別に分類するほか、システムの開発単位などで分類することもでき

ます。例えば、あるシステムの開発を終了する場合、それまで開発に使用してきた辞書を、新たに作成した別の保管用の辞書フォルダで管理するというような運用もできます。

1.2.3 データ項目とは

データ項目は、システム開発の情報の基本単位となる情報といえます。データ項目は、データ項目同士に継承や結合関係を定義することで階層化できます。

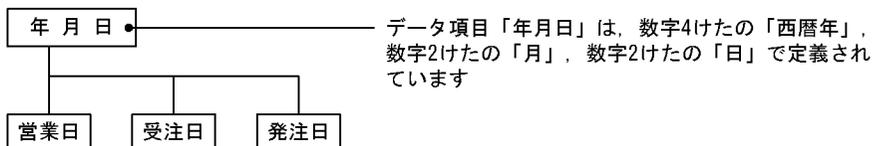
次にデータ項目の階層化について説明します。継承関係による階層化では、データ中心アプローチで提唱されているドメインの概念を辞書に採用できます。重複する内容をその度に定義する必要がないため、定義作業の効率化を図れます。データ項目の継承関係および結合関係は、相互に組み合わせることもできます。

• データ項目の継承

継承とは、共通の性質を持つ上位に位置づけられたデータ項目から、下位に位置づけられた複数のデータ項目にその性質やデータの表現形式を引き継ぐことです。例えば、日付を表すデータ項目「年月日」のように、システムで共通の性質と形式を持つデータ項目をグルーピングして、継承関係の上位のデータ項目（これがドメインに相当します）として定義します。そして、「営業日」、「受注日」および「発注日」を、同じ日付の性質を持つデータ項目として、下位のデータ項目に定義します。

この継承の例を、図 1-3 に示します。

図 1-3 データ項目の継承の例



この三つのデータ項目は、「年月日」の共通の形式と性質を引き継ぎます

(凡例)

□ : データ項目

— : 継承の関連

• データ項目の結合

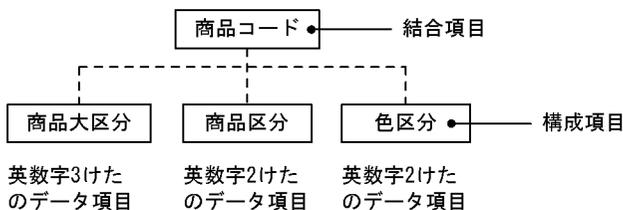
結合とは、複数のデータ項目を連結して定義し、そのデータ項目の集まりを一つのデータ項目として扱うことです。COBOLの集団項目、C言語の構造体の考え方に相当します。例えば、商品を識別するためのデータ項目「商品コード」では、まず、「商品大区分」、「商品区分」および「色区分」を定義したあと、これらのデータ項目を結合して定義します。

この結合の例を、図 1-4 に示します。この例の「商品コード」のように、複数のデータ項目で構成されるデータ項目を「結合項目」といいます。また、結合項目の構成要素となるデータ項目を「構成項目」といいます。

1. 概要

図 1-4 データ項目の結合の例

結合項目「商品コード」は、構成項目「商品大区分」、「商品区分」および「色区分」をまとめる、7けたのデータ項目となります



(凡例)

□ : データ項目

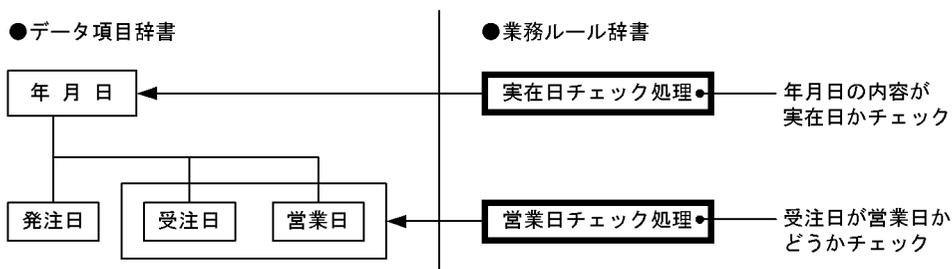
---- : 結合の関連

なお、レコード構造は、SEWB+/RECORD DEFINER でも定義できます。

1.2.4 業務ルールとは

業務ルールとは、データ項目に着目して、データ項目特有の処理を部品化したものです。SEWB+/CONSTRUCTION を使ったアプリケーション開発で共通に利用することを目的としています。業務ルールは必ずデータ項目と関連づけて登録し、業務ルール辞書で管理します。データ項目に関連する業務ルールの例を、図 1-5 に示します。

図 1-5 データ項目に関連する業務ルールの例



(凡例)

□ : データ項目 □ : 業務ルール

— : 継承の関連

← : 業務ルールとデータ項目の関連

なお、データ項目間の転記処理のような同じデータ項目を使った処理を記述するための特殊な業務ルールがあります。これを「同一項目用業務ルール」といいます。同一項目用業務ルールは、データ項目と関連づけないで、リポジトリ内のデータ項目で共用できます。

データ項目特有の処理を業務ルールとして登録しておけば、そのデータ項目を使用するさまざまなアプリケーションで再利用でき、処理を標準化できます。アプリケーションを作成するときには、あらかじめ用意された業務ルールから目的に合った業務ルールを選択するだけでよいので、効率良く作業できます。

データ項目特有の処理とは

次に、データ項目特有の処理（つまり、業務ルールに適した処理）とは、どのような処理なのかについて解説します。業務ルールに適した処理としては、「データ項目の値の妥当性をチェックする処理」および「データ項目から別のデータ項目の値を導き出す処理」が挙げられます。

<データ項目の値の妥当性をチェックする処理>

クライアント画面やファイルなどから入力された値が、正しいかどうかを判定する処理です。例えば、次に示すような処理などを業務ルールにできます。

- データ項目に入力された値の形式（英字、数字、漢字など）を判定する
- データ項目に入力された値の範囲（0以上、100以下など）を判定する
- 年月日のデータ項目の値が、実在する日付かどうかを判定する
- 出荷年月日は受注年月日よりもあとかどうか判定する

<データ項目から別のデータ項目の値を導き出す処理>

データベースやファイル、クライアント画面や帳票データなど、これらの中でデータをやり取り（入力または出力）する場合に必要な処理です。例えば、次に示すような、形式を変換する処理、編集処理、複数のデータ項目で成り立つ計算処理などを業務ルールにできます。

- 西暦年を和暦年に変換する
- 時刻を hh:mm:ss の形式に編集する
- 電話番号の市外局番，市内局番，加入者番号をハイフンで編集する
- 商品単価と売り上げ数量を基に，売り上げ金額を計算する
- 売り上げ金額と消費税率を基に，消費税（外税）を計算する
- 作業開始年月日と作業終了年月日から，実作業日数を求める
- 商品コードから商品名を取得する
- 所属事業所，入社年，従業員氏名から，氏名 NO を割り当てる

例に挙げたように、業務ルールには、データ項目の形式や範囲を判定するといった汎用性の高いチェック処理だけでなく、編集処理や計算処理など、業務の規則や制度などに密接な処理を蓄積して、アプリケーション開発に利用できます。業務の見直しや制度の変更に伴ってデータ項目に関する処理を変更しなければならない場合でも、業務ルールを使用していれば、変更波及のある開発資源を容易に調査でき、作業量も軽減できます。

1.3 辞書を利用したシステム開発

辞書は、SEWB+/CS-DESIGN を使った C/S (クライアントサーバ) システム設計、SEWB+/RECORD DEFINER を使ったレコード設計、SEWB+/CONSTRUCTION、EUR Professional Edition または APPGALLERY Enterprise を使ったアプリケーション開発で共通に利用されます。データ分析で標準化された再利用性の高い辞書の資源を、複数の開発者と開発工程で共通に利用することで、システム開発作業の効率を向上すると同時に、システムでのデータや処理の一貫性を保てます。

1.3.1 辞書を利用した C/S システム設計

SEWB+/CS-DESIGN とは

ORB (オブジェクトリクエストブローカ) を利用した C/S システムのアプリケーション (オブジェクト) の論理構成設計を支援するツールです。特に、3層 - 画面入出力などのユーザインタフェースを制御する「プレゼンテーション層」、中核的な業務処理を実行する「ファンクション層」、データベースの管理およびデータベースへのアクセスを実行する「データ層」 - で構成される C/S システムの設計に適しています。SEWB+/CS-DESIGN では、C/S システムのアプリケーション分割や関連の定義を、論理設計図を描きながらビジュアルに設計できます。論理設計図には、各アプリケーションの外部インタフェースとして、アプリケーションの操作関数、属性や例外を定義できます。また、オブジェクトに定義したインタフェースの情報からは、CORBA¹ (Common Object Request Broker Architecture) の IDL² (Interface Definition Language) を生成できます。

注

SEWB+/REPOSITORY との連携は、SEWB+/CS-DESIGN のバージョン 02-04 以降が前提となります。

注 1

CORBA は、OMG (オブジェクトマネジメントグループ) が標準化を進めている、ネットワークに分散したオブジェクト間のメッセージング機構の規約です。

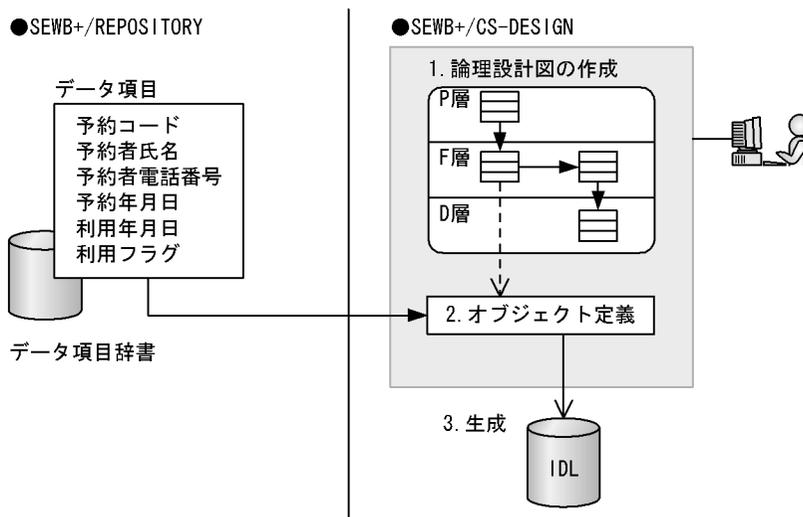
注 2

IDL は、ORB を利用する C/S システムで、オブジェクトのインタフェースを定義する言語です。ORB とは、クライアントとサーバに分散されたアプリケーション間の通信を制御し、要求を伝える役割を持つ分散オブジェクト実行環境です。

辞書の利用の流れ

次に、SEWB+/CS-DESIGN から辞書に登録した資源がどのように利用されるのかについておおまかに説明します。この流れを、図 1-6 に示します。

図 1-6 辞書の利用の流れ (SEWB+/CS-DESIGN)



1. 論理設計図の作成

C/S システムを論理的な 3 層に分割し、システムを構成するオブジェクトをどの階層に属するのかを決定します。そして、各オブジェクトおよびオブジェクト間の呼び出し関係を決定し、論理設計図を作成します。

2. オブジェクト定義

オブジェクトの詳細を設計して定義します。ここでは、そのオブジェクトがインタフェースを持つかどうかや、インタフェースを持つ場合の属性や引数などを定義します。このとき、辞書に用意したデータ項目が利用されます。

3. IDL の生成

開発する C/S システムで CORBA をサポートする ORB (日立製品では、TPBroker があります) を使用する場合は、個々のオブジェクトから、または論理設計図を構成するすべてのオブジェクトから IDL を生成します。インタフェース情報の定義で利用したデータ項目の情報が、SEWB+/CS-DESIGN の生成規則に従って IDL の中に展開されます。

論理設計図をサーバアプリケーション作成に連携する

SEWB+/CS-DESIGN で設計した論理設計図のファイルを SEWB+/CONSTRUCTION に読み込み、サーバアプリケーション開発に利用できます。SEWB+/CS-DESIGN の詳細は、マニュアル「SEWB+/CS-DESIGN ユーザーズガイド」を参照してください。SEWB+/CS-DESIGN での設計情報を利用したアプリケーション開発については、マニュアル「SEWB+/CONSTRUCTION アプリケーション開発ガイド」を参照してください。

1.3.2 辞書を利用したアプリケーション開発

(1) SEWB+/CONSTRUCTION を使ったアプリケーション開発

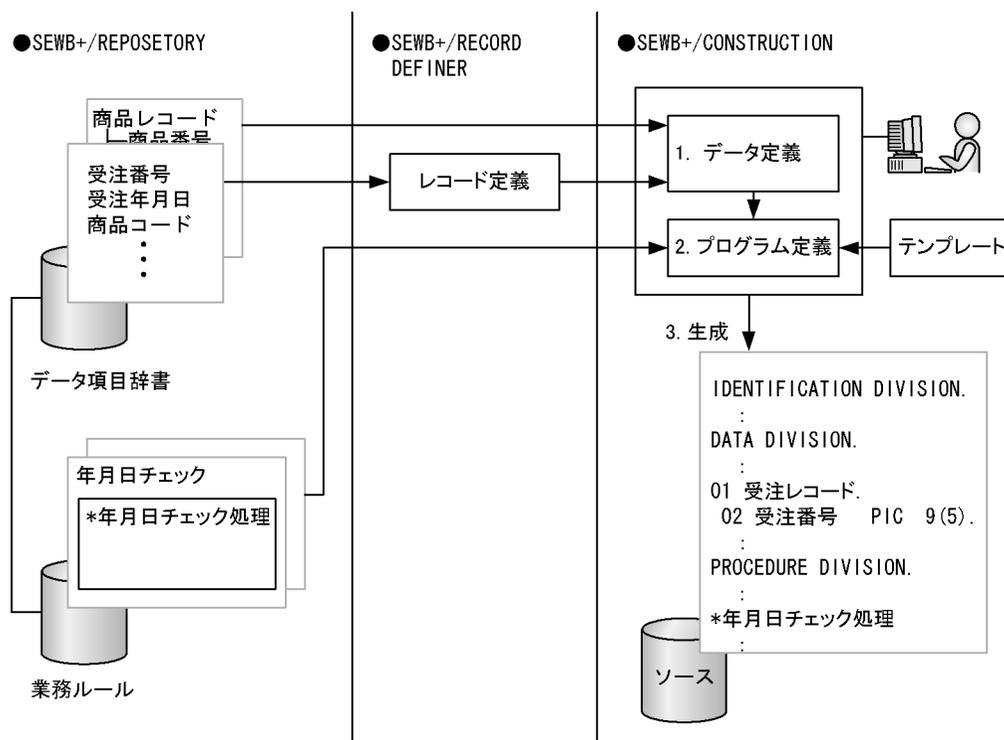
SEWB+/CONSTRUCTION とは

SEWB+/CONSTRUCTION は、C/S システムのアプリケーション開発を支援するツールです。特に、TP モニタやジョブ運用管理ツールなどを利用する大規模な C/S システム上で動作するサーバ側のプログラムおよびバッチシステムの開発に適しています。テンプレート、データ定義、プログラム定義の機能を使い、C/S システムのサーバプログラムと、クライアントプログラムに必要なサーバプログラムとのリンクのインタフェースを効率良く作成できます。また、CORBA 環境の C/S システムで、サーバオブジェクトの開発にも適しています。サーバオブジェクトは、SEWB+/CS-DESIGN の論理設計図を利用して開発できます。

辞書の利用の流れ

次に、SEWB+/CONSTRUCTION から辞書に登録した資源がどのように利用されるのかについて、おおまかに説明します。この流れを、図 1-7 に示します。

図 1-7 辞書の利用の流れ (SEWB+/CONSTRUCTION)



1. データ定義

データ定義では、アプリケーションに使用するファイル、データベースおよびクライアントとサーバ間の通信のインタフェースを定義します。ここでは、辞書の

レコード定義（最上位結合項目）や SEWB+/RECORD DEFINER で作成したレコード定義が利用されます。

2. プログラム定義

プログラム定義では、作成するアプリケーションに必要なデータ定義や、アプリケーション固有の処理を定義します。ここでは、テンプレート に記述された指示に従って、作成するアプリケーションに適した業務ルールの一覧が表示されます。その中から、プログラマが目的の業務ルールを選択することで、業務ルールが利用されます。

注

テンプレートとは、ソースプログラムの生成時にアプリケーションの枠組みとなる処理構成を記述したものです。テンプレートは、実際にアプリケーション作成の作業を開始する前に、専任者（テンプレート作成者）によって作成されます。テンプレート記述言語と、開発に使用するプログラミング言語で記述されます。

3. ソースプログラムの生成

プログラム定義からソースプログラムが生成される時、データ定義で利用したデータ項目の情報と、プログラム定義で利用した業務ルールの情報が、ソースプログラム中に展開されます。

このようにして、辞書の情報は、ソースプログラムまで引き継がれ、データ中心の開発を実現できます。

SEWB+/CONSTRUCTION を使ったアプリケーション開発の詳細は、マニュアル「SEWB+/CONSTRUCTION アプリケーション開発ガイド」を参照してください。

（2）EUR Professional Edition を使ったレポート設計

EUR Professional Edition とは

SEWB+/REPOSITORY と連携し、レポートの設計から印刷までを支援するツールです。EUR Professional Edition では、ワープロを操作する感覚で、自由な形式のレポートを効率良く設計できます。設計時には、SEWB+/REPOSITORY の辞書のレコード定義（最上位結合項目）から、レポートの設計に必要なフィールド定義情報を自動的に生成して利用できます。また、SEWB+/CONSTRUCTION で作成したアプリケーションと EUR Professional Edition で作成したレポートの設計情報を組み合わせれば、効率良くレポート出力アプリケーションを作成できます。

注

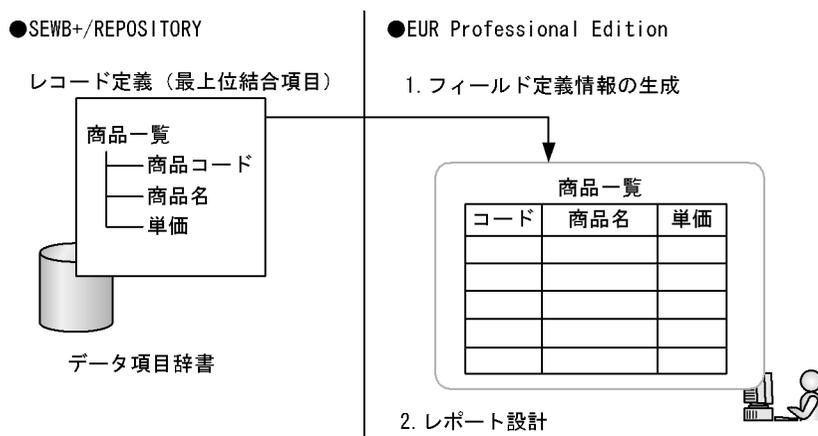
SEWB+/REPOSITORY との連携は、EUR Professional Edition のバージョン 04-01 以降が前提となります。

辞書の利用の流れ

次に、EUR Professional Edition から辞書に登録した資源がどのように利用されるのかについて、おおまかに説明します。この流れを、図 1-8 に示します。

1. 概要

図 1-8 辞書の利用の流れ (EUR Professional Edition)



1. フィールド定義情報の生成

辞書のレコード定義 (最上位結合項目) を選び、レポートの設計に必要なフィールド定義情報 (データのけた数やデータ種別などの情報をいいます) を生成します。

2. レポート設計

生成されたフィールド定義情報を基に、けい線や印字位置などのレイアウトを編集してレポートを設計します。

辞書と SEWB+/CONSTRUCTION で作成するアプリケーション、レポートの関係、レポートの設計手順などについては、「5. EUR Professional Edition のレポート設計」を参照してください。

(3) APPGALLERY Enterprise を使ったアプリケーション開発

APPGALLERY Enterprise とは

再利用性の高いさまざまな部品 (機能単位のソフトウェア) を組み合わせてアプリケーションを開発する、アプリケーション開発環境です。画面上で部品を組み合わせていくビジュアルプログラミングを採用しています。特に、GUI (グラフィカルユーザインタフェース) を中心としたアプリケーションの作成に適しています。大規模な C/S システムの開発では、クライアント側で動作するアプリケーションの作成に適しています。

APPGALLERY Enterprise では、SEWB+/REPOSITORY と連携して、大規模システムのアプリケーション開発を支援する機能を備えています。SEWB+/REPOSITORY の辞書の情報を開発に利用し、作成したドキュメントをリポジトリで管理することで、データの共有化および一元管理を図れます。

注

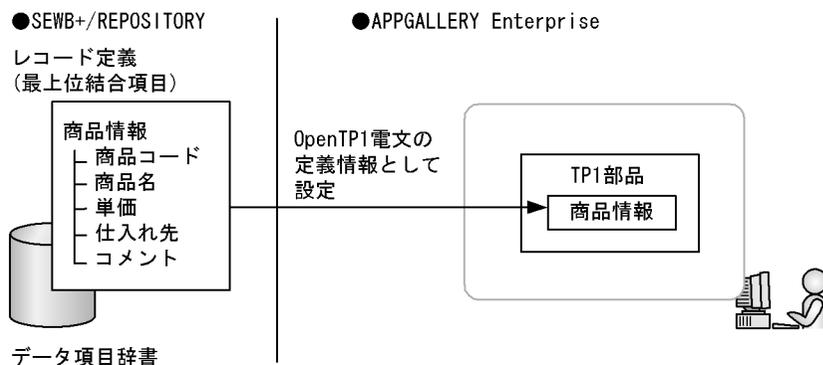
SEWB+/REPOSITORY との連携は、APPGALLERY Enterprise のバージョン 05-00 以降が前提となります。

辞書の利用の流れ

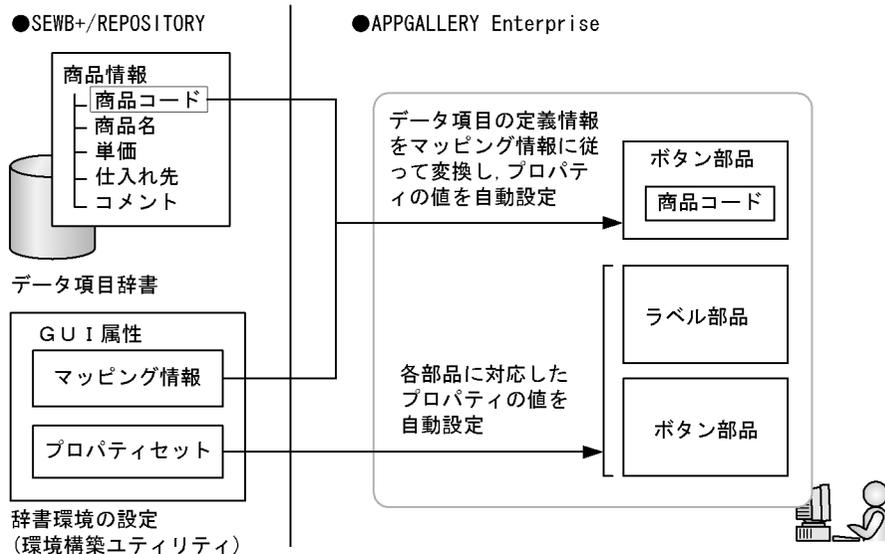
APPGALLERY Enterprise から辞書に登録した資源がどのように利用されるのかについて、おおまかに説明します。APPGALLERY Enterprise からは、2通りの利用形態があります。この利用の流れを、図 1-9 に示します。

図 1-9 辞書の利用の流れ (APPGALLERY Enterprise)

〈OpenTP1クライアントアプリケーションの開発に利用する〉



〈クライアント画面のGUI作成に利用する〉



- OpenTP1 クライアントアプリケーションの開発に利用する

OpenTP1 を利用した C/S システムのクライアントアプリケーションを開発するとき、辞書のレコード定義（最上位結合項目）を、通信でやりとりするための電文の定義情報として利用します。通信用のレコード定義は、SEWB+/CONSTRUCTION でのサーバアプリケーションの作成にデータ定義として利用さ

1. 概要

れます。このため、電文の定義情報を共有化でき、定義の不一致を回避できます。

注

OpenTP1 は、日立製の TP モニタです。C/S システム環境でトランザクション処理を実現します。

- クライアント画面の GUI の作成に利用する

SEWB+/REPOSITORY 側で、C/S システムのクライアントアプリケーションの GUI 作成に利用する部品の設定値（プロパティ）を一元管理します。GUI 作成時に、その設定情報を利用します。

マッピング情報の利用

データ項目の定義情報を、部品の設定値に置き換えて、GUI のオブジェクトを自動的に作成できます。

プロパティセットの利用

各部品の標準的な設定値を SEWB+/REPOSITORY 側でプロパティセットとして設定しておけます。これによって、複数の開発者でチーム開発する場合でも、システムや業務で GUI の統一を図れます。

なお、マッピング情報とプロパティセットは、アプリケーション開発前に、SEWB+/REPOSITORY の環境構築ユティリティを使って辞書の環境として設定しておく必要があります。

辞書と APPGALLERY Enterprise との関係や連携の概要については、「6. APPGALLERY Enterprise のクライアントアプリケーション作成」を参照してください。また、APPGALLERY Enterprise でのアプリケーション開発の詳細は、マニュアル「APPGALLERY プログラマーズガイド」を参照してください。

1.4 SEWB+/REPOSITORY の機能

ここでは、SEWB+/REPOSITORY に用意されている機能の中から、辞書の作成や管理に利用する機能を紹介します。SEWB+/REPOSITORY 全体の機能については、マニュアル「SEWB+/REPOSITORY 運用ガイド」を参照してください。

サーバの機能

サーバには、リポジトリの開発資源を管理・運用するためのユティリティが用意されています（Windows 2000 Server, Windows Server 2003 または Windows Server 2003 x64 で動作）。次に、辞書の管理に利用するユティリティを紹介します。なお、サーバのユティリティ群は、システム管理者だけが利用できます。

- 環境構築ユティリティ
辞書フォルダの作成や、独自の辞書環境、辞書に関する操作ログを出力するかどうかを設定します。
また、ODBC 入力や ERwin/ERX と連携した CSV 入力をするときのタイプ変換（タイプマッピング）の規則や、APPGALLERY Enterprise を使った GUI 作成で使用する GUI 属性を設定します。
- エクスポート・インポートユティリティ
リポジトリから辞書情報を取り出すことをエクスポート、エクスポートした辞書情報をリポジトリに取り込むことをインポートといいます。辞書を分散環境へ配布したり、複製したりします。
- CSV 形式ファイルの入力（CSV 入力）
CSV 形式ファイルからデータ項目または業務ルールの辞書情報を入力します。
- CSV 形式ファイルの出力（CSV 出力）
データ項目または業務ルールの辞書情報を CSV 形式ファイルへ出力します。
- ODBC 入力コマンド
ODBC インタフェースをサポートしている DBMS 上の既存データベースからスキーマ定義情報を取り出し、データ項目を生成します。
- SEWB3 辞書移行コマンド
SEWB3 データ項目辞書の辞書情報を SEWB+/REPOSITORY の辞書に移行します。

クライアントの機能

クライアントの機能を紹介します。

- リポジトリブラウザ
クライアントには、リポジトリの開発資源を操作、利用するための「リポジトリブラウザ」が用意されています（Windows で動作）。リポジトリブラウザを使うと、リポジトリのシステム開発資源をクライアント側でビジュアルに操作できます。リポジトリブラウザには、次の三つの機能があります。

リポジトリのブラウジング機能

リポジトリに格納されたシステム開発資源をツリー形式やリスト形式で表示し

1. 概要

ます。データ項目や業務ルールをリポジトリに登録したり，目的の辞書情報を検索したりできます。

結合項目のブラウジング機能

データ項目を，結合関係に基づいてブラウジングしたり編集したりできます。

関連情報のブラウジング機能

データ項目を，関連に基づいてブラウジングできます。

- CSV 形式ファイルの入力（CSV 入力）

CSV 形式ファイルからデータ項目または業務ルールの辞書情報を入力します。

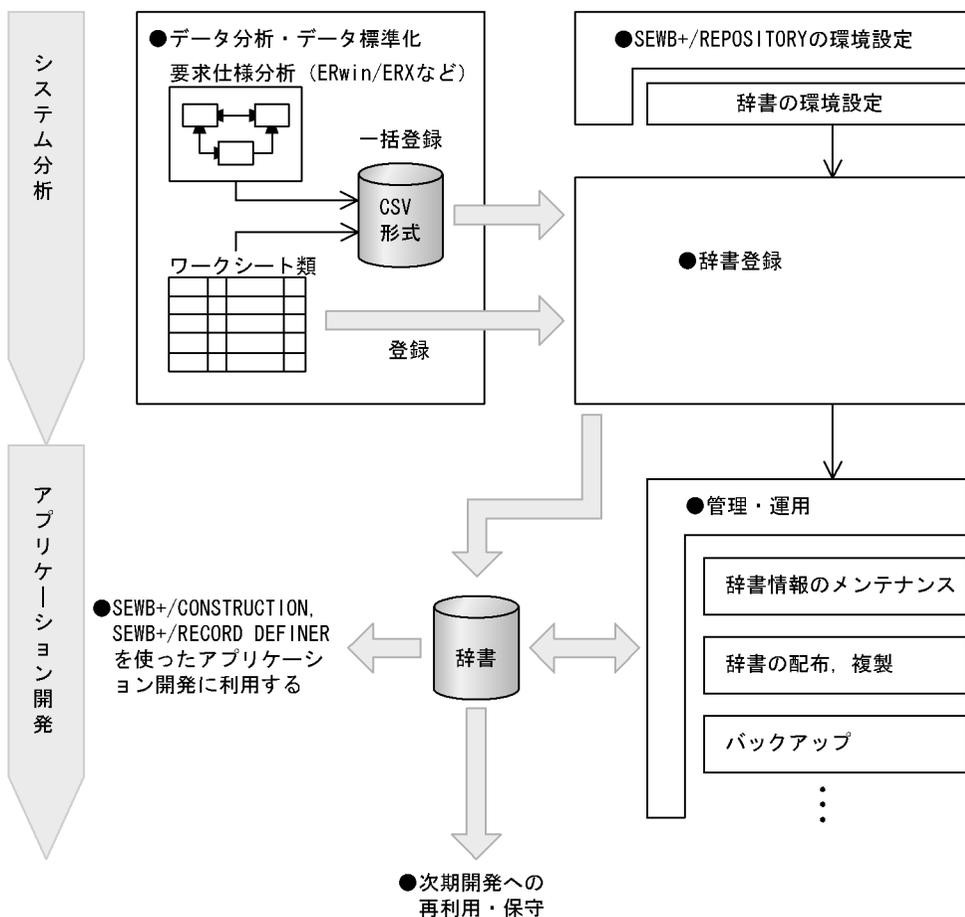
- CSV 形式ファイルの出力（CSV 出力）

データ項目または業務ルールの辞書情報を CSV 形式ファイルへ出力します。

1.5 辞書に関する作業の流れ

ここでは、SEWB+/REPOSITORY の環境に辞書を構築して、その辞書を管理・運用するまでの作業例を説明します。新システムの開発を例に、おおまかな作業の流れを図 1-10 に示します。

図 1-10 新規システム開発での作業例



図で示した新システムの開発を例に、各作業の概要を説明します。

(1) 辞書に蓄積するデータの分析と設計

システム分析段階で、辞書に蓄積するデータを分析し、情報をどのように辞書に蓄積するかを検討します。

辞書の設計の詳細は、「2. 辞書の設計」を参照してください。

辞書の構造と仕組みの理解

1. 概要

SEWB+/REPOSITORY では、一般的なシステム開発を想定して、標準の辞書の枠組みを提供しています。この標準的な枠組みを基に、プロジェクトの開発体制や開発する業務システムに合わせて、COBOL や C などのアプリケーション開発に使用するプログラミング言語、システムで使用する日本語や英語などの国語、命名ルール、検査項目などの辞書の構造をカスタマイズできます。

データ分析の結果を辞書にスムーズに引き継いで、辞書を有効に利用しながらデータ中心の開発を進めていくには、辞書の構造や仕組みを考慮したデータ分析が必要になります。このため、実際のデータ分析作業に入る前に、あらかじめ辞書の構造と仕組みについて理解しておくことをお勧めします。

データ分析と標準化、辞書の設計

- データ分析

データ中心アプローチの方法論に基づいて辞書に格納するデータ項目と業務ルールを分析・設計します。方法論に従って、正規化、名称・属性の標準化をして、一貫性のあるデータの体系を作ります。この作業では、辞書登録のために、分析した内容と、辞書に登録できる内容を一致させるよう考慮します。データ分析では、辞書に登録するデータ項目、ドメインを分析します。また、各データ項目に特有の処理（業務ルール）を抽出して整理します。

- データモデリングの情報を連携

システム開発の上流工程を支援する CASE ツールを使用している場合、この情報を引き継いだ辞書登録を検討します。

例えば、データモデリングを支援するツール「ERwin/ERX」で作成した設計情報を、CSV 形式ファイルを経由して辞書登録します。また、業務設計を支援するツール「Xupper」で作成した設計情報を ERwin/ERX に連携し、さらに CSV 形式ファイルを経由して辞書登録します。正規化によってデータの重複を排除したデータモデルから、エンティティやエンティティ属性といった情報を辞書に連携できます。

- 既存資産の利用

現行システムの開発資産の有効利用を検討します。SEWB+/REPOSITORY では、SEWB3 のデータ項目辞書から辞書情報を移行して再利用できます。また、ORACLE、HiRDB などの ODBC インタフェースをサポートしている DBMS の既存データベースから、スキーマ定義情報を取り出し、辞書のデータ項目を生成して登録できます。

どの資産を引き継いで辞書に登録して利用するかを、データ分析と併せて検討します。

リポジトリで辞書をどのように管理・運用するか検討

リポジトリで辞書をどのように管理・運用するか検討します。

- 辞書の構成の検討

管理方法や用途によってマルチ辞書（辞書フォルダ）をどのように構成するか、検討します。

- 辞書のカスタマイズの検討

アプリケーション開発に使用するプログラミング言語、データ項目や業務ルール

に対する命名ルールや検査項目，検索用のフィールドをどのように設けるのかなどを検討，決定します。

- 運用ルールの検討

辞書の作成者と利用者の権限や役割，辞書の作成者が辞書情報を変更する際の作業・変更通知ルール，分散配布のルールなど，プロジェクトでの運用ルールを明確にします。

(2) 環境設定

SEWB+/REPOSITORY をインストールし，サーバおよびクライアントで構成されるシステム開発環境を構築します。この作業は，システム開発環境全体を管理するシステム管理者が行います。なお，SEWB+/REPOSITORY は，SEWB+ 基本開発環境のインストールを実行することで，インストールできます。

SEWB+/REPOSITORY の基本環境の設定

まず，次の基本環境を構築します。詳細な手順や設定内容については，マニュアル「SEWB+/REPOSITORY 運用ガイド」を参照してください。

- OS の設定

SEWB+/REPOSITORY を Windows 2000 Server，Windows Server 2003 または Windows Server 2003 x64 で稼働させるための通信環境を設定します。

- データベースの初期設定

リポジトリを格納する前提データベースシステムの環境を設定します。

- ユーザ，グループの登録

リポジトリ内のシステム開発資源を操作できる利用者（ユーザ）をユーザ登録ファイルに登録します。ここで登録されたユーザが，辞書情報を作成，更新したり，辞書情報にアクセスしたりできます。また，各ユーザを，例えば，辞書の管理をするグループ，システム設計者のグループというように，プロジェクトや役割などでグルーピングします。このグループを，グループ登録ファイルに登録します。

- SEWB+/REPOSITORY の初期設定

リポジトリを格納するデータベース名称，ドキュメントの格納ディレクトリやバージョン管理システムなどを初期設定ユーティリティを使用して設定します。

- 管理者パスワードの設定

環境構築ユーティリティや各種コマンドなど，サーバに用意されている機能はシステム管理者だけが利用できます。このための，システム管理者用のパスワードを設定します。

辞書の準備

基本環境の設定を終えたあと，環境構築ユーティリティを使用してプロジェクトに合わせた辞書の環境を準備します。詳細は，「3.1 辞書の準備」を参照してください。

- 辞書フォルダの作成

辞書フォルダを作成し，マルチ辞書を構成します。

- 辞書の環境設定

プロジェクトや開発体制に合わせて辞書をカスタマイズします。プログラミング

1. 概要

言語、国語区分、タイプ、検索に使用するためのフィールド名、検査時に適用する検査項目、命名ルール、GUI 属性などを設定します。また、リポジトリブラウザを使ったデータ項目、業務ルール編集時の文字数制限を設定します。

なお、検査項目や命名ルールは、標準で用意されているもののほかに、独自の規則を記述した DLL を作成して登録することもできます。名称に使用する文字セットの制限など、プロジェクト独自の検査項目や命名ルールを定義できます。

- タイプマッピングの設定

ODBC 入力や ERwin/ERX と連携した CSV 入力を使って辞書登録をする際のタイプの変換規則を設定します。プロジェクトに合わせて、標準で用意されている変換規則をカスタマイズします。

- 標準提供の同一項目用業務ルールの登録

SEWB+/REPOSITORY では、実際の業務に使用することを想定した同一項目用業務ルール（同一項目間転記）を提供しています。プロジェクトの必要に応じて、あらかじめ辞書に登録しておきます。

（3）辞書登録

辞書の運用を始める前に、データ分析結果を辞書に登録します。リポジトリブラウザのデータ項目編集機能、業務ルール編集機能を使ってデータ項目、業務ルールを登録します。コマンドを使って一括登録することもできます。どちらの場合も、辞書情報が正しく定義されているか検査したり、継承や結合関係を定義したり、定義内容を変更したりするなどの作業や、登録結果の確認には、リポジトリブラウザを使用します。

（4）辞書の管理・運用

アプリケーション開発で辞書の利用が開始されてからの管理・運用作業について説明します。

辞書情報のメンテナンス

- 辞書情報の追加、変更、削除

リポジトリブラウザのデータ項目編集・作成機能、結合項目の構成機能、業務ルール編集・作成機能などを使用して辞書情報をメンテナンスします。

- 辞書情報の検査

環境設定で設定した検査項目に従って、辞書情報の内容を検査します。

- 関連検索、変更波及の調査

辞書情報を更新する際などに、その変更波及を調査します。リポジトリブラウザの検索機能、関連ブラウザ機能などを使用して変更波及の範囲を確認します。関連ブラウザでツリー形式で表示される資源の関連図は、SEWB+ で定義した情報の印刷を支援するツール「SEWB+/REPORT MANAGER」と連携することで、インパクトレポートとして印刷できます（インパクトレポートの印刷は、リポジトリブラウザの関連ブラウザから実行します）。

- 辞書情報の印刷、文書作成

SEWB+/REPORT MANAGER を使用して、データ項目または業務ルールの定義内容や一覧を印刷します（定義内容および一覧の印刷は、リポジトリブラウザから実行します）。

辞書情報を CSV 形式ファイルに出力したあと、Microsoft Excel や OFIS/POL などの帳票出力に適したアプリケーションを使って印刷する方法もあります。また、この CSV 形式ファイルを使って必要に応じて辞書情報を加工し、辞書の作成者とアプリケーションの開発者の間でやり取りするための文書を作成したりします。

辞書情報のセキュリティ

辞書フォルダ、データ項目または業務ルール単位にアクセス権を設定し、辞書情報を保全します。アクセス権の設定によって、辞書情報を作成・更新できるユーザ、辞書情報を参照してシステム開発に利用するユーザを分け、辞書の一貫性を守ります。

異なるサーバへの辞書の配布、複製

辞書を複数のサーバに分散して運用している場合、エクスポート・インポートユーティリティを使用して、辞書情報のサーバ間の配布、複製をします。エクスポート・インポートは、対象の範囲を全体または増分情報のどちらかを選択できるため、効率良く作業できます。

定期的なバックアップ

辞書は、システム開発プロジェクト共通の重要な情報資源といえます。運用ルールを定めて、定期的に（例えば、業務終了後に毎日、一週間おきなどに）辞書情報を外部媒体に退避（バックアップ）することをお勧めします。予期しない障害が発生して辞書が破壊された場合でも、バックアップしておけば、その内容をリポジトリに回復（リストア）できます。バックアップおよびリストアは、Windows 2000 Server、Windows Server 2003 または Windows Server 2003 x64 のファイルシステムに用意されている機能を使用します。環境構築ユーティリティを使用すれば、バックアップのためのコマンドファイルを生成できます。詳細は、マニュアル「SEWB+/REPOSITORY 運用ガイド」を参照してください。

(5) 次期開発への再利用、保守

システム開発の終了後、そのシステムの開発に利用した辞書を保守します。また、次期システムの開発に、継続して辞書を利用していきます。なお、辞書情報の見直しが必要な場合は、次期システム開発を始める前に（新システムの開発時と同様に）、辞書を整備しておくことをお勧めします。

- 開発システム単位の管理
システム開発に利用した辞書を凍結用に用意した別の辞書フォルダにエクスポート・インポートしたり、外部媒体に辞書をエクスポートして保守に備えます。
- システム保守（システムの拡張）への利用
次期システムの開発へ利用します。

2

辞書の設計

この章では、システム開発の上流工程で辞書を設計していく手順を紹介し、手順に沿いながらデータ項目と業務ルールの設計方法について説明します。

2.1 設計手順

2.2 ドメインの分析

2.3 名称の標準化

2.4 データ項目の設計

2.5 レコード構造の設計

2.6 業務ルールの設計

2.7 辞書登録の確認

2.1 設計手順

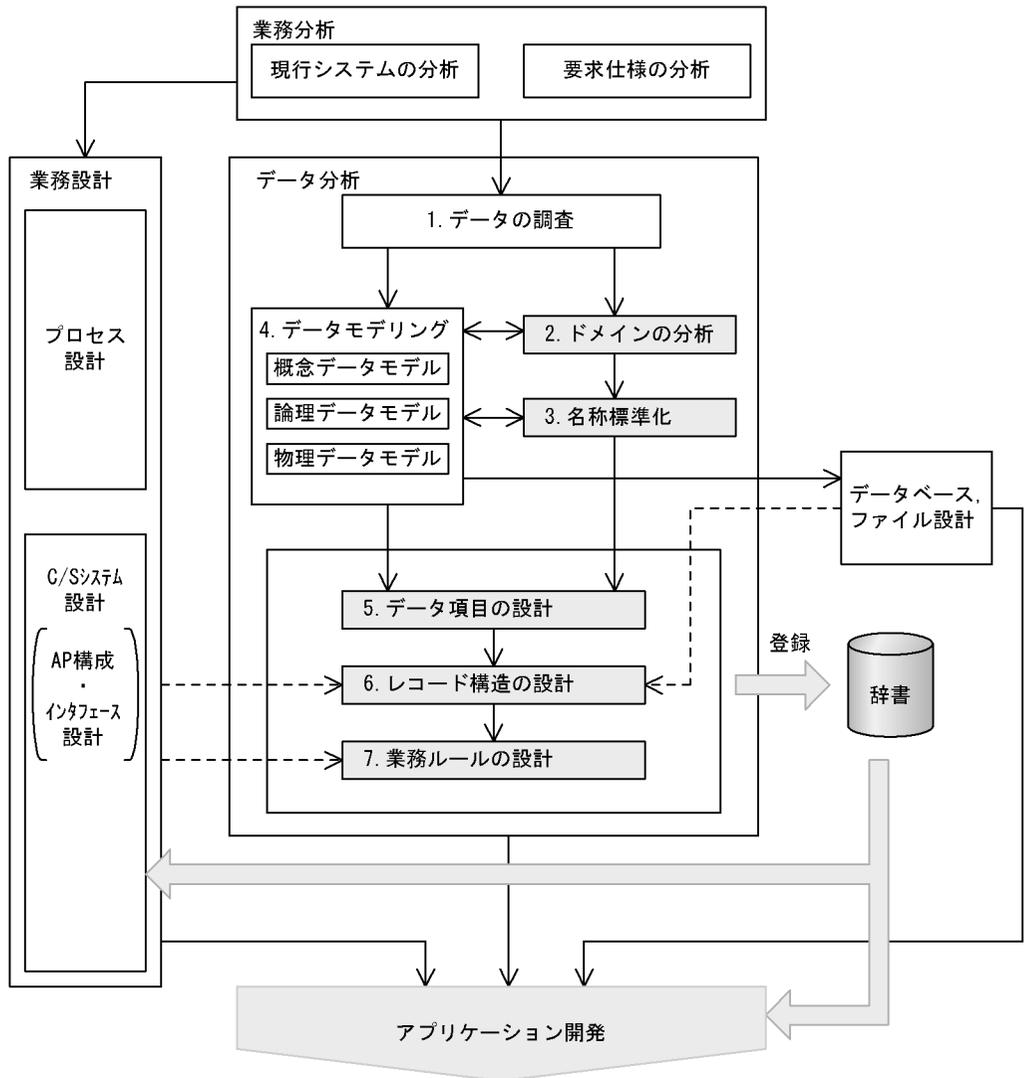
辞書に登録するデータ項目および業務ルールは、システム開発の基盤となる情報です。アプリケーションを標準化して、システム開発や保守の効率を向上させるためには、アプリケーション開発に先駆けて、データ中心アプローチの考え方に基づいたデータ分析が必要です。ここでは、そのデータ分析の作業の中で、辞書の仕組みを生かしてデータ項目や業務ルールを効果的に利用するための、辞書の設計手順を説明します。

(1) データ分析の中での辞書の設計手順

データ中心アプローチでは、現行システムや、目標とする新システム、全体的または長期的なニーズを基にデータ分析をします。データ分析とは、データを共有資源として活用するために業務システムで使用するデータの要素や相互関係を分析し、その名称、構造、意味、表現形式や値の制限などを定義する作業をいいます。このデータ分析の過程で、システム開発で共有するデータ項目や業務ルールを設計し、辞書に登録していきます。

ここでは、データ分析の中で、どのような手順で辞書を設計すればよいのかについて説明します。辞書の設計手順と、システム開発の上流工程での位置づけを、図 2-1 に示します。

図 2-1 辞書の設計手順



注 でマークされた作業については、このマニュアルの2.2~2.6で詳細に説明しています。

1. 業務で使用するデータを調査します

業務分析によって求められた、分析対象とする現行システムと、新システムの要求仕様の内容を引き継ぎ、業務で使用するデータを調査します。ここでは、システム開発のすべての工程で使用するデータが対象になります。現行システムからは、既存の帳票、画面、データベース、ファイルなどからデータ項目を抽出し、そのデータの属性、性質や制約条件を詳細に調査します。このとき、あらかじめ収集しておいた運用マニュアルなどを参考資料とします。また、システムで新しく使用するデータについても同様に、データの属性や性質などについて詳細に検討します。

2. ドメインを分析します

2. 辞書の設計

1. の調査結果を基に、業務システムに使用するデータから、共通の値域（タイプ、けた数、意味など）を持つデータをグルーピングします。これが、ドメインとなります（例：「年月日」、「金額」など）。システム開発に効果的に使用できる辞書を設計するには、あらかじめ、このドメインを分析する作業が必要です。ドメインの考え方や、分析の進め方については、「2.2 ドメインの分析」を参照してください。
3. 名称基準を決め、それに従って名称を標準化します
データ項目は、辞書の中で最も基本となる情報です。このデータ項目を効率良く利用するには、データ項目の名称によって各データを識別し、その意味を知ることができなければなりません。このために、あらかじめ名称基準を定めておき、それに基づいて名称を決めるようにします。詳細は、「2.3 名称の標準化」を参照してください。
4. データモデルを作成します（データモデリング）
データモデリングとは、業務システムの静的なデータの構造を、全体像としてまとめることをいいます。データを正規化する作業も、このデータモデリングに含まれます。作成するデータモデルは、実体を表すエンティティとエンティティ属性、エンティティ間の関連であるリレーションシップで構成されます。
大規模システムのデータモデリングには、これを支援するツールを利用すると効率良く作業できます。データモデリングの進め方については、使用するツールのマニュアルを参照してください。ERwin/ERXを使用する場合には、作成したデータモデルから、エンティティやエンティティ属性の情報を、CSV形式ファイルを経由してデータ項目辞書に登録できます。
5. データ項目を設計してデータ項目辞書に登録します
ドメインの分析、名称の標準化およびデータモデリングの結果を基に、各データ項目の名称、意味、タイプ、けた数などを検討し、データ項目辞書に登録します。まず、2. で分析したドメインに相当するデータ項目をデータ項目辞書に登録します。そのあと、ドメインのデータ項目を継承して、下位のデータ項目をデータ項目辞書に登録します。詳細は、「2.4 データ項目の設計」を参照してください。なお、データ項目の設計には、SEWB+/STANDARD-DICTIONARY が提供する標準データ項目を利用できます。概要については、「付録 A 標準データ項目辞書の紹介」を参照してください。
6. レコード構造を設計してデータ項目辞書に登録します
アプリケーション開発の準備として、サーバアプリケーション、およびクライアントとサーバのインタフェースに使用するレコードの構造やレポート出力に使用するレコードの構造をデータ項目辞書に用意します。このレコード構造には、データベースやファイルの設計結果、業務設計から求められた C/S（クライアントサーバ）システムのインタフェース仕様などを反映します。
SEWB+/CONSTRUCTION, EUR Professional Edition および APPGALLERY Enterprise でレコード構造をどのように利用するかを考慮しながら設計します。レコードの構造は 5. で登録したデータ項目を組み合わせ、登録します。詳細は、「2.5 レコード構造の設計」を参照してください。
なお、レコード構造は、SEWB+/RECORD DEFINER でも定義できます。SEWB+/RECORD DEFINER では、データ項目辞書に登録されていないデータ項目

(COBOL 言語の FILLER など) をレコードの構成要素として使用できます。

7. 業務ルールを設計して業務ルール辞書に登録します

ドメイン分析や業務設計の結果を基に、各データ項目特有の処理を抽出して、業務ルール辞書に登録します。業務ルールは、SEWB+/CONSTRUCTION で作成するアプリケーションにどのように利用するかを考慮しながら設計します。データ項目と同様に、まず、継承関係の上位のデータ項目(ドメインに相当するデータ項目)の業務ルールから登録し、それに基づいて下位のデータ項目の業務ルールに登録します。詳細は、「2.6 業務ルールの設計」を参照してください。なお、業務ルールの設計には、SEWB+/STANDARD-DICTIONARY が提供する標準業務ルールを利用できます。概要については、「付録 A 標準データ項目辞書の紹介」を参照してください。

(2) 効果的なデータ分析するには

データ分析の分析対象は、データ分析をする際の目的やプロジェクトの開発体制や制限によって、現行システムを主体とするか、現行システムに対するニーズとするか、またはその両方とするかが異なります。しかし、実際にデータ分析をするときには、現行システムや現行システムに対するニーズおよび目標を取り混ぜた分析が必要といえます。また、システムが大規模であったり、複雑なシステムを扱うときや、分析作業に多くの時間を費やせない場合など、ユーザのおかれるシステム開発の環境はさまざまです。

したがって、データ分析作業は、一律ではないと考えられます。データ分析を始める前に、自プロジェクトではどのような分析方法、手順を採用すれば開発の目的を達成できるか検討し、状況に応じた適切な手段でデータ分析を実施するようにしてください。

2.2 ドメインの分析

ドメインとは、共通の意味と属性を持つデータ要素の集合をいいます。例えば、業務で使用する年月日データ「入社年月日」、「異動年月日」、「従業員生年月日」の値をすべて和暦で表すとき、これらの共通要素をまとめ、ドメイン「和暦年月日」を求められます。SEWB+/REPOSITORYの辞書の仕組みを効果的にシステム開発に利用するには、辞書を設計する前に、このドメインを分析することが重要な作業になります。

業務で使用するデータ全体に対してドメインを分析し、整理しておくことで、辞書の設計作業の効率を向上できます。また、辞書に登録するデータ項目や業務ルールを標準化し、それぞれの品質を向上できます。

2.2.1 ドメインの分析とは

ドメインの分析とは、業務で使用するデータから、共通の値域と制約条件を抽出し、標準化することです。

(1) 値域、制約条件とは

ドメインを理解するために、まず値域と制約条件について解説します。

- 値域とは、データのタイプ、けた数、意味、構成、値の範囲などのことをいいます。
- 制約条件とは、データ項目を利用するに当たって守らなければならない制約をいいます。制約条件は、次の2種類に分類できます。

1. データの値が妥当であるかを判断する条件

例えば、「受注年月日」は、必ず企業の「営業日」でなければならないという業務の規則があるとき、受注年月日が営業日であるかどうかを検証する条件のことです。

2. データの値が導き出されるルール

そのデータが、ある別のデータから、計算、編集、形式変換などの処理を経て導き出されるルールのことです。例えば、「売り上げ金額」が、「単価×数量」で求められるものであれば、この計算式が「売り上げ金額」の導出ルールになります。

(2) ドメインを分析し、その値域と制約条件を整理する

ドメインを分析し、その値域と制約条件を整理する手順を、次に説明します。

1. ドメインを抽出する

ドメインを抽出するには、まず、業務に使用するデータを調査し、その中から共通の値域を持つデータをグルーピングします。値域が共通なデータは、その制約条件も共通であると考えられます。それぞれのグループに対して、共通の制約条件を洗い出します。このようにして、データ共通の値域と制約条件をまとめ、整理したものがドメインとなります。

2. 必要に応じて、ドメインを階層化する

ドメインによってグルーピングされたデータのなかで、さらに共通の値域や制約条件

を抽出できれば、ドメインを階層化し、下位ドメインとして整理しておきます。

3. ドメインに名称を付ける

それぞれのドメインには、業務の中で一意に識別できるよう、ユニークな名称を付けます。このドメインの名称は、データ項目の名称に反映します。詳細は「2.3 名称の標準化」を参照してください。

ドメインの抽出例を次に示します。

ドメイン名称	定義	データの例
コード	企業の主要な管理対象を識別するための文字列（値）で、必ず個々の値が示す対象が決められるデータ	顧客コード 製品コード 銀行コード
番号	企業の主要な管理対象を識別するための文字列（値）で、時々刻々と発生する出来事（取り引き）などを識別するデータ	社員番号 受注番号
区分	ある特定の条件を満たすかどうかを示すデータ	引当区分 割引区分
年月日	ある特定時点の年、月、日を表すデータ	営業年月日 発注年月日 登録年月日
金額	円やドルなどの通貨を表すデータ	売上げ金額 割引金額

このようにして整理したドメインと、その値域、制約条件を基に、辞書のデータ項目と業務ルールを設計します。

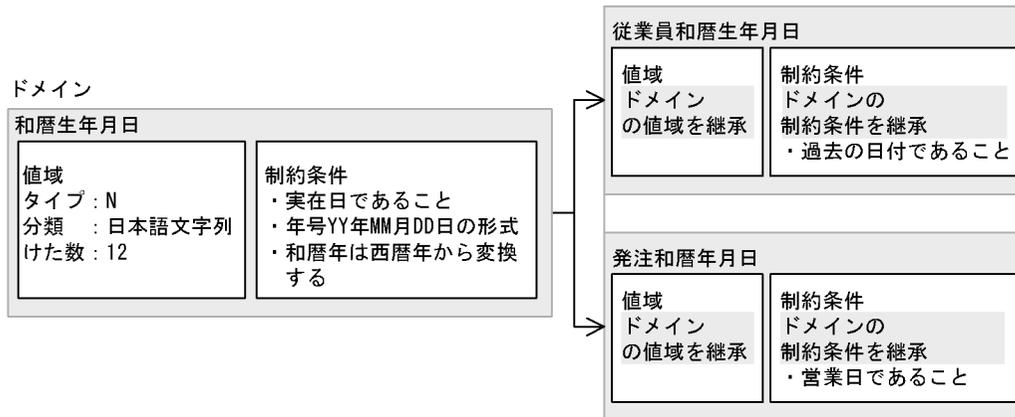
2.2.2 ドメインの値域と制約条件を基に、辞書を設計する

ドメインの分析で求められた、ドメインの名称と値域は、継承関係の上位に位置づけるデータ項目としてデータ項目辞書に登録します。また、ドメインの制約条件は、ドメインのデータ項目に対応する業務ルールとして業務ルール辞書に登録します。

ドメインにまとめられた共通の値域と制約条件は、データ項目の「継承」という仕組みによって、ドメインにグルーピングされたデータ項目に引き継がれます。この継承の考え方を、図 2-2 に示します。

2. 辞書の設計

図 2-2 ドメインの値域と制約条件の継承



この継承の仕組みを有効に利用するために、ドメインを分析して、その内容を確定する（値域や制約条件を確定する）ということは、重要な意味を持っています。

あらかじめドメインの共通の値域と制約条件を整理しておき、辞書の設計にドメインの内容を取り込めば、個々のデータ項目や業務ルールを設計するとき、そのデータ項目または業務ルールに固有の値域や制約条件だけを決定すればよいことになります。すでにドメインに整理されている値域や制約条件を重ねて決定する必要がなく、設計作業の効率を向上でき、標準化を促進できます。

また、ドメインの内容を変更しなければならないときでも継承の仕組みを利用することによって、その変更内容を自動的にデータ項目や業務ルールに展開できるため、作業の効率化を図れます。

SEWB+/STANDARD-DICTIONARY では、標準的なドメインのデータ項目（標準データ項目）と、そのデータ項目に対応する業務ルール（標準業務ルール）を用意した標準データ項目辞書を提供しています。また、この標準データ項目辞書を利用して各プロジェクト専用の辞書を自動生成する辞書生成機能を提供しています。このため、SEWB+/STANDARD-DICTIONARY を使用すれば、辞書を構築する作業の効率と、辞書の品質の向上を図れます。このプログラムプロダクトの概要については、「付録 A 標準データ項目辞書の紹介」を参照してください。

2.3 名称の標準化

データ項目は、辞書およびシステムの中で最も基本となる情報です。このデータ項目を共用資源とするために、データ項目には、次のような要素が求められます。

- 区別できる データ項目はすべてユニークな名称を持つ
- 安定している 組織の変更やシステムの変更の影響を受けない
- わかりやすい データ項目の名称を見れば、その意味がわかる
- 扱いやすい 使いたいデータ項目がすぐに探し出せる

この要素を兼ね備えたデータ項目を設計するには、あらかじめプロジェクトの名称基準を定め、それに基づいて適切なデータ項目の名称を付けることが大切です。データ項目の名称は、システム開発に携わる人およびシステムの利用者のだれもが、共通に内容や意味を理解し、使いこなせるものである必要があります。ここでは、お勧めする標準的なデータ項目の名称基準と、名称の標準化について説明します。

2.3.1 データ項目の名称基準

辞書に登録するデータ項目の名称は、「実体名 + 属性名 + ドメイン名」の形式で構成することをお勧めします。これは、「ある対象を、ある特性に限定して、ある範囲（単位）で」表現した名称です。このため、対象となるデータ本来の姿を表す名称を付けられます。この名称基準は、意味や内容がわかりやすいだけでなく、データ項目を辞書に登録したあと、名称でソートしたり、絞り込んだりしやすいという特長があります。さらに、データ項目そのものの内容に着目しているため、システムや業務の変更の影響を受けにくい名称になります。

（例） 名称基準に従ったデータ項目の名称

実体名：商品
 属性名：種別
 ドメイン名：コード
 データ項目の名称：商品種別コード

次に、データ項目の名称を構成する「実体名」、「属性名」および「ドメイン名」の意味を説明します。

実体名

業務およびシステムで管理する、人、物、場所または概念が実体です。この実体に名称を付けたものが、実体名となります。実体は、データ項目を抽出したあと、データモデリング（データの正規化）の過程で求められます。実体は、エンティティに相当します。

（例）

人：顧客、従業員
 物：商品
 場所：本店、営業所、事業所
 概念：取引、営業、仕入れ、発注、受注

2. 辞書の設計

属性名

実体の特性（特徴や性質）を表すものです。この属性に名称を付けたものが、属性名となります。属性によって、実体の表す意味を限定できます。一つの実体に対し、複数の属性を持てます。属性も実体と同様に、データモデリング（データの正規化）の過程で求められます。

（例）

店別，月別，最大，最小，平均，合計

ドメイン名

ドメインとは、共通の意味と属性を持つデータ項目の集合のことです。辞書を設計するために、このドメインの分析が重要な作業であることは「2.2 ドメインの分析」で見てきました。データ項目が属すドメインの名称を、データ項目の末尾に付けます。データ項目に共通の値域や、制約条件がまとめられたドメインの名称を反映することによって、データ項目の内容を適切に判断できるようになります。

（例）

年月日，コード，番号，金額，数量

2.3.2 名称を標準化する

データ項目の名称の定義項目には、「データ項目名」、「標準名称」、「フリガナ」、「言語別（プログラミング言語別）の名前」があります。データ項目の設計、登録を始める前に、それぞれの定義項目についてプロジェクト独自の名称基準を定めておきましょう。特に、データ項目名および標準名称は、前述した標準的な名称基準に沿った名称にすることをお勧めします。

SEWB+/REPOSITORY では、プロジェクトで定めた名称基準を命名ルール として定義することによって、データ項目に定義した名称が、その基準に合わせて付けられているかどうかを診断する機能を提供しています。詳細は、「2.7.1 定義内容を検査する」および「3.1.1(11) 命名ルールを定義する」を参照してください。

注

命名ルールは、環境構築ユティリティを使用して定義します。データ項目の名称のほか、業務ルールの定義項目「業務ルール名」、「標準名称」および「フリガナ」についても診断できます。

2.4 データ項目の設計

データ項目は、アプリケーション開発に使用する情報の最も基本的な単位です。辞書の設計は、このデータ項目の設計から始めます。

2.4.1 データ項目に定義する情報

データ項目にどのような情報を定義するかについて説明します。

(1) データ項目の定義項目

データ項目の定義項目の一覧を、表 2-1 に示します。データ項目名や標準名称などに定義する文字数の上限はカスタマイズできます。なお、「データ項目名」は、必ず定義しなければなりません。ほかの定義項目は、必要に応じて定義を省略できます。

表 2-1 データ項目の定義項目

	定義項目	文字数の上限		文字セット
		標準値	設定範囲	
名称	データ項目名	30	1 ~ 92	英数字, カナ文字, 日本語文字 ⁴
	標準名称 ¹	46	1 ~ 92	英数字, カナ文字, 日本語文字
	フリガナ	30	1 ~ 92	英数字, カナ文字, 日本語文字
属性	分類	-	-	(リストからの選択で定義)
	けた数	9	固定	数字
	小数部けた数	9	固定	数字
	反復回数	5	固定	数字
付加情報	フィールド 1 ~ 20	30	1 ~ 255	任意
	コメント ¹	6500	固定	任意
言語別	名前	46	1 ~ 92	英数字, カナ文字, 日本語文字
詳細情報 ²	タイプ	-	-	(リストからの選択で定義)
	フリー定義	6500	固定	英数字, カナ文字, 日本語文字
	タイプ修飾情報 ³	30	固定	英数字, カナ文字, 日本語文字
	取りうる値	6500	固定	英数字, カナ文字, 日本語文字 ⁵
	初期値	160	固定	英数字, カナ文字, 日本語文字
	言語別フィールド	30	1 ~ 255	任意

(凡例)

固定：文字数の上限は固定されていることを示します。

注 1

2. 辞書の設計

4種類までの国語区別に定義できます。

注 2

10種類までの言語区別に定義できます。

注 3

言語区分「COBOL 又は OOCOBOL」の編集文字列または「IDL(CORBA)」のシーケンスを定義します。

注 4

「:」および「|」は使えません。また、文字列の途中に空白文字を含めてはなりません。

注 5

記述規則に従います。詳しくは「(5)(d) 取りうる値」を参照してください。

(2) 名称

名称は、データ項目辞書で管理される個々のデータ項目を識別し、その意味を知るための定義情報です。名称の定義内容を基に、目的のデータ項目を検索したり、指定した条件でソートしたりするために利用できます。この情報を定義するリポジトリブラウザのダイアログ（[基本属性]タブ）を、図 2-3 に示します。

図 2-3 データ項目を作成するダイアログ（[基本属性]タブ）

(a) データ項目名

データ項目を識別するために付ける名称です。データ項目名は必ず定義しなければなりません。

(例)

SHOHIN-CODE

(b) データ項目種別

登録するデータ項目を結合項目にする場合には、データ項目種別に「結合項目」を選択します。さらに、レコードの構造を定義する結合項目の場合には「最上位結合項目」を指定します。結合項目、最上位結合項目については、「2.4.3 データ項目の結合」および「2.5 レコード構造の設計」を参照してください。

データ項目を結合項目にしない場合には「単項目」を選択します。

(c) 標準名称

データ項目の内容や意味をわかりやすくするために付ける、システム標準の名称です。4種類までの国語別に定義できます。

(例)

日本語の標準名称：商品コード

(d) フリガナ

日本語読みの名称（フリガナ）を付けます。国語区分「英語」の場合は、フリガナを定義できません。

(例)

ショウヒンコード

注

データ項目を作成するダイアログの「基本属性」タブに表示される「親データ項目名」について説明します。定義対象としているデータ項目は、親データ項目名に表示されたデータ項目から、定義情報を直接、継承していることを示します。継承については、「2.4.2 データ項目の継承」を参照してください。

(3) 属性

属性は、そのデータ項目で構成されるファイルやデータベースを定義するための開発言語共通の定義情報です。この情報を定義するリポジトリブラウザのダイアログ（「ソース生成属性」タブ）を、図 2-4 に示します。

2. 辞書の設計

図 2-4 データ項目を作成するダイアログ ([ソース生成属性] タブ)

(a) 分類

データ項目のデータの形式を、次に示す分類区分から選択します。

- 英数字文字列データ：英数字や記号（半角文字）で表す文字列のデータ
- 日本語文字列データ：日本語文字（全角文字）で表す文字列のデータ
- 整数データ：整数で表す数値のデータ
- 正整数データ：正の整数で表す数値のデータ
- 実数データ：実数で表す数値のデータ
- 時刻データ：時刻のデータ
- 日付データ：日付のデータ
- 分類なし：上記分類以外のデータまたは分類できないデータ

プログラムで使用する詳細なデータの形式は、言語別詳細情報のタイプに定義します。分類区分に応じて定義できる言語別のタイプも異なります。この対応関係は、環境構築ユーティリティでカスタマイズできます。標準で設定されている分類とタイプの対応を、表 2-2 から表 2-7 に示します。

また、言語区分 SQL のタイプを HiRDB として使用する場合のカスタマイズ例を表 2-8 に示します。

表 2-2 分類区分とタイプの対応 (COBOL 又は OOCOBOL)

分類区分	タイプ
英数字文字列データ	X 英数字項目, Z 数字編集項目, フリー定義
日本語文字列データ	N 漢字項目, フリー定義

分類区分	タイプ
整数データ	9 符号なし外部 10 進項目, S 符号付き外部 10 進項目, U 符号なし内部 10 進項目, P 符号付き内部 10 進項目, BU 符号なし 2 進項目, B 符号付き 2 進項目, E 外部浮動小数点項目, D 内部浮動小数点項目, フリー定義
正整数データ	9 符号なし外部 10 進項目, S 符号付き外部 10 進項目, U 符号なし内部 10 進項目, P 符号付き内部 10 進項目, BU 符号なし 2 進項目, B 符号付き 2 進項目, E 外部浮動小数点項目, D 内部浮動小数点項目, フリー定義
実数データ	9 符号なし外部 10 進項目, S 符号付き外部 10 進項目, U 符号なし内部 10 進項目, P 符号付き内部 10 進項目, E 外部浮動小数点項目, D 内部浮動小数点項目, フリー定義
時刻データ	X 英数字項目, フリー定義
日付データ	X 英数字項目, フリー定義
分類なし	X 英数字項目, Z 数字編集項目, N 漢字項目, 9 符号なし外部 10 進項目, S 符号付き外部 10 進項目, U 符号なし内部 10 進項目, P 符号付き内部 10 進項目, BU 符号なし 2 進項目, B 符号付き 2 進項目, E 外部浮動小数点項目, D 内部浮動小数点項目, 1 内部プール項目, 8 外部プール項目, T アドレスデータ項目, フリー定義

注

アンダーラインは、初期値を示します。

表 2-3 分類区分とタイプの対応 (C 又は C++)

分類区分	タイプ
英数字文字列データ	<u>char</u> 文字型, char* 文字型 (ポインタ), フリー定義
日本語文字列データ	char 文字型, <u>char*</u> 文字型 (ポインタ), フリー定義
整数データ	short 符号付き短整数型, unsigned short 符号なし短整数型, <u>int</u> 符号付き整数型, unsigned int 符号なし整数型, long 符号付き長整数型, unsigned long 符号なし長整数型, float 単精度浮動小数点型, double 倍精度浮動小数点型, long double 拡張精度浮動小数点型, フリー定義
正整数データ	short 符号付き短整数型, unsigned short 符号なし短整数型, <u>int</u> 符号付き整数型, <u>unsigned int</u> 符号なし整数型, long 符号付き長整数型, unsigned long 符号なし長整数型, float 単精度浮動小数点型, double 倍精度浮動小数点型, long double 拡張精度浮動小数点型, フリー定義

2. 辞書の設計

分類区分	タイプ
実数データ	<u>float</u> 単精度浮動小数点型, <u>double</u> 倍精度浮動小数点型, <u>long double</u> 拡張精度浮動小数点型, フリー定義
時刻データ	<u>char</u> 文字型, フリー定義
日付データ	<u>char</u> 文字型, フリー定義
分類なし	<u>char</u> 文字型, <u>char*</u> 文字型 (ポインタ), <u>short</u> 符号付き短整数型, <u>unsigned short</u> 符号なし短整数型, <u>int</u> 符号付き整数型, <u>unsigned int</u> 符号なし整数型, <u>long</u> 符号付き長整数型, <u>unsigned long</u> 符号なし長整数型, <u>float</u> 単精度浮動小数点型, <u>double</u> 倍精度浮動小数点型, <u>long double</u> 拡張精度浮動小数点型, フリー定義

注

アンダーラインは, 初期値を示します。

表 2-4 分類区分とタイプの対応 (SQL)

分類区分	タイプ
英数字文字列データ	<u>CHAR</u> 固定長文字列, <u>VARCHAR2</u> 可変長文字列, <u>LONG</u> 可変長長文字列, フリー定義
日本語文字列データ	<u>CHAR</u> 固定長文字列, <u>VARCHAR2</u> 可変長文字列, <u>LONG</u> 可変長長文字列, フリー定義
整数データ	<u>NUMBER</u> 固定小数点数, <u>NUMBERf</u> 浮動小数点数, フリー定義
正整数データ	<u>NUMBER</u> 固定小数点数, <u>NUMBERf</u> 浮動小数点数, フリー定義
実数データ	<u>NUMBER</u> 固定小数点数, <u>NUMBERf</u> 浮動小数点数, フリー定義
時刻データ	フリー定義
日付データ	<u>DATE</u> 日付, フリー定義
分類なし	<u>CHAR</u> 固定長文字列, <u>VARCHAR2</u> 可変長文字列, <u>LONG</u> 可変長長文字列, <u>NUMBER</u> 固定小数点数, <u>NUMBERf</u> 浮動小数点数, <u>DATE</u> 日付, フリー定義

注

アンダーラインは, 初期値を示します。

表 2-5 分類区分とタイプの対応 (IDL(CORBA))

分類区分	タイプ
英数字文字列データ	<u>char</u> 文字型, <u>string</u> バウンディッド文字列型, <u>unbounded string</u> アンバウンディッド文字列型, フリー定義

分類区分	タイプ
日本語文字列データ	char 文字型, string バウンディッド文字列型, unbounded string アンバウンディッド文字列型, wstring ワイド文字列型, wchar ワイド文字型, フリー定義
整数データ	short 符号付き短整数型, unsigned short 符号なし短整数型, long 符号付き長整数型, unsigned long 符号なし長整数型, float 単精度浮動小数点型, double 倍精度浮動小数点型, long long 符号付き長々整数型, unsigned long long 符号なし長々整数型, フリー定義
正整数データ	short 符号付き短整数型, unsigned short 符号なし短整数型, long 符号付き長整数型, unsigned long 符号なし長整数型, float 単精度浮動小数点型, double 倍精度浮動小数点型, long long 符号付き長々整数型, unsigned long long 符号なし長々整数型, フリー定義
実数データ	float 単精度浮動小数点型, double 倍精度浮動小数点型, long double 拡張倍精度長浮動小数点型, fixed 符号付き固定小数点型, フリー定義
時刻データ	char 文字型, string バウンディッド文字列型, wchar ワイド文字型, wstring ワイド文字列型, フリー定義
日付データ	char 文字型, string バウンディッド文字列型, wchar ワイド文字型, wstring ワイド文字列型, フリー定義
分類なし	char 文字型, string バウンディッド文字列型, unbounded string アンバウンディッド文字列型, unsigned short 符号なし短整数型, short 符号付き短整数型, unsigned long 符号なし長整数型, long 符号付き長整数型, double 倍精度浮動小数点型, float 単精度浮動小数点型, octet オクテット型, boolean 論理型, unsigned long long 符号なし長々整数型, any 任意データ型, fixed 符号付き固定小数点型, long long 符号付き長々整数型, wstring ワイド文字列型, long double 拡張倍精度長浮動小数点型, フリー定義, wchar ワイド文字型

注

アンダーラインは、初期値を示します。

表 2-6 分類区分とタイプの対応（帳票定義）

分類区分	タイプ
英数字文字列データ	C 文字データ, M 画像データ, フリー定義
日本語文字列データ	C 文字データ, M 画像データ, フリー定義

2. 辞書の設計

分類区分	タイプ
整数データ	<u>I</u> 数値データ, B 集計対象外数値データ, SP 特殊編集レベルデータ, フリー定義
正整数データ	<u>I</u> 数値データ, B 集計対象外数値データ, SP 特殊編集レベルデータ, フリー定義
実数データ	<u>I</u> 数値データ, B 集計対象外数値データ, フリー定義
時刻データ	<u>C</u> 文字データ, フリー定義
日付データ	CY 年次データ, MO 月次データ, DA 日次データ, HS 日付・時刻データ, C 文字データ, フリー定義
分類なし	C 文字データ, I 数値データ, B 集計対象外数値データ, CY 年次データ, MO 月次データ, DA 日次データ, HS 日付・時刻データ, M 画像データ, SP 特殊編集レベルデータ, フリー定義

注

アンダーラインは、初期値を示します。

表 2-7 分類区分とタイプの対応 (Java)

分類区分	タイプ
英数字文字列データ	<u>byte</u> バイト型, フリー定義
日本語文字列データ	<u>char</u> 文字型, フリー定義
整数データ	<u>byte</u> バイト型, <u>short</u> 符号付き短整数型, <u>int</u> 符号付き整数型, <u>long</u> 符号付き長整数型, フリー定義
正整数データ	<u>byte</u> バイト型, <u>short</u> 符号付き短整数型, <u>int</u> 符号付き整数型, <u>long</u> 符号付き長整数型, フリー定義
実数データ	<u>float</u> 単精度浮動小数点型, <u>double</u> 倍精度浮動小数点型, フリー定義
時刻データ	<u>フリー</u> 定義
日付データ	<u>フリー</u> 定義
分類なし	<u>byte</u> バイト型, <u>short</u> 符号付き短整数型, <u>int</u> 符号付き整数型, <u>long</u> 符号付き長整数型, <u>float</u> 単精度浮動小数点型, <u>double</u> 倍精度浮動小数点型, <u>char</u> 文字型, <u>boolean</u> 論理型, <u>フリー</u> 定義

注

アンダーラインは、初期値を示します。

表 2-8 分類区分とタイプの対応のカスタマイズ例（言語区分 SQL のタイプを HiRDB として使用する場合）

分類区分	タイプ
英数字文字列データ	CHAR 固定長文字列, VARCHAR 可変長文字列, MCHAR 固定長混在文字列, MVARCHAR 可変長混在文字列, フリー定義
日本語文字列データ	NCHAR 固定長各国文字列, NVARCHAR 可変長各国文字列, MCHAR 固定長混在文字列, MVARCHAR 可変長混在文字列, フリー定義
整数データ	SMALLINT 短整数, INT 整数, DEC 固定小数点数, SMALLFLT 単精度浮動小数点数, FLOAT 倍精度浮動小数点数, フリー定義
正整数データ	SMALLINT 短整数, INT 整数, DEC 固定小数点数, SMALLFLT 単精度浮動小数点数, FLOAT 倍精度浮動小数点数, フリー定義
実数データ	DEC 固定小数点数, SMALLFLT 単精度浮動小数点数, FLOAT 倍精度浮動小数点数, フリー定義
時刻データ	TIME 時刻, フリー定義
日付データ	DATE 日付, フリー定義
分類なし	CHAR 固定長文字列, VARCHAR 可変長文字列, NCHAR 固定長各国文字列, NVARCHAR 可変長各国文字列, MCHAR 固定長混在文字列, MVARCHAR 可変長混在文字列, SMALLINT 短整数, INT 整数, DEC 固定小数点数, SMALLFLT 単精度浮動小数点数, FLOAT 倍精度浮動小数点数, TIME 時刻, DATE 日付, フリー定義

注 1

アンダーラインは、初期値を示します。

注 2

言語区分 SQL のタイプを HiRDB として使用する場合は、上表の例のように環境構築ユーティリティでカスタマイズすることをお勧めします。また、言語区分 SQL のタイプを HiRDB として使用する場合は、標準に設定されているタイプの組み合わせを変更する必要があります。タイプの組み合わせの変更については「3.1.1(8) タイプの組み合わせを変更する」を参照してください。

(b) けた数

データのけた数（文字列データの場合は文字数）を定義します。分類区分「分類なし」を選択した場合は、けた数を定義しなくてもかまいません。

けた数の範囲 は、言語区分や言語別のタイプによって異なります。

標準に設定されているけた数の範囲を、表 2-9 から表 2-14 に示します。なお、けた数の範囲は必要に応じ、環境構築ユーティリティを使ってカスタマイズすることもできます。

2. 辞書の設計

注

検査機能でチェックされるけた数の範囲を指します。

表 2-9 タイプに対応するけた数 (COBOL 又は OOCOBOL)

タイプ	けた数範囲
X 英数字項目	1 ~ 16,777,215
Z 数字編集項目	1 ~ 18
N 漢字項目	1 ~ 16,383
9 符号なし外部 10 進項目	1 ~ 18
S 符号付き外部 10 進項目	1 ~ 18
U 符号なし内部 10 進項目	1 ~ 18
P 符号付き内部 10 進項目	1 ~ 18
BU 符号なし 2 進項目	1 ~ 18
B 符号付き 2 進項目	1 ~ 18
E 外部浮動小数点項目	1 ~ 15
D 内部浮動小数点項目	1 ~ 15
1 内部ブール項目	1 ~ 2,034
8 外部ブール項目	1 ~ 2,034
T アドレスデータ項目	(チェック対象外)

表 2-10 タイプに対応するけた数 (C 又は C++)

タイプ	けた数範囲
char 文字型	1 ~ 999,999,999 ¹
char * 文字型 (ポインタ)	1 ~ 999,999,999 ¹
short 符号付き短整数型	1 ~ 4
unsigned short 符号なし短整数型	1 ~ 4
int 符号付き整数型	1 ~ 9 ²
unsigned int 符号なし整数型	1 ~ 9 ²
long 符号付き長整数型	1 ~ 9
unsigned long 符号なし長整数型	1 ~ 9
float 単精度浮動小数点型	1 ~ 6
double 倍精度浮動小数点型	1 ~ 15
long double 拡張精度浮動小数点型	1 ~ 33 ³

注 1

SEWB+/REPOSITORY で定めた制限値を示しています。けた数の制限は実行するシステムによって異なるため、その制限に合わせてカスタマイズすることをお勧めします。

注 2

32 ビット表現の場合を示します。

注 3

128 ビット表現の場合を示します。

表 2-11 タイプに対応するけた数 (SQL)

タイプ	けた数範囲
CHAR 固定長文字列	1 ~ 255 ¹
VARCHAR2 可変長文字列	1 ~ 2,000
LONG 可変長長文字列	1 ~ 99,999,999 ²
NUMBER 固定小数点数	1 ~ 38
NUMBERf 浮動小数点数	1 ~ 38
DATE 日付	(チェック対象外)
VARCHAR 可変長文字列	1 ~ 32,000
NCHAR 固定長各国文字列	1 ~ 15,000
NVARCHAR 可変長各国文字列	1 ~ 16,000
MCHAR 固定長混在文字列	1 ~ 15,000
MVARCHAR 可変長混在文字列	1 ~ 16,000
SMALLINT 短整数	1 ~ 4
INT 整数	1 ~ 9
DEC 固定小数点数	1 ~ 29
SMALLFLT 単精度浮動小数点数	1 ~ 6
FLOAT 倍精度浮動小数点数	1 ~ 15
TIME 時刻	(チェック対象外)

注 1

言語区分 SQL のタイプを HiRDB として使用する場合は、けた数の範囲を 1 ~ 30,000 に変更しておくことをお勧めします。

注 2

SEWB+/REPOSITORY で定めた制限値を示しています。けた数の制限値は実行するシステムによって異なるため、その制限に合わせてカスタマイズすることをお勧めします。

表 2-12 タイプに対応するけた数 (IDL(CORBA))

タイプ	けた数範囲
char 文字型	1 ~ 999,999,999

2. 辞書の設計

タイプ	けた数範囲
string バウンディッド文字列型	1 ~ 999,999,999
unbounded string アンバウンディッド文字列型	(チェック対象外)
short 符号付き短整数型	1 ~ 4
unsigned short 符号なし短整数型	1 ~ 4
long 符号付き長整数型	1 ~ 9
unsigned long 符号なし長整数型	1 ~ 9
float 単精度浮動小数点型	1 ~ 6
double 倍精度浮動小数点型	1 ~ 15
boolean 論理型	(チェック対象外)
octet オクテット型	(チェック対象外)
any 任意データ型	(チェック対象外)
long long 符号付き長々整数型	1 ~ 19
unsigned long long 符号なし長々整数型	1 ~ 20
long double 拡張倍精度浮動小数点型	1 ~ 28
wchar ワイド文字型	1 ~ 2
wstring ワイド文字列型	1 ~ 999,999,999
fixed 符号付き固定小数点型	1 ~ 10

注

SEWB+/REPOSITORY で定めた制限値を示しています。けた数の制限値は実行するシステムによって異なるため、その制限に合わせてカスタマイズすることをお勧めします。

表 2-13 タイプに対応するけた数 (帳票定義)

タイプ	けた数範囲
C 文字データ	1 ~ 4,096
I 数値データ	1 ~ 4,096
B 集計対象外数値データ	1 ~ 4,096
CY 年次データ	4 ~ 4
MO 月次データ	6 ~ 6
DA 日次データ	8 ~ 8
HS 日付・時刻データ	4 ~ 19
M 画像データ	1 ~ 256
SP 特殊編集レベルデータ	1 ~ 2

表 2-14 タイプに対応するけた数 (Java)

タイプ	けた数範囲
byte バイト型	1 ~ 3
char 文字型	1 ~ 999,999,999
short 符号付き短整数型	1 ~ 4
int 符号付き整数型	1 ~ 9
long 符号付き長整数型	1 ~ 19
float 単精度浮動小数点型	1 ~ 6
double 倍精度浮動小数点型	1 ~ 15
boolean 論理型	(チェック対象外)

注

SEWB+/REPOSITORY で定めた制限値を示しています。けた数の制限値は実行するシステムによって異なるため、その制限に合わせてカスタマイズすることをお勧めします。

(c) 小数部けた数

分類区分「実数データ」を選択した場合に、小数点以下のけた数を定義します。小数部けた数には、定義項目「けた数」の定義内容より小さい値を定義しなければなりません。

(d) 反復回数

データが反復構造を持つ場合に、反復回数を定義します。反復回数を指定したデータ項目は、C 言語や IDL では配列を意味し、反復回数に指定した値が要素数になります。

(4) 付加情報

付加情報は、自由に使用できる付属的な定義情報です。この付加情報を基に、目的のデータ項目を検索することもできます。この情報は、データ項目を作成するダイアログの「基本属性」タブに定義します。

(a) フィールド 1 ~ 20

プロジェクト独自の情報を定義するなどして、自由に使用できるフィールドです。環境構築ユティリティを使用し、任意のフィールド名称を付けられます。また、リポジトリブラウザのダイアログに表示されるフィールドの数もカスタマイズできます。

(例)

1. 環境構築ユティリティを使って、フィールド 1 の名称に「管理部署」を設定する
2. リポジトリブラウザのデータ項目を作成するダイアログでフィールド「管理部署」に「製品企画部」を定義する

このように定義すると、リポジトリブラウザで「管理部署」に定義された名称をキーにしてデータ項目を検索できます。

2. 辞書の設計

(b) コメント

データ項目の使用目的、内容、注意事項などを記述します。4種類までの国語別に定義できます。

(例)

日本語のコメント：各商品に付ける識別用コード

(5) 言語別詳細情報

言語別詳細情報は、データ項目の定義内容を、実際のアプリケーションに使用するソースコード（COBOL ソースプログラムや IDL など）に反映するための定義情報です。10種類までの言語区別に、それぞれの言語で規定された規則に従って定義します。この情報は、データ項目を作成するダイアログの「ソース生成属性」タブに定義します。

(a) 名前

データの名前を言語別に定義します。例えば、「COBOL 又は OOCOBOL」ではデータ項目名を、「C 又は C++」では変数名を、「IDL(CORBA)」ではインタフェース名などを記述します。言語別の名前に記述する文字数の制限や文字の種類は、使用する言語の規則に従ってください。

- 「COBOL 又は OOCOBOL」、「C 又は C++」の名前の記述には
記述する文字数は、SEWB+/CONSTRUCTION で言語別の名前がソースコードに生成されるときに付加される接頭語または接尾語についても留意する必要があります。これらが付加された結果、文字数の制限を超えてしまわないように、あらかじめテンプレート作成者と言語別の名前に記述する文字数について検討することをお勧めします。

(例)

COBOL 又は OOCOBOL : SYOHIN-CODE

C 又は C++ : syohin-code

(b) タイプ

データの形式を表すタイプを、言語別に定義します。タイプは、英字の識別子と日本語名で表現され、設定されているレコード生成キーワードに従ってソースコードに対応づけられます。例えば、「COBOL 又は OOCOBOL」のタイプ「X 英数字項目」は、「PICTURE X()」に対応づけられています。なお、レコード生成キーワードは、環境構築ユティリティで編集できます。詳しくは「3.1.1(9) タイプに対するレコード生成キーワードを編集する」を参照してください。

分類区分に応じて、定義できる言語別のタイプが異なります。分類区分とタイプの対応の詳細は、「2.4.1(3)(a) 分類」を参照してください。

タイプ「フリー定義」を選択すると、各言語の文法に従ってデータの形式を直接、記述できます。フリー定義の言語別の記述方法を表 2-15 に示します。

表 2-15 タイプ「フリー定義」の記述方法

言語区分	記述内容	記述例
COBOL 又は OOCOBOL	PICTURE 句および USAGE 句の定義内容	PICTURE 9(5) PACKED-DECIMAL PICTURE ¥¥¥¥, ¥¥9
C 又は C++	基本型, ユーザ定義型, typedef 名など	BOOLEAN int*
SQL	SQL データ型	LARGE DECIMAL INTEGER
IDL (CORBA)	基本データ型, typedef 名など	unsigned short list
帳票定義	EUR Professional Edition のデータ種別記号	SP F
Java	ユーザ定義型, typedef 名など	int*

注

フリー定義に記述するデータ種別記号については、「表 5-3 データ種別記号とデータ種別」を参照してください。

(c) タイプ修飾情報 (編集文字列 / シーケンス)

タイプ修飾情報は、言語区分「COBOL 又は OOCOBOL」の編集文字列の定義項目と「IDL(CORBA)」のシーケンスの定義項目の総称です。この項目を定義できるのは、言語区分が「COBOL 又は OOCOBOL」または「IDL(CORBA)」の場合です。

リポジトリブラウザでは、言語区分に応じて「編集文字列」または「シーケンス」が定義項目の名称として表示されます。

- COBOL 又は OOCOBOL での編集文字列の定義
タイプに「Z 数字編集項目」または「E 外部浮動小数点項目」を定義した場合に、COBOL 言語の文法に従って編集文字列を記述します。

(例)

ZZZ,ZZZ,ZZ9

- IDL (CORBA) でのシーケンスの定義
タイプに指定したデータ型を、シーケンス (一次元可変長配列) とする場合にだけ定義します。
バウンディッド・シーケンスとして定義する場合には、最大要素数を指定します。また、アンバウンディッド・シーケンスとして定義する場合には、半角文字のアスタリスク (*) を指定します。
例えば、データ項目のタイプが「short 符号付き短整数型」で、シーケンスに「512」を指定した場合は、最大要素数 512 の short 型のバウンディッド・シーケンスという意味になります。これに対して、同じタイプでシーケンスに「*」を指定した場合には、short 型のアンバウンディッド・シーケンスという意味になります。シーケンス

2. 辞書の設計

に何も指定しないと、単なる short 型になります。

シーケンスは、結合項目に対して定義することもできます。この場合、その結合項目が表す構造体がシーケンスとして定義されていることを意味します。

注

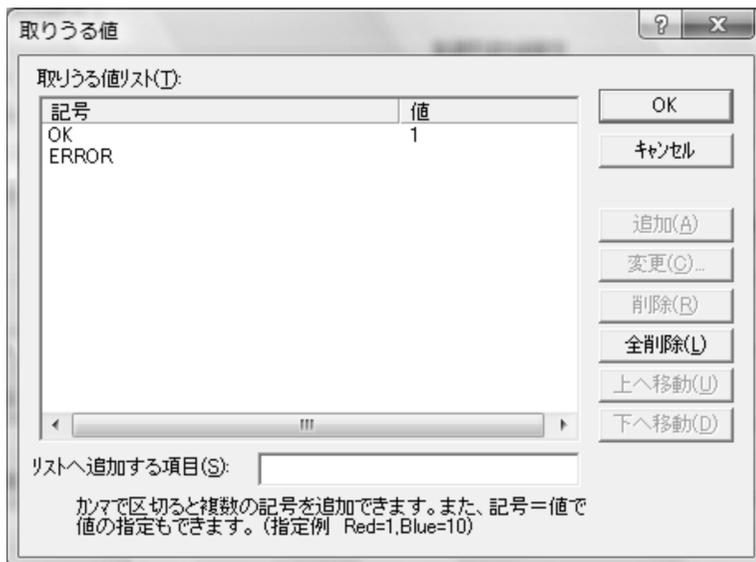
シーケンスの定義は IDL 特有のものです。シーケンスに相当する概念を「COBOL 又は OOCOBOL」、「C 又は C++」などの言語区分で定義することはできません。したがって、シーケンスを定義したデータ項目は、SEWB+/CS-DESIGN のオブジェクト定義専用のデータ項目として管理することをお勧めします。

(d) 取りうる値

扱うデータが限られるデータ項目で、それぞれの値に名前を付けて管理したい場合に定義する項目です。「取りうる値」では、識別のために付ける名前とデータ項目に格納する値の組み合わせを、言語別に定義します。

データ項目を作成するダイアログで [取りうる値] ボタンから表示されるダイアログを図 2-5 に示します。

図 2-5 取りうる値を定義するダイアログ



取りうる値は言語別に一覧で管理します。COBOL 言語での条件名や、C 言語や IDL の列挙定数（列挙型の識別子）のように、値に対応する名前を指定します。この名前を「記号」と呼びます。明示的に記号に値を割り当てる場合は C 言語の列挙子のように半角の「=」を使って値を記述します。複数の場合は半角のコンマ(,)で区切ります。

例えば、「実行結果」というデータ項目が「成功」と「失敗」のどちらかの状態だけを扱う場合、取りうる値の記号には、成功の「OK」と失敗の「ERROR」を指定します。さ

らに、OK にだけ値「1」を割り当てておくという指定 (OK = 1, ERROR) ができません。

なお、この項目を定義できるのは、言語区分が「SQL」、「帳票定義」以外で、タイプが「フリー定義」以外の場合です。

注

記号には全角および半角の空白と「,」,「=」は使えません。また、値に半角の「,」,「=」,「¥」を使う場合は、その直前に「¥」を付加する必要があります。

(e) 初期値

データ項目に初期値を定義する場合、各言語の文法に従って言語別に記述します。初期値の言語別の記述方法を次に示します。なお、「COBOL 又は OOCOBOL」での引用符の記述には、アポストロフィ (') を使うことをお勧めします。

なお、この項目を定義できるのは、言語区分が「SQL」、「IDL(CORBA)」、「帳票定義」以外の場合です。

- COBOL 又は OOCOBOL では、VALUE 句に定義する文字列または表意定数を記述します。

(例)

```
'顧客名' ALL '*' ZERO
```

- C 又は C++, Java では、初期化文字列を記述します。

(例)

```
100 "SHIKOKU"
```

(f) 言語別フィールド

自由に使用できる言語別のフィールドです。

2.4.2 データ項目の継承

データ分析で求められたドメインと、ドメインによってグルーピングされたデータ項目は、継承関係で結び付けてデータ項目辞書に登録します。継承とは、データ項目間で共通の性質を持つデータ項目から、その定義情報を複数のデータ項目に引き継ぐことです。この継承の仕組みを利用して、ドメインを継承関係の上位のデータ項目に登録し、ドメインによってグルーピングされたデータ項目を、その定義情報を引き継ぐ下位のデータ項目として登録します。

このように、ドメインの分析結果に基づいてデータ項目に登録すれば、データ項目を効率良く体系的に登録および管理できるとともに、定義情報の標準化を図れます。

ここでは、データ項目を設計および登録するための前提知識として、継承の考え方について解説します。

(1) ドメインと、ドメインにグルーピングされたデータ項目の登録

ドメインの分析結果をデータ項目辞書に登録するには、まず、最も共通的な値域を持つドメインを、継承関係の最上位のデータ項目として登録します。ドメインとなるデータ項目の登録を終えたあと、そのデータ項目から定義情報を継承して、ドメインにグルーピングされたデータ項目または下位のドメインに登録します。このようにして、データ項目同士に継承関係を定義したとき、データ項目の間には、関連種別「継承」の関連が自動的に付けられます。

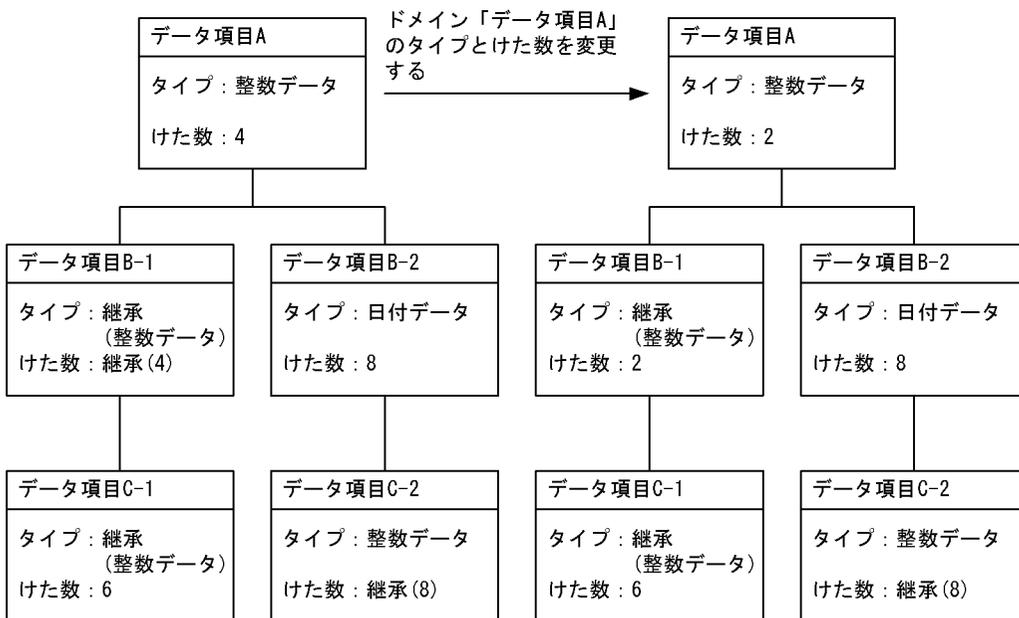
(2) ドメインから下位のデータ項目への定義情報の継承

ドメインのデータ項目の定義情報は、継承関係の下位のデータ項目に引き継がれます。したがって、ドメインにグルーピングされたデータ項目は、各データ項目固有の定義情報だけを定義し、登録すればよいことになります。

継承関係を定義しておけば、変更波及の管理も容易になります。継承関係を定義したあと、上位のデータ項目の内容を変更すると、下位のデータ項目に継承された定義情報にも、その変更が自動的に反映されます。ただし、下位のデータ項目固有の定義情報には、変更は反映されません。

定義情報の継承と変更波及の例を、図 2-6 に示します。

図 2-6 定義情報の継承と変更波及の例



「データ項目 B-1」の定義内容には、「データ項目 A」の変更がそのまま引き継がれます。これに対して、「データ項目 C-1」では、固有のけた数「6」を定義しているため、「データ項目 A」のけた数を変更しても、定義内容は変更されません。

(3) 定義情報を継承する定義項目の指定

それぞれの定義項目に対して、データ項目同士の継承関係を定義したときに、ドメイン（上位のデータ項目）の情報を必ず継承するか、リポジトリブラウザで任意に継承を指定するか、または継承を禁止するかを、環境構築ユティリティであらかじめ設定しておけます。これを「継承モード」といい、標準では任意に継承を指定することを示す「任意継承」が設定されています。

2.4.3 データ項目の結合

データ分析で求められたデータの構造は、複数のデータ項目同士を結合関係で結び付けて、データ項目辞書に登録します。結合とは、複数のデータ項目を順序づけて連結することです。この結合の仕組みを利用すれば、COBOL 言語での集団項目と基本項目の関係、C 言語での構造体とそれを構成する変数の関係、またはアプリケーションに使用するレコードとレコードを構成するデータ項目の関係といったデータの構造を、データ項目辞書に登録できます。

ここでは、データ項目を設計および登録するための前提知識として、結合の考え方を解説します。

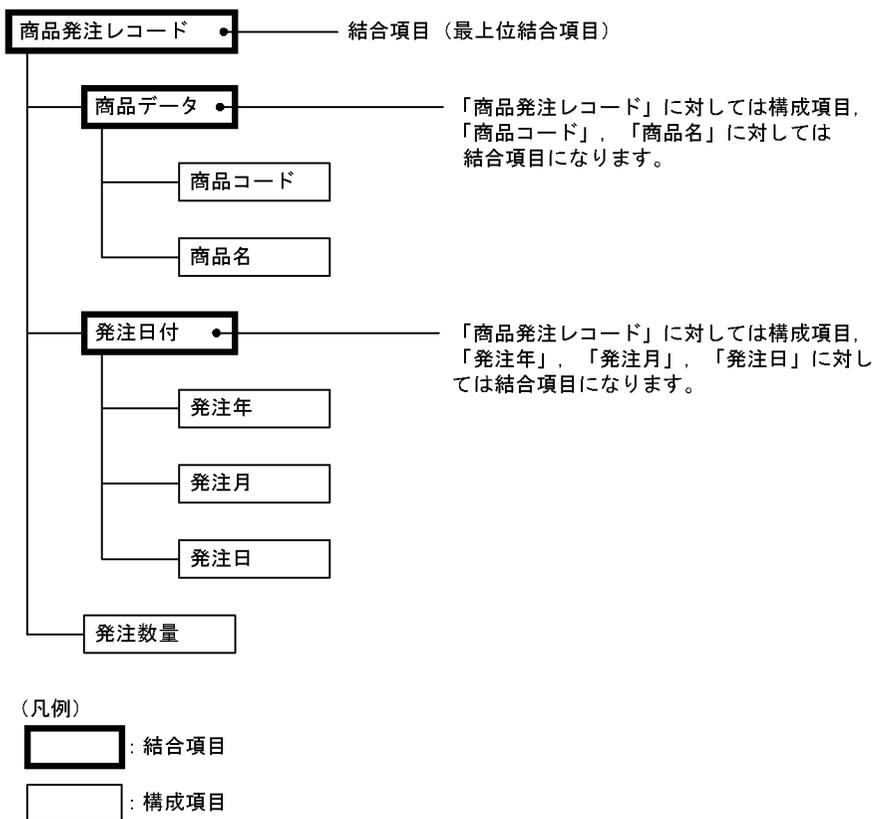
(1) 結合項目と構成項目

データ項目同士の関係を結合の関係でとらえる場合、複数のデータ項目をまとめるデータ項目を、結合項目といいます。結合項目は、COBOL 言語での集団項目、C 言語での構造体に相当します。また、結合項目の構成要素となるデータ項目を、構成項目といいます。例えば、データ項目「年月日」を複数のデータ項目「年」、「月」、「日」で構成する場合、「年月日」が結合項目、「年」、「月」、「日」が構成項目となります。

結合関係を定義するには、まず、基本的な要素となるデータ項目（構成項目）を設計および登録します。結合項目に登録するには、すでに登録されているデータ項目を順番に組み合わせて結合項目の構成を定義します。このようにして、データ項目同士に結合関係を定義したとき、データ項目の間には、関連種別「結合」の関連が自動的に付けられます。

結合関係は階層化できます。結合項目の構成の例を図 2-7 に示します。

図 2-7 結合項目の構成の例



この例では、「商品発注レコード」を最上位結合項目として定義しています。最上位結合項目とは、アプリケーションで使用するレコード構造が定義された結合項目を意味します。レコード構造の定義は、データ項目の設計および登録を終えたあと、データ項目辞書に登録します。詳細は、「2.5 レコード構造の設計」を参照してください。

(2) 結合項目の定義項目

結合項目は、一つのデータ項目として扱えますが、定義できる項目に限られます。結合項目の定義項目を、表 2-16 に示します。

表 2-16 結合項目の定義項目

定義項目		定義区分
名称	データ項目名 ¹	
	標準名称	
	フリガナ	
属性	分類 ²	-
	けた数	-

	定義項目	定義区分
	小数部けた数	-
	反復回数	
付加情報	フィールド 1 ~ 20	
	コメント	
言語別	名前	
詳細情報	タイプ	-
	フリー定義	-
	タイプ修飾情報	3
	取りうる値	-
	初期値	-
	言語別フィールド	

(凡例)

- : 必ず定義しなければならない定義項目
- : 定義を省略できる定義項目
- : 定義できない項目

注 1

データ項目名(結合項目の名称)は必ず定義しなければなりません。

注 2

「結合データ」が自動的に定義されます。

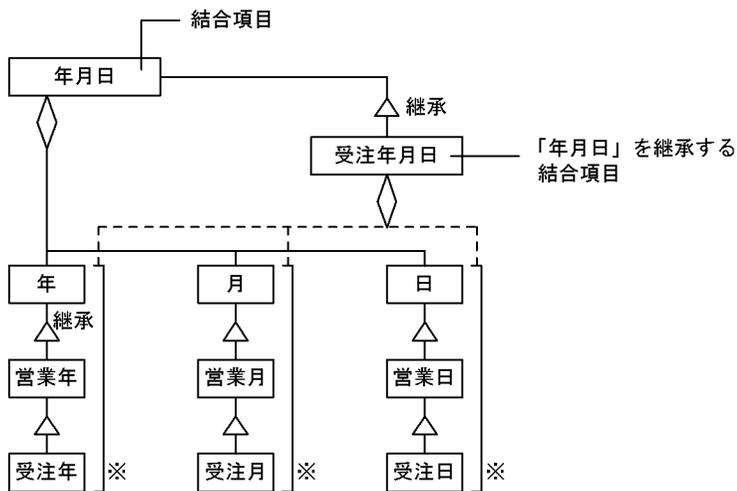
注 3

言語区分「IDL(CORBA)」のシーケンスだけ定義できます。

(3) 結合関係の継承

結合項目を、継承関係の上位のデータ項目として位置づけて、結合関係を継承できます。結合項目を継承するデータ項目は、その親となる結合項目と同じ構造(構成項目の並びや数が同じ)の結合項目になります。親となる結合項目の構成項目が継承関係の下位のデータ項目を持つ場合には、同じ位置の構成項目を、その下位のデータ項目の中から選択できます。この例を図 2-8 に示します。例えば、「受注年月日」の年を定義するための構成項目を、「年」、「営業年」または「受注年」から選択できます。

図 2-8 結合関係の継承の例



(凡例)

----- : 「受注年月日」が「年月日」から継承する結合関係

注

結合項目ではないデータ項目から、その定義情報を継承する結合項目を定義することはできません。

注

構成項目を選択できる範囲です。

2.4.4 ドメインを登録する

初めに、「西暦年月日」、「和暦年月日」、「コード」、「金額」などに代表されるような、最も共通的な要素を持つドメインを、データ項目辞書に登録します。

(1) ドメインを登録する辞書フォルダを準備する

まず、データ分析結果に従って、辞書フォルダをどのように分類し、辞書フォルダにどのようなドメインを登録するかを検討します。例えば、複数のシステム間で共通なドメインは、システムで共用する「システム共通辞書フォルダ」に登録し、システムに特有なドメインは、そのシステム専用の「Aシステム専用辞書フォルダ」に登録するようにします。

サーバの環境構築ユティリティを使い、データ項目辞書および業務ルール辞書を管理する辞書フォルダを準備します。詳細は、「3.1.2 辞書フォルダを作成する」を参照してください。

(2) ドメインのデータ項目を登録する

ドメインを登録する辞書フォルダを決定したあと、その辞書フォルダが管理するデータ項目辞書に、ドメインのデータ項目を整備していきます。次に、リポジトリブラウザを使ったデータ項目の定義について説明します。

1. リポジトリブラウザウィンドウで、データ項目を登録する辞書フォルダのデータ項目辞書を選びます。
2. [リポジトリ] - [新規作成] - [単項目]を選びます。
データ項目を作成するダイアログが表示されます。ドメインの名称をデータ項目名に定義し、継承関係の下位のデータ項目に引き継ぐための共通の定義情報を定義します。

(3) 結合項目を登録する

データ項目が結合項目の場合は、構成項目となるデータ項目をあらかじめ登録しておきます。次に、結合項目ブラウザおよびリポジトリブラウザを使った、結合項目の定義について説明します。

結合項目ブラウザを使った結合項目の定義手順

1. 結合項目ブラウザウィンドウで、結合項目を登録する辞書フォルダを選びます。
2. [リポジトリ] - [新規作成] - [結合項目]を選びます。
データ項目を作成するダイアログが表示されます。
3. データ項目名を定義したあと、[結合項目構成]タブを表示します。
「結合項目構成」欄で、すでに登録されたデータ項目を選択して結合項目の構成を編集します。

なお、結合項目の構成だけを編集したい場合は、ドラッグ & ドロップまたは[構成項目]メニューを使って操作してください。

リポジトリブラウザを使った結合項目の定義手順

1. リポジトリブラウザウィンドウで、結合項目を登録する辞書フォルダのデータ項目辞書を選びます。
2. [リポジトリ] - [新規作成] - [結合項目]を選びます。
データ項目を作成するダイアログが表示されます。
3. データ項目名を定義したあと、[結合項目構成]タブを表示します。
「結合項目構成」欄で、すでに登録されたデータ項目を選択して結合項目の構成を編集します。

なお、結合項目の構成だけを編集したい場合は、目的の結合項目を選択して[リポジトリ] - [結合項目の構成]を選んでください。

2.4.5 ドメインにグルーピングされたデータ項目を登録する

ドメインのデータ項目をデータ項目辞書に登録したあと、ドメインの定義情報を継承して下位のデータ項目を登録します。次にリポジトリブラウザを使った下位のデータ項目

2. 辞書の設計

の定義について説明します。

1. リポジトリブラウザウィンドウで、定義情報を継承したいドメインのデータ項目を選びます。
2. ドメインのデータ項目が単項目か結合項目かによって、[リポジトリ] - [新規作成] - [単項]を選ぶか、または[リポジトリ] - [新規作成] - [結合項目]を選びます。データ項目を作成するダイアログが表示されます。このダイアログには、1. で選んだデータ項目の定義情報が継承されて表示されています。そのデータ項目に固有の情報だけを定義してください。

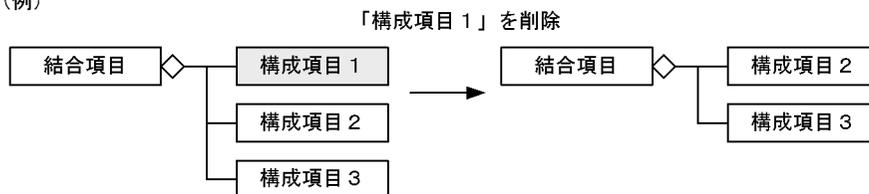
2.4.6 データ項目を削除する

データ項目を設計および登録する作業では、必要に応じて、いったん登録したデータ項目をリポジトリブラウザを使って削除しなければならない場合があります。

結合項目の構成としたデータ項目や業務ルールの適用データ項目としたデータ項目でも削除できます。それぞれの削除の例を次に示します。

- 結合項目の構成としたデータ項目を削除する
結合項目の構成としたデータ項目を削除すると、結合項目との関連が削除されデータ項目自体も削除されます。

(例)

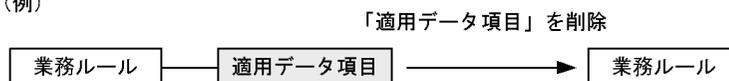


(凡例)

- : 削除するデータ項目
- : 関連

- 業務ルールの適用データとしたデータ項目を削除する
業務ルールの適用項目としたデータ項目を削除すると、業務ルールとの関連が削除されデータ項目自体も削除されます。

(例)



(凡例)

- : 削除するデータ項目
- : 関連

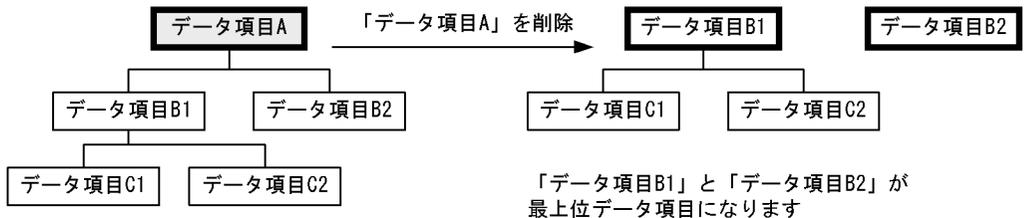
データ項目同士に継承関係または結合関係を定義しているデータ項目を削除した場合、相互の関連も自動的に削除および変更されます。次に、データ項目の削除に伴った関連

の変更規則について説明します。

- 最上位階層のデータ項目を削除する

削除対象のデータ項目から子の位置にあったデータ項目との継承の関連および結合の関連も同時に削除されます。そして、子の位置にあったデータ項目が自動的に最上位階層に繰り上げられます。

(例)



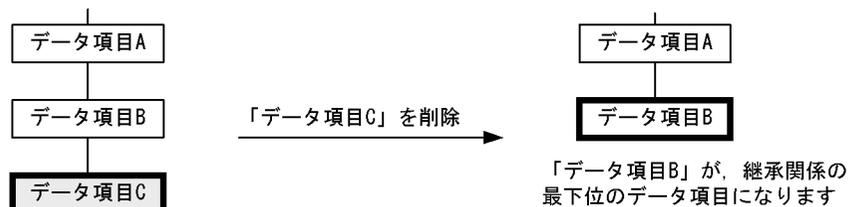
(凡例)

- : 継承関係の最上位のデータ項目
- : 削除するデータ項目
- : 継承の関連

- 最下位階層のデータ項目を削除する

削除対象のデータ項目から親の位置にあったデータ項目との継承の関連および結合項目であれば結合の関連も同時に削除されます。

(例)



(凡例)

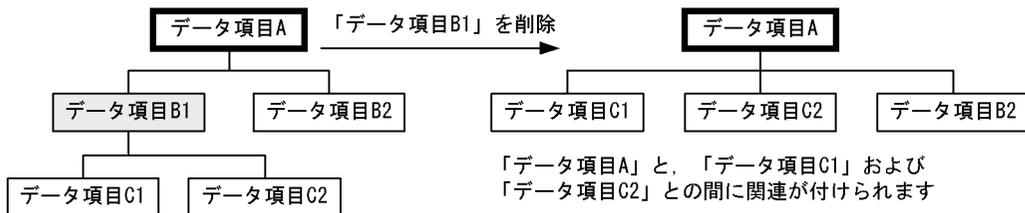
- : 継承関係の最下位のデータ項目
- : 削除するデータ項目
- : 継承の関連

- 中間層のデータ項目を削除する

削除対象のデータ項目から親の位置にあったデータ項目との継承の関連と、子の位置にあったデータ項目との継承の関連も削除されます。結合項目であれば結合の関連も削除されます。さらに、削除したデータ項目に関連していた双方のデータ項目同士を直接結び関連が付けられます。

2. 辞書の設計

(例)



(凡例)

-  : 継承関係の最上位のデータ項目
-  : 削除するデータ項目
-  : 継承の関連

2.5 レコード構造の設計

データ項目を整えたあと、RDB (Relational Data Base) やファイルのレコード構造や、通信のインタフェーステーブルなどアプリケーション開発に使用するレコードの構造を、最上位結合項目としてデータ項目辞書に登録、またはレコード定義ファイルを作成します。このマニュアルでは、これを「レコード定義」と呼びます。レコード定義は、次のアプリケーション開発の場面で利用されます。

- SEWB+/CONSTRUCTION でのデータ定義
- EUR Professional Edition でのレポート設計
- APPGALLERY Enterprise での OpenTP1 クライアントアプリケーションの作成

なお、アプリケーション開発で使用するツールによって、どちらのレコード定義を使用するかどうかを選択します。

使用するツール	最上位結合項目 (辞書のレコード定義)	レコード定義ファイル (SEWB+/RECORD DEFINER のレコード定義)
SEWB+/CONSTRUCTION		
EUR Professional Edition		×
APPGALLERY Enterprise		×

(凡例)

- : 使用できる
- ×

注

SEWB+/RECORD DEFINER では、最上位結合項目 (レコードの構成情報を含む) を、簡単にレコード定義に取り込めます。また、データ項目辞書に登録されていない項目 (COBOL 言語の FILLER など) についてもレコード定義の構成要素として使用できます。このため、SEWB+/CONSTRUCTION でのアプリケーション開発の場合には、SEWB+/RECORD DEFINER のレコード定義を使用することを推奨します。

2.5.1 辞書のレコード定義 (最上位結合項目) とプログラムの関係

SEWB+/CONSTRUCTION を使ったアプリケーション開発の準備として、サーバアプリケーション、およびクライアントとサーバのインタフェースに使用するレコードの構造をデータ項目辞書またはレコード定義ファイルに用意します。なお、レコード定義ファイルは、SEWB+/RECORD DEFINER で作成します。詳細は、マニュアル「SEWB+/RECORD DEFINER ユーザーズガイド」を参照してください。

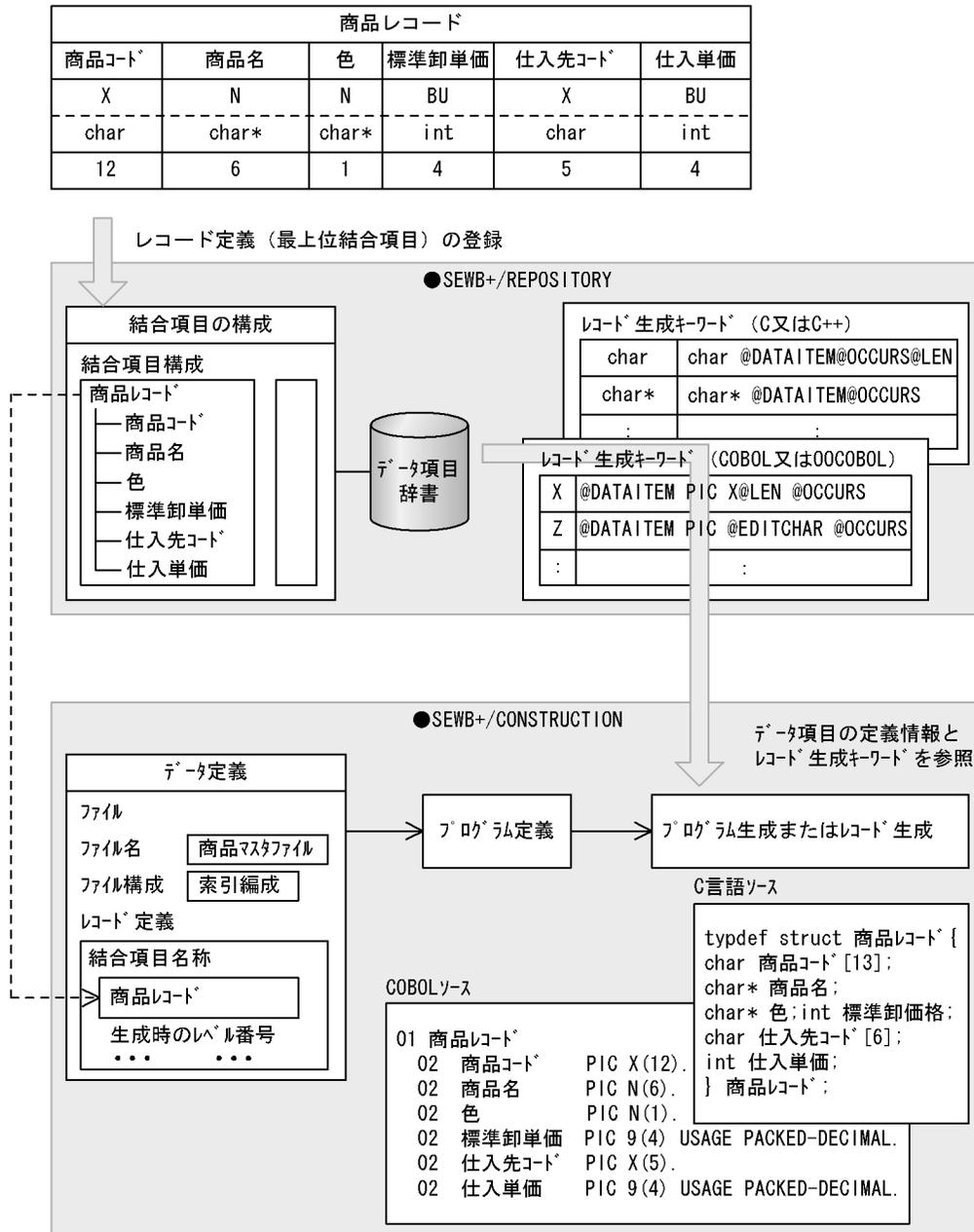
ここでは、レコードの構造をデータ項目辞書に用意する場合について説明します。

最初に、適切なレコードの構造をデータ項目辞書に用意するために、あらかじめ、辞書

2. 辞書の設計

のレコード定義（最上位結合項目）とプログラムの関係を理解しておきましょう。辞書のレコード定義（最上位結合項目）がアプリケーション開発に利用され、プログラムに反映されるまでの流れを、図 2-9 に示します。

図 2-9 辞書のレコード定義（最上位結合項目）がプログラムに反映されるまで



(1) レコード定義（最上位結合項目）の登録

データベースやファイル設計および業務設計から求められた C/S（クライアントサーバ）システムのインタフェース仕様などの情報を基に、辞書に登録するレコード構造を設計します。このレコード構造に従って、すでに辞書に登録してあるデータ項目を組み合わせ、結合項目を構成します。この結合項目は、SEWB+/CONSTRUCTION、EUR Professional Edition および APPGALLERY Enterprise でレコードとして扱われるために「最上位結合項目」として定義します。

SEWB+/CONSTRUCTION では、データ定義を使って、実際にアプリケーションで使用するデータを定義するときに、辞書に用意されているレコード定義（最上位結合項目）を参照します。例えば、RDB の定義では、RDB の表の構造としてレコード定義（最上位結合項目）が参照され、データが定義されます。このため、辞書の作成者は、アプリケーション開発に使用するデータの種類の考慮しながら、辞書に用意するレコード構造を決めてください。

SEWB+/CONSTRUCTION で定義できるデータ種別を次に示します。

- ファイル
- RDB
- DAM (Direct Access Method)
- TAM (Table Access Method)
- RPC (Remote Procedure Call)
- メッセージ
- ユーザジャーナル
- メッセージログ
- 共通作業領域

データ定義や、データ種別の詳細は、マニュアル「SEWB+/CONSTRUCTION アプリケーション開発ガイド」を参照してください。

(2) SEWB+/CONSTRUCTION のデータ定義との関係

実際にアプリケーションで使用するファイルや RDB の情報を、SEWB+/CONSTRUCTION のデータ定義機能で定義します。図 2-9 のデータ定義の例では、辞書に登録された「商品レコード」を、実際に使用する「商品マスタファイル」のレコードとして定義しています。データ定義情報をリポジトリにチェックインすると、レコード定義（最上位結合項目）とデータ定義のドキュメントとの間に、自動的に関連（関連種別「辞書参照」）が付けられます。

(3) ソースコードの生成

辞書のレコード定義情報は、データ定義と関連づけられたあと、プログラム定義機能を経て、ソースプログラムが生成されるときに、レコードの記述として展開されます。また、クライアントとサーバ間のインタフェーステーブルを作成したい場合には、データ定義機能から、個別の COPY メンバ（COBOL 言語の場合）またはヘッダファイル（C

2. 辞書の設計

言語の場合)を生成することもできます。生成時には、SEWB+/REPOSITORYのデータ項目の定義情報とともにレコード生成キーワードが参照されます。参照されるデータ項目の定義情報を表 2-17 に示します。

表 2-17 プログラム生成時に参照されるデータ項目の定義情報

定義項目		COBOL 又は OOCOBOL	C 又は C++
名称	データ項目名	×	×
	標準名称	×	×
	フリガナ	×	×
属性	分類	×	
	けた数		
	小数部けた数		
	反復回数		
付加情報	フィールド 1 ~ 20	×	×
	コメント	×	×
言語別 詳細情報	名前		
	タイプ		
	タイプ修飾情報 (編集文字列)		-
	取りうる値		
	初期値		×
	言語別フィールド	×	×

(凡例)

- : 生成時に参照される
- : 生成するかどうかを指定できる
- × : 生成時に参照されない
- : 定義できない

注

言語別の名前を定義していない場合には、データ項目名が参照されます。

- レコード生成キーワードとは
レコード生成キーワードとは、言語別のタイプごとに、データ項目の定義情報をどのようにソースコードに生成するかが指定されたものです。レコード生成キーワードの内容は、SEWB+/REPOSITORYの環境構築ユティリティで、独自のアプリケーション開発の運用に合わせて編集できます。詳細は「3.1.1(9) タイプに対するレコード生成キーワードを編集する」を参照してください。

2.5.2 辞書のレコード定義(最上位結合項目)を登録する

ここでは、辞書のレコード定義(最上位結合項目)を登録するする場合について説明し

ます。

(1) リポジトリブラウザを使ったレコード定義（最上位結合項目）の登録

辞書のレコード定義を構成するデータ項目と結合項目は、すでに辞書に登録されていることが前提になります。リポジトリブラウザを使ったレコード定義（最上位結合項目の定義）の登録手順は、結合項目を登録する場合と同様です。

レコード定義（最上位結合項目）を登録するには、データ項目を作成するダイアログの[基本属性]タブで、データ項目種別から「結合項目」を選択し、さらに「最上位結合項目」を指定してください。また、結合項目を構成する構成項目の順番は、レコード構造の設計で決めたデータ項目の並び順と一致させてください。

また、結合項目ブラウザを起動してレコード定義（最上位結合項目）を登録することもできます。結合項目ブラウザを起動するには、リポジトリブラウザの[リポジトリ] - [結合項目ブラウザ]を選びます。

(2) レポート出力に使用するレコード定義（最上位結合項目）を登録するには

レポート出力アプリケーションに使用するデータファイル用のレコード定義（最上位結合項目）には、アプリケーションからのデータの出力形態を考慮することをお勧めします。

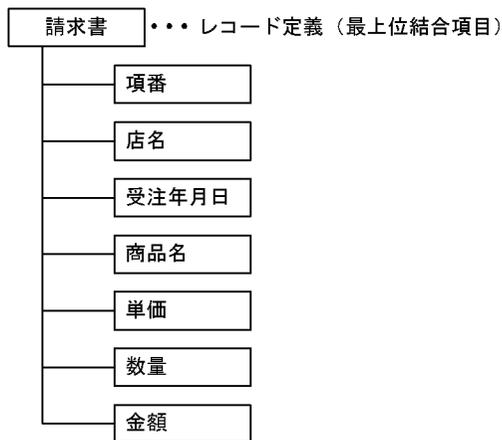
COBOL 言語でレポート用のデータファイル（CSV 形式ファイルまたは FIX 形式ファイル）を出力するアプリケーションを作成するには、表計算プログラム入出力機能またはテキストファイル機能を使用します。これらは日立 COBOL で提供している機能です。この機能では、集団項目に従属する基本項目のデータを出力します。このため、レコード定義（最上位結合項目）も、この出力形態に合わせて、1 階層のデータ項目の結合関係で構成するようにしてください。次の例のように、単独のデータ項目をフラットに並べた構造にします。

注

FIX 形式のデータファイルおよびテキストファイル機能を使用するのは、帳票設計ツールに EUR Professional Edition を使用している場合だけです。

2. 辞書の設計

(例)



(3) レコード定義 (最上位結合項目) を変更する

基本的には、アプリケーション開発を始める前にレコード定義 (最上位結合項目) を用意しておきますが、アプリケーション開発中でもレコード定義 (最上位結合項目) を変更しなければならない場合が考えられます。このようなとき、SEWB+/CONSTRUCTION のデータ定義から、レコード定義 (最上位結合項目) を構成するデータ項目を追加したり変更したりできます。しかし、SEWB+/CONSTRUCTION からは、新しいレコード定義 (最上位結合項目) を作成することはできません。新しいレコード定義 (最上位結合項目) が必要な場合には、リポジトリブラウザを使って辞書の作成者が登録してください。

2.6 業務ルール設計

業務ルールとは、データ項目に着目して、データ項目特有の処理を部品化したものです。データ項目辞書を整備したあと、この業務ルールを格納する業務ルール辞書を整備します。

業務ルールには、データ項目と関連づけて設計、登録する通常の業務ルールと同じデータ項目を使った処理を記述するための同一項目用業務ルールがあります。通常、業務ルールはデータ項目の継承の関連を考慮して設計、登録します。ただし、同一項目用業務ルールの設計と登録では、継承の関連を考慮する必要はありません。

なお、SEWB+/REPOSITORYでは、一般の業務に使用することを想定した同一項目用業務ルール（同一項目間転記）を提供しています。辞書への登録方法については「3.1.3 同一項目用業務ルールを辞書に準備する」を、内容については「付録 E 標準提供の同一項目用業務ルール」を参照してください。

2.6.1 業務ルールに定義する情報

業務ルールにどのような情報を定義するかについて説明します。

(1) 業務ルールの定義項目

業務ルールの定義項目の一覧を、表 2-18 に示します。業務ルール名や標準名称に定義する文字数の上限は、カスタマイズできます。なお、業務ルール名と適用条件は、必ず定義しなければなりません。ほかの定義項目は、必要に応じて定義を省略できます。

表 2-18 業務ルールの定義項目

	定義項目	文字数の上限		文字セット
		標準値	設定範囲	
属性	業務ルール名	63	1 ~ 92	英数字, カナ文字, 日本語文字 ³
	標準名称 ¹	63	1 ~ 92	英数字, カナ文字, 日本語文字
	フリガナ	30	1 ~ 92	英数字, カナ文字, 日本語文字
	コメント ¹	6500	固定	任意
	フィールド 1 ~ 3	30	1 ~ 255	任意
関連	キーワード	93	固定	英数字, カナ文字, 日本語文字
	適用条件	-	-	4
	ルールスクリプト ²	6500	固定	英数字, カナ文字, 日本語文字

(凡例)

固定：文字数の上限は固定されていることを示します。

注

2. 辞書の設計

文字数は半角文字の単位です。

注 1

4 種類までの国語区別に定義できます。

注 2

10 種類までの言語区別に定義できます。

注 3

「:」および「|」は使えません。また、文字列の途中に空白文字を含めてはなりません。

注 4

「入力」、「出力」または「-」から選択します。

(2) 属性

業務ルールを識別したり、検索したりするための情報を、次に説明する定義項目に定義します。この情報を定義するリポジトリブラウザのダイアログ（[属性] タブ）を、図 2-10 に示します。

図 2-10 業務ルールを作成するダイアログ（[属性] タブ）

業務ルール新規作成

ルールスクリプト | 属性 | データ項目

辞書フォルダ名: ManualSample

業務ルール名(N):

国語別(L): 日本語

標準名称(D):

フリガナ(E):

コメント(C):

フィールド1(1):

フィールド2(2):

フィールド3(3):

閉じる 上書き保存(V) 新規に保存(E)

(a) 業務ルール名

業務ルール辞書で管理される業務ルールを識別するために付ける名称です。業務ルール名は必ず定義しなければなりません。

(b) 標準名称

業務ルールの処理内容をわかりやすくするために付ける，システム標準の名称です。4種類までの国語別に定義できます。

(c) フリガナ

日本語読みの名称（フリガナ）を定義します。国語区分「英語」の場合は，フリガナを定義できません。

(d) コメント

処理の概要，ソースプログラムへの展開条件，注意事項などを記述します。SEWB+/CONSTRUCTION から適切に業務ルールを選択するためのポイントとなる情報であるため，必ず定義することをお勧めします。コメントは，4種類までの国語別に定義できます。リポジットブラウザでは，このコメントを基に目的の業務ルールを検索することもできます。

(e) フィールド 1 ~ 3

プロジェクト独自の情報を定義するなどして，自由に使用できるフィールドです。環境構築ユティリティを使用し，任意のフィールド名称を付けられます。

(3) 業務ルールに適用するデータ項目との関係

業務ルールは，通常，データ項目と必ず関連づけて登録します。このほか，同一データ項目間の処理専用の業務ルールである「同一項目用業務ルール」があります。この2種類の業務ルールは，適用するデータ項目との関係や定義のしかたが異なります。ここでは，この2種類を分けて説明します。

(a) 適用するデータ項目と業務ルールの関連

業務ルールに使用するデータ項目や，その適用形態などを，次に説明する定義項目に定義します。この情報を定義するリポジットブラウザのダイアログ（[データ項目]タブ）を，図 2-11 に示します。

図 2-11 業務ルールを作成するダイアログ ([データ項目] タブ)



(i) データ項目の適用形態

データ項目の適用形態を「専用にする」または「共用する」から選択します。

専用にする

特定のデータ項目（単独のデータ項目または複数のデータ項目の組み合わせ）に対応する業務ルールであることを示します。

共用する

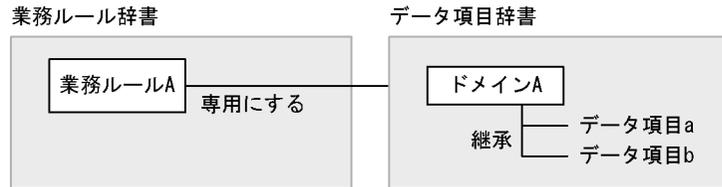
複数のデータ項目に共通に対応する業務ルールであることを示します。

• データ項目の適用形態の考え方

データ項目と業務ルールとの関係を明確にし、データ項目の体系に基づいて効率良く業務ルールを登録、管理するために、通常は「専用にする」を選択することをお勧めします。

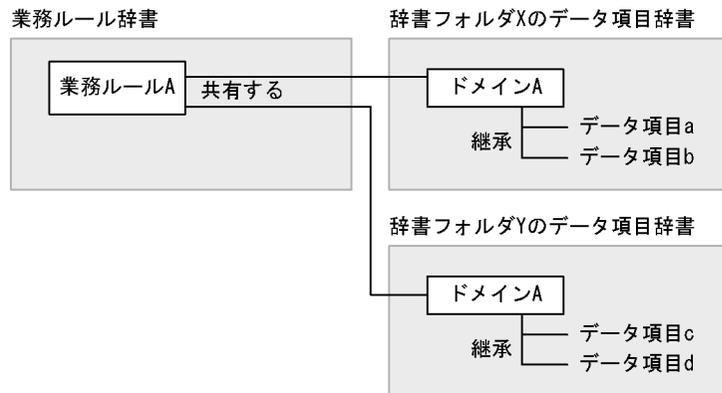
業務ルールに使用するデータ項目がドメイン、つまり継承関係の上位のデータ項目であるとき、業務ルールとの対応もドメインにグルーピングされた下位のデータ項目に継承されます。したがって、複数のデータ項目で業務ルールを共用したい場合には、継承の仕組みを利用して共用するようにしてください。

(例)



複数の辞書フォルダに登録した同一のドメインに対応する業務ルールを登録したい場合には、「共用する」を選択します。

(例)



(ii) データ項目

業務ルールに使用するデータ項目を定義します。

データ項目の適用形態に「専用にする」を選んだ場合には、単独のデータ項目または組み合わせるデータ項目を定義します。「共用する」を選んだ場合は、その業務ルールを共通に使用するデータ項目を定義します。業務ルールを登録したとき、ここに定義したデータ項目と業務ルールの間には、関連種別「ルール」の関連が自動的に付けられます。

(iii) キーワード

業務ルールに使用するデータ項目の仮の名称です。業務ルールに使用するデータ項目を定義すると、記号「@」にデータ項目名を加えた名称が初期値として表示されます。必要に応じてキーワードを変更してください。ここで定義したキーワードは、ルールスクリプトを記述する際に必ず使用してください。

データ項目の適用形態に「専用にする」を選んだ場合には、組み合わせるデータ項目それぞれに対してキーワードを定義します。「共用する」を選んだ場合は、すべてのデータ項目に対して共通のキーワードを定義します。

(iv) 適用条件

データ項目がどのような条件（処理の入力または出力）にあるときに、その業務ルールをアプリケーションに適用させるのか、という情報を指定します。一般に、

2. 辞書の設計

業務ルールの処理の入力となるデータ項目には適用条件に「入力」を、業務ルールの処理結果を出力するデータ項目には、適用条件に「出力」を指定します。どちらにも役割を限定しない場合や、役割を明確にしたい場合には、指定しなくてもかまいません（指定しない場合は「 - 」となります）。

データ項目の適用形態に「専用にする」を選んだ場合には、組み合わせるデータ項目それぞれに対して適用条件を指定します。「共用する」を選んだ場合は、すべてのデータ項目に対して共通の適用条件を指定します。

適用条件は、SEWB+/CONSTRUCTION からアプリケーションに適用する業務ルールを抽出する場合にキーとなる情報です。したがって、SEWB+/CONSTRUCTION からの利用も考慮した指定が必要となります。指定の考え方については、「2.6.4 適用条件の指定」を参照してください。

(b) 同一項目用業務ルール

同一項目用業務ルールは、二つの同じデータ項目間の処理を記述するための業務ルールです。主に、COBOL 言語で MOVE 文、C 言語で strepy 関数で記述できるような、入力側のデータ項目から出力側のデータ項目へのデータの転記処理に使用します。同一項目用業務ルールとして作成した業務ルールは、リポジトリ内のすべてのデータ項目で共用できます。同一項目用業務ルールでは、データ項目との関連は付けません。しかし、業務ルールを利用する SEWB+/CONSTRUCTION からは、アプリケーションに使用するデータ項目と同一項目用業務ルールの間に関連が付けられているとみなされ、通常の業務ルールと同様に扱えます。この情報を定義する場合のリポジトリブラウザのダイアログ（[データ項目] タブ）を、図 2-12 に示します。

図 2-12 同一項目用業務ルールを作成するダイアログ ([データ項目] タブ)



(i) データ項目の適用形態

同一項目用業務ルールでは、データ項目の適用形態は、必ず「専用にする」になります。この業務ルールに仮想的に関連づけるデータ項目を「すべての単項目」または「すべての項目」から選択します。

すべての単項目

リポジトリ内の結合項目ではないデータ項目（単項目）に対して仮想的な関連を付け、適用できるようにします。

すべての項目

リポジトリ内の結合項目を含めたすべてのデータ項目に対して仮想的な関連を付け、適用できるようにします。

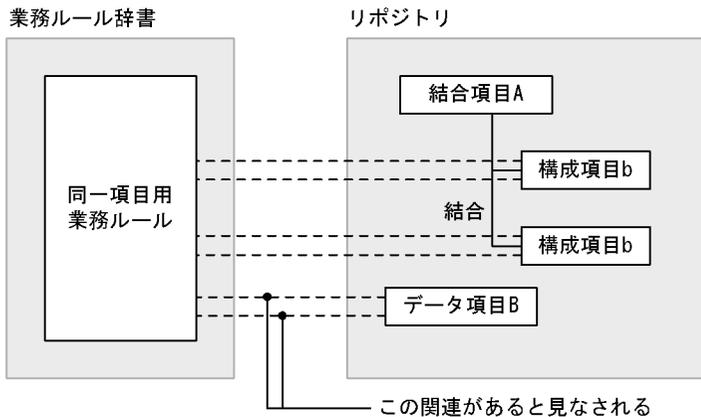
- 「すべての単項目」、「すべての項目」の選択の考え方

適用するデータ項目の範囲で考えます。どちらの場合も、データ項目の継承関係を考慮する必要はありません。

COBOL 言語での基本項目同士の転記処理、C 言語での構造体を構成する変数同士の転記処理だけを適用したい場合は、「すべての単項目」を選択します。この場合には、次に示すような仮想的な関連が付けられたものとみなされます。

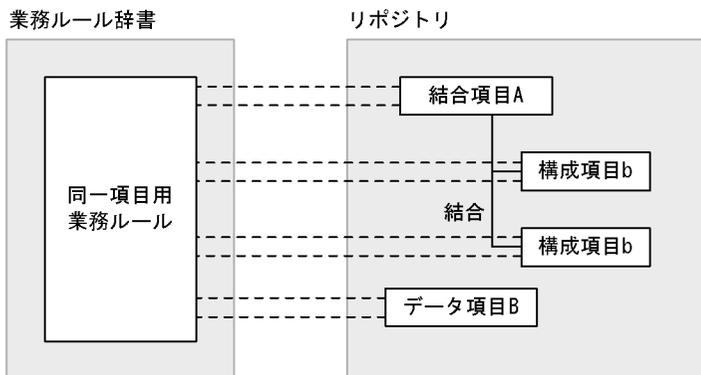
2. 辞書の設計

(例)



また、結合項目や構成項目に関係なく、すべてのデータ項目を適用範囲にしたい場合、つまり、COBOL 言語での集団項目同士の転記処理、C 言語での構造体同士の転記処理を適用範囲に含めたい場合には、「すべての項目」を選択します。この場合には、次に示すような仮想的な関連が付けられたものとみなされます。

(例)



(ii) データ項目

組み合わせるデータ項目を定義します。組み合わせるデータ項目は、必ず「すべての単項目」同士または「すべての項目」同士でなければなりません。

(iii) キーワード

キーワードの考え方は、通常の業務ルールを作成する場合と同様です。同一項目用業務ルールでは、組み合わせる各データ項目にキーワードを付けます。キーワードは、例えば、転記処理用の業務ルールの場合、転記元と転記先のデータ項目をそれぞれ識別できるように、重複しない名称を付けてください。

(iv) 適用条件

適用条件の考え方は、通常の業務ルールを作成する場合と同様です。同一項目用業

務ルールでは、一般に、入力側のデータ項目に「入力」を、出力側データ項目に「出力」を指定します。

(4) ルールスクリプト

ルールスクリプトには、ソースプログラムに展開する業務ルールの処理を、アプリケーション開発に使用するプログラミング言語別にコーディングします。この情報を定義するリポジトリブラウザのダイアログ（「ルールスクリプト」タブ）を、図 2-13 に示します。

図 2-13 業務ルールを作成するダイアログ（「ルールスクリプト」タブ）



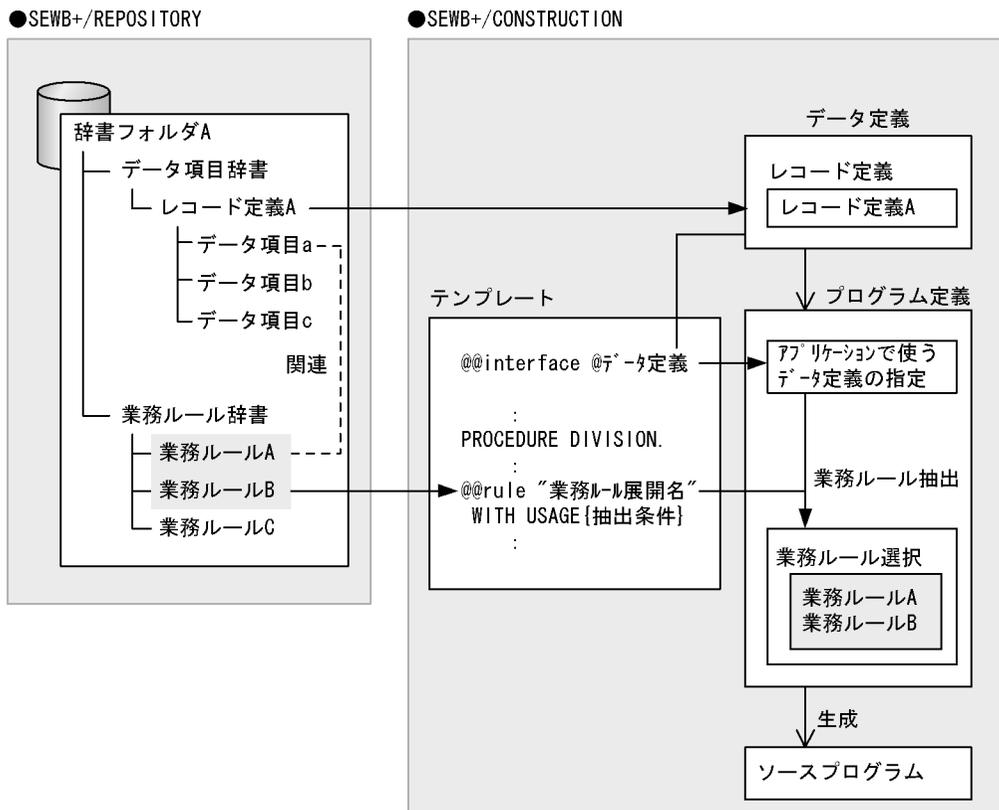
ルールスクリプトは 10 種類までの言語区別に定義できます。記述方法の詳細は、「2.6.3 ルールスクリプトの記述」を参照してください。

2.6.2 業務ルールとプログラムの関係

SEWB+/CONSTRUCTION を使ったアプリケーション開発の準備として、データ項目それぞれに特有の処理を抽出し、業務ルールとして辞書に登録します。業務ルールを効果的にアプリケーション開発に利用してもらうために、あらかじめ、辞書の業務ルールとプログラムの関係を理解しておきましょう。辞書の業務ルールがアプリケーション開発に利用され、プログラムに反映されるまでの流れを図 2-14 に示します。

2. 辞書の設計

図 2-14 業務ルールがプログラムに反映されるまで



SEWB+/CONSTRUCTION を使って作成するアプリケーションで業務ルールをどのように利用するのは、テンプレート作成者がテンプレートに記述します。このため、SEWB+/CONSTRUCTION から効果的に利用できる業務ルールを設計し、業務ルール辞書に登録するには、テンプレート作成者と辞書の作成者がコミュニケーションをとりながら作業することが大切です。業務ルールの設計やテンプレートの作成を始める前に、プロジェクト独自の基準を決定してドキュメントにまとめ、これに基づいて作業することをお勧めします。基準の決め方については、「(1) 業務ルールの基準を決める」を参照してください。

- テンプレートとは

業務ルールとの関係を理解するために、テンプレートについて少し説明しましょう。テンプレートは、プログラムのひな形のような役割を持っています。テンプレートには、例えば、検索や更新、帳票出力など、業務システムのアプリケーションの構造がパターン化されて記述されています。アプリケーションにどんな処理をさせるか、それにはどんなデータ定義が必要かなどのプログラムの処理および構成が書かれます。テンプレートは、通常、業務システム全体を把握する専任者（テンプレート作成者）によって作成されます。

業務ルールを SEWB+/CONSTRUCTION から利用する場合には、テンプレートに、

生成されるソースプログラムでルールスクリプトの展開位置を指示する文（これを @@rule 文といいます）と、業務ルールの抽出条件（@@rule 文の WITH USAGE 句の指定）が書かれます。プログラム定義では、この指定に従って辞書から抽出された業務ルールの一覧が表示されます。

詳細は、マニュアル「SEWB+/CONSTRUCTION アプリケーション開発ガイド」を参照してください。

（１）業務ルールの基準を決める

業務ルールの設計およびテンプレートの作成に当たり、あらかじめ次の２点についてプロジェクト独自の基準を決定します。

1. どのような処理を業務ルールにするか
2. 業務ルールをアプリケーションにどのように適用させるのか

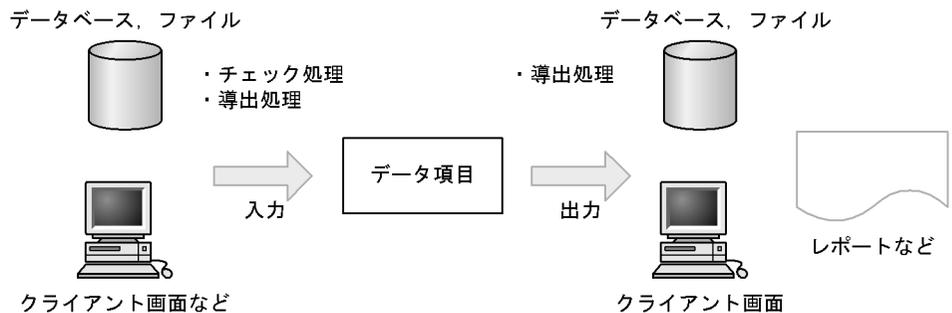
この結果をドキュメントにまとめ、これに基づいて、業務ルールを設計し、テンプレートを作成することをお勧めします。このほか、ルールスクリプトの記述についてもあらかじめ統一が必要な事項があります。詳細は、「2.6.3 ルールスクリプトの記述」を参照してください。

（a）どのような処理を業務ルールにするか

まず、データ分析で抽出された各データ項目の制約条件や、業務設計から求められた業務の規則、制度などを基に、どのような処理を業務ルールとして辞書に登録するかを検討します。このとき、テンプレート作成者の、アプリケーションを作成する立場から、どのような処理を業務ルールとして用意してほしいか、という意見も重要になります。

データ項目を中心に、データ項目に関するどのような処理を業務ルールにすればよいのかを決定します。業務ルールとする処理の例を次に示します。

（例）



チェック処理・・・データ項目の値が入力されたときに、その妥当性をチェックする処理。

導出処理・・・データ項目の値から、別のデータ項目の値を導き出す処理。
値が入力される時、または出力される時で分類できる。
編集処理、計算処理など。

2. 辞書の設計

(b) 業務ルールをアプリケーションにどのように適用させるか

業務ルールの適用条件と、テンプレートの @@rule 文の抽出条件の指定基準を決めます。

業務ルールの適用条件とは、アプリケーションに業務ルールを適用する条件を指定するものです。辞書の作成者は、「データ項目がどのような条件（処理の入力なのか、出力なのか）にあるときに、この業務ルールをアプリケーションに適用させるのか」という観点で指定します。

SEWB+/CONSTRUCTION から業務ルールを使用する場合、この適用条件が指定されていることが前提になります。テンプレート作成者は、テンプレートの @@rule 文の抽出条件に、「データ定義に含まれるデータ項目が、ある条件（アプリケーションの入力または出力）にあるとき、条件に一致する業務ルールを抽出する」ということを指定します。

このため、業務ルールを SEWB+/CONSTRUCTION から適切に利用するためには、辞書の作成者およびテンプレート作成者それぞれが、統一した基準に従って適用条件や抽出条件を扱う必要があります。

この基準を決定するまでの手順を次に示します。

1. 業務ルールを機能（処理内容）別に分類する
2. 分類ごとに、アプリケーション中でデータ項目がどのような条件にあるときに、業務ルールを適用させるのかを検討する
3. 検討した適用のしかたに従って、業務ルールおよびテンプレートの両面から、指定基準を決定する
 - その分類の業務ルールを作成するときの、データ項目の適用条件「入力」、「出力」および「 - （指定しない）」の指定方法
 - その分類の業務ルールをアプリケーションに適用するときの、テンプレートに記述する @@rule 文の抽出条件（WITH USAGE 句）の指定方法

1. ~ 3. の過程では、必要に応じて分類の詳細化や統合を行います。実際に業務ルールを設計、登録するときには、その業務ルールが含まれる分類の基準に従って適用条件を指定してください。

(2) 業務ルールの設計、登録

業務ルールについてのプロジェクトの基準に従って、業務ルールを設計、登録します。データ項目の登録の手順と同様に、まず、ドメインの業務ルールを登録し、そのあとにドメインにグルーピングされた下位のデータ項目の業務ルールを登録していきます。また、必要に応じて同一項目用業務ルールを登録します。

なお、通常の業務ルールは必ずデータ項目と関連づけて登録します。したがって、業務ルール辞書を整備するには、使用するデータ項目、それらの継承関係、結合関係が整えられていることが前提となります。

(3) SEWB+/CONSTRUCTION からの業務ルールの利用

次に、業務ルールが SEWB+/CONSTRUCTION から利用される流れを説明します。

業務ルールを使う仕組みの詳細は、マニュアル「SEWB+/CONSTRUCTION アプリケーション開発ガイド」を参照してください。

(a) テンプレートの記述

テンプレート作成者によって、業務ルールをアプリケーションにどのように利用するかが、テンプレートに記述されます。

(b) データ定義

データ定義では、アプリケーションで使用するファイル、データベース、およびクライアントとサーバ間の通信のインタフェースが定義されます。データ定義では、レコード定義が参照されます。

(c) プログラム定義

プログラム定義では、プログラマが使用するテンプレートやデータ定義を選び、それらに定義されている情報を基に、ソースプログラム生成に必要な情報を定義していきます。

1. プログラマは、まず、作成するアプリケーションに使用するデータ定義を指定します。データ定義の使用は、テンプレートの `@@interface` 文で宣言されています。
2. 選択したテンプレートに `@@rule` 文が書いてある場合には、1. で指定したデータ定義のレコード定義と `@@rule` 文の `WITH USAGE` 句に指定した抽出条件に従って、アプリケーションに適用できる業務ルールの候補が抽出され、一覧に表示されます。プログラマは、この候補の中から、ソースプログラムに展開させたい業務ルールを選択します。このとき、プログラマは、業務ルール名とコメントの情報を手がかりにして業務ルールを選択します。したがって、辞書の作成者が業務ルールを登録するときには、プログラマに対しても、わかりやすく検索しやすい業務ルール名、コメントを定義するようにしましょう。

なお、プログラム定義情報をリポジトリにチェックインすると、選択した業務ルールとプログラム定義のドキュメントの間に、自動的に関連（関連種別「辞書参照」）が付けられます。

(d) ソースプログラムの生成

プログラム定義からソースプログラムを生成すると、プログラム定義で選ばれた業務ルールのルールスクリプトがソースプログラムの中に展開されます。

2.6.3 ルールスクリプトの記述

ルールスクリプトは、アプリケーション開発に使用するプログラミング言語（COBOL 言語や C 言語など）の文法に従って記述します。これに加えて、ソースプログラムへの展開位置を指定する文や、業務ルールやデータ項目の情報を取り出せる予約キーワードなど、ルールスクリプトを記述するための独自の文法が用意されています。さらに、SEWB+/CONSTRUCTION のテンプレート記述言語をルールスクリプトの記述に使用することもできます。

2. 辞書の設計

業務ルールの内容に応じた書き方については「2.6.5 システムで共用する業務ルールを登録する」および「2.6.6 特定業務で扱う業務ルールを登録する」を参照してください。

(1) キーワードでデータ項目の名称を記述する

業務ルールで使用するデータ項目名は、必ずキーワードで記述します。キーワードは、SEWB+/CONSTRUCTION でソースプログラムが生成されるときに、データ項目に定義されている言語別の名前に置換されます。

(例)

適用するデータ項目とキーワードの対応

データ項目名	言語別の名前	キーワード
金額	KINGAKU	@金額
単価	TANKA	@単価
数量	SUURYO	@数量

ルールスクリプト

```
COMPUTE @金額 = @単価 * @数量
```

SEWB+/CONSTRUCTIONで生成されるソースプログラム

```
COMPUTE KINGAKU = TANKA * SUURYO
```

生成

注

なお、言語別の名前を定義していない場合には、キーワードはデータ項目名に置換されます。また、ソースプログラムに生成されるとき、テンプレートに記述した @@rule 文の指定に従って、言語別の名前に接頭語または接尾語が付加される場合があります。@@rule 文の詳細は、マニュアル「SEWB+/CONSTRUCTION アプリケーション開発ガイド」を参照してください。

(2) 記述形式

COBOL 言語でルールスクリプトを記述するときの規則について説明します。SEWB+/CONSTRUCTION で作成するアプリケーションのコーディング形式に固定形式（正書法）を採用している場合でも、ルールスクリプト中に一連番号（正書法 1 ~ 6 カラム目）を記述する必要はありません。7 カラム目の標識領域から記述してください。一連番号は、SEWB+/CONSTRUCTION でソースプログラムを生成したときに自動的に付加されます。

(3) 記述方法

ルールスクリプトの記述には、ルールスクリプト独自の文法とプログラミング言語で記述する方法と、これにテンプレート記述言語を併用して記述する方法の 2 通りがあります。

前者の記述方法は、データ項目と業務ルールが一对一に対応する業務ルール、またはデータ項目の属性や性質がそのまま下位のデータ項目に継承するドメインの業務ルールの記述に適しています。

これに対して、後者のテンプレート記述言語の展開制御文（可変記号に値を代入する @@set 文や、可変記号の値によって条件判定する @@if 文など）を併用する方法は、異なる複数のドメインやデータ項目に対応させる業務ルールの記述に適しています。汎用性が高い業務ルールを作成できるため、業務ルールの再利用性を向上できます。ただし、この場合は、辞書を設計する知識に加えて、テンプレートを作成する知識が必要になります。また、業務ルールとテンプレートの関係も複雑になります。

どちらの方法でルールスクリプトを記述すればよいかは、プロジェクトの状況や、どれだけ業務ルールに汎用性を持たせるかなどによって異なります。テンプレート作成者と十分検討したうえで、最適な記述方法を採用するようにしてください。

(4) 作業領域を記述する

ルールスクリプトには、手続きだけではなく、処理内容に応じてデータの作業領域（データ項目の内部構造を分割または統合するための領域、処理に使う一時領域、外部サブルーチンとのインタフェース領域など）を記述できます。作業領域の記述については、「(5) ルールスクリプトを分割して展開する」、「(6) 作業領域のデータ項目名の重複を回避する」および「(7) 作業領域を 1 回だけ展開する」を参考にしてください。

(5) ルールスクリプトを分割して展開する

ルールスクリプトは通常、SEWB+/CONSTRUCTION のテンプレートの @@rule 文で、業務ルールが挿入指示された位置にソースコードが展開されます。しかし、@@section 文を利用すれば、ルールスクリプト中のコードの一部を、指定した別の位置に展開できます。例えば、COBOL 言語で記述する場合には、ルールスクリプト中の作業領域を、手続きとは別の位置（WORKING-STORAGE SECTION や LINKAGE SECTION など）へ展開しなければなりません。このようなときに「@@section 文」を使用します。

なお、分割展開の必要がないときには @@section 文を記述しなくてもかまいません。ルールスクリプト中に @@section 文がない場合、テンプレートの @@rule 文でその業務ルールが引き当てられた位置に展開されます。

(a) @@section 文

形 式

@@section {MAIN | コード名}

文

注

{ } 内のどちらかを選んで記述します。

規 則

- @@section に続く文には、テンプレート記述言語以外の任意のテキストを記述で

2. 辞書の設計

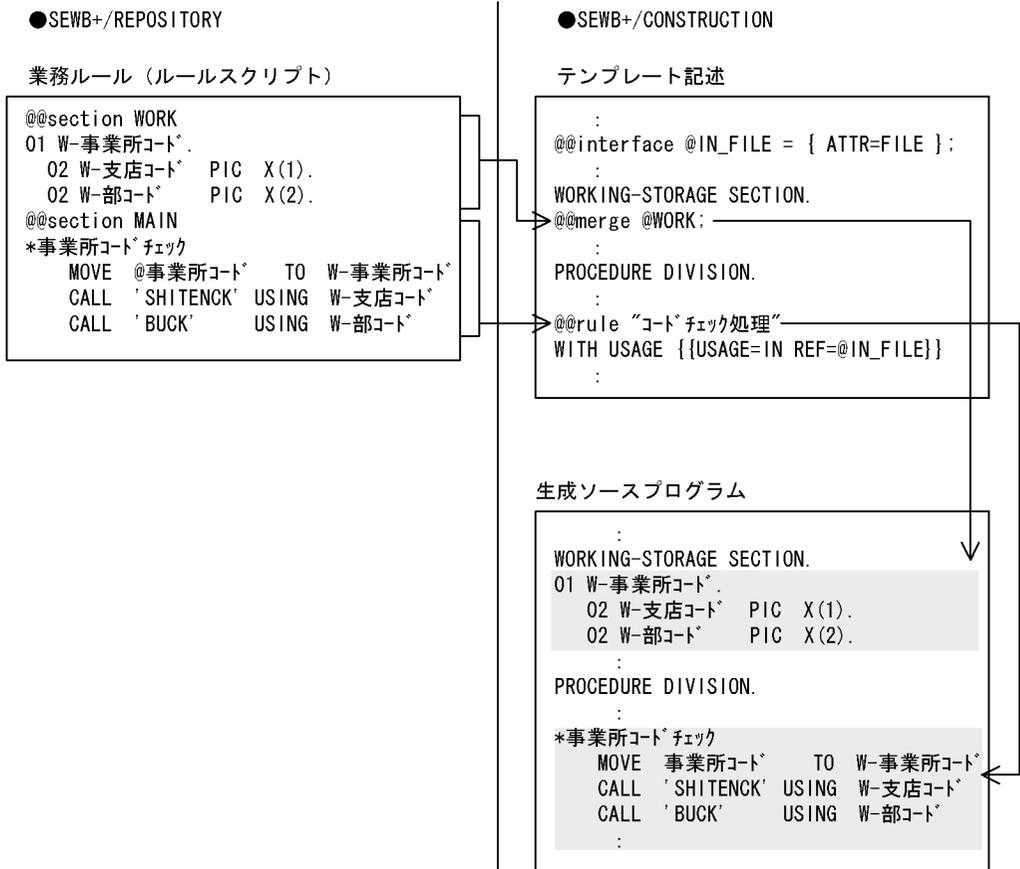
きます。

- 次のどちらかの方法でソースプログラムへの展開先を指定します。
MAIN : SEWB+/CONSTRUCTION のテンプレートの @@rule 文で、その業務ルールが挿入指示された位置に展開されます。
コード名 : 29 文字以内の文字列を指定します。テンプレート中の、コード名の先頭に @ を付加した可変記号が指定された @@merge 文の位置に展開されます。この関係については「(b) テンプレートに記述する @@merge 文との関係」を参照してください。
- @@section 文の指定は、次の @@section 文が同じルールスクリプト上に記述されるまで有効になります。
- @@section 文のコード名と一致するものが @@merge 文の可変記号に指定されていない場合には、展開されません。

(b) テンプレートに記述する @@merge 文との関係

@@section 文で @@rule 文以外の場所に展開先を指定されたルールスクリプトは、テンプレートの @@merge 文の位置に展開されます。この関係を次の例で見てみましょう。

(例)



ルールスクリプトの作業領域「W- 事業所コード」は、同一のコード名を持つ可変記号「@WORK」が指定された @@merge 文の位置に展開されています。

このように、適切に分割展開するには、ルールスクリプトの「@@section コード名」とテンプレートの「@@merge 文 可変記号」のコード名が一致していなければなりません。したがって、辞書の作成者はテンプレート作成者の協力を得て、ルールスクリプトとテンプレートで、どのように分割展開し、どのようなコード名を使うかをあらかじめ決めておく必要があります。

@@merge 文の詳細は、マニュアル「SEWB+/CONSTRUCTION アプリケーション開発ガイド」を参照してください。

(6) 作業領域のデータ項目名の重複を回避する

作業領域を記述するときに予約キーワード「@rulecount」を使用すれば、ソースプログラム中に同じルールスクリプトが複数個展開されても、データ項目名の重複を回避できます。@rulecount は、ソースプログラムの生成時に、その業務ルールが展開された回数

2. 辞書の設計

と置換されます。

(a) @rulecount

作業領域のデータ項目名に記述します。回数は、通し番号（4けたの整数）で生成されま
す。

(例)

```
01 作業領域-@rulecount  PIC X(8).  
:  
MOVE @データ項目  TO 作業領域-@rulecount
```

@rulecount を使った作業領域の展開例を次に示します。

(例)

●SEWB+/REPOSITORY

業務ルール (ルールスクリプト)

```

@@section WORK
01 W-時刻-@rulecount.
02 W-時-@rulecount PIC 9(2).
02 W-分-@rulecount PIC 9(2).
01 W-編集時刻-@rulecount.
02 W-編集時-@rulecount PIC 99.
02 FILLER PIC X(1) VALUE ' '.
02 W-編集分-@rulecount PIC 99.
@@section MAIN*
時刻出力編集
MOVE @時刻 TO W-時刻-@rulecount
MOVE W-時-@rulecount TO
      W-編集時-@rulecount
MOVE W-分-@rulecount TO
      W-編集分-@rulecount
MOVE W-編集時刻-@rulecount TO @編集時刻

```

●SEWB+/CONSTRUCTION

テンプレート記述

```

:
@@interface @IN_FILE =
      { ATTR = FILE };
@@interface @OUT_FILE =
      { ATTR = FILE };
:
WORKING-STORAGE SECTION.
@@merge @WORK;
PROCEDURE DIVISION.
:
@@rule "時刻編集処理" WITH USAGE
  {{ USAGE=IN REF=@IN_FILE }
  { USAGE=OUT REF=@OUT_FILE}};
:

```

生成ソースプログラム

```

:
WORKING-STORAGE SECTION.
01 W-時刻-0001.
02 W-時-0001 PIC 9(2).
02 W-分-0001 PIC 9(2).
01 W-編集時刻-0001.
02 W-編集時-0001 PIC 99.
02 FILLER PIC X(1) VALUE ' '.
02 W-編集分-0001 PIC 99.
:
PROCEDURE DIVISION.
:
*時刻出力編集
MOVE 時刻 TO W-時刻-0001
MOVE W-時-0001 TO W-編集時-0001
MOVE W-分-0001 TO W-編集分-0001
MOVE W-編集時刻-0001 TO 編集時刻
:

```

(7) 作業領域を1回だけ展開する

ルールスクリプトにテンプレート記述言語の展開制御文を使用すれば、@@section 文と同様の分割展開ができます。さらに、ソースプログラム中に同じルールスクリプトが複数展開されるときに、作業領域を1回だけ展開するように記述できます。ただし、この記述では、テンプレートに、最初の業務ルールの展開かどうかを判断するための可変記号を定義する必要があります。

2. 辞書の設計

(例)

●SEWB+/REPOSITORY

業務ルール (ルールスクリプト)

```

@@if (!(@defined(@RULE_FLAG[時刻編集])))
  @@put @WORK <<
  01 W-時刻
  02 W-時 PIC 9(2).
  02 W-分 PIC 9(2).
  01 W-編集時刻.
  02 W-編集時 PIC 9(2).
  02 FILLER PIC X(1) VALUE ':'.
  02 W-編集分 PIC 9(2).
  @@end;
  @@set @RULE_FLAG[時刻編集] = 'Y';
@@end;
*時刻出力編集
MOVE @時刻 TO W-時刻
MOVE W-時 TO W-編集時
MOVE W-分 TO W-編集分
MOVE W-編集時刻 TO @編集時刻

```

●SEWB+/CONSTRUCTION

テンプレート記述

```

:
@@interface @IN_FILE = { ATTR = FILE };
@@interface @OUT_FILE = { ATTR = FILE };
@@global @RULE_FLAG;
:
WORKING-STORAGE SECTION.
@@merge @WORK;
PROCEDURE DIVISION.
@@rule "時刻編集処理" WITH USAGE
  {{ USAGE=IN REF=@IN_FILE }
  { USAGE=OUT REF=@OUT_FILE}};
:

```

生成ソースプログラム

```

:
WORKING-STORAGE SECTION.
01 W-時刻
02 W-時 PIC 9(2).
02 W-分 PIC 9(2).
01 W-編集時刻.
02 W-編集時 PIC 9(2).
02 FILLER PIC X(1) VALUE ':'.
02 W-編集分 PIC 9(2).
:
PROCEDURE DIVISION.
:
MOVE 時刻 TO W-時刻
MOVE W-時 TO W-編集時
MOVE W-分 TO W-編集分
MOVE W-編集時刻 TO 編集時刻
:

```

この例では、展開制御文の一つである @@put 文を使用して分割展開しています。ソースプログラム中に最初に業務ルールが展開されたとき、@@put 文で指定された作業領域が、テンプレート上の同じ可変記号「@WORK」が指定された @@merge 文の位置に展開されます。同時に、可変記号 @RULE_FLAG[時刻編集] に文字列「Y」がセットされます。以降にこの業務ルールが展開されても、@@if 文の判定によって作業領域の展開が抑止されます。なお、ルールスクリプトに記述されている @@if 文、@@put 文、@@set 文は、すべてテンプレート記述言語の展開制御文です。詳細は、マニュアル「SEWB+/CONSTRUCTION アプリケーション開発ガイド」を参照してください。

(8) データ項目の定義情報を取り出す

データ項目に定義されているけた数、タイプ、反復回数または修飾情報付データ項目名を、それぞれに対応した予約キーワードで取り出せます。取得した値を比較条件に利用したり、処理に渡したりすることで、汎用性の高い業務ルールを作成できます。例えば、同じドメインを継承するデータ項目であっても、けた数、タイプなどが異なる場合があります。このようなとき、定義情報が異なる部分を予約キーワードを使って記述すれば、ドメインに関連する共通の業務ルールを作成できます。

(a) @len

データ項目に定義されているけた数を取り出します。

形式

@len(キーワード)

規則

- キーワードに対応するデータ項目が結合項目の場合、またはけた数が定義されていない場合には、取得されません。
- ルールスクリプトを記述している言語が「C 又は C++」で、分類に日本語文字列データを定義したデータ項目のけた数では、けた数の 2 倍の値が取得されます。

(例)

キーワード「@コード」に対応するデータ項目のけた数を取り出し、外部サブルーチンに引き渡すインタフェース領域にセットします。

```
@@section WORK
01 W-LEN PIC 9.
@@section MAIN
MOVE @len(@コード) TO W-LEN ...@コードのけた数を数値で取り出す
CALL 'CDLENCK' USING @コード W-LEN
```

(b) @type

データ項目に定義されているタイプを取り出します。

形式

@type(キーワード)

規則

- キーワードに対応するデータ項目が結合項目の場合は、取得されません。
- @type で取得される値とタイプの対応を次に示します。@type で取得される値は、SEWB+/CONSTRUCTION のテンプレート @@lang 文の FOR_REPOSITORY 句に指定された言語区分に対応した値となります。
また、追加したユーザ定義のタイプで取得される @type の値は、そのタイプの英字識別子となります。例えば、「B 符号付き 2 進項目」の場合は「B」が取得されます。

2. 辞書の設計

表 2-19 タイプと @type の値の対応 (COBOL 又は OOCOBOL)

タイプ	@type の値
X 英数字項目	X
Z 数字編集項目	Z
N 漢字項目	N
9 符号なし外部 10 進項目	9
S 符号付き外部 10 進項目	S
U 符号なし内部 10 進項目	U
P 符号付き内部 10 進項目	P
BU 符号なし 2 進項目	BU
B 符号付き 2 進項目	B
E 外部浮動小数点項目	E
D 内部浮動小数点項目	D
1 内部ブール項目	1
8 外部ブール項目	8
T アドレスデータ項目	T
フリー定義	フリー定義の内容

表 2-20 タイプと @type の値の対応 (C 又は C++)

タイプ	@type の値
char 文字型	char
char* 文字型 (ポインタ)	char*
short 符号付き短整数型	short
unsigned short 符号なし短整数型	unsigned short
int 符号付き整数型	int
unsigned int 符号なし整数型	unsigned int
long 符号付き長整数型	long
unsigned long 符号なし長整数型	unsigned long
float 単精度浮動小数点型	float
double 倍精度浮動小数点型	double
long double 拡張精度浮動小数点型	long double
フリー定義	フリー定義の内容

表 2-21 タイプと @type の値の対応 (SQL)

タイプ	@type の値
CHAR 固定長文字列	CHAR
VARCHAR2 可変長文字列	VARCHAR2
LONG 可変長長文字列	LONG
NUMBER 固定小数点数	NUMBER
NUMBERf 浮動小数点数	NUMBERf
DATE 日付	DATE
VARCHAR 可変長文字列	VARCHAR
NCHAR 固定長各国文字列	NCHAR
NVARCHAR 可変長各国文字列	NVARCHAR
MCHAR 固定長混在文字列	MCHAR
MVARCHAR 可変長混在文字列	MVARCHAR
SMALLINT 短整数	SMALLINT
INT 整数	INT
DEC 固定小数点数	DEC
SMALLFLT 単精度浮動小数点数	SMALLFLT
FLOAT 倍精度浮動小数点数	FLOAT
TIME 時刻	TIME
フリー定義	フリー定義の内容

表 2-22 タイプと @type の値の対応 (IDL (CORBA))

タイプ	@type の値
char 文字型	char
string バウンディッド文字列型	string
unbounded string アンバウンディッド文字列型	unbounded string
unsigned short 符号なし短整数型	unsigned short
short 符号付き短整数型	short
unsigned long 符号なし長整数型	unsigned long
long 符号付き長整数型	long
float 単精度浮動小数点型	float
double 倍精度浮動小数点型	double
boolean 論理型	boolean
octet オクテット型	octet

2. 辞書の設計

タイプ	@type の値
any 任意データ型	any
long long 符号付き長々整数型	long long
unsigned long long 符号なし長々整数型	unsigned long long
long double 拡張倍精度浮動小数点型	long double
wchar ワイド文字型	wchar
wstring ワイド文字列型	wstring
fixed 符号付き固定小数点型	fixed
フリー定義	フリー定義の内容

表 2-23 タイプと @type の値の対応 (帳票定義)

タイプ	@type の値
C 文字データ	C
I 数値データ	I
B 集計対象外数値データ	B
CY 年次データ	CY
MO 月次データ	MO
DA 日次データ	DA
HS 日付・時刻データ	HS
M 画像データ	M
SP 特殊編集レベルデータ	SP
フリー定義	フリー定義の内容

表 2-24 タイプと @type の値の対応 (Java)

タイプ	@type の値
char 文字型	char
short 符号付き単整数型	short
int 符号付き整数型	int
long 符号付き長整数型	long
float 単精度浮動小数点型	float
double 倍精度浮動小数点型	double
byte バイト型	byte
boolean 論理型	boolean
フリー定義	フリー定義の内容

(例)

キーワード「@氏名」に対応するデータ項目のタイプを取り出し、「X 英数字項目」、
「N 漢字項目」のタイプ別に処理を分岐します。

```

@@section WORK
 77 W-TYPE PIC X.
@@section MAIN
  MOVE '@type(@氏名)' TO W-TYPE ...@氏名のタイプを文字で取り出す
  EVALUATE W-TYPE
    WHEN 'X' CALL 'EIJICK' USING @氏名
    WHEN 'N' CALL 'KANJICK' USING @氏名
  END-EVALUATE

```

(c) @occurs

データ項目に定義されている反復回数を取り出します。

形式

@occurs(キーワード)

規則

- キーワードに対応するデータ項目に反復回数が定義されていない場合には、取得されません。

(例)

キーワード「@金額」に対応するデータ項目の反復回数を取り出し、反復回数分、
処理をループします。

```

@@section WORK
 77 W-COUNT PIC 9(2).
@@section MAIN
  MOVE ZERO TO W-COUNT
  PERFORM WITH TEST AFTER
    VARYING W-COUNT FROM 1 BY 1
    UNTIL W-COUNT = @occurs(@金額)...反復回数(数値)を、作業用カウンタの値と比較する
  COMPUTE @合計金額 = @合計金額 + @金額(W-COUNT)
  END-PERFORM

```

(d) @occurs_dimension

結合関係の最も上位に位置するデータ項目から、キーワードに対応するデータ項目までの、
配列の次元数を取り出します。結合関係の最も上位にある結合項目からそのデータ
項目までに反復回数が定義されていない場合は、0が展開されます。

形式

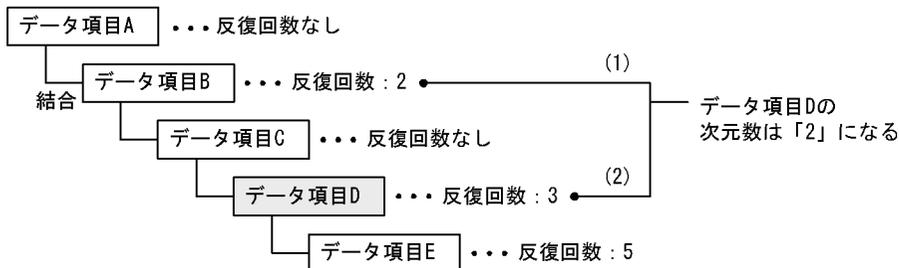
@occurs_dimension(キーワード)

(例)

データ項目が次のような結合関係にあるときを例に、次元数がどのように取り出さ

2. 辞書の設計

れるかについて考えてみましょう。キーワード「@ データ項目 d」には「データ項目 D」が対応するとします。例えば、ルールスクリプトに「@occurs_dimension(@ データ項目 d)」を記述すると、「データ項目 D」の次元数として「2」が取り出されます。



規則

- SEWB+/RECORD DEFINER のレコード定義に取り込まれたデータ項目の場合、レコード定義で定義された上位項目の配列は、次元数に反映されません。

(e) @occurs1 ~ @occurs9

キーワードに対応するデータ項目の反復回数、または結合関係の上位にあるデータ項目の反復回数を取り出します。主に @occurs_dimension で取り出した次元数と組み合わせで使用します。

形式

@occurs1(キーワード) ~ @occurs9(キーワード)

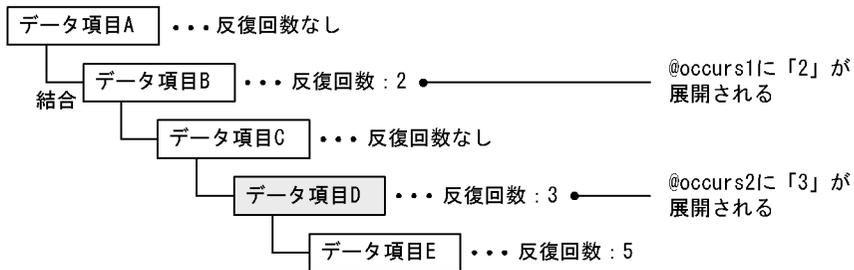
規則

- @occurs1 は、キーワードに対応するデータ項目が属する結合関係の中で、反復回数が定義されたデータ項目のうち、最も上位にある結合項目の反復回数を取り出します。反復回数が定義されたデータ項目が、結合関係の下位になるに従って、@occurs2, @occurs3, ...@occurs9 が対応します。
- 予約キーワードに指定した次元数 (@occursN の、N に記述した数字) が、データ項目の配列の次元数 (@occurs_dimension で取り出した次元数) を超える場合は、何も展開されません。
- SEWB+/RECORD DEFINER のレコード定義に取り込まれたデータ項目の場合、レコード定義で定義された上位項目の反復回数は展開されません。

(例1)

「データ項目 D」の次元数として「2」が取り出された場合を考えてみましょう。キーワード「@ データ項目 d」は「データ項目 D」に対応するとします。ルールスクリプトに「@occurs1(@ データ項目 d)」と記述すると、反復回数が定義されたデータ項目のうち、結合関係の最も上位にある「データ項目 B」の反復回数を取り出されます。また、「@occurs2(@ データ項目 d)」と記述すると、一つ下位の反復回数が定義されたデータ項目、つまり「データ項目 D」の反復回数を取り出されます。な

お、「データ項目 D」の配列の次元数は「2」となっているため、@occurs3 を使用しても何も展開されません。



(例 2)

@occurs_dimension と @occurs1 ~ @occurs9 を組み合わせたルールスクリプトの記述例を示します。まず、@occurs_dimension を使って、キーワード「@コード」に対応するデータ項目の、結合関係の最も上位にあるデータ項目からの配列の次元数を取り出します。各次元の結合項目に定義された反復回数分、処理をループして、配列に含まれるすべてのコードの数字チェックを行います。

```

@@if (@occurs_dimension(@コード) == 0)
  IF @コード IS NOT NUMERIC
    MOVE 1011 TO ERR-CODE
  END-IF
@@elseif (@occurs_dimension(@コード) == 1)
  @@put @WORK <<
  01 WCNT1@rulecount PIC S9(4) COMP.
  @@end;
PERFORM VARYING WCNT1@rulecount FROM 1 BY 1
  UNTIL WCNT1@rulecount > @occurs1(@コード)
  IF @コード(WCNT1@rulecount) IS NOT NUMERIC
    MOVE 1011 TO ERROR-CODE
  END-IF
END-PERFORM
@@else
  @@msg "次元数が2次元を超える数字チェックを以降に記述します。"
@@end;

```

(f) @modify

キーワードに対応するデータ項目名を修飾情報（全バスの項目名）付きで取り出します。

形式

@modify (キーワード)

規則

- 取得するデータ項目名の修飾順序は、SEWB+/CONSTRUCTION のテンプレート @@lang 文の MODIFY_ORDER 句の指定に従います。MODIFY_ORDER 句の指定がなく、ルールスクリプトを記述している言語が「C 又は C++」, 「COBOL 又は OOCOBOL」 以外の場合、修飾順序は DESCEND（上位レベル項目から下位

2. 辞書の設計

レベル項目の順に修飾)になります。

- 取得するデータ項目の言語区分は SEWB+/CONSTRUCTION のテンプレート @@lang 文の FOR_REPOSITORY 句の指定に従います。指定された言語別の名前が定義されていない場合、共通のデータ項目名となります。指定された言語区分が SEWB+/REPOSITORY の辞書に登録されているものと一致しない場合、ソースプログラムの生成時にエラーとなります。
- 修飾情報の連結文字は SEWB+/CONSTRUCTION のテンプレート @@lang 文の MODIFY_CONNECT 句の指定に従います。MODIFY_CONNECT 句の指定がなく、ルールスクリプトを記述している言語が「C 又は C++」、「COBOL 又は OOCOBOL」以外の場合、修飾情報（データ項目名）がそのまま連結されて置換されます。
- 最上位結合項目名は SEWB+/CONSTRUCTION のテンプレート @@rule 文の TOP 句で指定した文字列に変更して取得できます。
- 取得するデータ項目名には、SEWB+/CONSTRUCTION のテンプレート @@rule 文で指定した接頭語および接尾語を付けられます。
- 取得するデータ項目の上位項目に繰り返しがある場合、ソースプログラムの生成時にエラーとなります。
- SEWB+/RECORD DEFINER のレコード定義に取り込まれたデータ項目の場合、レコード定義で定義された上位項目についての修飾は展開されません。

(例)

キーワード「@受注日」および「@手配日」に対応するデータ項目名を修飾情報付きの項目名に置き換えて、代入文を生成します。

```
MOVE @modify (@受注日)
      TO @modify (@手配日) .
```

(9) チェック処理を記述する場合には

データの妥当性をチェックする処理での、チェック結果を受けたエラー処理の記述について説明します。エラー処理は一般に、アプリケーション間で共通化することが望ましいと考えられます。したがって、エラー処理の内容や、エラー処理に使うデータ項目は、SEWB+/CONSTRUCTION のテンプレート上に記述することをお勧めします（さらに、エラー処理部分は複数のテンプレートから汎用的に使えるよう、部品テンプレートにしておくことをお勧めします）。

業務ルールのチェック処理とテンプレートのエラー処理を適切に連携するには、データ項目名や相互でやり取りする値を正しく記述することが重要です。このため、業務ルールやテンプレートを作成する前に、あらかじめエラー処理の仕様を検討して、そのインタフェース（エラー処理に使用するデータ項目、やり取りする値など）を決めておきます。辞書の作成者およびテンプレート作成者は、この仕様に基づいて、業務ルールやテンプレートを作成するようにしてください。

(例)

業務ルールAおよびBでは共通に、「RTN-CODE」に処理結果を、「ERR-FLG」にエラーの有無を、「ERR-VALUE」にチェックしたデータの種類を設定しています。これらのデータ項目の定義とチェック結果に従って実行するエラー処理「ERR-DATA-EDIT-PROC」と「ERR-LIST-WRITE-PROC」は、テンプレート上に記述しています。

●SEWB+/REPOSITORY

業務ルールA (ルールスクリプト)

```
*氏名コードチェック
CALL 'SHICDCK' USING @氏名コード RTN-CODE
IF RTN-CODE = SPACE
  THEN CONTINUE
  ELSE MOVE '1'          TO ERR-FLG
        MOVE '氏名コード' TO ERR-VALUE
END-IF
```

業務ルールB (ルールスクリプト)

```
*顧客コードチェック
CALL 'KOKCDCK' USING @顧客コード RTN-CODE
IF RTN-CODE = SPACE
  THEN CONTINUE
  ELSE MOVE '1'          TO ERR-FLG
        MOVE '顧客コード' TO ERR-VALUE
END-IF
```

●SEWB+/CONSTRUCTION

テンプレート記述

```
:
@@interface @IN_FILE = {ATTR=FILE};
WORKING-STORAGE SECTION.
01 RTN-CODE PIC X(3).
01 WK-ERR. 02 ERR-FLG PIC X(1).
  02 ERR-VALUE PIC N(5).
:
PROCEDURE DIVISION.
:
--- MOVE '0' TO ERR-FLG
  @@rule "コードチェック処理" WITH USAGE
    {[USAGE=IN REF=@IN_FILE]};
  IF ERR-FLG = '1'
    THEN PERFORM ERR-DATA-EDIT-PROC
    PERFORM ERR-LIST-WRITE-PROC
  END-IF
:
```

(10) 予約キーワードの留意事項

予約キーワード(「@type」,「@len」など)を記述する場合は、次の点に留意してください。

- 予約キーワードは、小文字または大文字のどちらでも記述できます。ただし、両方を混在して記述することはできません。
- 括弧「()」を使って値を取得する予約キーワードでは、括弧の前後に空白や改行を挿入してはなりません。

2.6.4 適用条件の指定

適用条件とは、アプリケーションに業務ルールを適用する条件を指定するものです。辞書の作成者は、「データ項目がどのような条件(処理の入力なのか、出力なのか)にあるときに、この業務ルールをアプリケーションに適用させるのか」という観点で指定します。適用条件を指定するには、「2.6.2(1) 業務ルールの基準を決める」で説明したように、テンプレートに記述される @@rule 文の抽出条件の指定と考え方が統一されていることが大切です。

(1) 業務ルールの分類と適用条件の指定

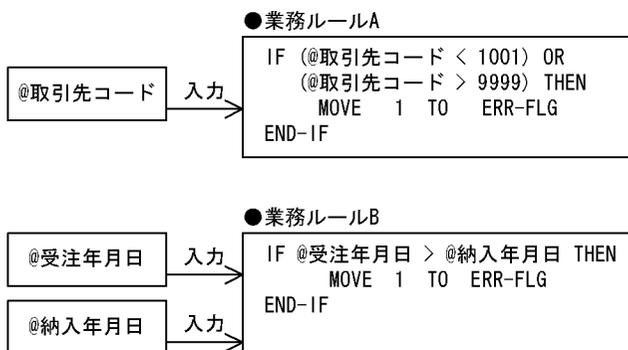
業務ルールの処理は一般に、「データの値の妥当性をチェックする処理」と「データの値を導き出す処理」に大別されると考えられます。ここでは、この分類を例に、適用条件の指定について考えてみましょう。

(a) データの値の妥当性をチェックする処理

単一のデータ項目または複数のデータ項目間のデータの値の妥当性をチェックする処理には、次のように適用条件を指定します。

- チェック対象のデータ項目に、「入力」を指定する

(例)

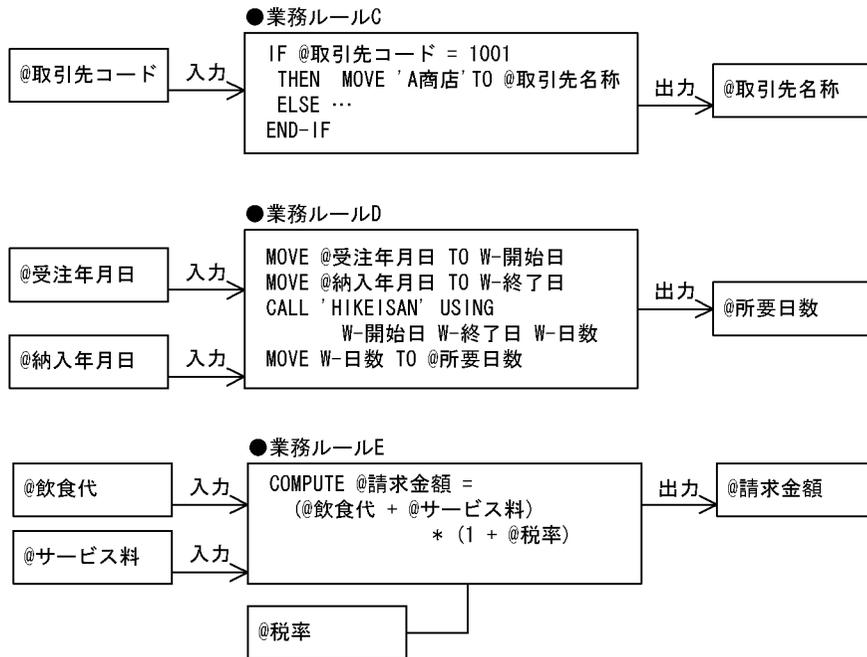


(b) データの値を導き出す処理

編集処理や計算処理といった、あるデータ項目の値から別のデータ項目の値を導き出す処理には、次のように適用条件を指定します。

- データの入力元となるデータ項目に「入力」を指定する
- 導き出された結果を格納するデータ項目に「出力」を指定する

(例)



なお、業務ルール E のデータ項目「@税率」のように、処理をする際に補助的に利用するデータ項目については、適用条件を指定しないで「-」のままにしておきます。

(2) @@rule 文で抽出される業務ルール

業務ルールの適用条件の指定を検討するときには、テンプレートの @@rule 文の指定によって、目的の業務ルールが適切に抽出されるかどうかについても考慮する必要があります。@@rule 文による業務ルールの抽出には、WITH USAGE 句を使って抽出条件を基に業務ルールを絞り込む方法と、絞り込みを行わない方法の 2 通りがあります。@@rule 文の抽出条件の指定や、抽出の考え方については、マニュアル「SEWB+/CONSTRUCTION アプリケーション開発ガイド」を参照してください。

(1) の業務ルール A ~ E を抽出するときの @@rule 文の記述について、例を見ながら考えてみましょう。次に、@@rule 文の記述例と、データ定義（ファイルおよびレコード定義）にファイル a ~ ファイル i を仮定したとき抽出される業務ルールを示します。

2. 辞書の設計

<データ定義例>

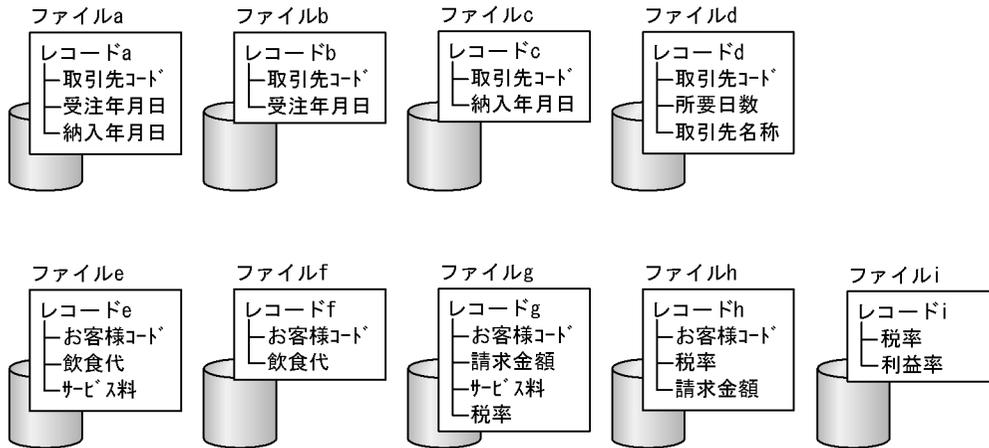


表 2-25 @@rule 文の記述例と抽出される業務ルール

項番	@@rule 文の記述例	抽出される業務ルール				
		A	B	C	D	E
1	@@rule "展開1" WITH USAGE{{USAGE=IN REF=ファイルa}};			x	x	x
2	@@rule "展開2" WITH USAGE{{USAGE=IN REF=ファイルb}{USAGE=IN REF=ファイルc}};	x		x	x	x
3	@@rule "展開3" WITH USAGE{{USAGE=IN REF=ファイルb REF=ファイルc}};			x	x	x
4	@@rule "展開4" WITH USAGE{{USAGE=IN REF=ファイルa}{USAGE=OUT REF=ファイルd}};	x	x			x
5	@@rule "展開5" WITH USAGE{{USAGE=IN REF=ファイルb}{USAGE=IN REF=ファイルc}{USAGE=OUT REF=ファイルd}};	x	x	x		x
6	@@rule "展開6" WITH USAGE{{USAGE=IN REF=ファイルb REF=ファイルc}{USAGE=OUT REF=ファイルd}};	x	x			x
7	@@rule "展開7" WITH USAGE{{USAGE=IN REF=ファイルe}{USAGE=OUT REF=ファイルg}};	x	x	x	x	
8	@@rule "展開8" WITH USAGE{{USAGE=IN REF=ファイルe}{USAGE=OUT REF=ファイルh}};	x	x	x	x	
9	@@rule "展開9" WITH USAGE{{USAGE=IN REF=ファイルe}{USAGE=OUT REF=ファイルg}REF=ファイルi};	x	x	x	x	
10	@@rule "展開10" {REF=ファイルb REF=ファイルc REF=ファイルd REF=ファイルe REF=ファイルh};					

(凡例)

○ : 抽出される
 × : 抽出されない

注

取引先コードがファイル b 側に含まれるものとファイル c 側に含まれるものの 2 通り抽出されます。

(a) データの値の妥当性をチェックする業務ルールの抽出

@@rule 文に抽出条件を指定して、この業務ルールを抽出するには、記述例 1, 2 および 3 のような記述が考えられます。

記述例 2 では、ファイル b とファイル c の両方に入力側のデータ項目がある業務ルールが対象になるため、業務ルール B が抽出されます。また、記述例 3 では、ファイル b とファイル c のどちらかに入力側のデータ項目がある業務ルールが対象になるため、業務ルール A と B の両方が抽出されます。

(b) データの値を導き出す業務ルールの抽出

@@rule 文に抽出条件を指定して、この業務ルールを抽出するには、記述例 4 ~ 9 のような記述が考えられます。

@@rule 文の記述によって、記述例 5 のように、使用するデータ項目が三つのレコードに含まれる業務ルールだけを絞り込むこともできます。また、記述例 6 のように、どちらかの入力レコードに含まれる入力側のデータ項目があれば抽出されるように指定することもできます。

業務ルール E では、データ項目「@ 税率」に適用条件を指定していませんでした。こうした場合、そのデータ項目が、@@rule 文の USAGE=IN で指定したデータ定義、USAGE=OUT で指定したデータ定義、あるいは USAGE 指定をしていないデータ定義のうち、どれかのデータ定義中に含まれれば、その業務ルールが抽出されます。したがって、業務ルール E は、記述例 7, 8, 9 のどれからでも抽出されることになります。

(c) 絞り込みをしない抽出

記述例 10 のように、@@rule 文に抽出条件の指定をしない場合には、データ項目が関連するすべての業務ルールが抽出されます。この抽出方法では、業務ルールに指定した適用条件は、関係ありません。記述例 10 では、データの値の妥当性をチェックする業務ルール（業務ルール A, B）およびデータの値を導き出す業務ルール（業務ルール C ~ E）のどれもが抽出されます。

2.6.5 システムで共用する業務ルールを登録する

実際に業務ルール辞書を整備するには、まず、ドメイン、つまり、継承関係の上位のデータ項目の業務ルールから登録を始めます。ドメインと業務ルールの関連は、データ項目の体系と同じようにドメインを継承するすべてのデータ項目に引き継がれ、業務ルールを共用できます。したがって、ドメインの業務ルールとは、システムで共用する

2. 辞書の設計

ことを目的とした汎用性の高い業務ルールといえます。

ドメインの業務ルールは、次の手順で登録します。

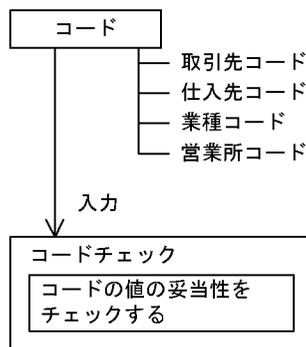
1. データの値の妥当性をチェックする業務ルール
2. データの値を導き出す業務ルール

ここでは、例題に沿ってドメインの業務ルールの登録を説明します。

(1) データの妥当性をチェックする業務ルールを登録する

(a) 例題の概要

データ分析の結果、ドメインのデータ項目「コード」には、「0000」,「9999」以外の数字で入力されなければならない、という制約条件が抽出されました。この制約条件は、ドメインを継承するデータ項目に共通です。例題では、入力された値の妥当性をチェックする業務ルール「コードチェック」を登録します。



(b) 業務ルールの定義

リポジトリブラウザを使った定義について説明します。業務ルールを新しく登録するには、業務ルールのアイコンを選び、[リポジトリ] - [新規作成] - [業務ルール] を選んでください。業務ルールの情報を定義するダイアログが表示されます。

1. 業務ルール名、コメントを定義する

[属性] タブの情報を定義します。業務ルール名とコメントは、SEWB+/CONSTRUCTION でプログラマが目的の業務ルールを選択するときの大切な情報となります。理解しやすい業務ルール名、処理内容や展開条件を適切に把握できるようなコメントを定義しましょう。

- 業務ルール名に、「コードチェック」
コメントに、「入力されたコードの値が、「0000」以上「9999」未満および数字かどうかをチェックする。」

2. 適用するデータ項目と業務ルールの関連を定義する

[データ項目] タブの情報を定義します。「コード」特有の制約条件を反映するため、データ項目の適用形態は「専用にする」を選びます。

- 組み合わせるデータ項目に、「コード」
キーワードに、「@コード」
適用条件に、「入力」

3. ルールスクリプトを記述する

[ルールスクリプト] タブの情報を定義します。

```
* コードチェック
**'0000'..'9999'以外の数字*
  IF (@コード IS NUMERIC AND
      '0000' < @コード AND @コード < '9999')
    THEN CONTINUE
    ELSE MOVE '1001' TO R-CHECK-STAT
    END-IF
```

記述のポイントを次に示します。

- データ項目の名称をキーワード「@コード」で記述します。
- エラーの記述に使うデータ項目や値は、あらかじめテンプレート作成者と決めておいたエラー処理の仕様に従います。ここでは、エラー状態 '1001' を「R-CHECK-STAT」にセットしています。このデータ項目の定義は、テンプレート上に記述されることを前提としています。

(c) 業務ルールの適用例

業務ルール「コードチェック」が、SEWB+/CONSTRUCTION からアプリケーションに適用される例を示します。

2. 辞書の設計

1. テンプレート記述

```
@@interface @IN_FILE = { ATTR=IN_FILE };
:
WORKING-STORAGE SECTION.
01 R-CHECK-STAT PIC X(4).
PROCEDURE DIVISION.
:
MOVE SPACE TO R-CHECK-STAT
@@rule "チェック処理" WITH USAGE
  {{USAGE=IN REF=@IN_FILE PREFIX='IN-'}};
IF R-CHECK-STAT NOT = SPACE
  THEN PERFORM ERR-PROC
END-IF
```

2. プログラム定義

業務ルール展開設定
コードチェック

3. 生成ソースプログラム

```
:
WORKING-STORAGE SECTION.
01 R-CHECK-STAT PIC X(4).
PROCEDURE DIVISION.
:
MOVE SPACE TO R-CHECK-STAT
*コードチェック
**'0000','9999'以外の数字*
IF (IN-業種コード IS NUMERIC AND
  '0000' < IN-業種コード AND IN-業種コード
  < '9999')
  THEN CONTINUE
  ELSE MOVE '1001' TO R-CHECK-STAT
END-IF
IF R-CHECK-STAT NOT = SPACE
  THEN PERFORM ERR-PROC
END-IF
```

1. テンプレート記述

エラー状態を格納するデータ項目「R-CHECK-STAT」は、ここに記述されます。

@@rule 文では、次の指定をしています。

- データ定義 IN_FILE に含まれるデータ項目を使用している業務ルールを抽出する。このとき、データ項目の適用条件に「入力」が指定された業務ルールだけを対象にする。
- データ項目の言語別の名前に接頭語「IN-」を付加する。

2. プログラム定義

プログラム定義で 1. のテンプレートを選択し、IN_FILE に指定されたデータ定義のレコード定義に、データ項目「コード」、「取引先コード」、「仕入先コード」、「業種コード」または「営業所コード」のどれかが含まれている場合、業務ルールの候補として「コードチェック」が表示されます。

3. 生成ソースプログラム

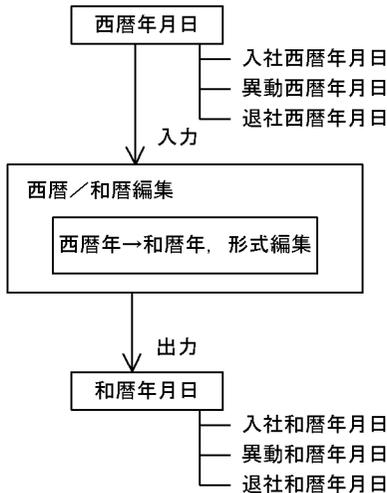
@@rule 文の位置にルールスクリプトが展開されます。この例は、アプリケーションで使用するデータ項目が「業種コード」の場合です。

(2) データの値を導き出す業務ルールを登録する

(a) 例題の概要

例題業務では、年月日データをクライアント画面に表示したりレポートに印刷したりする際には、西暦年を和暦年に変換し、さらに出力用の形式に編集します（例：19970301

平成 9 年 3 月 1 日)。西暦年月日の形式を持つデータ項目は、ドメインのデータ項目「西暦年月日」にまとめられています。また、和暦年月日の形式を持つデータ項目は、ドメインのデータ項目「和暦年月日」にまとめられています。例題では、この二つのドメインを組み合わせ、データの値を導き出す業務ルール「西暦 / 和暦編集」を登録します。



(b) 業務ルールの定義

リポジトリブラウザを使った定義について説明します。業務ルールを新しく登録するには、業務ルールのアイコンを選び、[リポジトリ] - [新規作成] - [業務ルール]を選んでください。業務ルールの情報を定義するダイアログが表示されます。

1. 業務ルール名、コメントを定義する

[属性] タブの情報を定義します。理解しやすい業務ルール名を付け、ドメインの導出関係、処理内容や展開条件が適切に把握できるようなコメントを定義しましょう。

- 業務ルール名に、「西暦 / 和暦編集」

コメントに、「西暦年を和暦年に変換し、「年号 YY 年 MM 月 DD 日」の形式に編集する。和暦年、月、日が 1 けたの場合、左側は空白にする。編集処理は外部サブルーチン「UXDATE」を使う」

2. 適用するデータ項目と業務ルールの関連を定義する

[データ項目] タブの情報を定義します。「西暦年月日」および「和暦年月日」の組み合わせに特有の処理のため、データ項目の適用形態は「専用にする」を選びます。

- 組み合わせるデータ項目に、「西暦年月日」
キーワードに、「@ 西暦年月日」
適用条件に、「入力」
- 組み合わせるデータ項目に、「和暦年月日」
キーワードに、「@ 和暦年月日」
適用条件に、「出力」

2. 辞書の設計

3. ルールスクリプトを記述する

[ルールスクリプト]タブの情報を定義します。

```
@@section WORK
01 W-編集年月日.
   02 W-西暦年月日-IN PIC X(8).
   02 W-和暦年月日-OUT PIC N(12).
@@section MAIN
MOVE @西暦年月日 TO W-西暦年月日-IN
CALL 'UXDATE' USING W-編集年月日
MOVE W-和暦年月日-OUT TO @和暦年月日
```

記述のポイントを次に示します。

- 入力するデータ項目をキーワード「@西暦年月日」で、編集結果を出力するデータ項目をキーワード「@和暦年月日」で記述します。
- 外部サブルーチンとのインタフェース領域は、@@section 文で WORKING-STORAGE SECTION への分割展開を指定します。@@section 文に指定するコード名は、あらかじめテンプレート作成者と決めておいたコード名を使います。ここでは、「WORK」を指定しました。このコード名に対応する @@merge 文が、テンプレート上に記述されることを前提としています。

(c) 業務ルールの適用例

業務ルール「西暦 / 和暦編集」が、SEWB+/CONSTRUCTION からアプリケーションに適用される例を示します。

1. テンプレート記述

```
@@interface @IN_FILE = {ATTR=FILE};
@@interface @OUT_FILE = {ATTR=FILE};
:
WORKING-STORAGE SECTION.
@@merge @WORK;
PROCEDURE DIVISION.
:
@@rule "編集処理" WITH USAGE
  {{USAGE=IN REF=@IN_FILE}
  {USAGE=OUT REF=@OUT_FILE}};
:
```

2. プログラム定義

業務ルール展開設定
西暦 / 和暦編集

3. 生成ソースプログラム

```
:
WORKING-STORAGE SECTION.
01 W-編集年月日.
   02 W-西暦年月日-IN PIC X(8).
   02 W-和暦年月日-OUT PIC N(12).
:
PROCEDURE DIVISION.
MOVE 入社西暦年月日 TO W-西暦年月日-IN
CALL 'UXDATE' USING W-編集年月日
MOVE W-和暦年月日-OUT TO 入社和暦年月日
:
```

1. テンプレート記述

ルールスクリプトの作業領域（インタフェース領域）の展開位置は、「@@merge

@WORK」で指定されています。@@rule 文では、次の二つの条件を満たす業務ルールを抽出することが指定されています。

- データ定義 IN_FILE に含まれるデータ項目を使用している業務ルール。このとき、データ項目の適用条件に「入力」が指定された業務ルールだけを対象にする。
- データ定義 OUT_FILE に含まれるデータ項目を使用している業務ルール。このとき、データ項目の適用条件に「出力」が指定された業務ルールだけを対象にする。

2. プログラム定義

プログラム定義で 1. のテンプレートを選択し、@@rule 文の指定に一致するデータ項目をアプリケーションに使用している場合、業務ルールの候補として「西暦 / 和暦編集」が表示されます。ドメインの下位のデータ項目を使用している場合も抽出されます。

3. 生成ソースプログラム

テンプレートで指定された位置にルールスクリプトが展開されます。この例は、アプリケーションで使用するデータ項目が「入社西暦年月日」と「入社和暦年月日」の場合です。

2.6.6 特定業務で扱う業務ルールを登録する

ドメインの業務ルールを業務ルール辞書に整備したあと、継承関係の下位にあるデータ項目の業務ルールを登録します。特定のデータ項目に関する業務の規則、制度などを業務ルールとして登録します。

特定業務で扱う業務ルールも、ドメインの業務ルールと同様に、次の手順で登録します。

1. データの値の妥当性をチェックする業務ルール
2. データの値を導き出す業務ルール

(1) データの値の妥当性をチェックする業務ルールを登録する

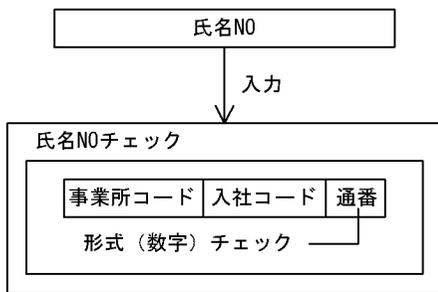
(a) 例題の概要

データ項目「氏名 NO」は、データ分析の結果、次の制約条件が抽出されました。

- 氏名 NO は、事業所コード、入社コードおよび通番の値で構成される
- 通番は必ず数字で入力する

例題では、「氏名 NO」に対して、入力された値の妥当性をチェックする業務ルール「氏名 NO チェック」を登録します。

2. 辞書の設計



(b) 業務ルールの定義

ここでは、例題業務ルールの定義情報を説明します。なお、リポジトリブラウザの定義手順は、ドメインの業務ルールを登録するときと同様です。「2.6.5 システムで共用する業務ルールを登録する」を参照してください。

1. 業務ルール名、コメント

- 業務ルール名に、「氏名 NO チェック」
コメントに、「入力された氏名 NO のデータを、事業所コード、入社年、通番に分割し、通番が数字で入力されているかをチェックする。」

2. 適用するデータ項目と業務ルールの関連

「氏名 NO」特有の制約条件を反映するため、データ項目の適用形態は「専用にする」を選びます。

- 組み合わせるデータ項目に、「氏名 NO」
キーワードに、「@ 氏名 NO」
適用条件に、「入力」

3. ルールスクリプトの記述

```
@@section WORK
01 W-氏名NO.
   02 W-事業所コード PIC X(3).
   02 W-入社コード   PIC 9(2).
   02 W-通番         PIC 9(3).
@@section MAIN
*氏名NOチェック
  MOVE @氏名NO TO W-氏名NO
  IF W-通番 = NUMERIC
    THEN CONTINUE
  ELSE MOVE '5001' TO R-CHECK-STAT
  END-IF
```

記述のポイントを次に示します。

- データ項目の名称をキーワード「@ 氏名 NO」で記述します。
- 通番だけをチェックするため、業務ルール内部の作業領域を使って氏名 NO のデータを分割します。作業領域の記述部分は、@@section 文を使用して WORKING-STORAGE SECTION への分割展開を指定します。ここでは、あらか

じめテンプレート作成者と決めておいたコード名「WORK」を指定しています。

- エラーの記述に使うデータ項目や値は、あらかじめテンプレート作成者と決めておいたエラー処理の仕様に従います。ここでは、エラー状態 '5001' を「R-CHECK-STAT」にセットしています。このデータ項目の定義は、テンプレート上に記述されることを前提としています。

(c) 業務ルールの適用例

業務ルール「氏名 NO チェック」が、SEWB+/CONSTRUCTION からアプリケーションに適用される例を示します。

1. テンプレート記述

```

@@interface @IN_FILE = {ATTR=FILE} ;
:
WORKING-STORAGE SECTION.
01 R-CHECK-STAT PIC X(4).
@@merge @WORK:
PROCEDURE DIVISION.
:
INPUT-CHECK SECTION.
CHECK-START.
MOVE SRACE TO R-CHECK-STAT
@@rule "入力チェック処理" WITH USAGE
  {{USAGE=IN REF=@IN_FILE
    PREFIX='IN-'}};
IF R-CHECK-STAT NOT = SPACE
  THEN PERFORM ERR-PROC
END-IF

```

2. プログラム定義

業務ルール展開設定
氏名NOチェック

3. 生成ソースプログラム

```

:
WORKING-STORAGE SECTION.
01 R-CHECK-STAT PIC X(4).
01 W-氏名NO.
02 W-事業所コード PIC X(3).
02 W-入社コード PIC 9(2).
02 W-通番 PIC 9(3).
PROCEDURE DIVISION.
:
INPUT-CHECK SECTION.
CHECK-START.
MOVE SRACE TO R-CHECK-STAT
:
*氏名NOチェック
MOVE IN-氏名NO TO W-氏名NO
IF W-通番 = NUMERIC
  THEN CONTINUE
  ELSE MOVE '5001' TO R-CHECK-STAT
END-IF
IF R-CHECK-STAT NOT = SPACE
  THEN PERFORM ERR-PROC
END-IF

```

1. テンプレート記述

エラー状態を格納するデータ項目「R-CHECK-STAT」は、ここに記述されます。また、データ分割に使用した作業領域の展開位置は、「@@merge @WORK」で指定されています。@@rule 文では、次の指定をしています。

- データ定義 IN_FILE に含まれるデータ項目を使用している業務ルールを抽出する。このとき、データ項目の適用条件に「入力」が指定された業務ルールだけを対象にする。

2. 辞書の設計

- データ項目の言語別の名前に接頭語「IN-」を付加する。

2. プログラム定義

プログラム定義で 1. のテンプレートを選択し、IN_FILE に指定されたデータ定義のレコード定義にデータ項目「氏名 NO」が含まれている場合、業務ルールの候補として「氏名 NO チェック」が表示されます。

3. 生成ソースプログラム

@@rule 文の位置にルールスクリプトが展開されます。

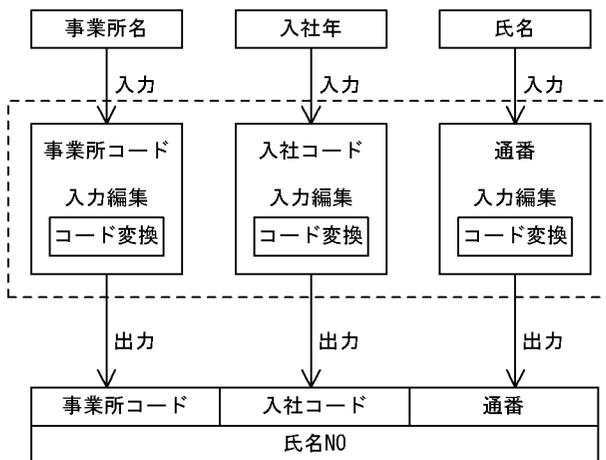
(2) データの値を導き出す業務ルールを登録する

(a) 例題の概要

例題業務では、従業員個人の識別に「氏名 NO」を使用します。このデータを格納するデータ項目「氏名 NO」は、データ項目「事業所コード」、「入社コード」および「通番」で構成されます。「氏名 NO」の三つの構成項目それぞれについて、次のような作成規則があります。

- 事業所コード...入力されたデータ項目「事業所名」の値からコード変換する
- 入社コード...入力されたデータ項目「入社年」の値からコード変換する
- 通番...入力されたデータ項目「氏名」の値から入社年別の通番を割り当てる

例題では、それぞれ導出関係にあるデータ項目を組み合わせ、データの値を導き出す業務ルールを登録します。SEWB+/CONSTRUCTION から、この三つの業務ルールを適用することで、結合項目「氏名 NO」の値を割り当てる処理を実現できます。



(b) 業務ルールの定義

ここでは、例題業務ルールの定義情報について説明します。リポジットブラウザの定義手順は、ドメインの業務ルールを登録するときと同様です。「2.6.5 システムで共用する業務ルールを登録する」を参照してください。

事業所コード入力編集

1. 業務ルール名, コメント

- 業務ルール名に,「事業所コード入力編集」
コメントに,「事業所名を事業所コードに変換する。変換処理は外部サブルーチン「CDHENKN1」を使う。」

2. 適用するデータ項目と業務ルールの関連

- 適用形態に,「専用にする」
組み合わせるデータ項目に,「入社年」
キーワードに,「@入社年」
適用条件に,「入力」
- 適用形態に,「専用にする」
組み合わせるデータ項目に,「入社コード」
キーワードに,「@入社コード」
適用条件に,「出力」

3. ルールスクリプトの記述

```

@@section WORK
  01 事業所コード変換 PIC X(8) VALUE 'CDHENKN1'.
  01 W-事業所コード.
    02 W-事業所コード-IN PIC N(10).
    02 W-事業所コード-OUT PIC X(5).
@@section MAIN
*事業所コード入力編集
  MOVE @事業所名 TO W-事業所コード-IN
  CALL 事業所コード変換 USING W-事業所コード
  MOVE W-事業所コード-OUT TO @事業所コード

```

入社コード入力編集

1. 業務ルール名, コメント

- 業務ルール名に,「入社コード入力編集」
コメントに,「入社年を入社コードに変換する。変換処理は外部サブルーチン「CDHENKN2」を使う。」

2. 適用するデータ項目と業務ルールの関連

- 適用形態に,「専用にする」
組み合わせるデータ項目に,「事業所名」
キーワードに,「@事業所名」
適用条件に,「入力」
- 適用形態に,「専用にする」
組み合わせるデータ項目に,「事業所コード」
キーワードに,「@事業所コード」
適用条件に,「出力」

3. ルールスクリプトの記述

2. 辞書の設計

```
@@section WORK
  01 入社コード変換 PIC X(8) VALUE 'CDHENKN2'.
  01 W-入社コード.
    02 W-入社コード-IN PIC 9(4).
    02 W-入社コード-OUT PIC 9(2).
@@section MAIN
*入社コード入力編集
  MOVE @入社年 TO W-入社コード-IN
  CALL 入社コード変換 USING W-入社コード
  MOVE W-入社コード-OUT TO @入社コード
```

通番入力編集

1. 業務ルール名, コメント

- 業務ルール名に, 「通番入力編集」
コメントに, 「氏名を通番に変換する。変換処理は外部サブルーチン「CDHENKN3」を使う。」

2. 適用するデータ項目と業務ルールの関連

- 適用形態に, 「専用にする」
組み合わせるデータ項目に, 「氏名」
キーワードに, 「@氏名」
適用条件に, 「入力」
- 適用形態に, 「専用にする」
組み合わせるデータ項目に, 「通番」
キーワードに, 「@通番」
適用条件に, 「出力」

3. ルールスクリプトの記述

```
@@section WORK
  01 通番変換 PIC X(8) VALUE 'CDHENKN3'.
  01 W-通番.
    02 W-通番-IN PIC N(10).
    02 W-通番-OUT PIC 9(4).
@@section MAIN
**通番入力編集
  MOVE @氏名 TO W-通番-IN
  CALL 通番変換 USING W-通番
  MOVE W-通番-OUT TO @通番
```

(c) 業務ルールの適用例

業務ルール「事業所コード入力編集」, 「氏名コード入力編集」および「通番入力編集」が, SEWB+/CONSTRUCTION からアプリケーションに適用される例を示します。

1. テンプレート記述

```

@@interface @IN_FILE = {ATTR=FILE};
@@interface @OUT_FILE = {ATTR=FILE};
:
WORKING-STORAGE SECTION.
  @@merge @WORK ;
PROCEDURE DIVISION.
:
INPUT-EDIT SECTION.
EDIT-START.
@@rule "入力編集処理" WITH USAGE
  {{USAGE=IN REF=@IN_FILE
    PREFIX=' IN-' }
  {USAGE=OUT REF=@OUT_FILE
    PREFIX=' OUT-' }};
:

```

2. プログラム定義

業務ルール展開設定

```

事業所コード入力編集
入社コード入力編集
通番入力編集

```

3. 生成ソースプログラム

```

:
WORKING-STORAGE SECTION.
01 事業所コード変換 PIC X(8) VALUE 'CDHENKN1'.
01 W-事業所コード*.
02 W-事業所コード-IN PIC N(10).
02 W-事業所コード-OUT PIC X(5).
01 入社コード変換 PIC X(8) VALUE 'CDHENKN2'.
01 W-入社コード*.
02 W-入社コード-IN PIC 9(4).
02 W-入社コード-OUT PIC 9(2).
01 通番変換 PIC X(8) VALUE 'CDHENKN3'.
01 W-通番*.
02 W-通番-IN PIC N(10).
02 W-通番-OUT PIC 9(4).
PROCEDURE DIVISION.
:
INPUT-EDIT SECTION.
EDIT-START.
*事業所コード入力編集
MOVE IN-事業所名 TO W-事業所コード-IN
CALL 事業所コード変換 USING W-事業所コード*
MOVE W-事業所コード-OUT TO OUT-事業所コード*
*入社コード入力編集
MOVE IN-入社年 TO W-入社コード-IN
CALL 入社コード変換 USING W-入社コード*
MOVE W-入社コード-OUT TO OUT-入社コード*.
*通番入力編集
MOVE IN-氏名 TO W-通番-IN
CALL 通番変換 USING W-通番
MOVE W-通番-OUT TO OUT-通番
:

```

1. テンプレート記述

@@rule 文では、次の二つの条件を満たす業務ルールを抽出することが指定されています。

- データ定義 IN_FILE に含まれるデータ項目を使用している業務ルール。このとき、データ項目の適用条件に「入力」が指定された業務ルールだけを対象にする。
- データ定義 OUT_FILE に含まれるデータ項目を使用している業務ルール。このとき、データ項目の適用条件に「出力」が指定された業務ルールだけを対象にする。

2. プログラム定義

1. のテンプレートを利用したプログラム定義で、アプリケーションの入力となるデー

2. 辞書の設計

タ定義のレコード定義にデータ項目「事業所名」、「入社年」および「氏名」が含まれており、また、出力となるデータ定義のレコード定義にデータ項目「事業所コード」、「入社コード」および「氏名コード」が含まれているとき、業務ルールの候補として三つの業務ルールが表示されます。

3. 生成ソースプログラム

@@rule 文の位置にルールスクリプトが展開されます。プログラマがこの三つの業務ルールを選んだ場合、例に示したようなソースプログラムが生成されます。

2.6.7 同一項目用業務ルールを登録する

ここでは、同一項目用業務ルールの登録を例題に沿って説明します。

(a) 例題の概要

同一のデータ項目間でデータを転記する同一項目用業務ルールを登録します。この業務ルールを、リポジトリ内のすべてのデータ項目に対応するようにします。

(b) 同一項目用業務ルールの定義

リポジトリブラウザを使った定義について説明します。業務ルールを新しく登録するには、業務ルールのアイコンを選び、[ツール] - [同一項目業務ルールの作成]を選んでください。同一項目用業務ルールを作成するダイアログが表示されます。

1. 業務ルール名、コメントを定義する

[属性]タブの情報を定義します。

- 業務ルール名に、「項目間転記処理」
コメントに、「同一のデータ項目間でデータを転記する。」

2. 適用するデータ項目を選択し、キーワードと適用条件を定義する

[データ項目]タブの情報を定義します。この例では、リポジトリ内のすべてのデータ項目に適用する[すべての項目]ボタンを選びます。

- 組み合わせるデータ項目に、「すべての項目」
キーワードに、「@入力項目」
適用条件に、「入力」
- 組み合わせるデータ項目に、「すべての項目」
キーワードに、「@出力項目」
適用条件に、「出力」

3. ルールスクリプトを記述する

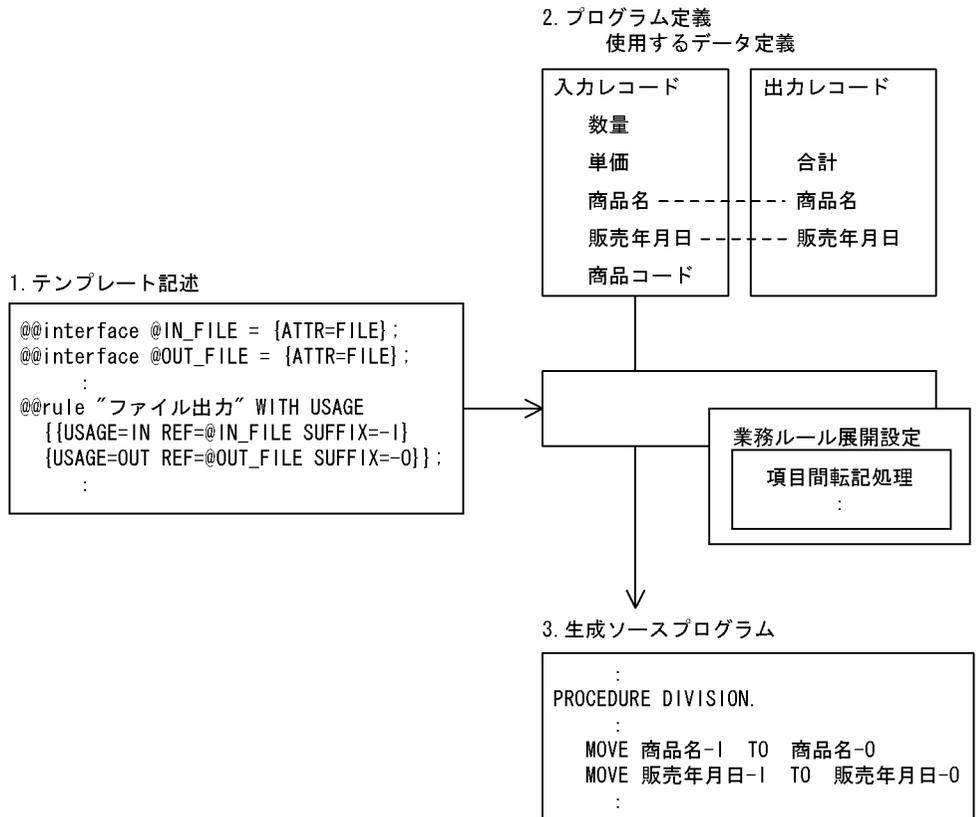
[ルールスクリプト]タブの情報を定義します。

```
MOVE @入力項目 TO @出力項目
```

(c) 同一項目用業務ルールの適用例

同一項目用業務ルール「項目間転記処理」が、SEWB+/CONSTRUCTION からアプリケーションに適用される例を示します。SEWB+/CONSTRUCTION からは、通常の業務

ルールと同様に同一項目用業務ルールを使用できます。



1. テンプレート記述

@@rule 文に、同一項目用業務ルールを含めた業務ルールの抽出条件を指定します。

なお、入力側と出力側のデータ項目のキーワードが置換される名称に、接頭語または接尾語を付加する指定が必要です。同一項目用業務ルールがソースプログラムに展開されたとき、転記元と転記先のデータ項目の名称が一意になるようにします。

2. プログラム定義

1. のテンプレートを選択し、アプリケーションプログラムの入力側と出力側で例に示したデータ定義を使用している場合、業務ルールの候補として「項目間転記処理」が表示されます。

3. 生成ソースプログラム

プログラム定義で同一項目用業務ルールを選択すると、同じ2項目間の転記処理をする MOVE 文が生成されます。

2.7 辞書登録の確認

クライアントのリポジトリブラウザでは、辞書登録時やメンテナンス時に必要に応じて、定義内容を検査したり、データ項目や業務ルールを検索したり、資源の相互関係を調査したりできます。ここでは、リポジトリブラウザを使った辞書の登録内容の確認について解説します。

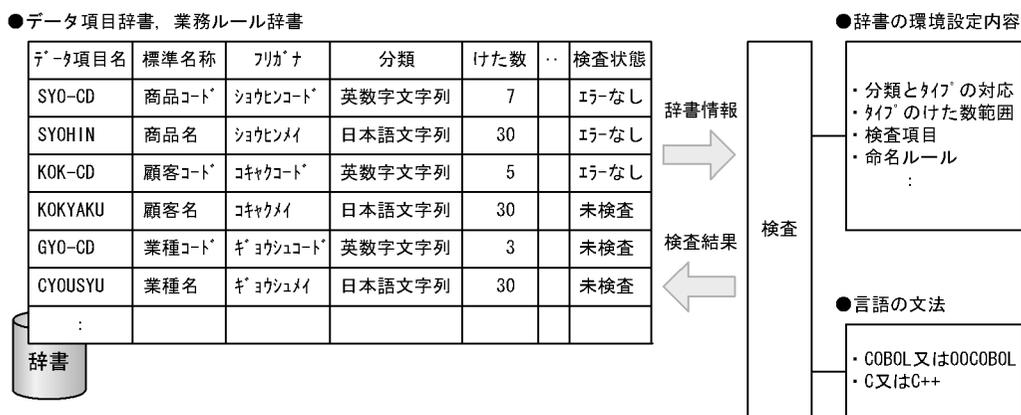
2.7.1 定義内容を検査する

(1) 検査の仕組み

リポジトリブラウザでは、データ項目や業務ルールの定義内容について、プロジェクトの基準やプログラミング言語の文法などに従って適切に定義されているかどうかを検査できます。

検査では、辞書の環境設定内容（環境構築ユーティリティで設定した、検査に適用する検査項目、命名ルール、分類とタイプの対応など）および言語の文法を基にして、データ項目または業務ルール別に定義内容の妥当性が診断されます。検査の仕組みを、図 2-15 に示します。

図 2-15 検査の仕組み



(a) 適用する検査項目とは

データ項目または業務ルールの検査内容別に、検査項目が用意されています。リポジトリブラウザでの検査に適用する検査項目を、あらかじめサーバの環境構築ユーティリティを使って設定しておきます。また、ユーザが作成した検査 DLL を登録して、独自の検査項目を追加することもできます。

プロジェクト内で定める辞書情報の基準に沿うよう、検査項目を設定してください。検査項目の詳細は、「3.1.1 環境を設定する」を参照してください。

データ項目の検査には、次の検査項目が用意されています。

データ項目名の一意性チェック
 名称と命名ルールのマッチングチェック
 けた数とタイプの整合性チェック
 タイプの組み合わせチェック
 初期値の記述チェック
 編集文字列の記述チェック
 構成項目の名前の重複チェック

業務ルールの検査項目には、次の検査項目が用意されています。

業務ルール名の一意性チェック
 名称と命名ルールのマッチングチェック
 ルールスクリプトのキーワード使用チェック

このうち、「データ項目名の一意性チェック」および「業務ルール名の一意性チェック」は、リポジトリブラウザでデータ項目または業務ルールを作成した時と、データ項目名または業務ルール名を変更した時に自動的に検査が実行されます。SEWB+/REPOSITORYでは、データ項目名および業務ルール名を、リポジトリ内で一意に定義しなければならないという制限はありません。したがって、同じ辞書フォルダや別の辞書フォルダに同一のデータ項目名または業務ルール名を定義することもできます（例えば、ドメインとなるデータ項目「年月日」を複数の辞書フォルダに登録した場合）。この検査項目でチェックされたデータ項目名または業務ルール名に対しては、プロジェクト独自の名称基準を確認したうえで、必要に応じて別の名称を付けてください。

その他の検査項目は、リポジトリブラウザのメニューの[検査]または[一括検査]を使用して検査します。

(b) 命名ルールとは

データ項目および業務ルールの名称の規則を命名ルールといいます。命名ルールは、データ項目および業務ルールの名称の標準化を図ることを目的としています。検査のために、プロジェクトの名称基準を、あらかじめ環境構築ユティリティを使って設定しておきます。命名ルールの条件の値は、辞書フォルダ単位で定義項目別に定義できます。

(c) 検査状態とは

個々のデータ項目および業務ルールには、それぞれ「検査状態」が設定されています。ここには、そのデータ項目または業務ルールがすでに検査されているかどうか、また、検査の結果、エラーがあったのかどうかを知るための情報が設定されます。これを「状態値」と呼びます。状態値の種類と意味は次のとおりです。

- エラーあり：定義内容にエラーが発見された
- エラーなし：定義内容は妥当である
- 未検査：定義内容の検査をしていない

検査をすると、状態値には「エラーあり」または「エラーなし」のどちらかが設定されます。また、データ項目の定義内容を変更したり、継承関係にある上位データ項目の定義内容が変更されたりすると、状態値は「未検査」に戻ります。

(2) 目的のデータ項目または業務ルールを指定して検査する

すでにリポジトリに格納されている辞書情報が検査の対象になります。目的に応じて、特定のデータ項目や業務ルールだけを指定して検査したり、辞書フォルダを指定してデータ項目と業務ルールを一括して検査したりできます。

ここでは、リポジトリブラウザから検査する方法について説明します。この方法のほかに、関連ブラウザウィンドウまたは検索ウィンドウの検索結果一覧から、データ項目や業務ルールを指定して検査することもできます。

特定のデータ項目や業務ルールだけを検査する

リポジトリブラウザウィンドウのツリービューまたはリストビューで、データ項目もしくは業務ルールを選択します。次に、[リポジトリ] - [検査] を選ぶと、検査のオプションを指定するダイアログが表示されます。ここで、検査対象とするデータ項目の検査状態を指定し [OK] ボタンを選ぶと検査が実行されます。

継承関係の下位データ項目も含めて検査する

リポジトリブラウザウィンドウのツリービューまたはリストビューで、データ項目を選択します。次に、[リポジトリ] - [一括検査] を選ぶと、検査のオプションを指定するダイアログが表示されます。ここで、検査対象とするデータ項目の検査状態を指定し [OK] ボタンを選ぶと検査が実行されます。選択したデータ項目と、継承関係の下位データ項目が検査されます。

辞書フォルダ内のすべてのデータ項目と業務ルールを検査する

リポジトリブラウザウィンドウのツリービューまたはリストビューで、辞書フォルダを一つ選択します。次に、[リポジトリ] - [一括検査] を選ぶと、検査のオプションを指定するダイアログが表示されます。ここで、検査対象とするデータ項目の検査状態を指定し [OK] ボタンを選ぶと検査が実行されます。

(3) 検査結果を確認し、定義内容を修正する

定義内容にエラーがあった場合は、その内容が検査結果ダイアログに一覧表示されます。検査結果を確認して、必要に応じて修正を加えてください。この検査結果の一覧から、エラーのあったデータ項目または業務ルールを選ぶと、編集のためのダイアログを開くこともできます。

2.7.2 目的のデータ項目や業務ルールを検索する

辞書に登録されているデータ項目や業務ルールの検索には、リポジトリブラウザで [リポジトリ] - [検索] を選ぶと表示される [検索] ダイアログを使います。このダイアログでは、大きく分けて次の 5 種類の条件を指定して、条件に一致するデータ項目や業務ルールを検索できます。

- 名前：データ項目の検索ではデータ項目名または標準名称を、業務ルールの検索では

- 業務ルール名または標準名称を指定できます。
- 更新日付：データ項目または業務ルールが作成もしくは更新された日付の範囲を指定できます。
- 所有者・更新者：ユーザを指定できます。
- コメント：コメントの定義内容を指定できます。
- フィールド：フィールドの定義内容を指定できます。

2.7.3 関連を基に，資源の相互関係を調査する

資源間に付けられている関連を基にして，データ項目同士の関係や，辞書の資源とリポジトリに登録されているドキュメントとの相互の依存関係などを調査できます。

(1) 変更波及を調査する

辞書の作成者が，データ項目や業務ルールの定義内容などを変更する場合には，その変更が，ほかのどの資源に波及するのかを把握しておく必要があります。リポジトリブラウザの関連ブラウザウィンドウを使用すれば，資源同士の相互関係を，資源間に付けられた関連を基に調査できます。

リポジトリブラウザで調査の起点とする資源のアイコンを選び，[リポジトリ] - [関連ブラウザ] - [すべての関連] (関連種別を指定したい場合は [他の関連]) を選択すると，関連ブラウザウィンドウが表示されます。

例えば，あるレコード定義を構成するデータ項目を変更するとき，そのレコード定義を調査の起点にすれば，SEWB+/CONSTRUCTION で作成されたどのドキュメントに波及するのかをツリー形式の関連図で確認できます。

関連ブラウザウィンドウのツールバーで [インパクトレポート] ボタンを選ぶと，表示された資源の関連図 (インパクトレポート) を印刷することもできます。

辞書の資源の調査に利用できる関連の例を次に示します。

- データ項目の結合の関連 (関連種別「結合」)
- データ項目と業務ルールの関連 (関連種別「ルール」)
- データ項目と SEWB+/RECORD DEFINER のレコード定義の関連 (関連種別「辞書参照」)
- 辞書のレコード定義 (最上位結合項目) とデータ定義の関連 (関連種別「辞書参照」)
- 業務ルールとプログラム定義の関連 (関連種別「辞書参照」)
- データ項目と，そのほかの資源に付けたユーザ関連

注

インパクトレポートの印刷機能を使用するには，クライアントに SEWB+/REPORT MANAGER をインストールしていることが前提となります。詳細は，マニュアル「SEWB+/REPORT MANAGER ドキュメント作成支援 使用の手引」を参照してください。

(2) データ項目の継承関係を調査する

データ項目の継承関係を調査したい場合は、リポジトリブラウザウィンドウを使用します。

(3) 業務ルールをブラウジングする

業務ルールのブラウジングには、「業務ルールブラウザ」を使用します。データ項目を調査の起点にして、そのデータ項目を使用している業務ルールを調査できます。また、業務ルールを起点にして、その業務ルールで使用しているデータ項目を一覧表示することもできます。

2.7.4 データ項目，業務ルールを印刷する

クライアントに SEWB+/REPORT MANAGER をインストールしている場合には、リポジトリブラウザから、データ項目、業務ルールの定義内容または一覧を印刷できます。印刷形式については、マニュアル「SEWB+/REPORT MANAGER ドキュメント作成支援 使用の手引」を参照してください。

(1) データ項目または業務ルールの定義内容を印刷する

各データ項目または業務ルールの定義内容を印刷します。

まず、リポジトリブラウザウィンドウから、定義内容を印刷したいデータ項目または業務ルールを選択します（複数を選択することもできます）。次に、[リポジトリ] - [SEWB+/REPORT MANAGER から印刷] を選ぶと、SEWB+/REPORT MANAGER が起動されます。そして、SEWB+/REPORT MANAGER のウィンドウから定義内容の印刷を実行します。

(2) データ項目または業務ルールの一覧を印刷する

データ項目辞書または業務ルール辞書が管理する資源の一覧を印刷します。

まず、リポジトリブラウザウィンドウから、データ項目辞書のアイコン（または業務ルール辞書のアイコン）を選択します。次に、[リポジトリ] - [SEWB+/REPORT MANAGER から印刷] を選ぶと、SEWB+/REPORT MANAGER が起動されます。そして、SEWB+/REPORT MANAGER のウィンドウから一覧の印刷を実行します。

なお、一覧の印刷では、リポジトリブラウザウィンドウのリストビューにそのとき表示されている資源が印刷の対象になります。

3

辞書の環境設定と運用

SEWB+/REPOSITORY では、プロジェクトやシステム開発環境に合わせて辞書の環境を設定したり、辞書を効率良く管理・運用するためのユティリティや機能を提供しています。この章では、辞書の環境設定と、管理・運用の作業について説明します。

3.1 辞書の準備

3.2 辞書情報のセキュリティ

3.3 辞書の配布，複製

3.1 辞書の準備

辞書の運用を開始するために、プロジェクトや開発体制に合わせた環境を準備します。ここでは、環境構築ユティリティを使用した辞書の準備について概要を説明します。この機能は、サーバでシステム管理者だけが利用できます。環境構築ユティリティの詳細な操作方法については、ヘルプを参照してください。なお、辞書の準備を始める前に、あらかじめサーバの環境設定と SEWB+/REPOSITORY の初期設定が必要です。詳細については、マニュアル「SEWB+/REPOSITORY 運用ガイド」を参照してください。また、サーバの環境設定を変更した場合には、複数のリポジトリに配布する必要があります。

3.1.1 環境を設定する

ここでは、辞書の環境設定の内容について説明します。なお、辞書の環境は、運用を開始したあとでも必要に応じて変更できます。この場合は、環境構築ユティリティで環境を変更したあと、クライアントのリポジトリブラウザを再起動してください。リポジトリに作成されている、すべての辞書フォルダの環境が変更されます。

(1) 国語区分を設定する

プロジェクトで使用する各国語の名称を、国語区分に設定します。国語区分の名称は、4種類まで設定できます。国語区分「日本語」および「英語」が標準に設定されています。

新しい国語区分の名称を付けるときは、「未設定 n」の名称を変更します。

なお、標準に設定されている国語区分の設定順序は、SEWB+/REPOSITORY で参照されます。変更しないよう留意してください。

(2) 言語区分を設定する

アプリケーションの開発に使用するプログラミング言語の名称を、言語区分に設定します。言語区分の名称は、10種類まで設定できます。次に示す言語区分が標準に設定されています。

- COBOL 又は OOCOBOL...COBOL 言語と OOCOBOL 共通の言語区分
- C 又は C++...C 言語と C++ 共通の言語区分
- SQL...SQL 言語の言語区分
- IDL(CORBA)...CORBA の IDL の言語区分
- 帳票定義...EUR Professional Edition で作成するレポートのフィールド定義情報を設定するための言語区分
- Java...Java の言語区分

新しい言語区分の名称を付けるときは、「言語 n」の名称を変更します。また、SQL 言語を「HiRDB の SQL 言語の言語区分」にするには、標準に設定されている分類区分とタイプの対応をカスタマイズする必要があります。カスタマイズの詳細については、「表

2-8 分類区分とタイプの対応のカスタマイズ例（言語区分 SQL のタイプを HiRDB として使用する場合）」を参照してください。

なお、標準に設定されている言語区分の設定順序は、SEWB+/REPOSITORY や SEWB+/CONSTRUCTION などの辞書を利用する SEWB+ ツールから参照されます。設定順序を変更しないよう留意してください。

（3）定義項目のフィールドの属性を設定する

（a）文字数の上限（フィールド長）の設定

リポジトリブラウザでの定義項目の文字数の上限（テキストボックスのフィールド長）を変更します。変更できる定義項目、標準に設定されている値および設定範囲については、「2.4.1 データ項目に定義する情報」または「2.6.1 業務ルールに定義する情報」を参照してください。

（b）データ項目の継承モードの設定

継承関係を定義したとき、データ項目の各定義項目の情報を、下位のデータ項目にどのように継承するかを継承モードで設定します。この設定は、リポジトリブラウザでのデータ項目の操作、およびコマンドを使った CSV 形式ファイル（標準フォーマット）の入出力時に反映されます。

（4）フィールドの名称を設定する

データ項目および業務ルールの定義項目「フィールド」の名称を変更して、プロジェクト独自の名称を付けます。また、リポジトリブラウザに表示するフィールドを設定します。標準に設定されている値および設定範囲については、「2.4.1 データ項目に定義する情報」または「2.6.1 業務ルールに定義する情報」を参照してください。

（5）ユーザ定義のタイプを追加する

標準で用意されているタイプに加えて、プロジェクトに応じたユーザ定義のタイプを登録できます。例えば、言語区分「COBOL 又は OOCOBOL」で使用するタイプとして、「J 右寄せ英数字項目」を追加登録できます。

追加したユーザ定義のタイプについては、分類区分との対応づけやけた数の範囲など、タイプに関する設定をしてください。

ユーザ定義のタイプは、削除することもできます。ただし、ユーザ定義のタイプを削除するには、そのタイプが分類区分のデフォルトタイプではなく、さらに、リポジトリ内で使用されていない（データ項目の定義に使用されていない）ことが必要です。ユーザ定義のタイプを削除すると、そのタイプに関する分類区分の対応づけなどの設定も削除されます。

注

エクスポート・インポート機能を使用するような、複数の SEWB+/REPOSITORY サーバを使用する環境で言語別タイプを追加する場合には、エクスポート・イン

3. 辞書の環境設定と運用

ポート機能で言語別タイプ（データ項目，業務ルール辞書の環境情報）も配布してください。また，それぞれの SEWB+/REPOSITORY サーバで言語別タイプを追加するのではなく，言語別タイプを追加する SEWB+/REPOSITORY サーバを決めて管理するようにしてください。

(6) 分類区分とタイプの対応を変更する

プロジェクトに応じて，分類区分に対応するタイプを変更します。プロジェクトで使用するユーザ定義のタイプを追加したり，標準で設定されているタイプのうち使用しないものを削除したりします。また，タイプの初期値（デフォルトタイプ）を設定することもできます。

例えば，COBOLでアプリケーションを開発するプロジェクトで浮動小数点データ形式を扱わないとき，分類区分「整数データ」に対応するタイプから「E 外部浮動小数点項目」および「D 内部浮動小数点項目」を除きます。

標準に設定されている分類区分とタイプの対応については，「2.4.1 データ項目に定義する情報」を参照してください。

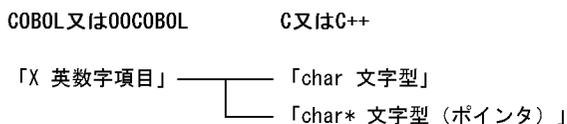
(7) タイプのけた数の範囲を変更する

検査でチェックされるけた数の範囲は，あらかじめ各プログラミング言語の仕様に基づいて設定されています。さらにプロジェクト固有の制限を設けたい場合には，けた数の上限値と下限値を設定して範囲を変更します。

標準に設定されているタイプに対応するけた数については，「2.4.1 データ項目に定義する情報」を参照してください。

(8) タイプの組み合わせを変更する

言語間でデータの形式を表すタイプの整合性を保つため，「COBOL 又は OOCOBOL」，「C 又は C++」，「SQL」，「IDL(CORBA)」，「帳票定義」，「Java」のタイプの組み合わせが標準に設定されています。例えば，「COBOL 又は OOCOBOL」のタイプ「X 英数字項目」に着目した場合，次の組み合わせが規則に従って設定されています。



規則

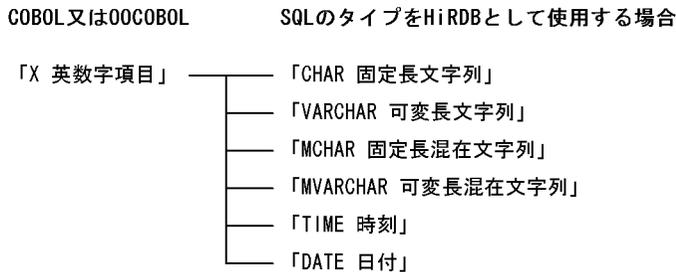
「COBOL 又は OOCOBOL」のタイプ「X 英数字項目」は分類区分「英数字文字列データ」，「英数字項目」，「日付データ」に対応しています。「COBOL 又は OOCOBOL」と「C 又は C++」間のデータ形式を表すタイプの整合性を保つために，「C 又は C++」の分類区分「英数字文字列データ」，「英数字項目」，「日付デー

タ」に対応するタイプが標準に設定されています。つまり、「C 又は C++」のタイプ「char 文字型」、「char* 文字型 (ポインタ)」が設定されています。分類区分とタイプの対応については「2.4.1(3)(a) 分類」を参照してください。

ほかの「言語区分」と「タイプの種類」に対する標準に設定されているタイプの組み合わせも、同様に、この規則に従っています。ただし、分類区分の「分類なし」は標準のタイプの組み合わせには設定されていません。

必要に応じて、整合性を取るタイプの組み合わせを変更してください。

また、言語区分 SQL のタイプを HiRDB として使用する場合、標準に設定されているタイプの組み合わせを変更する必要があります。例えば「COBOL 又は OOCOBOL」のタイプ「X 英数字項目」に着目した場合、次の組み合わせを設定します。



(9) タイプに対するレコード生成キーワードを編集する

レコード生成キーワードは、「COBOL 又は OOCOBOL」および「C 又は C++」の言語別のタイプごとに、データ項目の定義情報を、どのようにソースコードに生成するかを指定するものです。SEWB+/CONSTRUCTION でアプリケーション開発を始める前に、レコード生成キーワードの内容を確認し、必要であれば編集してください。ただし、この内容は、SEWB+/CONSTRUCTION および SEWB+/RECORD DEFINER でのすべてのプログラム生成およびレコード生成に有効になりますので、編集に際しては、あらかじめ十分な検討が必要です。

- レコード生成キーワードの内容
レコード生成キーワードは、プログラミング言語で規定されたデータの定義と、データ項目の定義情報を置換するための可変記号 (@ で始まる文字列) で記述されます。標準設定の内容を表 3-1 および表 3-2 に示します。また、可変記号の生成規則を表 3-3 に示します。

表 3-1 レコード生成キーワードの内容 (COBOL 又は OOCOBOL)

タイプ	レコード生成キーワード
X 英数字項目	@DATAITEM PIC X@LEN @OCCURS @VALUE
Z 数字編集項目	@DATAITEM PIC @EDITCHAR @OCCURS @VALUE

3. 辞書の環境設定と運用

タイプ	レコード生成キーワード
N 漢字項目	@DATAITEM PIC N@LEN @OCCURS @VALUE
9 符号なし外部 10 進項目	@DATAITEM PIC 9@ILEN@DLEN @OCCURS @VALUE
S 符号付き外部 10 進項目	@DATAITEM PIC S9@ILEN@DLEN @OCCURS @VALUE
U 符号なし内部 10 進項目	@DATAITEM PIC 9@ILEN@DLEN USAGE PACKED-DECIMAL @OCCURS @VALUE
P 符号付き内部 10 進項目	@DATAITEM PIC S9@ILEN@DLEN USAGE PACKED-DECIMAL @OCCURS @VALUE
BU 符号なし 2 進項目	@DATAITEM PIC 9@ILEN@DLEN USAGE BINALY @OCCURS @VALUE
B 符号付き 2 進項目	@DATAITEM PIC S9@ILEN@DLEN USAGE BINALY @OCCURS @VALUE
E 外部浮動小数点項目	@DATAITEM PIC @EDITCHAR @OCCURS
D 内部浮動小数点項目	@DATAITEM @COMP @OCCURS @VALUE
1 内部ブール項目	@DATAITEM PIC 1@LEN USAGE BIT @OCCURS @VALUE
8 外部ブール項目	@DATAITEM PIC 1@LEN USAGE DISPLAY @OCCURS @VALUE
T アドレスデータ項目	@DATAITEM USAGE ADDRESS @OCCURS @VALUE
フリー定義	@DATAITEM @FREE

表 3-2 レコード生成キーワードの内容 (C 又は C++)

タイプ	レコード生成キーワード
char 文字型	char @DATAITEM@OCCURS@LEN
char* 文字型 (ポインタ)	char* @DATAITEM@OCCURS
short 符号付き短整数型	short @DATAITEM@OCCURS
unsigned short 符号なし短整数型	unsigned short @DATAITEM@OCCURS
int 符号付き整数型	int @DATAITEM@OCCURS
unsigned int 符号なし整数型	unsigned int @DATAITEM@OCCURS
long 符号付き長整数型	long @DATAITEM@OCCURS
unsigned long 符号なし長整数型	unsigned long @DATAITEM@OCCURS
float 単精度浮動小数点型	float @DATAITEM@OCCURS
double 倍精度浮動小数点型	double @DATAITEM@OCCURS
long double 拡張精度浮動小数点型	long double @DATAITEM@OCCURS
フリー定義	@FREE @DATAITEM

表 3-3 可変記号の生成規則

可変記号	COBOL 又は OOCOBOL	C 又は C++
@DATAITEM	言語別の名前 ¹	言語別の名前 ¹
@LEN	けた数が n のとき「(n)」が生成される	けた数が n のとき「[n+1]」が生成される 分類が日本語文字列データの場合、1 文字を 2 バイトとして扱い「[n × 2+1]」となる
@PLEN	けた数が n のとき「(n)」が生成される	けた数が n のとき「[n]」が生成される 分類が日本語文字列データの場合、1 文字を 2 バイトとして扱い「[n × 2]」となる
@ILEN	けた数が m 小数部けた数が n のとき 「(m - n の値)」が生成される	-
@DLEN	小数部けた数が n のとき「V9(n)」が生成される	-
@OCCURS	反復回数が n のとき「OCCURS n」が生成される	反復回数が n のとき「[n]」が生成される
@VALUE	データ項目に初期値があるとき、 「VALUE 初期値」が生成される ²	-
@EDITCHAR	編集文字列	-
@COMP	タイプが D のとき、次のどちらかが生成される 「COMP-1」: けた数が 1 ~ 7 の場合 「COMP-2」: けた数が 8 ~ 16 の場合	-
@FREE	フリー定義の内容	フリー定義の内容

(凡例)

- : 指定できません (何も生成されません)

注 1

可変記号は必ず大文字で記述してください。

注 2

可変記号に対応するデータ項目の定義情報がない場合には、何も生成されません。

注 1

言語別の名前が定義されていない場合は、データ項目名と置換されます。

注 2

SEWB+/CONSTRUCTION のテンプレートの @@expand 文に @VALUE の指定がない場合には、生成されません。

日立製以外の COBOL 言語を使用する場合には

言語区分「COBOL 又は OOCOBOL」のレコード生成キーワードの内容は日立 COBOL の文法に従って設定されています。日立製以外の COBOL 言語を使用する場合は、タイプごとにデータを定義する文法に相違がないか確認し、違いがあればレコード生成キーワードを変更してください。例えば、PICTURE 句の日本語データを定義する文字が異なる場合、タイプ「N 漢字項目」のレコード生成キーワード

3. 辞書の環境設定と運用

「@DATAITEM PIC N@LEN @OCCURS」の「N」を目的の文字に変更します。

(10) 検査項目を定義する

データ項目および業務ルールの検査項目から、適用する検査項目を選びます。

(a) データ項目の検査項目

データ項目用に用意されているの各検査項目の内容を次に説明します。

- データ項目名の一意性チェック
データ項目名を重複して付けていないかどうかをチェックします。同じデータ項目名があると、リポジトリブラウザでのデータ項目の編集の操作が自動的に中断されます。標準では、リポジトリの中での一意性がチェックされます。リポジトリの辞書フォルダをグループに分けて指定し、その中での一意性をチェックする設定もできます。詳細については、「(13) データ項目名・業務ルール名の一意性チェック範囲を設定する」を参照してください。
- 名称と命名ルールのマッチングチェック
データ項目名、国語別の標準名称、フリガナおよび言語別の名前を、命名ルールに照らし合わせてチェックします。命名ルールの条件に従っていないと、エラーと判断されます。この検査項目を適用するには、あらかじめ命名ルールを定義しておく必要があります。命名ルールの定義については、「(11) 命名ルールを定義する」を参照してください。
- けた数とタイプの整合性チェック
けた数を、タイプに対応するけた数の範囲以内で定義しているかどうかをチェックします。範囲外のけた数を定義していると、エラーと判断されます。
- タイプの組み合わせチェック
データ項目には、言語共通の分類区分に対して、言語別にタイプを定義できます。この言語間のタイプの整合性をチェックします。組み合わせに従っていないと、エラーと判断されます。
- 初期値の記述チェック
標準提供の検査 DLL を使用し、「COBOL 又は OOCOBOL」および「C 又は C++」で言語別の初期値が各言語の文法に従って記述されているかどうかをチェックします。次に示すチェックの内容に従っていないと、エラーと判断されます。チェックの内容は分類区分に応じて異なります。

英数字文字列データ¹、時刻データ¹、日付データ¹

文字列の先頭と末尾を引用符で囲んでいるか

半角文字で記述しているか

記述した文字数が、けた数の定義内容を超えていないか

日本語文字列データ¹

日本語文字の先頭と末尾を引用符で囲んでいるか²

記述した文字数が、けた数の定義内容を超えていないか

整数データ¹

整数値を記述しているか

記述した文字数が、けた数の定義内容を超えていないか

正整数データ¹

正の整数値を記述しているか

記述した文字数が、けた数の定義内容を超えていないか

実数データ¹

実数値を記述しているか

記述した文字数が、けた数、小数部けた数の定義内容を超えていないか

注

SEWB+/REPOSITORY の辞書の検査では、「COBOL 又は OOCOBOL」ではアポストロフィ (') が、「C 又は C++」ではダブルクォーテーション (") が引用符とみなされます。

注 1

「COBOL 又は OOCOBOL」で記述した 16 進の定数および表意定数はチェックの対象になりません。また、浮動小数点項目のけた数もチェックの対象になりません。

注 2

「COBOL 又は OOCOBOL」の場合、日本語文字定数を表す分離符（例：'N' 日本語'）はエラーになりません。

- 編集文字列の記述チェック

標準提供の検査 DLL を使用し、「COBOL 又は OOCOBOL」のタイプに「Z 数字編集項目」または「E 外部浮動小数点項目」を定義したときに記述する編集文字列の定義内容について、次の項目をチェックします。

- 言語の文法に従って編集文字列を記述しているか
- 編集文字列の記述が、けた数の定義内容と一致しているか

- 構成項目の名前の重複チェック

結合項目だけを検査対象とします。

検査対象の結合項目を構成している構成項目の言語別詳細情報の名前が重複して付けられていないかどうかを言語区分ごとにチェックします。名前が重複して付けられていると、エラーと判断されます。構成項目が、さらに結合項目でもある場合、その結合項目の構成項目についても、名前が重複して付けられていないかどうかをチェックします。このように構成項目が結合項目を兼ねている間は、同様にチェックを繰り返していきます。

なお、言語別詳細情報の名前が定義されていない構成項目に限り、データ項目名を代わりにチェックします。

(b) 業務ルールの検査項目

業務ルール用に用意されている各検査項目の内容を、次に説明します。

- 業務ルール名の一意性チェック

業務ルール名を重複して付けていないかどうかをチェックします。同じ業務ルール名があると、リポジットブラウザでの業務ルールの編集の操作が自動的に中断されます。

3. 辞書の環境設定と運用

標準では、リポジトリの中での一意性がチェックされます。リポジトリの辞書フォルダをグループに分けて指定し、その中での一意性をチェックする設定もできます。詳細については、「(13) データ項目名・業務ルール名の一意性チェック範囲を設定する」を参照してください。

- 名称と命名ルールのマッチングチェック

業務ルール名、国語別の標準名称、フリガナを、命名ルールに照らし合わせてチェックします。命名ルールの条件に従っていないと、エラーと判断されます。この検査項目を適用するには、あらかじめ命名ルールを定義しておく必要があります。命名ルールの定義については、「(11) 命名ルールを定義する」を参照してください。

- ルールスクリプトのキーワード使用チェック

業務ルールに使用する各データ項目に定義したキーワードが、ルールスクリプトに記述されているかどうかをチェックします。記述されていない場合は、エラーと判断されます。

(c) 検査項目のカスタマイズ

標準で用意されている検査項目は、DLLとして登録されています。これに加えて、SEWB+/REPOSITORYでは、プロジェクト独自の基準を反映して検査DLLを作成、登録するためのインタフェースを提供しています。検査項目インタフェースと検査DLLの作成方法については、「付録D 検査項目インタフェース」を参照してください。

(11) 命名ルールを定義する

検査で適用する命名ルールの条件を選択して、辞書フォルダ単位で定義項目別に値を設定します。次の定義項目に対する命名ルールを定義できます。

- データ項目の名称...データ項目名、標準名称、フリガナ、言語別の名前
- 業務ルールの名称...業務ルール名、標準名称、フリガナ

例えば、辞書フォルダ「システム開発辞書」のデータ項目名、辞書フォルダ「社内向け辞書」の業務ルール名というように、辞書フォルダ単位で定義項目別に命名ルールを定義できます。プロジェクトの名称基準を基に命名ルールを定義することで、名称の標準化を図れます。

(a) 命名ルールの条件

命名ルールには、複数の条件を組み合わせて設定します。次に示す条件が標準に用意されています。

- 最大長 (バイト数)
名称の最大の長さを統一する値を設定します。例えば、特定の辞書フォルダのデータ項目名の長さについて、20 バイト以内という基準を設定したいときに利用します。
- プリフィックス
名称のプリフィックス (接頭語) を設定します。例えば、辞書フォルダ「顧客管理システム辞書」のデータ項目名のプリフィックスを「KOKYAKU-」で統一するという基準を設定したいときに利用します。
- カナ文字以外の使用

カナ文字以外の使用を制限するかどうかを設定します。例えば、定義項目「フリガナ」の定義内容をチェックする場合に使用します。

(b) 条件のカスタマイズ

標準で用意されている命名ルールの条件は、DLL として登録されています。これに加えて、SEWB+/REPOSITORY では、プロジェクト独自の名称基準を反映して条件 DLL を作成、登録するためのインタフェースを提供しています。命名ルールインタフェースと条件 DLL の作成方法については、「付録 C 命名ルールインタフェース」を参照してください。

(12) GUI 属性を設定する

GUI 属性は、APPGALLERY Enterprise を使ったアプリケーション開発で使用する部品（再利用性を高めるために機能単位に分割されたソフトウェア）のプロパティに関する設定です。APPGALLERY Enterprise でリポジトリの資源を利用したアプリケーション開発をする場合には、開発を始める前に、GUI 属性の内容を確認してください。開発の内容に応じて必要であれば、標準設定を変更したり、設定を追加します。

GUI 属性では、次の 2 種類の設定を管理しています。

- マッピング情報
- プロパティセット

APPGALLERY Enterprise では、部品に設定された動作や表示などに関する属性を「プロパティ」と呼びます。APPGALLERY Enterprise でのアプリケーション開発では、部品のプロパティに値を設定したり参照したりしながら、目的とする GUI のオブジェクトを作成していきます。

SEWB+/REPOSITORY で GUI 属性を一元管理し、アプリケーション開発に利用することで、GUI 作成時のプロパティ設定作業を軽減でき、複数の開発者によるチーム開発でも GUI の統一を図れます。

APPGALLERY Enterprise からマッピング情報とプロパティセットが利用される詳細については、マニュアル「APPGALLERY プログラマーズガイド」を参照してください。

(a) マッピング情報

- マッピング情報とは

APPGALLERY Enterprise では SEWB+/REPOSITORY と連携し、辞書のデータ項目の定義情報を部品のプロパティとして利用できます。マッピング情報は、データ項目の定義情報を部品のプロパティ値にどのように置き換えるかといった対応関係（マッピング）を定義するものです。

マッピング情報には、部品ごとに、データ項目の定義項目「データ項目名」、「標準名称」、「分類」、「けた数」、「小数部けた数」に対して、変換規則を設定します。それぞれの定義情報をプロパティに変換するかどうかを指定し（変換しない場合は「なし」を指定）、変換するプロパティのデータ型を指定します。各定義項目に指定できるプロ

3. 辞書の環境設定と運用

パティのデータ型を、表 3-4 に示します。

表 3-4 指定できるプロパティのデータ型

データ項目の定義項目	指定できるプロパティのデータ型
データ項目名	string, variant
標準名称 (1 ~ 4)	string, variant
分類 ¹	integer, long, string, variant, boolean, 列挙型 ²
けた数	integer, long, variant, boolean, 列挙型 ²
小数部けた数	integer, long, variant, boolean, 列挙型 ²

注 1

分類の場合は、データ型を指定し、さらに、分類区分ごとに変換する値を指定します。

注 2

列挙型とは、整数定数値の有限集合を指します。例えば、APPGALLERY Enterprise のテキストボックス部品の Type プロパティが該当します。

- マッピング情報の編集

SEWB+/REPOSITORY では、標準設定としてボタン部品とテキストボックス部品のマッピング情報を用意しています。これらの標準設定は、必要に応じて編集し、変更できます。また、開発内容に応じて、GUI 作成のためのほかの部品（メニューバー、ポップアップ部品など）に対するマッピング情報を追加できます。

(b) プロパティセット

- プロパティセットとは

APPGALLERY Enterprise での GUI 作成のために、部品ごとに標準とする色やフォントなどの設定値をまとめたものです。アプリケーション開発を始める前に、GUI を統一する対象ごとに、プロパティセットを準備しておきます。APPGALLERY Enterprise でのアプリケーション開発で、開発対象に応じたプロパティセットを選択しておく、GUI 作成で部品を配置した時点で自動的にプロパティ値が設定されます。

- プロパティセットの準備

GUI の統一を図る対象（システム、業務、開発環境など）を検討し、プロパティセットを登録します。複数のプロパティセットを登録できます。SEWB+/REPOSITORY では、標準設定として「サンプル（APPGALLERY の標準部品）」を用意していますので、設定の参考にしたたり、実際の業務に流用したりしてご利用ください。プロパティセットの設定例を、表 3-5 に示します。この例では、リポジトリブラウザで「標準プロパティセット」に「顧客管理システム用プロパティセット」を選択しておけば、GUI 作成でボタン、ラベルまたはテキスト部品を配置した時点で、自動的に次のプロパティ値が反映されます。

表 3-5 プロパティセットの設定例

プロパティ セット名	部品名	プロパティ名	プロパティ値
顧客管理 システム用 プロパティ セット	ボタン	BackColor	灰色
		Caption3D	3 凸型 3-D (立体) にする
		ForeColor	黒
		Font	MS ゴシック, 標準スタイル, 文字サイズ: 11 ポイント
	ラベル	BackColor	白
		Caption3D	0 3-D (立体) にしない
		ForeColor	黒
		Font	MS ゴシック, 標準スタイル, 文字サイズ: 8 ポイント
	テキスト	BackColor	青
		ForeColor	白
		Font	MS ゴシック, 標準スタイル, 文字サイズ: 8 ポイント

- APPGALLERY Enterprise からの部品情報の登録

APPGALLERY Enterprise 側から, 部品パレットの部品情報を SEWB+/REPOSITORY に転送し, プロパティセットとして登録することもできます。これによって, 市販ソフトウェアを利用した部品やユーザが独自に作成した部品についても, プロパティセットに部品のプロパティ値を設定できます。

操作は, APPGALLERY Enterprise から行います。転送方法については, マニュアル「APPGALLERY プログラマーズガイド」を参照してください。

注

APPGALLERY Enterprise のバージョン 04-00 以降で利用できる機能です。

(13) データ項目名・業務ルール名の一意性チェック範囲を設定する

データ項目名および業務ルール名の一意性チェック範囲の設定について説明します。この設定を有効にするには, データ項目の場合は検査項目「データ項目名の一意性チェック」を, 業務ルールの場合は検査項目「業務ルール名の一意性チェック」を選択しておく必要があります。

- チェック範囲の設定

リポジトリ内の辞書フォルダから, 一意性チェックを実行したい辞書フォルダを選択してグループ分けすることで, チェック範囲を設定します。データ項目と業務ルール別に, それぞれ最大 20 のチェック範囲を設定できます。

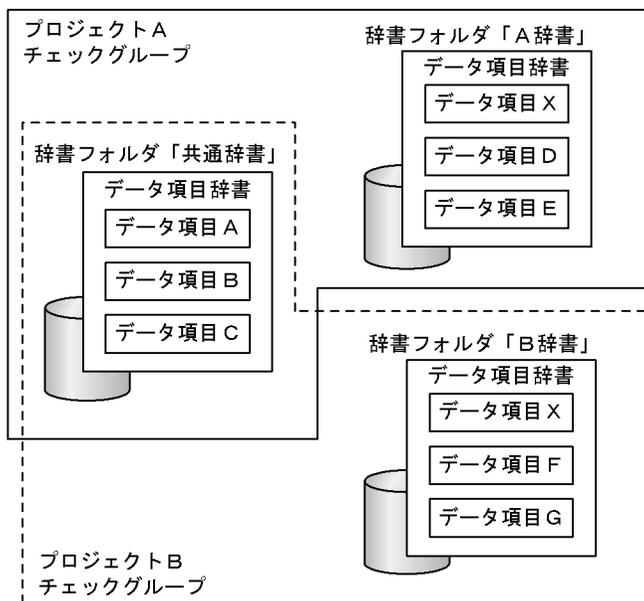
- 一意性チェック範囲の設定例

一意性チェック範囲の設定例を示します。一意性チェックの検査は, 範囲を設定した

3. 辞書の環境設定と運用

辞書フォルダ内で、データ項目や業務ルールを作成、移動または名称変更したときに実行されます。

辞書の構成例



データ項目の一意性チェック範囲の設定例

辞書の構成例に示した辞書フォルダに対する、データ項目の一意性チェック範囲の設定例を示します。

- 一意性チェック範囲名「プロジェクト A チェックグループ」
一意性チェックする辞書フォルダ：共通辞書，A 辞書
- 一意性チェック範囲名「プロジェクト B チェックグループ」
一意性チェックする辞書フォルダ：共通辞書，B 辞書

3.1.2 辞書フォルダを作成する

環境構築ユティリティを使用して、辞書フォルダをリポジトリに作成します。辞書フォルダにはリポジトリでユニークな名称を付け、運用状況に応じたアクセス権を設定してください。辞書フォルダを作成すると、辞書フォルダが管理するデータ項目辞書および業務ルール辞書も自動的に作成されます。

辞書フォルダを作成する場合には、その辞書フォルダでどのくらいの数の資源を管理するのかについても検討する必要があります。一つの辞書フォルダでは、最大 30,000 のデータ項目を管理できます。

(1) 辞書フォルダの定義項目

(a) 辞書フォルダ名

プロジェクトでの辞書の用途や管理内容に合わせて、辞書フォルダを識別する名称を付

けます。辞書フォルダにはリポジトリでユニークな名称を付けます。

辞書フォルダ名は、必ず定義しなければなりません。

- 文字数の上限（半角文字単位）は、1 ~ 92 で設定できます。標準値は 63 です。
- 英数字、カナ文字、日本語文字を使えます。ただし、「:」、「|」は使えません。また、文字列の途中に空白文字を含めてはなりません。

(b) コメント

その辞書フォルダがどのような辞書情報を管理しているのか、使用目的や内容、使用時の注意事項などを記述します。4 種類までの国語区分別に記述できます。

- 文字数の上限（半角文字単位）は、6500 です。変更はできません。
- 任意の文字を使えます。

(2) マルチ辞書を構築するには

リポジトリに複数の辞書フォルダを作成することで、マルチ辞書を構築できます。システム開発形態、プロジェクトの体制や運用などに合わせて、最適な辞書フォルダの構成を検討してください。構成を検討する場合には、次に説明する辞書の資源間の関連を考慮することをお勧めします。

(a) データ項目同士の関連

- データ項目の継承関係は、辞書フォルダ内にだけ定義できます。したがって、継承関係を定義する場合には、ドメインとするデータ項目が、同じ辞書フォルダ内に登録されていなければなりません。同一の定義情報を持つドメインでも、マルチ辞書を構成するそれぞれの辞書フォルダに登録する必要があります。
- データ項目の結合関係は、複数の辞書フォルダを渡って定義することもできます。例えば、辞書のレコード定義（最上位結合項目）を専用に管理する辞書フォルダ、レコード定義を構成する基本的なデータ項目を専用に管理する辞書フォルダというような分類ができます。

(b) 業務ルールとデータ項目の関連

業務ルールとデータ項目は、複数の辞書フォルダを渡って関連づけることもできます。例えば、あるプロジェクト専用のデータ項目を「辞書フォルダ A」で管理し、別のプロジェクト専用のデータ項目を「辞書フォルダ B」で管理する場合に、両方のプロジェクトから共用できる業務ルール専用の辞書フォルダを作成できます。

(3) 辞書フォルダを変更または削除するときの留意事項

(a) 辞書フォルダの環境を変更するとき

辞書フォルダの環境を変更（作成や削除）する前にクライアントのリポジトリブラウザを起動しているときには、変更後、再起動してください。

(b) 辞書フォルダを削除するとき

- 辞書フォルダを削除すると、その辞書フォルダが管理しているデータ項目辞書および

3. 辞書の環境設定と運用

業務ルール辞書も併せて削除されます。

- 複数の辞書フォルダに渡って結合項目を定義している場合、削除する辞書フォルダに構成項目となるデータ項目がある場合には、辞書フォルダを削除できません。
- 複数の辞書フォルダに渡って業務ルールを定義している場合、削除する辞書フォルダに適用条件「専用にする」が定義された業務ルールがある場合には、辞書フォルダを削除できません。

3.1.3 同一項目用業務ルールを辞書に準備する

SEWB+/REPOSITORY では、実際の業務に使用することを想定した同一項目用業務ルールを提供しています。この同一項目用業務ルールを利用するには、辞書の準備をする際に、あらかじめ辞書に登録しておく必要があります。ここでは、同一項目用業務ルールの登録方法について説明します。

標準提供される同一項目用業務ルールの内容については、「付録 E 標準提供の同一項目用業務ルール」を参照してください。

- 提供形態
CSV 形式ファイルで提供されています。提供ディレクトリとファイル名を次に示します。

ディレクトリ：サーバの SEWB+ 基本開発環境の組み込み先パス名

¥Repository¥Sample

ファイル名：moverule.csv

なお、組み込み先パス名は、デフォルトでは次のように設定されています。システムドライブとは、Windows 2000 Server, Windows Server 2003 または Windows Server 2003 x64 が組み込まれているドライブです。

- Windows 2000 Server または Windows Server 2003 の場合
システムドライブ：¥Program Files¥HITACHI¥Sewb
- Windows Server 2003 x64 の場合
システムドライブ：¥Program Files(x86)¥HITACHI¥Sewb
- 登録方法
CSV 形式ファイルを入力する機能を使い、提供ファイルを辞書に登録します。この作業は、サーバでシステム管理者が行ってください。
[スタート]メニューの [プログラム] または [すべてのプログラム] から [Sewb+] - [Repository] - [CSV 入力] を選びます。[CSV 入力] ダイアログで、次の指定をして実行すると、指定した辞書フォルダの業務ルール辞書に同一項目用業務ルールが登録されます。
 - 「登録先辞書フォルダ」で、同一項目用業務ルールに登録する辞書フォルダを指定する
 - 「CSV 入力ファイル名」に「サーバの SEWB+ 基本開発環境の組み込み先パス名 ¥Repository¥Sample¥moverule.csv」を指定する
 - 「入力する辞書種別」で「業務ルール」を選ぶ

3.1.4 タイプマッピングの規則を定義する

ODBC 入力や、ERwin/ERX と連携した CSV 入力を使って辞書登録をすると、元のタイプはマッピング規則に従って SEWB+/REPOSITORY の分類区分や言語別タイプ、けた数、小数部けた数に変換されます。

この変換に適用するタイプマッピング規則は、環境構築ユーティリティでカスタマイズできます。プロジェクトに応じて、変換元のタイプと、変換先の分類区分や言語別タイプなどの対応を定義できます。また、変換元のスケールや精度（長さ）に基づく条件を指定して、条件ごとの変換規則も定義できます。

なお、標準で用意されているタイプマッピング規則を適用する場合は、カスタマイズは必要ありません。標準のタイプマッピング規則の内容については、「4.2.2(2)(c) ODBC タイプマッピングの規則」および「付録 B.2(5) ERwin/ERX タイプマッピングの規則」を参照してください。

3.2 辞書情報のセキュリティ

ここでは、辞書情報を保護するためのアクセス権の設定と、排他制御の概要について説明します。

3.2.1 アクセス権の設定

辞書の作成者や利用者というようにユーザを分類して、アクセス権を設定すれば辞書情報に対する操作を制限できます。アクセス権の設定は、プロジェクトの大切な共有資源である辞書情報を保護し、整合性を保つことを目的としています。次に示す単位にアクセス権を設定できます。

- 辞書フォルダ
- データ項目
- 業務ルール

ユーザの分類方法と権限の種類については、マニュアル「SEWB+/REPOSITORY 運用ガイド」を参照してください。アクセス権の種類と許可される操作の詳細は、ヘルプを参照してください。

(1) 辞書フォルダに設定するアクセス権

辞書フォルダに設定できるアクセス権の種類と、初期値を説明します。辞書フォルダのアクセス権は、サーバの環境構築ユティリティで変更できます。

(a) アクセス権の種類

辞書フォルダのアクセス権は、「読み取り」、「書き込み」、「作成」および「削除」の4種類の権限の組み合わせで構成されます。辞書フォルダを作成するには、「書き込み」権限が必要です。「作成」権限では、辞書フォルダが管理する下位のデータ項目や業務ルールの作成ができます。

アクセス権の種類	権限の内容
アクセス権なし	アクセスできない
読み取り	読み取りの権限を持つ
変更	読み取り、作成の権限を持つ
フルコントロール	読み取り、作成、書き込みの権限を持つ
特殊なアクセス権	各権限をユーザの必要に応じて、任意に組み合わせたアクセス権

注

環境構築ユティリティ、リポジトリブラウザのどちらにも辞書フォルダの情報（名称やアクセス権）は表示されません。

(b) アクセス権の初期値

環境構築ユーティリティで辞書フォルダを作成するとき、あらかじめ設定されている初期値を次に示します。コマンドを使った一括登録で新しい辞書フォルダを作成する場合には、アクセス権の初期値が自動的に設定されます。

- 所有者：フルコントロール
- グループ：変更
- すべてのユーザ：読み取り

(2) データ項目に設定するアクセス権

データ項目に設定できるアクセス権の種類と、初期値を説明します。データ項目のアクセス権は、クライアントのリポジトリブラウザで変更できます。変更の際には、下位データ項目も含めて設定することもできます。

(a) アクセス権の種類

データ項目のアクセス権は、「読み取り」、「書き込み」および「削除」の3種類の権限の組み合わせで構成されます。

アクセス権の種類	権限の内容
アクセス権なし	アクセスできない
読み取り	読み取りの権限を持つ
変更	読み取り、書き込みの権限を持つ
フルコントロール	読み取り、書き込み、削除のすべての権限を持つ
特殊なアクセス権	各権限をユーザの必要に応じて、任意に組み合わせたアクセス権

(b) アクセス権の初期値

リポジトリブラウザでデータ項目を作成するとき、あらかじめ設定されている初期値を次に示します。コマンドを使った一括登録で新しいデータ項目を作成する場合には、アクセス権の初期値が自動的に設定されます。なお、CSV形式ファイル入力コマンドで新しいデータ項目を作成するとき、CSV形式ファイル上に直接アクセス権の値を記述している場合は、その値が設定されます。

- 所有者：フルコントロール
- グループ：変更
- すべてのユーザ：読み取り

(3) 業務ルールに設定するアクセス権

業務ルールに設定できるアクセス権の種類と、初期値を説明します。業務ルールのアクセス権は、クライアントのリポジトリブラウザで変更できます。

3. 辞書の環境設定と運用

(a) アクセス権の種類

業務ルールのアクセス権は、「読み取り」、「書き込み」および「削除」の3種類の権限の組み合わせで構成されます。

アクセス権の種類	権限の内容
アクセス権なし	アクセスできない
読み取り	読み取りの権限を持つ
変更	読み取り、書き込みの権限を持つ
フルコントロール	読み取り、書き込み、削除のすべての権限を持つ
特殊なアクセス権	各権限をユーザの必要に応じて、任意に組み合わせたアクセス権

(b) アクセス権の初期値

リポジトリブラウザで業務ルールを作成するとき、あらかじめ設定されている初期値を次に示します。コマンドを使った一括登録で新しい業務ルールを作成する場合には、アクセス権の初期値が自動的に設定されます。なお、CSV形式ファイル入力コマンドで新しい業務ルールを作成するとき、CSV形式ファイル上に直接アクセス権の値を記述している場合は、その値が設定されます。

- 所有者：フルコントロール
- グループ：変更
- すべてのユーザ：読み取り

3.2.2 排他制御

複数のユーザが同じ辞書情報に対して操作しているとき、自動的に排他制御が働き、辞書情報の二重更新が防止されます。排他制御の機能は、辞書情報の整合性を保つことを目的としています。

- 排他制御の単位
辞書フォルダ、データ項目および業務ルール単位の操作に排他制御が働きます。
- データ項目の階層単位の排他制御
データ項目は継承関係によって階層化されるため、上位データ項目の定義情報を変更すると、下位データ項目の定義情報にも波及します。そこで、データ項目では、階層単位の操作についても排他制御が働きます。操作しているデータ項目の変更がほかに波及する場合や、ほかのデータ項目変更がそのデータ項目に波及する場合には、メッセージが表示がされるため、留意してください。

3.3 辞書の配布，複製

分散された複数のサーバ（リポジトリ）間で辞書を運用するには、エクスポート・インポートユティリティを使用します。この機能は、サーバでシステム管理者だけが利用できます。

リポジトリの資源をファイル形式で外部媒体に取り出すことをエクスポートといいます。また、取り出したエクスポートファイルをリポジトリに取り込むことをインポートといいます。

ここでは辞書を中心に説明していますが、エクスポート・インポートはシステム管理者が、分散環境、ドキュメントと辞書、相互の資源に付けられた関連を合わせて検討したうえで運用していきます。エクスポート・インポートユティリティの全体像と運用例については、マニュアル「SEWB+/REPOSITORY 運用ガイド」を参照してください。また、詳しい操作方法については、ヘルプを参照してください。

3.3.1 複数のリポジトリで辞書を運用する

(1) 運用の目的に応じたエクスポート・インポートの利用

エクスポート・インポートは、次の2通りの運用が想定されています。この考え方は、辞書とドキュメント共通のものです。

(a) リポジトリの資源の配布

ほかの開発環境にオリジナルの資源を配布する運用です。辞書フォルダや資源同士の関連などを、すべてオリジナルと同じ状態で配布先のリポジトリに再現することを目的とします。

例えば、複数のリポジトリで同じ辞書を使って開発を進めたいとき、まず初めに、オリジナルの辞書を管理するリポジトリで、辞書登録をします。次に、完成した辞書をエクスポートし、配布先のリポジトリへインポートします。これによって、複数の開発環境や拠点へ速やかに辞書の環境を構築できます。さらに配布を繰り返す運用をしたい場合には、一度全体をエクスポートすれば、2回目からは前回にエクスポートした時点から追加または変更のあった情報だけをエクスポートするといったこともでき、作業の効率化を図れます。

また、システムの開発やコンポーネント機能の開発が終了したときに、その辞書をリポジトリから外部媒体に取り出して管理したい場合にも利用できます。辞書をエクスポートでファイルに取り出し、あとでシステムの保守や拡張が必要になったときリポジトリにインポートで取り込めば、以前と同じ環境に戻せます。

(b) リポジトリの資源の複製

オリジナルの資源を基に、別の開発環境を構築するときの運用です。複製元（エクスポート元）のリポジトリの資源と複製先（インポート先）のリポジトリの資源は、運用の異なる別の資源として扱います。

3. 辞書の環境設定と運用

例えば、新規システムの辞書を構築するとき、データ分析をした結果、既存システムの辞書を流用できる場合には、新規システムの開発環境に辞書を複製して効率の良い辞書登録ができます。

配布、複製のどちらの運用でも、インポート先に新規に追加する資源に対してエクスポート元とは別のアクセス権を設定できます。例えば、辞書を配布する際に、インポート先の辞書の資源に対して変更操作を制限するアクセス権を設定します。このようにして、オリジナル辞書と配布した辞書との整合性を守るといような運用ができ、辞書の管理が容易になります。

(2) エクスポート・インポートの流れ

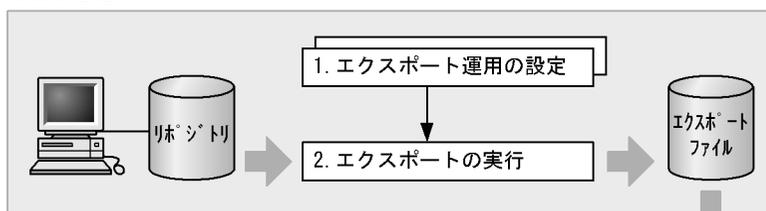
エクスポート・インポートをするには、まず複数の運用パターンを分散環境や運用状況に応じて設定し、用意しておきます。これを、エクスポートに対してはエクスポート運用、インポートに対してはインポート運用といいます。エクスポート・インポート運用を作成することで、辞書の配布や複製を定型化できます。また、一度運用を作成しておけば、ユティリティのウィンドウから実行するほか、コマンドで実行することもできるため作業の効率化を図れます。

エクスポート・インポートの実行時にはログ情報を取得できます。実行日時、運用名、エクスポートファイル名、対象となった資源の一覧情報などのログが出力されます。辞書の管理者は、分散されているサーバ間の辞書情報が相互に整合性があるかどうか、適切なタイミングでエクスポート・インポート作業が行われたかどうかを把握していなければなりません。ログ情報を取得して、辞書の管理に役立てることをお勧めします。

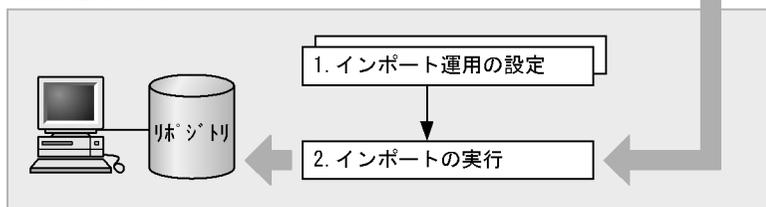
エクスポート・インポートユティリティを使ったエクスポート・インポートの流れを図3-1に示し、おおまかな手順を説明します。

図 3-1 エクスポート・インポートの流れ

●資源を提供するサーバ



●資源を受け取るサーバ



(a) エクスポートの手順

1. エクスポート運用を設定する

目的に応じて運用名を決め、エクスポートする資源などを設定します。辞書については、エクスポートする辞書フォルダ、どの資源や関連情報をエクスポートするか、また、データ項目や業務ルールの絞り込み条件などを設定します。

2. エクスポートを実行する

運用名を指定してエクスポートを実行します。エクスポート運用の設定内容に従って、リポジトリの資源がエクスポートファイルに取り出されます。このあと、インポートに備えて、エクスポートファイルを資源の受け取り側のサーバに転送するか、またはそのサーバから参照できるようにします。

(b) インポートの手順

1. インポート運用を設定する

目的に応じて運用名を決め、インポートするエクスポートファイル、インポート方法、インポート先の辞書フォルダ、追加される資源に与えるアクセス権などを設定します。

2. インポートを実行する

運用名を指定してインポートを実行します。インポート運用の設定内容に従って、エクスポートファイルの資源がリポジトリに取り込まれます。

(3) 効率良く運用するには

システム開発資源のすべてを定期的にバックアップするような運用をしたい場合は、Windows 2000 Server、Windows Server 2003 または Windows Server 2003 x64 のファイルシステムに用意されているバックアップ機能を使用します。環境構築ユティリティを使用すれば、バックアップのためのコマンドファイルを生成できます。詳細は、マニュアル「SEWB+/REPOSITORY 運用ガイド」を参照してください。

また、辞書の配布に際して毎回、全体をエクスポートすると、処理時間がかかり、作業効率下がります。2 回目からは、追加または変更のあった情報だけをエクスポート・インポートして作業の効率化を図ることをお勧めします。

(4) 辞書をエクスポート・インポートするときの留意事項

(a) 辞書をインポートできるサーバ

辞書に関する情報をインポートできるサーバを次に示します。

- 自サーバ
- 自サーバと異なるノード ID を持つ、ほかのサーバ

この条件に合わないサーバへリポジトリの資源をインポートしたいときには、次のどちらかの方法でインポートしてください。ただし、この作業をした場合、「インポート方法 1」では、配布先のドキュメントとデータ項目および業務ルールの関連を復元できません。また、「インポート方法 2」では、配布元のドキュメントとデータ項目および業務ルールの関連をインポートできません。

3. 辞書の環境設定と運用

注

自サーバの資源をエクスポートしたあと、自サーバのデータベースを初期化すると、エクスポートした資源を配布の目的でインポートすることはできません。この場合は、「インポート方法 2」の 2. を行ってください。

<インポート方法 1>

1. 配布先のリポジトリの資源をエクスポートする。
2. 配布先のサーバを初期化する。
データベースの環境を設定するとき、必ずほかのサーバと異なるノード ID を設定してください。ノード ID の設定については、マニュアル「Groupmax Object Server Version 6 システム管理者ガイド」を参照してください。サーバの初期化については、マニュアル「SEWB+ /REPOSITORY 運用ガイド」を参照してください。
3. 1. の資源を、元のサーバに復元する。
このときの辞書のインポート運用の設定では「指定した辞書フォルダへ複製する」を選択し、ドキュメントのインポート運用の設定では「ドキュメントとデータ項目間の関連をインポートしない」および「ドキュメントと業務ルール間の関連をインポートしない」を選択してください。

注

自サーバの資源をエクスポートしたあと、自サーバのデータベースを初期化すると、エクスポートした資源を配布の目的でインポートすることはできません。必ず複製の目的でインポートしてください。

4. 配布元から配布先へエクスポート・インポートする。

<インポート方法 2>

1. 配布元のリポジトリの資源をエクスポートする。
2. 1. をインポートする。
このときの辞書のインポート運用の設定では「指定した辞書フォルダへ複製する」を選択し、ドキュメントのインポート運用の設定では「ドキュメントとデータ項目間の関連をインポートしない」および「ドキュメントと業務ルール間の関連をインポートしない」を選択してください。

(b) 辞書フォルダをサーバ間で相互に配布する場合

辞書フォルダをサーバ間で相互に配布する場合は、エクスポート運用の「他の辞書フォルダのデータ項目を一緒にエクスポートする」の設定を解除してください。標準の設定では、結合項目をエクスポートしたときに構成項目もエクスポートされ、資源の管理元となっている辞書フォルダ内のデータ項目が上書きされてしまう場合があります。「他の辞書フォルダのデータ項目を一緒にエクスポートする」の設定を解除することでこれを回避できます。

ただし、この設定では、構成項目となるデータ項目がインポート先に存在しなかった場合はエラーとなり、インポートが中断されます。

3.3.2 辞書のエクスポート

(1) エクスポート運用の設定

辞書に関する設定を中心に、エクスポート運用の設定内容を説明します。

次に示すエクスポート方法は、エクスポート運用を実行するときに指定します。

- 全体エクスポート：エクスポートの度に、設定した範囲内のすべての資源がエクスポートの対象になります。
- 増分エクスポート：前回のエクスポート以降に追加または変更のあった情報だけがエクスポートの対象になります。ただし、環境構築情報は変更の有無に関係なく対象になります。

運用名、エクスポートファイル

辞書およびドキュメント共通の設定内容です。目的に応じて固有の運用名を定義し、資源を取り出すエクスポートファイルを指定します。

エクスポートする辞書フォルダ、対象資源

- エクスポートする辞書フォルダを指定します。マルチ辞書を構成している複数の辞書フォルダを指定できます。この場合、エクスポートする資源や絞り込み条件の指定は共通になります。したがって、辞書フォルダごとに別の設定が必要なときには、エクスポート運用を分けてください。
- 辞書のどの資源を対象にするかを選択します。次の3通りから対象資源の範囲を選択できます。
 1. 辞書フォルダのすべての資源
 2. 条件に一致するデータ項目とそのデータ項目に関連する業務ルール
 3. 条件に一致する業務ルールとその業務ルールに関連するデータ項目

データ項目の絞り込み条件

対象資源に2.を選んだ場合、エクスポートするデータ項目を条件を指定して絞り込みます。必要に応じ、絞り込み条件（名称の文字列、国語区分、言語区分など）を指定してください。さらに、絞り込み条件に一致したデータ項目を継承している下位のデータ項目もエクスポートするかどうかを指定します。

業務ルールの絞り込み条件

対象資源に3.を選んだ場合、エクスポートする業務ルールを条件を指定して絞り込みます。必要に応じ、絞り込み条件（名称の文字列、国語区分など）を指定してください。

エクスポートするドキュメント

ルートドキュメントフォルダまたはシステムバージョンで、エクスポートするドキュメントを指定します。ドキュメントと辞書を同時にエクスポートする場合には、必要に応じて、ドキュメントとデータ項目、またはドキュメントと業務ルールの関連のエクスポートを指定します。

3. 辞書の環境設定と運用

ログ情報の取得方法

メッセージログおよび実行結果リストの出力について指定します。

(2) 辞書と関連情報のエクスポート規則

辞書と辞書の関連情報がどのようにエクスポートされるのかを説明します。

(a) データ項目のエクスポート

- データ項目とデータ項目に関連する業務ルール

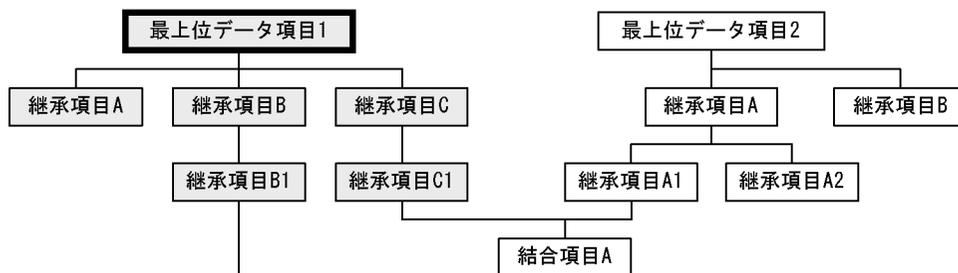
対象資源に、「条件に一致するデータ項目とそのデータ項目に関連する業務ルール」を指定した場合は、エクスポート対象のデータ項目だけでなく、そのデータ項目に関連する業務ルールもエクスポートされます。ただし、業務ルールに定義したデータ項目の適用形態が「専用にする」の場合には、その業務ルールに関連するデータ項目がすべてエクスポートの対象になっていることが前提となります。エクスポート運用に設定していない辞書フォルダの業務ルールでも、エクスポート対象のデータ項目と関連があれば、エクスポートされます。

- 継承関係の下位データ項目

絞り込み条件に下位データ項目のエクスポートを指定した場合、エクスポートの対象となったデータ項目を継承するすべての下位データ項目がエクスポートされます。この場合のエクスポート範囲の例を、図 3-2 に示します。

図 3-2 下位データ項目を指定した場合のエクスポート範囲の例

●データ項目辞書



●業務ルール辞書

業務ルール1

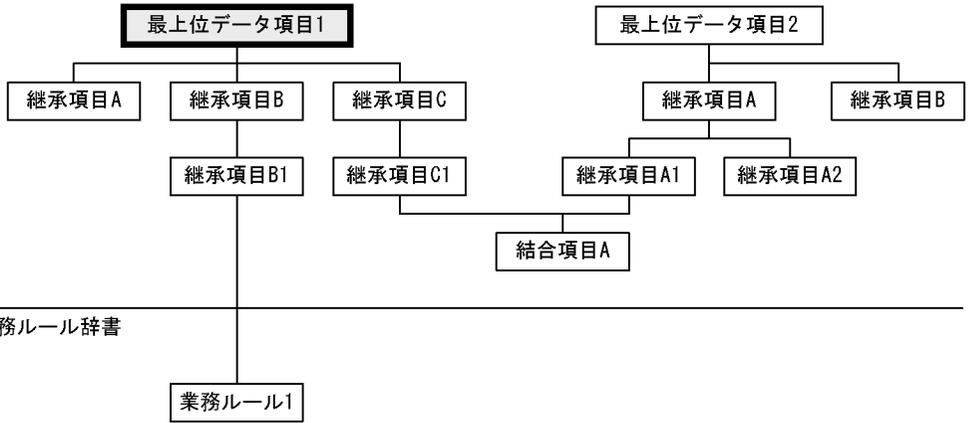
(凡例)

- : 絞り込み条件に一致するデータ項目
- : エクスポート範囲
- : 関連

同様の関連が定義された辞書で下位データ項目のエクスポートを指定しない場合の例を、図 3-3 に示します。

図 3-3 下位データ項目を指定しない場合のエクスポート範囲の例

●データ項目辞書



●業務ルール辞書

(凡例)

- : 絞り込み条件に一致するデータ項目
- : エクスポート範囲
- : 関連

- 結合関係、継承関係にあるデータ項目
エクスポート対象のデータ項目が、結合項目または継承関係の下位データ項目の場合、次に示すデータ項目もエクスポートされます。

エクスポート対象の結合項目の構成項目

結合関係の最下位にある構成項目までが対象になります。この構成項目のうち、エクスポート運用に設定していない辞書フォルダにあるものについてはエクスポートの対象にするかどうかを指定できます。

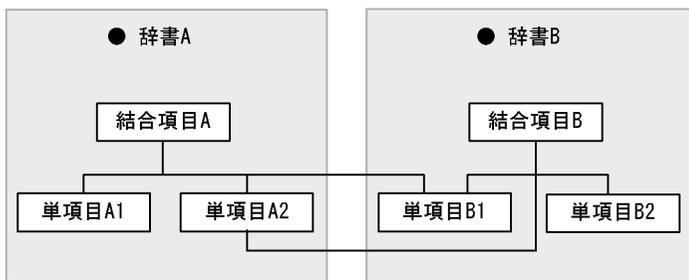
注

エクスポート運用で、「辞書」タブの「関連のある他の辞書フォルダの資源をエクスポートする」を指定しない場合、そのエクスポート運用に指定した複数の辞書フォルダ間で相互に関連があるときはエクスポートは実行できません。そのような場合には、「関連のある他の辞書フォルダの資源をエクスポートする」を指定してください。

例を次に示します。

3. 辞書の環境設定と運用

図 3-4 辞書フォルダ間に相互関連がある場合



(凡例)

—— : 関連

この例では、辞書 A の資源である結合項目 A は、辞書 B の資源である単項目 B1 を構成項目にしています。また、辞書 B の資源である結合項目 B は、辞書 A の資源である単項目 A2 を構成項目にしています。この場合、「辞書」タブの「関連のある他の辞書フォルダの資源をエクスポートする」を指定しなければ、エクスポートは実行できません。

エクスポート対象のデータ項目の上位データ項目

継承関係にある最上位データ項目までが対象になります。

- 関連情報

エクスポート運用の設定によって対象になったデータ項目の範囲内に付けられた次の関連情報がエクスポートされます。

- データ項目間の継承の関連、結合の関連
- データ項目と業務ルールの関連
- データ項目とドキュメントの関連

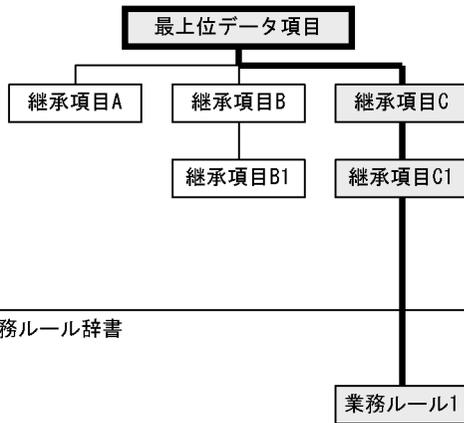
注

辞書とドキュメントを同時にエクスポートするとき、エクスポート運用にこの関連情報を指定していることが前提になります。

関連情報のエクスポートの例を、図 3-5 に示します。

図 3-5 関連情報のエクスポートの例

●データ項目辞書



●業務ルール辞書

(凡例)

- : エクスポート対象
- : 関連
- (thick) : エクスポート対象の関連情報

(b) 業務ルールのエクスポート

- 業務ルールと業務ルールに関連するデータ項目
対象資源に、「条件に一致する業務ルールとその業務ルールに関連するデータ項目」を指定した場合は、エクスポート対象の業務ルールだけでなく、その業務ルールに関連するデータ項目もエクスポートされます。
- 関連情報
エクスポート運用の設定によって対象になった業務ルールの範囲内に付けられた次の関連情報がエクスポートされます。
業務ルールとデータ項目の関連
業務ルールとドキュメントの関連

(c) 環境構築情報のエクスポート

辞書に関する環境構築情報として、次の情報をエクスポートできます。環境構築情報のエクスポートについては、マニュアル「SEWB+/REPOSITORY 運用ガイド」を参照してください。

- 辞書フォルダ
辞書資源をエクスポート対象にした場合にエクスポートされます。
- データ項目情報
データ項目や業務ルールに関する各種の環境構築情報がエクスポートされます。ただし、命名ルール情報や検査項目情報は対象になりません。
- タイプマッピング情報

3. 辞書の環境設定と運用

ODBC 入力や、ERwin/ERX と連携した CSV 入力をするときのタイプの変換規則がエクスポートされます。

3.3.3 辞書のインポート

(1) インポート運用の設定

辞書に関する設定を中心に、インポート運用の設定内容を説明します。

運用名、インポートするエクスポートファイル

辞書およびドキュメント共通の設定内容です。目的に応じて固有の運用名を定義し、インポートで取り込むエクスポートファイルを指定します。

辞書のインポート方法、インポート先の辞書フォルダ

- 目的に応じたインポート方法を選択します。

挿入インポート

インポート先にはない情報をエクスポートファイルから追加します。インポート先と同じ情報があっても、置換しません。

上書きインポート

インポート先にはない情報をエクスポートファイルから追加します。また、インポート先と同じ情報があれば、エクスポートファイルの情報と置換します。

- 運用の目的に応じてインポート先の辞書フォルダを指定します。

辞書を配布する

「すべてエクスポート元と同じ辞書フォルダへ配布する」を選びます。

辞書を複製する

「指定した辞書フォルダに複製する」を選び、エクスポート元の辞書フォルダ単位にインポート先の辞書フォルダを指定します。このとき、インポート先の辞書フォルダを指定しないと情報はインポートされないことを留意してください。

- インポートで新規に追加される辞書フォルダに設定するアクセス権を指定します。

データ項目のアクセス権

インポートで新規に追加されるデータ項目に、エクスポート元と同じアクセス権の設定を指定するか、または新たに設定するアクセス権を指定します。

業務ルールのアクセス権

インポートで新規に追加される業務ルールに、エクスポート元と同じアクセス権の設定を指定するか、または新たに設定するアクセス権を指定します。

ドキュメントのインポート方法、インポート先、対象資源など

辞書と同様に、インポート方法などを指定します。辞書とドキュメントを同時にインポートする場合には、必要に応じて、ドキュメントとデータ項目間、またはドキュメントと業務ルール間の関連のインポートを指定してください。

ログ情報の取得方法

メッセージログおよび実行結果リストの出力について指定します。

(2) 辞書と関連情報のインポート規則

辞書と辞書の関連情報がどのようにインポートされるのかを説明します。

(a) インポートで回復されるエクスポート元の情報

エクスポート方法および辞書のインポート方法によって異なります。データ項目、業務ルール、データ項目間の関連、およびデータ項目と業務ルールの関連情報の回復方法を次に示します。

- 全体エクスポートした情報
挿入インポートでは、設定範囲内のすべての情報を基に追加します。
上書きインポートでは、設定範囲内のすべての情報を基に追加および置換します。
- 増分エクスポートした情報
挿入インポートでは、設定範囲内から増分として抽出された情報を基に追加します。
上書きインポートでは、設定範囲内から増分として抽出された情報を基に追加および置換します。

注 1

追加とは、インポート先に同じ情報がない場合に追加することです。また、置換とは、インポート先に同じ情報があればその情報を置換することです。

注 2

データ項目および業務ルールの検査状態は、インポートしてもエクスポート元の情報は回復されません。

(b) インポート先の辞書フォルダ

インポート先のリポジトリに配布または複製の目的に対応した辞書フォルダがない場合には、次の規則で自動的に作成されます。

配布を目的にエクスポート元と同じ辞書フォルダへインポートする場合には、エクスポート元と同じ名称の辞書フォルダが自動的に作成されます。また、複製を目的に辞書フォルダを指定してインポートする場合には、インポート運用に設定した名称の辞書フォルダが自動的に作成されます。どちらの場合も作成された辞書フォルダには、エクスポート元の辞書フォルダと同じコメントが付けられます。

(c) エクスポート運用に設定していない辞書フォルダ資源のインポート

辞書と関連情報のエクスポート規則にあるように、運用に設定しなかった辞書フォルダの資源でも、結合およびデータ項目と業務ルールの関連を基にエクスポートされる場合があります。この資源がどのようにインポートされるのかを次に説明します。なお、どの場合もインポートされる資源が関連を持っていれば、その関連もインポートされます。

- 挿入インポートでの扱い
エクスポート元と同じ名称の辞書フォルダに資源が作成されます。ただし、この資源が継承関係の下位データ項目および構成項目で、上位データ項目がエクスポート元とは異なる辞書フォルダにある場合は、上位データ項目と同じ辞書フォルダに作成されます。エクスポート元と同じ名称の辞書フォルダがない場合は、その辞書フォルダが

3. 辞書の環境設定と運用

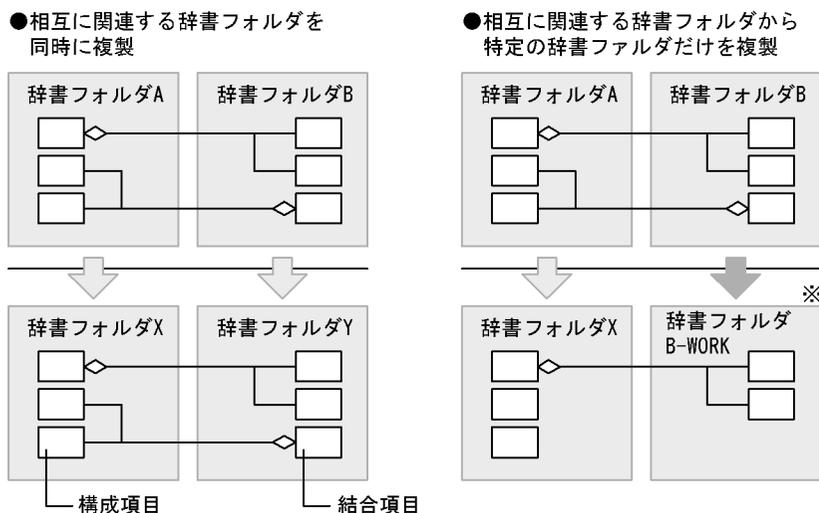
自動的に作成され、この中に構成項目が作成されます。

- 上書きインポートでの扱い
 エクスポート元と同じ名称の辞書フォルダに資源が作成されます。すでにその辞書フォルダに資源がある場合は、エクスポート元と同じ名称の辞書フォルダに資源が移動し、置換されます。なお、エクスポート元と同じ名称の辞書フォルダがない場合は、その辞書フォルダが自動的に作成されます。
- 複製を目的に、辞書フォルダを指定してインポートした場合の扱い
 指定外の辞書フォルダにあった資源については、新しい辞書フォルダが自動的に作成され、その中にインポートされます。この辞書フォルダには「エクスポート元の辞書フォルダ名-WORK」の名称が付けられます。

(d) 複製を目的とした、相互に関連する辞書フォルダのインポート

この場合の運用は2通り考えられます。別の辞書フォルダのデータ項目同士が結合関係にある場合の例を図3-6に示します。

図3-6 相互に関連する辞書フォルダのインポート



注※ 関連を基に作成された辞書フォルダには「エクスポート元の辞書フォルダ名-WORK」の名称が付けられます。

- 相互に関連する辞書フォルダを同時に複製する
 図は、辞書フォルダ A, B を同時に複製する場合の例です。辞書フォルダ A, B を同時にエクスポートします。そして、インポートではインポート先にそれぞれ X, Y を指定しています。辞書フォルダの資源および相互の関連が、そのままインポート先に指定した辞書フォルダに複製されます。
- 相互に関連する辞書フォルダから、特定の辞書フォルダだけを複製する
 図は、辞書フォルダ A, B を同時にエクスポートしていますが、インポートでは A に対してだけインポート先に X を指定しています。この例のように、インポートで指定

しなかった辞書フォルダに関連する資源がある場合には、辞書フォルダが自動的に作成され、その辞書フォルダとの関連と、関連する資源だけが複製されます。

(e) 辞書（データ項目、業務ルール）とドキュメント間の関連

辞書とドキュメントを複製の目的でインポートした場合には、辞書のデータ項目または業務ルールとドキュメント間の関連はインポートされません。

(f) 環境構築情報のインポート

インポート先の環境構築情報（辞書フォルダやデータ項目などの情報）の情報は、エクスポート元の環境構築情報に置換されます。

(g) インポートの中止

インポートを中止した場合、中止する前までの資源はインポート先に登録されているので、エクスポート元のリポジトリとインポート先のリポジトリで矛盾が生じてしまいます。インポート先のリポジトリを回復するためには、中止したインポートを再実行してインポートを完了させる必要があります。

4

辞書情報の一括登録と出力

SEWB+/REPOSITORY には、辞書情報を一括して登録したり出力したりするための機能が用意されています。この機能を利用すれば一度に複数の辞書情報を扱えるため、作業の効率化と既存の開発資源の有効利用を図れます。

この章では、一括して辞書情報を扱う作業と機能の使い方を説明します。

4.1 CSV 形式ファイルを使って辞書登録と出力をする

4.2 データベースから辞書のレコード定義（最上位結合項目）を生成する

4.3 SEWB3 資産を移行する

4.1 CSV 形式ファイルを使って辞書登録と出力をする

CSV 形式とは、表計算プログラムやリレーショナルデータベースで扱える一般的なテキストデータの形式です。この CSV 形式でデータが記述されたファイルを CSV 形式ファイルと呼びます。

SEWB+/REPOSITORY では、CSV 形式ファイルを使って一括して辞書情報を登録または出力できる機能を用意しています。この機能を「CSV 入力」、「CSV 出力」といいます。コマンドで実行することもでき、入力に使用するコマンドを CSV 形式ファイル入力コマンド、出力に使用するコマンドを CSV 形式ファイル出力コマンドといいます。

4.1.1 アプリケーション間の辞書情報の受け渡し

CSV 形式ファイルを経由したアプリケーション間の辞書情報の受け渡しの例を図 4-1 に示し、利用例を次に説明します。

データ分析やデータベース設計ツールから情報を連携する

システム開発の上流工程を支援するアプリケーションから、データ分析結果やデータ構造に関する情報を引き継いで辞書に登録します。アプリケーションで扱える CSV 形式のファイルフォーマットはそれぞれ異なるため、情報を連携したいアプリケーションに対応したフォーマットを使用します。

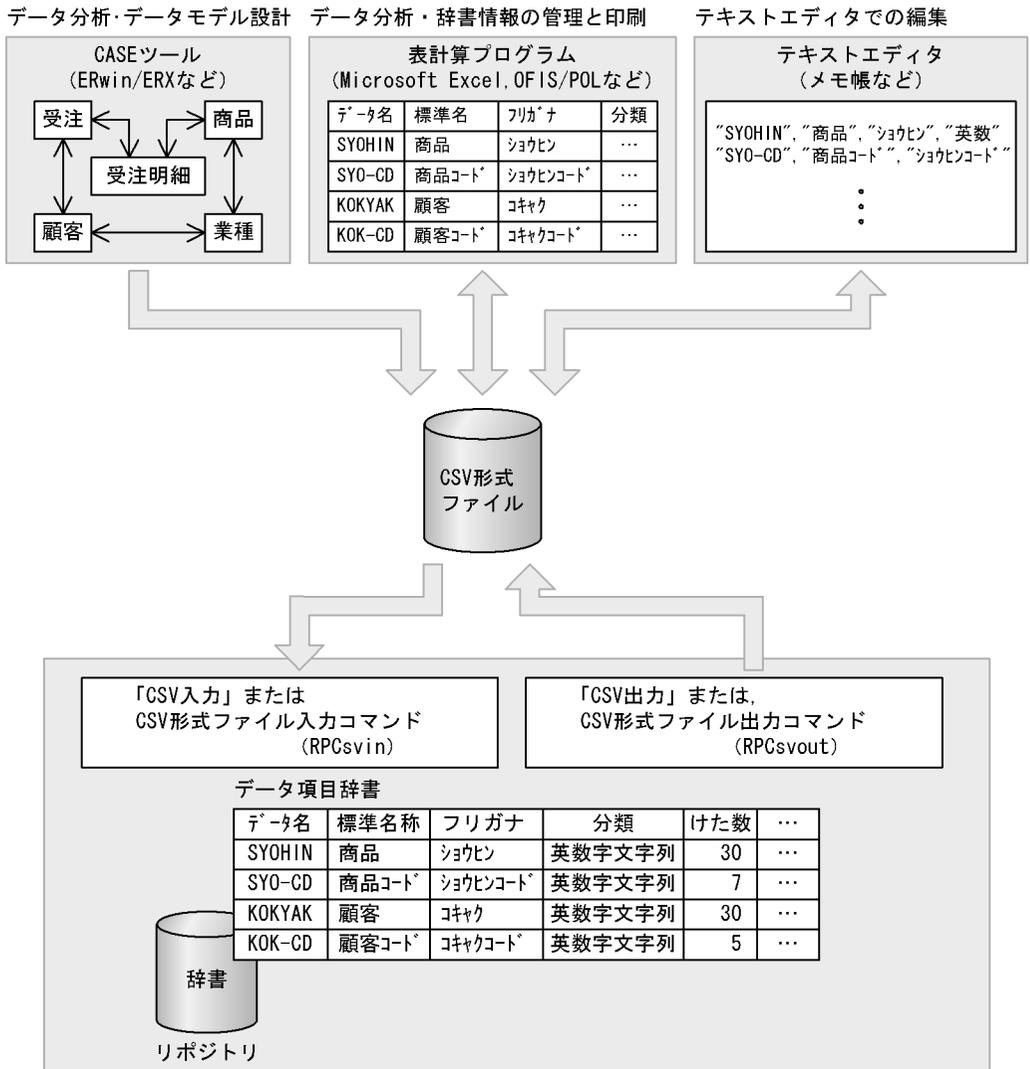
表計算プログラムを利用して一覧表示や印刷をする

表計算プログラムを使ってデータ分析結果をまとめている場合、その情報を引き継いで辞書に登録します。また、実際に辞書を運用しているときのメンテナンス作業では、辞書情報を印刷したり、一覧表で確認したりします。

テキストエディタで直接、編集する

メモ帳などのテキストエディタで直接、辞書情報を編集したり CSV 形式ファイルのフォーマットを変更したりします。

図 4-1 アプリケーション間での情報の受け渡しの例

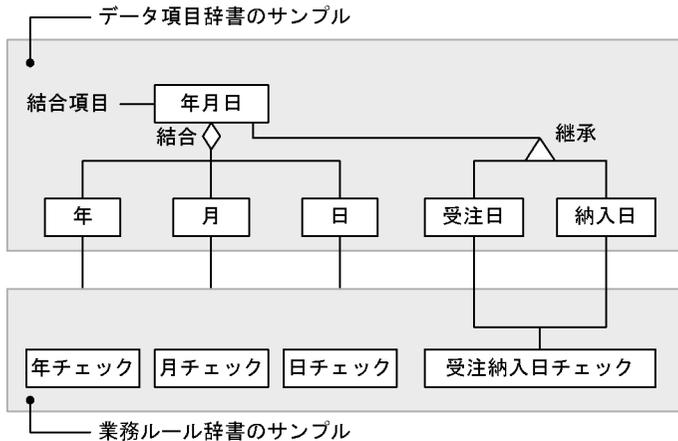


SEWB+/REPOSITORYで扱うフォーマットや使い方の概要を理解いただくため、CSV形式ファイルのサンプルが用意されています。表計算プログラムやテキストエディタを使って参照したり、辞書登録を体験したりすることをお勧めします。

データ項目辞書と業務ルール辞書の2種類のサンプルファイルがあります。それぞれのファイルの概要を図4-2に示します。

4. 辞書情報の一括登録と出力

図 4-2 CSV 形式ファイルのサンプル



- サンプルの提供ディレクトリとファイル名を次に示します。

ディレクトリ：

SEWB+ 基本開発環境の組み込み先パス名 ¥Repository¥sample (サーバの場合)

または

SEWB+ 基本開発環境の組み込み先パス名 ¥Repository-Browser¥sample (クライアントの場合)

ファイル名：

sampledi.csv (データ項目辞書)

samplebr.csv (業務ルール辞書)

- サンプルの登録手順

「CSV 入力」を使い、CSV 形式ファイルを辞書登録します。次の手順で登録すると、辞書フォルダ「Sample」の下にデータ項目と業務ルールが作成されます。

1. データ項目辞書のサンプル入力

[スタート]メニューの[プログラム]または[すべてのプログラム]から [Sewb+] - [Repository] - [CSV 入力]を選び、表示されるダイアログで次の指定をして実行します。

- [CSV 入力]ダイアログ

[登録先辞書フォルダ]の[新規に作成する]をオンにし、「Sample」を指定。

[入力 CSV 形式ファイル名]に、データ項目辞書のサンプルファイル名を指定。

[入力する辞書種別]に「データ項目」を指定。

- [CSV 入力オプション]ダイアログ

[CSV ファイルのヘッダを識別する]をオン指定。

2. 業務ルール辞書のサンプル入力

[スタート]メニューの[プログラム]または[すべてのプログラム]から [Sewb+] -

[Repository] - [CSV 入力] を選び、表示されるダイアログで次の指定をして実行します。

- [CSV 入力] ダイアログ
 [登録先辞書フォルダ] の [既存の辞書フォルダに登録する] をオンにし、「Sample」を指定。
 [入力 CSV 形式ファイル名] に業務ルール辞書のサンプルファイル名を指定。
 [入力する辞書種別] に「業務ルール」を指定。
- [CSV 入力オプション] ダイアログ
 [CSV ファイルのヘッダを識別する] をオン指定。

なお、組み込み先パス名は、デフォルトでは次のように設定されています。システムドライブとは、Windows が組み込まれているドライブです。

サーバの組み込み先パス名

- Windows 2000 Server または Windows Server 2003 の場合
 システムドライブ : ¥Program Files¥HITACHI¥Sewb
- Windows Server 2003 x64 の場合
 システムドライブ : ¥Program Files (x86) ¥HITACHI¥Sewb

クライアントの組み込み先パス名

- Windows 2000, Windows XP, Windows Server 2003 または Windows Vista の場合
 システムドライブ : ¥Program Files¥HITACHI¥Sewb
- Windows Server 2003 x64 の場合
 システムドライブ : ¥Program Files (x86) ¥HITACHI¥Sewb

データ項目辞書、業務ルール辞書のそれぞれに対応した CSV 形式のファイルフォーマットは、入力と出力で共通に使用できます。詳細は、「4.1.7 CSV 形式ファイルフォーマット」を参照してください。

4.1.2 CSV 形式ファイルの入力

ここでは、CSV 形式ファイルから辞書情報を入力する場合の入力規則について説明します。

ダイアログでの操作については、ヘルプを参照してください。また、CSV 形式ファイル入力コマンドの指定方法については、「4.1.3 CSV 形式ファイル入力コマンド (RPCsvin) の使い方」を参照してください。

(1) データ項目辞書の入力規則

データ項目辞書に CSV 形式ファイルが入力される規則について説明します。

4. 辞書情報の一括登録と出力

(a) システム内部 ID によるデータ項目の識別

リポジトリ内の各データ項目は、SEWB+/REPOSITORY によって一意の値が割り当てられた「システム内部 ID」で識別されます。したがって、データ項目名が同じでも、このシステム内部 ID の値が異なれば、別のデータ項目として扱われます。このシステム内部 ID は、データ項目自身を示すほか、継承関係にある上位データ項目や結合関係にある構成項目など、関連するデータ項目の識別にも使われます。

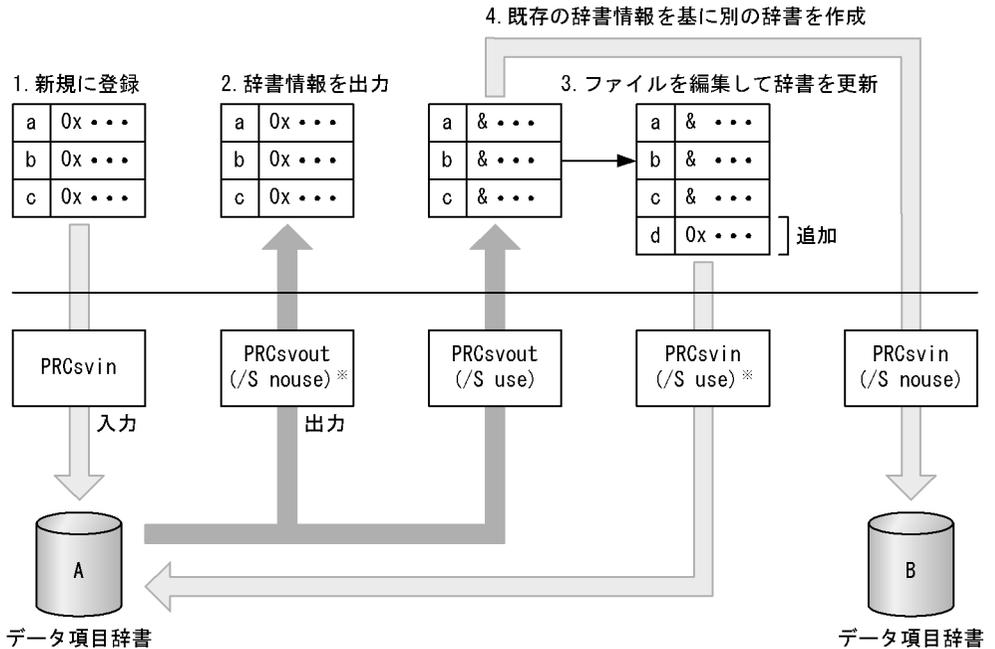
CSV 形式ファイルを作成して入力するときや、既存の CSV 形式ファイルにデータ項目を追加するときには、システム内部 ID の代わりにユーザが仮の ID を決めて割り当てます。これを「仮システム内部 ID」といい、先頭が「0x」で始まる 16 文字以内の 16 進文字列で記述します（システム内部 ID の先頭には「&」が付き、これによって形式が区別されます）。仮システム内部 ID は、CSV 形式ファイルを入力すると自動的にシステム内部 ID に変換されます。

注

これらの ID は、標準フォーマットの CSV 形式ファイルで使います。

- 作業の目的に応じた ID の扱い方
ユーザ作業の例と CSV 形式ファイルの入出力の関係（コマンドを使用する場合）を、
図 4-3 に示します。

図 4-3 ユーザ作業の例と CSV 形式ファイルの入出力の関係



(凡例)

PRCsvin : CSV形式ファイル入力コマンド

PRCsvout : CSV形式ファイル出力コマンド

() : コマンドに指定する引数

注※ 引数「/S」の指定を省略すると、この値が仮定されます。

図の 1. ~ 4. の作業に応じた ID の扱いについて解説します。入力時の ID の扱いは、CSV 形式ファイル入力コマンドの引数「/S 辞書の登録方法」、または [CSV オプション] ダイアログの「辞書の登録方法」で指定します。また、出力時の ID の扱いは、CSV 形式ファイル出力コマンドの引数「/S ID の出力形式」、または [CSV 出力オプション] ダイアログの「出力 ID の形式」で指定します。

1. 新規に CSV 形式ファイルを作成してデータ項目を登録する

各データ項目には、CSV 形式ファイル内で一意となるユーザ任意の仮システム内部 ID を割り当てます。

2. 辞書情報を出力する（更新を目的としないとき）

表計算プログラムで辞書情報を印刷したいときなど、辞書情報の更新を目的としない場合には、仮システム内部 ID の形式で出力します。

3. CSV 形式ファイルで辞書情報を編集したあと、再入力して辞書を更新する

出力では、システム内部 ID の形式で出力します。CSV 形式ファイル上で既存の辞書情報を編集するときには、システム内部 ID を変更しないように注意してください。再入力時には、CSV 形式ファイルのシステム内部 ID を使用して入力します。

4. 既存の辞書情報を基に別の辞書を作成する

4. 辞書情報の一括登録と出力

出力では、システム内部 ID の形式で出力します。再入力時には、システム内部 ID を新しく割り当てて入力します。

- システム内部 ID の注意事項

CSV 形式ファイルに出力されたシステム内部 ID は、変更しないでそのまま使用してください。変更した場合、CSV 形式ファイルのデータと辞書の整合性は保証されません。

- (b) データ項目名によるデータ項目の識別

CSV 形式ファイルでのデータ項目の識別には、システム内部 ID を使用するほかに、データ項目名を使用することもできます。ただし、この場合には、CSV 形式ファイル、および CSV 形式ファイルを入力する辞書フォルダ内で、データ項目名を必ず一意に定義していることが前提になります。このため、データ項目名の継承フラグには「1」を指定してはなりません。データ項目名をデータ項目の識別に使用する場合、システム内部 ID および仮システム内部 ID は、ほかのデータ項目からの関連先としてだけ使用されます。

データ項目名によるデータ項目の識別は、CSV 形式ファイル入力コマンドの引数「/S 辞書の登録方法」、または [CSV 入力オプション] ダイアログの「辞書への登録方法」で指定します。

- (c) データ項目の登録および更新

入力時の指定によって、データ項目の登録および更新の方法を次の 3 通りから選択できます。これは、CSV 形式ファイル入力コマンドの引数「/S 辞書の登録方法」、または [CSV 入力オプション] ダイアログの「辞書への登録方法」で指定します。

- a. 新規に登録する

- b. システム内部 ID で更新する

- c. データ項目名で更新する

新規に登録したい場合は、a. の方法で入力を実行します。b. または c. の方法を選んだ場合、システム内部 ID またはデータ項目名によって、更新する対象のデータ項目が特定されます。一致するデータ項目が辞書に存在しない場合は新規データ項目として登録されます。

- (d) 関連情報の扱い

いったん、CSV 形式ファイルに辞書フォルダをわたる結合項目をシステム内部 ID を使用して出力したあと、再びその CSV 形式ファイルを入力すると、出力時の関連情報を基にして結合項目の関連が復元されます。

したがって、結合先のデータ項目が CSV 形式ファイルまたはリポジトリ内にないと、CSV 形式ファイルを入力することができません。この場合には、CSV 形式ファイルの内容を見直して、再度入力をしてください。

(e) 国語区分, 言語区分, フィールドの指定

入力するデータ項目の定義情報の国語区分, 言語区分およびフィールドを指定できます。CSV 形式ファイル入力コマンドの引数「/C 登録先国語別」, 「/N 登録先言語別」, 「/I 登録先フィールド別」, または [CSV 入力オプション] ダイアログの「入力する辞書情報」で指定します。指定した区分に対応する行の値だけが辞書に入力されます。

(f) 継承フラグ

CSV 形式ファイル中の定義情報を継承できる定義項目にはそれぞれ継承フラグが設けられています。したがって, 項目ごとに上位データ項目から値を継承するかどうかを指定できます。標準フォーマットで継承フラグを省略した場合や, 継承フラグを指定できないフォーマットで入力する場合には, 継承関係の最上位以外にあるデータ項目の継承フラグに「継承する」が仮定されます。

ただし, 環境構築ユーティリティで継承モードに「必ず継承」または「継承禁止」を設定している場合は, 継承モードが優先されます。

(g) 継承関係のデータ項目を扱うときの注意

継承関係の下位のデータ項目を登録する場合, 次の点に注意してください。

- データ項目の定義情報のうち, 上位データ項目から継承するように指定したものについては, CSV 形式ファイルで値が定義されていても無視され, 上位データ項目の値が継承されます。
- 下位にある結合項目の構成項目を指定する場合は, 親の構成項目と同じ順番にある構成項目, または下位のデータ項目でなければなりません。

(h) 継承関連の更新有無の指定

データ項目名を使用して更新登録する場合, ユーザが既存のデータ項目に定義していた継承の関連を更新しない指定ができます。CSV 形式ファイル入力コマンドの引数「/P 継承関連の更新有無」, または [CSV 入力オプション] ダイアログの「辞書への登録方法」で指定します。

例えば, ERwin/ERX からの辞書登録では通常, データ項目は継承関係の最上位のデータ項目として登録されます。しかし, ERwin/ERX から新規に辞書登録したあと, ユーザがデータ項目の継承関係を変更していても, この指定をすれば, この関係を保てます。

(i) 親データ項目と構成項目の指定

親データ項目や構成項目は, システム内部 ID や仮システム ID で指定する代わりに, プリフィックスを付けたデータ項目名を指定して辞書に入力できます。

同一名称のデータ項目が, 同じ CSV ファイルや辞書フォルダに複数ある場合, プリフィックスを付けたデータ項目名で親データ項目や構成項目を指定することはできません。その場合は, システム内部 ID, または仮システム ID で指定してください。

4. 辞書情報の一括登録と出力

(j) 連携する ERwin/ERX の情報だけの更新

ERwin/ERX の分析情報で既存のデータ項目を更新する場合、ERwin/ERX で変更された属性情報だけをデータ項目に反映できます。CSV 形式ファイルコマンドの引数「/W 更新対象属性」で指定、または [CSV 入力オプション] ダイアログの「CSV フォーマット種別」で「ERwin/ERX 属性のみを更新する」を選びます。

ERwin/ERX の属性の更新だけを指定すれば、通常の変換規則で初期値やシステムで変換した値が設定される定義情報は変更されません。

ERwin/ERX のタイプ情報から変換されたデータ項目種別（単項目 / 結合項目）が、更新の対象になったデータ項目と異なる場合は、登録時にエラーとなります。

(k) データ項目を一括更新する辞書フォルダの範囲

データ項目名でデータ項目を更新する場合、辞書フォルダの更新の対象範囲を次の三つから選択できます。

- 登録先の辞書フォルダ
- 一意性チェック範囲の辞書フォルダ
- すべての辞書フォルダ

登録先の辞書フォルダをデータ項目の一括更新の対象にするには、CSV 形式ファイル入力コマンドの引数「/V 更新対象とする辞書フォルダの範囲」に「only」を指定するか、または [CSV 入力オプション] ダイアログの「辞書への登録方法」で「登録先辞書フォルダのみを更新対象にする」を選びます。

一意性チェック範囲の辞書フォルダを一括更新の対象にするには、CSV 形式ファイル入力コマンドの引数「/V 更新対象とする辞書フォルダの範囲」で「scope」を指定するか、または [CSV 入力オプション] ダイアログの「辞書への登録方法」で「一意性チェック範囲を更新対象にする」を選びます。環境構築ユーティリティで登録した一意性チェック範囲の辞書が更新の対象になります。一意性チェック範囲は、CSV 形式ファイル入力コマンドの引数「/H データ項目の一意性チェックの範囲」、または [CSV 入力オプション] ダイアログの「データ項目の一意性チェック範囲」のコンボボックスで指定します。

リポジトリのすべての辞書フォルダをデータ項目の一括更新の対象にするには、CSV 形式ファイル入力コマンドの引数「/V 更新対象とする辞書フォルダの範囲」で「all」を指定するか、または [CSV 入力オプション] ダイアログの「辞書への登録方法」で「全辞書フォルダを更新対象にする」を選びます。全辞書フォルダを更新対象にすれば、新規登録したあとに、データ項目の辞書フォルダを変更しても、定義情報を更新できます。

一意性チェック範囲の辞書フォルダまたはすべての辞書フォルダを対象とした場合、対象となる辞書フォルダに存在する、データ項目名が一致するデータ項目が更新されます。したがって、データ項目を新規登録したあとに、所属する辞書フォルダを変更しても、同じデータ項目の定義情報を更新できます。なお、CSV 形式ファイルで定義されたデータ項目名がどの辞書フォルダにも存在しない場合には、指定した辞書フォルダに新規

データ項目として登録されます。

(2) 業務ルール辞書の入力規則

業務ルール辞書に CSV 形式ファイルが入力される規則について説明します。

(a) システム内部 ID による業務ルールの識別

リポジトリ内の各業務ルールは、SEWB+/REPOSITORY によって一意の値が割り当てられた「システム内部 ID」で識別されます。したがって、業務ルール名が同じでも、このシステム内部 ID の値が異なれば、別の業務ルールとして扱われます。

CSV 形式ファイルを作成して入力するときや、既存の CSV 形式ファイルに業務ルールを追加するときは、システム内部 ID の代わりにユーザが仮の ID を割り当てます。これを「仮システム内部 ID」といい、先頭が「0x」で始まる 16 文字以内の 16 進文字列で記述します（システム内部 ID の先頭には、「&」が付き、これによって形式が区別されます）。仮システム内部 ID は、CSV 形式ファイルを入力すると自動的にシステム内部 ID に変換されます。

- 作業の目的に応じた ID の扱い方
作業に応じた ID の扱いや入力時の指定の考え方については、データ項目の場合と同様です。「(1)(a) システム内部 ID によるデータ項目の識別」を参考にしてください。
- システム内部 ID の注意事項
CSV 形式ファイルに出力されたシステム内部 ID は、変更しないでそのまま使用してください。変更した場合、CSV 形式ファイルのデータと、辞書の整合性は保証されません。

(b) 業務ルール名による業務ルールの識別

CSV 形式ファイルでの業務ルールの識別には、システム内部 ID を使用するほかに、業務ルール名を使用することもできます。ただし、この場合には、CSV 形式ファイル、および CSV 形式ファイルを入力する辞書フォルダ内で、業務ルール名を必ず一意に定義していることが前提になります。業務ルール名を識別に使用する場合、システム内部 ID および仮システム内部 ID は、無視されます。

業務ルール名による業務ルールの識別は、CSV 形式ファイル入力コマンドの引数「/S 辞書の登録方法」、または [CSV 入力オプション] ダイアログの「辞書への登録方法」で指定します。

(c) 業務ルールに関連するデータ項目の指定

業務ルールに関連するデータ項目を指定するときは、データ項目辞書から出力された CSV 形式ファイル中のシステム内部 ID、またはプリフィックスを付けたデータ項目名を使用します。システム内部 ID を使用して業務ルールの識別をする場合には、ほかの辞書フォルダにあるデータ項目への関連も指定できます。

関連するデータ項目名と同一名称のデータ項目が複数登録されている場合、プリフィッ

4. 辞書情報の一括登録と出力

クスを付けたデータ項目名では指定できません。その場合は、システム内部 ID で指定してください。

(d) 関連情報の扱い

いったん、CSV 形式ファイルに、辞書フォルダを渡って関連するデータ項目をシステム内部 ID を指定して出力したあと、再びその CSV 形式ファイルを入力すると、出力時の関連情報を基にして関連が復元されます。したがって、関連するデータ項目が CSV 形式ファイルまたはリポジトリ内にないと、CSV 形式ファイルを入力できないため、注意が必要です。

(e) 業務ルールの登録および更新

入力時の指定によって、業務ルールの登録および更新の方法を次の 3 通りから選択できます。これは、CSV 形式ファイル入力コマンドの引数「/S 辞書の登録方法」、または [CSV 入力オプション] ダイアログの「辞書への登録方法」で指定します。

- a. 新規に登録する
- b. システム内部 ID で更新する
- c. 業務ルール名で更新する

新規に登録したい場合は、a. の方法で入力を実行します。b. または c. の方法を選んだ場合、システム内部 ID または業務ルール名によって、更新する対象の業務ルールが特定されます。一致する業務ルールが辞書に存在しない場合は新規業務ルールとして登録されます。

(f) 業務ルールを一括更新する辞書フォルダの範囲

業務ルール名で業務ルールを更新する場合、辞書フォルダの更新の対象範囲を次の三つから選択できます。

- 登録先の辞書フォルダ
- 一意性チェック範囲の辞書フォルダ
- すべての辞書フォルダ

登録先の辞書フォルダを業務ルールの一括更新の対象にするには、CSV 形式ファイル入力コマンドの引数「/V 更新対象とする辞書フォルダの範囲」に「only」を指定するか、または [CSV 入力オプション] ダイアログの「辞書への登録方法」で「登録先辞書フォルダのみを更新対象にする」を選びます。

一意性チェック範囲の辞書フォルダを一括更新の対象にするには、CSV 形式ファイル入力コマンドの引数「/V 更新対象とする辞書フォルダの範囲」で「scope」を指定するか、または [CSV 入力オプション] ダイアログの「辞書への登録方法」で「一意性チェック範囲を更新対象にする」を選びます。環境構築ユティリティで登録した一意性チェック範囲の辞書が更新の対象になります。一意性チェック範囲は、CSV 形式ファイル入力コマンドの引数「/J 業務ルールの一意性チェックの範囲」、または [CSV 入力オプション] ダ

イアログの「業務ルールの一意性チェック範囲」のコンボボックスで指定します。

リポジトリのすべての辞書フォルダを業務ルールの一括更新の対象にするには、CSV形式ファイル入力コマンドの引数「/V 更新対象とする辞書フォルダの範囲」で「all」を指定するか、または[CSV入力オプション]ダイアログの「辞書への登録方法」で「全辞書フォルダを更新対象にする」を選びます。全辞書フォルダを更新対象にすれば、新規登録したあとに、業務ルールの辞書フォルダを変更しても、定義情報を更新できます。

一意性チェック範囲の辞書フォルダまたはすべての辞書フォルダを対象とした場合、対象となる辞書フォルダに存在する、業務ルール名が一致する業務ルールが更新されます。したがって、業務ルールを新規登録したあとに、所属する辞書フォルダを変更しても、同じ業務ルールの定義情報を更新できます。なお、CSV形式ファイルで定義された業務ルール名がどの辞書フォルダにも存在しない場合には、指定した辞書フォルダに新規業務ルールとして登録されます。

(g) 国語区分、言語区分、フィールドの指定

入力する業務ルールの定義情報の国語区分、言語区分およびフィールドを指定できます。CSV形式ファイル入力コマンドの引数「/C 登録先国語別」、「/N 登録先言語別」、「/I 登録先フィールド別」、または[CSV入力オプション]ダイアログの「入力する辞書情報」で指定します。指定した区分に対応する行の値だけが辞書に入力されます。

4.1.3 CSV形式ファイル入力コマンド(RPCsvin)の使い方

(1) CSV形式ファイル入力コマンドが格納されているディレクトリ

CSV形式ファイル入力コマンドが格納されているディレクトリを示します。

(a) SEWB+/REPOSITORY(サーバ)の場合

コマンドは、次のディレクトリに格納されています。

SEWB+ 基本開発環境の組み込み先パス名¥Repository¥bin

SEWB+ 基本開発環境の組み込み先パス名は、デフォルトでは次のように設定されています。システムドライブとは、Windows 2000 Server, Windows Server 2003 または Windows Server 2003 x64 が組み込まれているドライブです。

組み込み先パス名

- Windows 2000 Server または Windows Server 2003
システムドライブ : ¥Program Files¥HITACHI¥Sewb
- Windows Server 2003 x64 の場合
システムドライブ : ¥Program Files(x86)¥HITACHI¥Sewb

(b) SEWB+/REPOSITORY-BROWSER(クライアント)の場合

コマンドは、次のディレクトリに格納されています。

4. 辞書情報の一括登録と出力

SEWB+ 基本開発環境の組み込み先パス名¥Repository-Browser

SEWB+ 基本開発環境の組み込み先パス名は、デフォルトでは次のように設定されています。システムドライブとは、Windows が組み込まれているドライブです。

組み込み先パス名

- Windows 2000 , Windows XP , Windows Server 2003 または Windows Vista の場合
システムドライブ : ¥Program Files¥HITACHI¥Sewb
- Windows Server 2003 x64 の場合
システムドライブ : ¥Program Files (x86) ¥HITACHI¥Sewb

(2) コマンド実行時の注意事項

(a) コマンドを実行する前に

サーバでコマンドを実行するときは、サーバでは SEWB+/REPOSITORY からログアウトしておいてください。

(b) コマンド実行中にほかの作業をする場合

CSV 形式ファイル入力コマンドでは、実行中に排他制御を行っていません。したがって、コマンドの実行中でも並行してほかの作業をすることができます。ただし、次に示す点に注意してください。

- コマンドの実行中に対象となっている資源や関連情報を更新または削除してはなりません。不整合が検出され、正常に終了しない場合があります。この場合には、CSV 形式ファイル入力コマンドの再実行が必要です。
- コマンドの実行中に、CSV 形式ファイルの入力先で更新の対象となっている資源を更新してはなりません。CSV 形式ファイル入力コマンドが正常に終了しても、資源の内容が更新されていない場合があります。

(c) ほかの辞書フォルダに構成項目を持つ結合項目を CSV 形式ファイルに出力し、このファイルを再び辞書登録する場合

この CSV 形式ファイルには構成項目 ID が含まれるため、そのまま辞書に登録しようとすると構成項目が要因となってエラーが発生します。また、ID をシステム内部 ID の形式で出力した場合、出力されたシステム内部 ID が CSV 形式ファイルまたはリポジトリ内にないときにもエラーとなります。構成項目 ID またはその構成項目を参照している結合項目を削除してから CSV 形式ファイル入力コマンドを実行してください。

(d) システム内部 ID の注意事項

システム内部 ID の形式で出力した CSV 形式ファイルを、次の条件で入力してはなりません。

- 同じノード ID でデータベースを初期化したあと、初期化する前に出力した CSV 形式ファイルを入力する

- データベースのノード ID が重複して指定されているサーバ間で、CSV 形式ファイルを入力したり出力したりする

この条件で入力してエラーが発生した場合には、「3.3.1(4) 辞書をエクスポート・インポートするときの留意事項」を参照し、<インポート方法 1>1. ~ 3.の手順を実行してください。そのあとに、CSV 形式ファイルを入力してください。

(3) CSV 形式ファイル入力コマンド (RPCsvin) の指定

コマンドラインに、次に示す構文に従ってコマンドと引数を指定して実行します。

サーバ上では、リポジトリのシステム管理者の権限で実行します。起動時に表示されるダイアログに、システム管理者のパスワードを入力する必要があります。クライアント上では、SEWB+ ログインに指定されたユーザの権限で実行します。起動時に表示されるダイアログに、ユーザ名、ログイン先およびユーザのパスワードを入力する必要があります。

実行中には、データ項目または業務ルールの登録件数が表示されます。また、詳細情報がログファイルに出力されます。

(a) 形式

```
RPCsvin /F 入力 CSV 形式ファイル名 /D 辞書フォルダ名
        [ /O 所有者名 ] [ /G グループ名 ]
        [ /K 辞書種別 ] [ /C 登録先国語別 ] [ /N 登録先言語別 ]
        [ /I 登録先フィールド別 ]
        [ /E ファイルフォーマット種別 (02-01 互換用) ]
        [ /M ファイルフォーマット種別 ] [ /B データベース種別 ]
        [ /Z 物理名の異なる外部キーを識別する ]
        [ /A 属性レポートファイル名 ]
        [ /T ヘッダの有無 ]
        [ /S 辞書への登録方法 ]
        [ /W 更新対象属性 ]
        [ /R 同一カラム名の属性のマージ有無 ]
        [ /P 継承関連の更新有無 ]
        [ /V 更新対象とする辞書フォルダの範囲 ]
        [ /H データ項目の一意性チェックの範囲 ]
        [ /J 業務ルールの一意性チェックの範囲 ]
        [ /L ログファイル名 ]
        [ /? ]
```

(b) 引数

/F 入力 CSV 形式ファイル名
 入力する CSV 形式のファイル名を指定します。

4. 辞書情報の一括登録と出力

/D 辞書フォルダ名

入力した辞書情報の登録先となる辞書フォルダ名を指定します。

/O 所有者名¹

登録する辞書の所有者名を指定します。省略時はログインしたユーザが仮定されます。ただし、入力する CSV 形式ファイルに所有者名が定義されている場合は、定義内容がこの引数の指定より優先されます。

/G グループ名¹

登録する辞書のグループ名を指定します。省略時は、システム管理者でログインしている場合は“everyone”が、システム管理者以外の場合はログインしたユーザが所属するグループが仮定されます。ただし、入力する CSV 形式ファイルにグループ名が定義されている場合は、定義内容がこの引数の指定より優先されます。

/K 辞書種別

登録する辞書の種別を指定します。省略時は「DI」が仮定されます。

DI：データ項目辞書

BR：業務ルール辞書

/C 登録先国語別

登録する国語区分を指定します。環境構築ユティリティで設定した国語区分の順序に対応して1～4の数字で指定します。省略時は、CSV形式ファイルに記述されている値が、記述順で国語区分の設定順に辞書へ登録されます。

/N 登録先言語別

登録する言語区分を指定します。環境構築ユティリティで設定した言語区分の順序に対応して1～10の数字で指定します。省略時は、CSV形式ファイルに記述されている値が、記述順で言語区分の設定順に辞書へ登録されます。

/I 登録先フィールド別

登録するフィールドを指定します。環境構築ユティリティで設定したフィールドの順序に対応してデータ項目の場合は1～20、業務ルールの場合は1～3の数字で指定します。省略時は、CSV形式ファイルに記述されている値が、記述順でフィールドの設定順に辞書へ登録されます。

/E ファイルフォーマット種別（02-01 互換用）

入力する CSV 形式ファイルのフォーマット種別を次の値で指定します。省略時は「std」が仮定されます。

ただし、この引数は、SEWB+ リポジトリセットの SEWB+/REPOSITORY 02-01 以前での指定と互換性を保つためのものです。したがって、SEWB+ リポジトリセットの SEWB+/REPOSITORY 02-02 以降でのフォーマット種別の指定には、引数「/M ファイルフォーマット種別」で指定することをお勧めします。

std：標準フォーマット（「/M std」と同様）

- ERwin/ERX のデータ分析情報を利用する場合（データベース：ORACLE）

ewe_o : エンティティレポート形式フォーマット (「/M ewe /B ora_c」と同様)

ewa_o : 属性レポート形式フォーマット (「/M ewa /B ora_c」と同様)

ewe_ox : エンティティレポート形式フォーマット (「/M ewe /B ora」と同様)

ewa_ox : 属性レポート形式フォーマット (「/M ewa /B ora」と同様)

バージョン 2.6 以前の ERwin/ERX で ORACLE を使用する場合「ewe_o」, 「ewa_o」は SEWB+ リポジトリセットの SEWB+/REPOSITORY 02-00 との互換用フォーマットです。SEWB+ リポジトリセットの SEWB+/REPOSITORY 02-01 以降の場合、より効率的な「ewe_ox」, 「ewa_ox」の使用をお勧めします。

- ERwin/ERX のデータ分析情報を利用する場合 (データベース: HiRDB)

ewe_h : エンティティレポート形式フォーマット (「/M ewe /B hir」と同様)

ewa_h : 属性レポート形式フォーマット (「/M ewa /B hir」と同様)

/M ファイルフォーマット種別

入力する CSV 形式ファイルのフォーマット種別を次の値で指定します。省略時には「std」が仮定されます。

std : 標準フォーマット

ewe : ERwin/ERX (バージョン 2.6 以前) エンティティレポート形式

ewa : ERwin/ERX (バージョン 2.6 以前) 属性レポート形式

ewt : ERwin/ERX (バージョン 3.5) テーブルレポート形式

/B データベース種別

ERwin/ERX で使用するデータベース種別を次の値で指定します。省略時は「ora」が仮定されます。

ora : ORACLE

ora_c : ORACLE (SEWB+ リポジトリセットの SEWB+/REPOSITORY 02-00 との互換用)

hir : HiRDB

この引数は、引数「/M ファイルフォーマット種別」または「/E ファイルフォーマット種別 (02-01 互換用)」で ERwin/ERX 専用のフォーマット種別が指定されている場合に有効となります。

/Z 物理名の異なる外部キーを識別する

ERwin/ERX で PK 属性と物理名の異なる FK 属性を識別するかどうかを、次の値で指定します。省略時は「use」が仮定されます。

use : 識別する

uouse : 識別しない

この引数は、引数「/M ファイルフォーマット種別」または「/E ファイルフォーマット種別 (02-01 互換用)」で ERwin/ERX 専用のフォーマット種別が指定されている場合に有効となります。

/A 属性レポートファイル名

引数「/F 入力 CSV 形式ファイル名」にエンティティレポートのファイル名を指定し

4. 辞書情報の一括登録と出力

た場合、同時に入力する属性レポートのファイル名を指定します。この引数を省略すると、属性レポートは同時入力されません。この引数は、引数「/M ファイルフォーマット種別」または「/E ファイルフォーマット種別 (02-01 互換用)」で、ERwin/ERX 専用のフォーマット種別としてエンティティレポート形式が指定されている場合に有効となります。

/T ヘッダの有無

入力する CSV 形式ファイルのヘッダの有無を次の値で指定します。省略時は「nouse」が仮定されます。

use : ヘッダあり

nouse : ヘッダなし

/S 辞書への登録方法

辞書から出力した CSV 形式ファイルを再び入力したり、辞書のデータ項目または業務ルールを CSV 形式ファイル中のデータ項目名や業務ルール名で特定して更新したりする場合に指定します。CSV 形式ファイル中に定義されている辞書の資源（データ項目または業務ルール）を新規に登録するかどうかの判別と、更新する場合の資源の特定方法を指定します。データ項目または業務ルールの定義情報を更新する場合、システム内部 ID で識別するには「use」、データ項目または業務ルール名で識別するには「name」を指定します。元のデータ項目や業務ルールとは別に新しく作成する場合は「nouse」を指定してください。省略時は「use」が仮定されます。

use : CSV 形式ファイル出力コマンドによって出力されたシステム内部 ID で辞書の資源を特定し、既存の資源を更新する

nouse : CSV 形式ファイル出力コマンドによって出力されたシステム内部 ID を使わず、別のシステム内部 ID で資源を新規に登録する

name : CSV 形式ファイル出力コマンドで出力されたシステム内部 ID を使わず、指定したデータ項目名または業務ルール名から辞書の資源を特定し、既存の資源を更新する

/W 更新対象属性

ERwin/ERX からの辞書登録で、データ項目名から特定したデータ項目を更新する場合に、更新対象とする属性を指定します。省略時には「all」が仮定されます。

erwin : ERwin/ERX で定義した情報だけをデータ項目の属性として更新する

all : データ項目のすべての属性を更新する

この引数は、データ項目辞書への入力で、引数「/S 辞書の登録方法」へ「name」を指定し、さらに、引数「/M ファイルフォーマット種別」または「/E ファイルフォーマット種別」で ERwin 専用のフォーマット種別を指定している場合に有効となります。

/R 同一カラム名の属性のマージ有無

ERwin/ERX の定義情報の中に、同一カラム名の属性情報が複数ある場合、それらの情報をマージしてデータ項目辞書に登録するかどうかを指定します。省略時は

「nouse」が仮定されます。

use：同一カラム名の属性をマージする

nouse：同一カラム名の属性をマージしない

/P 継承関連の更新有無

引数「/S 辞書の登録方法」に「name」を指定したときに、CSV形式ファイルの中で継承の最上位に定義されたデータ項目の継承の関連を更新するかどうかを指定します。省略時は「use」が仮定されます。

use：継承の関連を最上位に更新する

nouse：継承の関連を更新しない

なお、引数「/S 辞書の登録方法」に「name」を指定しなかったときには、この引数は指定できません。また、この引数は、データ項目辞書の入力時にだけ有効になります。

/V 更新対象とする辞書フォルダの範囲

データ項目または業務ルールを一括更新する辞書フォルダの範囲を指定します。省略時は「only」が仮定されます。

only：引数「/D 辞書フォルダ名」で指定した辞書フォルダだけを更新対象とする

all：リポジトリのすべての辞書フォルダを更新対象とする

scope：一意性チェックの範囲内の辞書フォルダを更新対象とする

この引数は、引数「/S 辞書の登録方法」に「name」が指定されている場合に有効となります。

/H データ項目の一意性チェックの範囲

環境構築ユティリティで登録してあるデータ項目の一意性チェック範囲名を指定します。指定できる一意性チェック範囲名は、引数「/D 辞書フォルダ名」で指定した辞書フォルダが存在する範囲名です。

この引数は、引数「/V 更新対象とする辞書フォルダの範囲」に「scope」が指定されている場合に有効となります。

/J 業務ルールの一意性チェックの範囲

環境構築ユティリティで登録してある業務ルールの一意性チェック範囲名を指定します。指定できる一意性チェック範囲名は、引数「/D 辞書フォルダ名」で指定した辞書フォルダが存在する範囲名です。

この引数は、引数「/V 更新対象とする辞書フォルダの範囲」に「scope」が指定されている場合に有効となります。

/L ログファイル名²

ログファイルの出力先を指定します。省略時は登録する辞書の種別に応じて次のファイルがログファイルとして作成されます。

パス：環境変数「TEMP」または「TMP」に代入されている値

ファイル名：rpdicsvi.log（データ項目辞書）、rpbrcsvi.log（業務ルール辞書）

4. 辞書情報の一括登録と出力

/?

コマンドの構文と引数の指定方法を表示します。

注 1

新しい辞書フォルダ、データ項目または業務ルールを登録するときに指定します。ここで指定した所有者（ユーザ）名とグループ名は、辞書フォルダ、データ項目または業務ルールに共通に設定されます。引数に指定する名称は、辞書登録するサーバに登録されているユーザ、グループの名称を使用してください。ユーザ、グループの登録についてはマニュアル「SEWB+/REPOSITORY 運用ガイド」を参照してください。なお、この指定では、既存の辞書情報に設定されている所有者およびグループ名は変更できません。

注 2

ログファイルはコマンドの実行ごとに上書きされます。残しておきたい場合は、異なるログファイル名を指定してください。

(c) リターン値

コマンドの実行が終了すると、次のリターン値が返されます。

- 0: 正常終了した
- 1: その他のエラーが発生した
- 2: メモリが確保できない
- 3: 引数が不正である
- 4: ログファイルが作成できない
- 5: 辞書に対するアクセス権がない
- 6: 入力ファイルがない
- 7: 入力ファイルのフォーマットが不正である

4.1.4 CSV 形式ファイルの出力

ここでは、辞書情報を CSV 形式ファイルに出力する場合の出力規則について説明します。

ダイアログでの操作については、ヘルプを参照してください。また、CSV 形式ファイル出力コマンドの指定方法については、「4.1.5 CSV 形式ファイル出力コマンド (RPCsvout) の使い方」を参照してください。

(1) データ項目辞書の出力規則

データ項目辞書を CSV 形式ファイルに出力する規則について説明します。

(a) データ項目の ID の出力

コマンドの ID の出力形式の指定に従って、データ項目の ID が仮システム内部 ID また

はシステム内部 ID の形式で出力されます。CSV 形式ファイル出力コマンドの引数「/S ID の出力形式」、または [CSV 出力オプション] ダイアログの「出力 ID の形式」で指定します。ID の扱い方については、「4.1.2(1)(a) システム内部 ID によるデータ項目の識別」を参照してください。

(b) 関連情報の扱い

データ項目辞書に辞書フォルダにわたる結合項目がある場合の出力規則を次に示します。

- 指定した辞書フォルダのデータ項目辞書内のデータ項目だけを CSV 形式ファイルに出力します。
- ID の形式にシステム内部 ID を使用する場合、指定していない辞書フォルダにある結合先のデータ項目に対しては、CSV 形式ファイルを入力したときに結合関係を復元できるように関連情報だけを出力します。

(c) 国語区分、言語区分、フィールドの指定

出力するデータ項目の定義情報の国語区分、言語区分およびフィールドを指定できます。CSV 形式ファイル出力コマンドの引数「/C 出力国語別」、「/N 出力言語別」、「/I 出力フィールド別」、または [CSV 出力オプション] ダイアログの「出力する辞書情報」で指定します。指定した区分の定義内容だけが、CSV 形式ファイルに出力されます。データ項目の最初の行から、指定した区分に対応する行までを出力するように指定することもできます。

(d) 親データ項目と構成項目の指定

指定によって、親データ項目や構成項目を、システム内部 ID や仮システム ID で出力する代わりに、プリフィックスを付けたデータ項目名で CSV 形式ファイルに出力できます。CSV 形式ファイル出力コマンドの引数「/R 関連するデータ項目の出力形式」、または [CSV 出力オプション] ダイアログの「出力 ID の形式」で指定します。

関連するデータ項目が同一の名称で複数登録されている場合、プリフィックスを付けたデータ項目名で出力した CSV 形式ファイルを、再び入力することはできません。その場合は、データ項目名を一意に変更するか、システム内部 ID または仮システム ID で CSV 形式ファイルに出力してください。

(e) 出力基準とする日時の指定

指定した日時よりもあとに更新・追加したデータ項目だけを CSV 形式ファイルに出力できます。CSV 形式ファイル出力コマンドの引数「/U 出力基準とする日時」、または [CSV 出力オプション] ダイアログの「出力基準日時」で指定します。

(2) 業務ルール辞書の出力規則

業務ルール辞書を CSV 形式ファイルに出力する規則を説明します。

4. 辞書情報の一括登録と出力

(a) 業務ルールの ID の出力

コマンドの ID の出力形式の指定に従って、業務ルールの ID が仮システム ID またはシステム内部 ID の形式で出力されます。CSV 形式ファイル出力コマンドの引数「/S ID の出力形式」、または [CSV 出力オプション] ダイアログの「出力 ID の形式」で指定します。ID については、「4.1.2(2)(a) システム内部 ID による業務ルールの識別」を参照してください。

(b) 関連情報の扱い

業務ルール辞書に、辞書フォルダを渡ってデータ項目と関連する業務ルールがある場合の出力規則を説明します。

- 指定した辞書フォルダの業務ルール辞書内の業務ルールだけを CSV 形式ファイルに出力します。
- 関連するデータ項目については、業務ルールとの関連情報だけを出力します。CSV 形式ファイルを入力したとき、この関連情報を基にして業務ルールとデータ項目の関連が復元されます。

(c) 国語区分、言語区分、フィールドの指定

出力する業務ルールの定義情報の国語区分、言語区分およびフィールドを指定できます。CSV 形式ファイル出力コマンドの引数「/C 出力国語別」、「/N 出力言語別」、「/I 出力フィールド別」、または [CSV 出力オプション] ダイアログの「出力する辞書情報」で指定します。指定した区分の定義情報だけが、CSV 形式ファイルに出力されます。業務ルール定義の最初の行から、指定した区分に対応する行までを出力するように指定することもできます。

(d) 業務ルールに関連するデータ項目の指定

指定によって、業務ルールに関連するデータ項目を、システム内部 ID、またはプリフィックスを付けたデータ項目名で CSV 形式ファイルに出力できます。CSV 形式ファイル出力コマンドの引数「/R 関連するデータ項目の出力形式」または [CSV 出力オプション] ダイアログの「出力 ID の形式」で指定します。

関連するデータ項目が同一の名称で複数登録されている場合、プリフィックスを付けたデータ項目名で出力した CSV 形式ファイルを、再び入力することはできません。その場合は、データ項目をシステム内部 ID で CSV 形式ファイルに出力してください。

(e) 出力基準とする日時

指定した日時よりもあとに更新・追加した業務ルールだけを CSV 形式ファイルに出力できます。CSV 形式ファイル出力コマンドの引数「/U 出力基準とする日時」、または [CSV 出力オプション] ダイアログの「出力基準日時」で指定します。

4.1.5 CSV 形式ファイル出力コマンド (RPCsvout) の使い方

CSV 形式ファイル出力コマンドが格納されているディレクトリを示します。

(1) CSV 形式ファイル出力コマンドが格納されているディレクトリ

(a) SEWB+/REPOSITORY (サーバ) の場合

コマンドは、次のディレクトリに格納されています。

SEWB+ 基本開発環境の組み込み先パス名¥Repository¥bin

SEWB+ 基本開発環境の組み込み先パス名は、デフォルトでは次のように設定されています。システムドライブとは、Windows 2000 Server, Windows Server 2003 または Windows Server 2003 x64 が組み込まれているドライブです。

組み込み先パス名

- Windows 2000 Server または Windows Server 2003 の場合
システムドライブ : ¥Program Files¥HITACHI¥Sewb
- Windows Server 2003 x64 の場合
システムドライブ : ¥Program Files (x86)¥HITACHI¥Sewb

(b) SEWB+/REPOSITORY-BROWSER (クライアント) の場合

コマンドは、次のディレクトリに格納されています。

SEWB+ 基本開発環境の組み込み先パス名¥Repository-Browser

SEWB+ 基本開発環境の組み込み先パス名は、デフォルトでは次のように設定されています。システムドライブとは、Windows が組み込まれているドライブです。

組み込み先パス名

- Windows 2000, Windows XP, Windows Server 2003 または Windows Vista の場合
システムドライブ : ¥Program Files¥HITACHI¥Sewb
- Windows Server 2003 x64 の場合
システムドライブ : ¥Program Files (x86)¥HITACHI¥Sewb

(2) コマンド実行時の注意事項

(a) コマンドを実行する前に

サーバでコマンドを実行するときは、サーバでは SEWB+/REPOSITORY からログアウトしておいてください。

(b) コマンド実行中にほかの作業をする場合

CSV 形式ファイル出力コマンドでは、実行中に排他制御を行っていません。したがって、

4. 辞書情報の一括登録と出力

コマンドの実行中でも並行してほかの作業をすることができます。ただし、次に示す点に注意してください。

- コマンドの実行中に対象となっている資源や関連情報を更新または削除してはなりません。不整合が検出され、正常に終了しない場合があります。この場合には、CSV形式ファイル出力コマンドの再実行が必要です。
- コマンドの実行中に、CSV形式ファイルの出力先で出力の対象となっている資源を更新してはなりません。CSV形式ファイル出力コマンドが正常に終了しても、その更新内容がCSV形式ファイルに反映されていない場合があります。

(3) CSV形式ファイル出力コマンド (RPCsvout) の指定

サーバのコマンドラインに、次に示す構文に従ってコマンドと引数を指定して実行します。サーバ上では、リポジトリのシステム管理者の権限で実行します。起動時に表示されるダイアログに、システム管理者のパスワードを入力する必要があります。クライアント上では、SEWB+ ログインに指定されたユーザの権限で実行します。起動時に表示されるダイアログに、ユーザ名、ログイン先およびユーザのパスワードを入力する必要があります。

実行中には、データ項目または業務ルールの出力件数が表示されます。また、詳細情報がログファイルに出力されます。

(a) 形式

```
RPCsvout /F 出力 CSV 形式ファイル名 /D 辞書フォルダ名  
  [ /K 辞書種別 ] [ /C 出力国語別 ] [ /N 出力言語別 ]  
  [ /I 出力フィールド別 ]  
  [ /T ヘッダの有無 ]  
  [ /S ID の出力形式 ]  
  [ /R 関連するデータ項目の出力形式 ]  
  [ /U 出力基準とする日時 ]  
  [ /L ログファイル名 ]  
  [ /? ]
```

(b) 引数

/F 出力 CSV 形式ファイル名
出力先の CSV 形式ファイル名を指定します。既存のファイルを指定した場合は、コマンド実行後に表示されるメッセージで上書きするかどうかを確認してください。

/D 辞書フォルダ名
出力する辞書情報が格納されている辞書フォルダ名を指定します。

/K 辞書種別
出力する辞書の種別を指定します。省略時は「DI」が仮定されます。
DI: データ項目辞書

BR：業務ルール辞書

/C 出力国語別

出力する辞書項目の国語区分を指定します。環境構築ユティリティで設定した国語区分の順序に対応して 1 ~ 4 の数字で指定します。値の前に「:」を付けると、最初の行から指定した国語区分までが出力されます。省略時はすべての国語別情報が、国語別の登録順に CSV 形式ファイルへ出力されます。

/N 出力言語別

出力する辞書項目の言語区分を指定します。環境構築ユティリティで設定した言語区分の順序に対応して 1 ~ 10 の数字で指定します。値の前に「:」を付けると、最初の行から指定した言語区分までが出力されます。省略時はすべての言語別情報が、言語別の登録順に CSV 形式ファイルへ出力されます。

/I 出力フィールド別

出力するフィールドを指定します。環境構築ユティリティで設定したフィールドの順序に対応して、データ項目の場合は 1 ~ 20、業務ルールの場合は 1 ~ 3 の数字で指定します。値の前に「:」を付けると、最初の行から指定したフィールドまでが出力されます。省略時はすべてのフィールドの情報が、フィールド別の登録順に CSV 形式ファイルへ出力されます。

/T ヘッダの有無

CSV 形式ファイルに、ヘッダを出力するかどうかを次の値で指定します。省略時は「nouse」が仮定されます。

use：ヘッダあり

nouse：ヘッダなし

/S ID の出力形式

CSV 形式ファイル中のデータ項目および業務ルールの ID の出力形式を次の値で指定します。省略時は「nouse」が仮定されます。

use：システム内部 ID を使用して出力する

nouse：システム内部 ID を使用しないで、仮システム内部 ID にして出力する

/R 関連するデータ項目の出力形式

CSV 形式ファイルに、関連するデータ項目をデータ項目の ID で出力するか、プリフィックス付きデータ項目名で出力するかを次の値で指定します。省略時は「id」が仮定されます。

id：関連するデータ項目をデータ項目の ID で出力する。

name：関連するデータ項目をプリフィックス付きデータ項目名で出力する。

「id」を指定した場合、業務ルールの CSV 形式ファイル中の関連データ項目 ID は、引数「/S ID の出力形式」の指定に関係なく、常にシステム内部 ID で出力されます。

/U 出力基準とする日時

4. 辞書情報の一括登録と出力

指定した日時よりもあとに更新・追加したデータ項目または業務ルールだけを CSV 形式ファイルに出力します。コマンドの引数に出力基準とする日時を指定してください。

省略時は、すべてのデータ項目または業務ルールを CSV 形式ファイルに出力します。日付を指定して、時刻を省略した場合、時刻は「00:00:00」と仮定されません。

出力基準とする日時は、次の形式で指定してください。

- YYYY/MM/DD [hh:mm:ss][TYPE]

は半角スペースを示します。[] 内は省略できます。

YYYY/MM/DD は西暦年月日を示します。hh:mm:ss は時刻を示します。

TYPE は、指定した日時よりもあとに更新・追加したもののうち、どのデータ項目または業務ルールを CSV 形式ファイルに出力するかを次の値で指定します。省略時は、「all」が仮定されます。

all：更新・追加したすべてのデータ項目または業務ルール

up：更新したデータ項目または業務ルールだけ

new：追加したデータ項目または業務ルールだけ

ル ログファイル名

ログファイルの出力先を指定します。省略時は、出力する辞書の種別に応じて次のファイルがログファイルとして作成されます。

パス：環境変数「TEMP」または「TMP」に代入されている値

ファイル名：rpdicsvo.log (データ項目辞書), rpbrcsvo.log (業務ルール辞書)

/?

コマンドの構文と引数の指定方法を表示します。

注

ログファイルはコマンドの実行ごとに上書きされます。残しておきたい場合は、異なるログファイル名を指定してください。

(c) リターン値

コマンドの実行が終了すると、次のリターン値が返されます。

- 0：正常終了した
- 1：その他のエラーが発生した
- 2：メモリが確保できない
- 3：引数が不正である
- 4：ログファイルが作成できない
- 5：辞書に対するアクセス権がない
- 6：辞書フォルダがない
- 7：出力ファイルが作成できない

4.1.6 CSV 形式の記述規則

ここでは、辞書情報を CSV 形式で表すための共通の記述規則について説明します。

(1) レコードの記述

各定義項目に対応する値は、数値または文字列で記述します。値は区切り文字となるコンマ(,)で区切ります。空白やタブは区切り文字とみなされません。

「,,」のように区切り文字を連続して記述すると、その中の値は省略されたものとみなされます。この場合、対応する定義項目には数値または文字列の初期値があてはめられます。数値または文字列のどちらが省略されたのかは、対応する定義項目の属性によって判断されます。詳細は、「(4) 定義項目の初期値」を参照してください。

(2) 数値の記述

整数および実数の記述規則を次に説明します。

(a) 整数

次の形式で整数を表記します。

Snnnn

- S は符号 (+ または -) を示します。ただし + の場合は省略できます。
- nnnn は仮数部を示します。整数値で表記します。

(b) 実数

次の形式で実数を表記します。

SnnnnESnn

- S は符号 (+ または -) を示します。ただし + の場合は省略できます。
- nnnn は仮数部を示します。整数値で表記します。小数点を含めて表記することもできます。
- E は指数部の始まりを示します。「E」または「e」で表記します。
- nn は指数部を示します。整数値で表記します。

(3) 文字列の記述

文字列の記述には、ダブルクォーテーション(")で囲む方法と、ダブルクォーテーションで囲まないでそのまま記述する方法の2通りがあります。

文字列をダブルクォーテーションで囲む方法は、標準フォーマットの記述に使用します。また、ダブルクォーテーションで囲まない方法は、標準フォーマットのほかに、ERwin/ERX などのほかのアプリケーションと連携するためのフォーマットの記述に使用します。フォーマットの詳細は、「4.1.7 CSV 形式ファイルフォーマット」を参照してください。

4. 辞書情報の一括登録と出力

(a) ダブルクォーテーションで囲む文字列

次の方法で文字列を記述します。全角文字を記述する場合には、シフト JIS 標準漢字コードを使用します。

- 文字列中のダブルクォーテーションは「\"」と記述します。
- ダブルクォーテーションで囲まれた文字列はすべて値となります。したがって、コンマ(,)や空白文字も値とみなされます。
- 文字列を囲むダブルクォーテーションは、値とみなされません。

(b) ダブルクォーテーションで囲まない文字列

次の方法で文字列を記述します。全角文字を記述する場合には、シフト JIS 標準漢字コードを使用します。

- 区切り文字の間のすべての文字が値とみなされます。
- コンマ(,)を文字列の値として記述することはできません。
- 区切り文字であるコンマ以外の文字はすべて値となります。したがって、アポストロフィ(')や空白文字も値とみなされます。

(c) 共通規則

コメントおよびルールスクリプトの値には、文字列中に改行とタブを記述できます。このほかの定義項目に改行とタブを記述することはできません。

- 改行の記述
改行は、識別子「¥n」で表します。CSV 形式ファイルの入力時に「¥n」が改行に変換され、出力時には改行が「¥n」に変換されます。「¥n」自体を文字列として記述する場合には¥の記述に従って「¥¥n」を使用します。
- タブの記述
タブは、識別子「¥t」で表します。CSV 形式ファイルの入力時に「¥t」がタブに変換され、出力時にはタブが「¥t」に変換されます。「¥t」自体を文字列として記述する場合には¥の記述に従って「¥¥t」を使用します。
- ¥の記述

「¥」は、改行やタブを記述するために特別な扱いをしています。このため、CSV 形式ファイルの入力時には「¥¥」が「¥」に変換され、出力時には「¥」が「¥¥」に変換されます。

(4) 定義項目の初期値

区切り文字を連続して記述し、値を省略した場合の初期値を次に示します。

(a) データ項目の初期値

データ項目の定義項目、アクセス権および関連情報の初期値を、表 4-1 に示します。

表 4-1 データ項目の初期値

	項目	初期値
定義項目	データ項目名	- ¹
	標準名称	長さ 0 の文字列 (NULL)
	フリガナ	長さ 0 の文字列 (NULL)
	分類	「分類なし」 ²
	けた数	特定の値は設定されない
	小数部けた数	特定の値は設定されない
	反復回数	特定の値は設定されない
	最上位結合項目フラグ	最上位結合項目ではない
	フィールド 1 ~ 20	長さ 0 の文字列 (NULL)
	コメント	長さ 0 の文字列 (NULL)
	名前	長さ 0 の文字列 (NULL)
	タイプ	「フリー定義」(長さ 0 の文字列 (NULL))
	タイプ修飾情報 ³	長さ 0 の文字列 (NULL)
	取りうる値	長さ 0 の文字列 (NULL)
	初期値	長さ 0 の文字列 (NULL)
言語別フィールド	長さ 0 の文字列 (NULL)	
セキュリティ情報	所有者名	ログインしたユーザまたは指定した所有者 ⁴
	グループ名	システム管理者でログインしている場合は “everyone” が、システム管理者以外の場合は ログインしたユーザが所属するグループまたは 指定したグループ ⁵
	アクセス権	アクセス権の初期値 ⁶
関連情報	親データ項目	辞書フォルダを親とする
	構成項目	構成項目なし

注 1
必ず定義しなければならないため、初期値はありません。

注 2
結合項目の場合は「結合データ」となります。

注 3
言語区分「COBOL 又は OOCOBOL」で定義する編集文字列，言語区分「IDL(CORBA)」で定義するシーケンスを指します。

注 4
CSV 形式ファイル入力コマンドの引数，または [CSV 入力オプション] ダイアログで指定した所有者名です。

4. 辞書情報の一括登録と出力

注 5

CSV 形式ファイル入力コマンドの引数, または [CSV 入力オプション] ダイアログで指定したグループ名です。

注 6

初期値については, 「3.2.1 アクセス権の設定」を参照してください。

(b) 業務ルールの初期値

業務ルールの定義項目, アクセス権および関連情報の初期値を, 表 4-2 に示します。

表 4-2 業務ルールの初期値

	項目	初期値
定義項目	業務ルール名	なし ¹
	標準名称	長さ 0 の文字列 (NULL)
	フリガナ	長さ 0 の文字列 (NULL)
	フィールド 1 ~ 3	長さ 0 の文字列 (NULL)
	共用情報	「共用する」
	キーワード	キーワードなし
	適用条件	適用条件なし
	関連データ項目	関連データ項目なし
	ルールスクリプト	長さ 0 の文字列 (NULL)
	コメント	長さ 0 の文字列 (NULL)
セキュリティ情報	所有者名	ログインしたユーザ, または指定した所有者 ²
	グループ名	システム管理者でログインしている場合は “everyone” が, システム管理者以外の場合はログインしたユーザが所属するグループまたは指定したグループ ³
	アクセス権	アクセス権の初期値 ⁴

注 1

必ず定義しなければならないため, 初期値はありません。

注 2

CSV 形式ファイル入力コマンドの引数, または [CSV 入力オプション] ダイアログで指定した所有者名です。

注 3

CSV 形式ファイル入力コマンドの引数, または [CSV 入力オプション] ダイアログで指定したグループ名です。

注 4

初期値については, 「3.2.1 アクセス権の設定」を参照してください。

4.1.7 CSV 形式ファイルフォーマット

CSV 形式ファイルのフォーマットは、用途に応じて、標準フォーマットとアプリケーション専用フォーマットの 2 種類に分類されます。標準フォーマットは通常、SEWB+/REPOSITORY の辞書情報を入出力するために使用します。また、アプリケーション専用フォーマットは、SEWB+/REPOSITORY の辞書情報をほかのアプリケーションに連携するために使用します。

ここでは、基本的なファイル内部の構成と、標準フォーマットの詳細について説明します。アプリケーション専用フォーマット (ERwin/ERX) の詳細は、「付録 B ERwin/ERX からの辞書登録」を参照してください。

(1) ファイル内部の構成

基本的なファイル内部の構成を次に示します。

ヘッダ
定義情報1
⋮
定義情報N

先頭の 1 行目には、ヘッダがあります。ヘッダには、定義項目の名称を記述してその CSV 形式ファイルにどのような定義情報があるのかを示す役割があります。ヘッダは、標準フォーマット (データ項目辞書)、アプリケーション専用フォーマットで記述内容が異なります。なお、コマンド実行時の引数の指定によって、必要に応じ、ヘッダの入出力を抑止することもできます。

2 行目以降からは、定義情報を記述します。定義情報の記述内容も標準フォーマット (データ項目辞書)、アプリケーション専用フォーマットで異なります。

(2) データ項目辞書の標準フォーマット

データ項目辞書のファイル内部の構成を次に示します。

ヘッダ
データ項目1 定義情報1
⋮
データ項目1 定義情報m
⋮
データ項目n 定義情報1
⋮
データ項目n 定義情報m
⋮

一つのデータ項目の定義情報を内容に応じて1から複数行で記述します

4. 辞書情報の一括登録と出力

1行目にはヘッダを，2行目からは個々のデータ項目の定義情報を記述します。一つのデータ項目に対応する定義情報は，内容に応じて1から複数行で記述します。国語区分，言語区分，フィールドおよび結合項目の関連の数のように，一つのデータ項目で複数の定義内容を持つ場合は，複数行になります。

(a) データ項目のヘッダの記述内容

データ項目のヘッダの記述内容を，表 4-3 に示します。CSV 形式ファイル入力コマンドでヘッダの入力を指定した場合には，必ずヘッダが記述されていなければなりません。

表 4-3 データ項目のヘッダの記述内容

記述順	項目名	ヘッダの記述内容
1	データ項目名継承フラグ	DataItemNameIFlg
2	データ項目名	DataItemName
3	標準名称継承フラグ	StdNameIFlg
4	標準名称	StdName
5	フリガナ継承フラグ	SymbolNameIFlg
6	フリガナ	SymbolName
7	分類継承フラグ	TypeIFlg
8	分類	Type
9	けた数継承フラグ	DigitIFlg
10	けた数	Digit
11	小数部けた数継承フラグ	FloatDigitIFlg
12	小数部けた数項目名	FloatDigit
13	反復回数継承フラグ	RepeatNumberIFlg
14	反復回数	RepeatNumber
15	フィールド継承フラグ	FieldIFlg
16	フィールド	Field
17	名前継承フラグ	SourceNameIFlg
18	名前	SourceName
19	タイプ継承フラグ	SourceTypeIFlg
20	タイプ	SourceType
21	タイプ修飾情報継承フラグ	TypeModifyInfoIFlg
22	タイプ修飾情報	TypeModifyInfo
23	初期値継承フラグ	InitValueIFlg
24	初期値項目名	InitValue
25	言語別フィールド継承フラグ	SourceFieldIFlg
26	言語別フィールド	SourceField
27	最上位結合項目フラグ	TopCompositeFlg

記述順	項目名	ヘッダの記述内容
28	所有者	Owner
29	グループ	Group
30	アクセス権	AccessRight
31	データ項目 ID	DataItemID
32	親データ項目 ID	PrntDataItemID
33	構成項目 ID	ElmtDataItemID
34	コメント	Comment
35	取りうる値	AvailableValue

注

この表の記述順に従って記述してください。

(b) データ項目の定義情報の記述内容

データ項目の定義項目に対応する値を記述するためのフォーマットを、表 4-4 に示します。国語区分、言語区分、フィールドおよび結合項目の構成項目のように、一つのデータ項目で複数の定義内容を持つ場合は、最大の個数の数だけ、複数行に渡って同じ列に値を記述します。

CSV 形式ファイルを入力する場合、データ項目名およびデータ項目名継承フラグには必ず値が記述されていなければなりません。ほかの定義項目の値の記述は任意です。なお、一つのデータ項目を複数行に渡って記述する場合は、開始行にだけデータ項目名とデータ項目名継承フラグを記述します。2 行目からは、この値は記述してはなりません。

表 4-4 データ項目の定義情報の記述内容

記述順	記述する定義情報	記述順	記述する定義情報
1	データ項目名継承フラグ ¹ ²	18	名前
2	データ項目名 ¹	19	タイプ継承フラグ ²
3	標準名称継承フラグ ²	20	タイプ ³
4	標準名称	21	タイプ修飾情報継承フラグ ² ⁴
5	フリガナ継承フラグ ²	22	タイプ修飾情報 ⁴
6	フリガナ	23	初期値継承フラグ ²
7	分類継承フラグ ²	24	初期値
8	分類	25	言語別フィールド継承フラグ ²
9	けた数継承フラグ ²	26	言語別フィールド
10	けた数	27	最上位結合項目フラグ ⁵
11	小数部けた数継承フラグ ²	28	所有者 ⁶

4. 辞書情報の一括登録と出力

記述順	記述する定義情報	記述順	記述する定義情報
12	小数部けた数	29	グループ ⁶
13	反復回数継承フラグ ²	30	アクセス権 ⁷
14	反復回数	31	データ項目 ID ⁸
15	フィールド継承フラグ ²	32	親データ項目 ID ^{8 9}
16	フィールド	33	構成項目 ID ^{8 9}
17	名前継承フラグ ²	34	コメント ¹⁰
		35	取りうる値 ¹¹

注

この表の記述順に従って記述してください。

注 1

CSV 形式ファイルを入力するときは、必ず値を記述します。一つのデータ項目を複数行に渡って記述する場合は、開始行にだけ記述します。

注 2

「データ項目の継承フラグ」を参照してください。

注 3

タイプの英字の識別子を記述します。識別子の前後に空白を記述してはなりません。複数のキーワードで構成される識別子（例：unsigned int）は、半角の空白 1 文字で区切ります。タイプの英字の識別子以外の文字が指定された場合は、フリー定義とみなされます。

注 4

言語区分「COBOL 又は OOCOBOL」で定義する編集文字列、言語区分「IDL(CORBA)」で定義するシーケンスを指します。

注 5

最上位結合項目であることを示す「1」、または最上位結合項目ではないことを示す「0」を指定します。CSV 形式ファイルの入力時には、初期値として「0」が設定されます。

注 6

サーバに登録されている所有者（ユーザ）名、グループ名を記述します。

注 7

「データ項目のアクセス権」を参照してください。

注 8

「データ項目の ID」を参照してください。

注 9

「データ項目名の指定」を参照してください。

注 10

文字列の途中に改行またはタブを記述できます。

注 11

「取りうる値の指定」を参照してください。

データ項目の ID

各 ID の内容を次に示します。

- データ項目 ID ...データ項目を一意に識別する ID
- 親データ項目 ID ...そのデータ項目が定義情報を継承している上位データ項目のデータ項目 ID
- 構成項目 ID ...その結合項目を構成している構成項目のデータ項目 ID

注

構成項目は空白行をはさまないで連続して記述してください。上位データ項目と同じにする場合はすべて省略してください。

ID の値は、次の 2 種類の形式で記述します。

- 仮システム内部 ID :「0x」で始まる最大 16 けたの 16 進の文字列。
- システム内部 ID :「&」で始まる英数字と記号から成る文字列。SEWB+/REPOSITORY が割り当てたシステム内部 ID を文字列に変換した値。

データ項目名の指定

親データ項目、構成項目にはデータ項目名を指定できます。指定形式を次に示します。

* [辞書フォルダ名:] データ項目名

[] 内は省略できます。データ項目名は省略できません。

記述する文字列に「¥」、「*」、「?」、「,」、「;」、「/」、「<」、「>」および半角空白を含む場合は、「¥」に続けて各文字を記述してください。

構成項目を定義している辞書フォルダを指定する場合は、データ項目名の直前に必ず辞書フォルダ名を記述してください。辞書フォルダ名とデータ項目名は「:」で区切ってください。

辞書フォルダ名を含めたデータ項目名の指定の直前には、「*」を付加します。CSV 形式ファイルの入力時、辞書フォルダ名を省略した親データ項目および構成項目は、全辞書フォルダを対象として一括更新する場合、全辞書フォルダから特定されます。全辞書フォルダを対象として一括更新する場合以外は、親データ項目に辞書フォルダ名を指定してはなりません。

データ項目の継承フラグ

継承フラグは、各定義項目の値を上位データ項目から引き継ぐかどうかを示します。継承フラグには、次の値を指定します。

- 1: 上位データ項目から値を継承する
- 0: そのデータ項目で定義する値を示す

• 継承モードとの関係

継承フラグは、その定義項目の継承モードに「任意継承」を設定している場合だけ有効になります。継承モードが「必ず継承」の場合は継承フラグの値に関係なく「1」が設定されたものとみなされます。継承モードが「継承禁止」の場合は継承フラグの値に関係なく「0」が設定されたものとみなされます。

• 初期値

継承フラグが有効なとき、CSV 形式ファイルの入力時には初期値として「1」が

4. 辞書情報の一括登録と出力

設定されます。ただし、最上位データ項目には「0」が設定されます。

注 CSV形式ファイルの入力時、継承関係の下位のデータ項目に指定した値のうち、継承フラグに「1」が指定されているときの定義情報の値は無視されます。

データ項目のアクセス権

アクセス権は、次に示す形式で記述します。

所有者のアクセス権：グループのアクセス権：すべてのユーザのアクセス権

それぞれのアクセス権は、次に示す値の組み合わせで設定します。

r：読み取り権 w：書き込み権 d：削除権

(例)「rwd:rwr:r」は、次に示すアクセス権の設定を表します。

所有者...読み取り権，書き込み権，削除権

グループ...読み取り権，書き込み権

すべてのユーザ...読み取り権

取りうる値の指定

取りうる値には、記号だけ、または記号と値を「=」で結んだ記号の定義を定義数分繰り返した文字列を記述します。

繰り返す記号の定義は「,」で区切り、定義文字列全体は「"」で囲みます。なお、記号には、半角の空白、「,」および「=」を記述できません。ただし、記号の前後にある半角空白または全角空白は無視されます。

値の記述に「,」、「=」、「¥」を含む場合は、「¥」に続けて各文字が記述されていなければなりません。

(例) "NAGOYA=3" "OK=1,NG"

(3) 業務ルール辞書の標準フォーマット

業務ルール辞書のファイル内部の構成を次に示します。

ヘッダ		
業務ルール1 定義情報1	一つの業務ルールの定義情報を内容に応じて1から複数行で記述します	
:		
業務ルール1 定義情報m		
:		
業務ルールn 定義情報1		
:		
業務ルールn 定義情報m		
:		

1行目にはヘッダを、2行目からは個々の業務ルールの定義情報を記述します。一つのデータ項目に対応する定義情報は、内容に応じて1から複数行で記述します。国語区分、

言語区分、フィールドおよび業務ルールに関連するデータ項目のように、一つの業務ルールで複数の定義内容を持つ場合は、複数行になります。例えば、1行目の標準名称の定義情報は国語区分「日本語」、3行目のルールスクリプトの定義情報は言語区分「SQL」とみなされます。ある特定の区分だけを入力する時、CSV形式ファイルコマンドの引数に指定する数字も定義情報の定義行に対応していますので、注意してください。

(a) 業務ルールのヘッダの記述内容

業務ルールのヘッダの記述内容を、表 4-5 に示します。CSV形式ファイル入力時にヘッダの入力を指定した場合には、必ずヘッダが記述されていなければなりません。

表 4-5 業務ルールのヘッダの記述内容

記述順	項目名	ヘッダの記述内容
1	業務ルール名	BusiRuleName
2	標準名称	StdName
3	フリガナ	SymbolName
4	フィールド	Field
5	所有者	Owner
6	グループ	Group
7	アクセス権	AccessRight
8	業務ルール ID	BusiRuleID
9	共用情報	ShareInfo
10	キーワード	Keyword
11	適用条件	ApplyCond
12	関連データ項目 ID	ReltnDataItemID
13	ルールスクリプト	RuleScript
14	コメント	Comment

注

この表の記述順に従って記述してください。

(b) 業務ルールの定義情報の記述内容

業務ルールの定義項目に対応する値を記述するためのフォーマットを、表 4-6 に示します。国語区分、言語区分、フィールドおよび業務ルールに関連するデータ項目のように、一つの業務ルールで複数の定義内容を持つ場合は、最大の個数の数だけ、複数行に渡って同じ列に値を記述します。

表 4-6 業務ルールの定義情報の記述内容

記述順	記述する定義情報	記述順	記述する定義情報
1	業務ルール名 ¹	8	業務ルール ID ⁴

4. 辞書情報の一括登録と出力

記述順	記述する定義情報	記述順	記述する定義情報
2	標準名称	9	共用情報 ⁵
3	フリガナ	10	キーワード
4	フィールド	11	適用条件 ⁶
5	所有者 ²	12	関連データ項目 ID ^{7, 8}
6	グループ ²	13	ルールスクリプト ⁹
7	アクセス権 ³	14	コメント ⁹

注

この表の記述順に従って記述してください。

注 1

CSV形式ファイルを入力するときには、必ず値を記述します。一つの業務ルールを複数行に渡って記述する場合は、開始行にだけ記述します。

注 2

サーバに登録されている所有者（ユーザ）名、グループ名を記述します。

注 3

アクセス権の設定方法は、データ項目と同様です。「(2)(b) データ項目の定義情報の記述内容」のデータ項目のアクセス権の説明を参照してください。

注 4

「業務ルールの ID」を参照してください。

注 5

データ項目の適用形態が「共用する」の場合は何も記述しません。「専用にする」の場合は、すべての関連データ項目 ID を記述する行に「M」を記述します。同一項目用業務ルールでは、「専用にする」を示す「M」を記述します。

注 6

適用条件が入力の場合は「I」、出力の場合は「O」を記述します。

注 7

業務ルールに関連するデータ項目のシステム内部 ID を記述します。ただし、同一項目用業務ルールの場合は、「すべての単項目」を関連づけるときは「?S」、「すべての項目」を関連づけるときは「?A」を記述します。同一項目用業務ルールでは関連データ項目を 2 行に記述します。この 2 行はどちらも同じ形式で記述しなければなりません。

注 8

「関連するデータ項目の ID」を参照してください。

注 9

文字列の途中で改行またはタブを記述できます。

業務ルールの ID

ID の値は、次の 2 種類の形式で記述します。

- 仮システム内部 ID : 「0x」で始まる最大 16 けたの 16 進の文字列。
- システム内部 ID : 「&」で始まる英数字と記号から成る文字列。SEWB+/

REPOSITORY が割り当てたシステム内部 ID を文字列に変換した値。

関連するデータ項目の ID

関連するデータ項目の ID はシステム内部 ID と同じ形式、または次の形式で記述します。

* [辞書フォルダ名:] データ項目名

[] 内は省略できます。データ項目名は省略できません。リポジトリに登録されているデータ項目を識別するデータ項目名を記述してください。

記述する文字列に「¥」、「*」、「?」、「,」、「;」、「/」、「<」、「>」および半角空白を含む場合は、「¥」に続けて各文字を記述してください。

関連データ項目を定義している辞書フォルダを指定する場合は、データ項目名の直前に必ず辞書フォルダ名を記述してください。辞書フォルダ名とデータ項目名は「:|」で区切ってください。

また、指定する関連データ項目が入出力対象の辞書フォルダに定義されている場合は、辞書フォルダ名の指定は必要ありません。

4.2 データベースから辞書のレコード定義（最上位結合項目）を生成する

ORACLE などの ODBC インタフェース機能をサポートしているデータベースから表と列の情報を取り出して、結合項目および構成項目を生成できます。この機能はコマンドとして用意され（ODBC 入力コマンドといいます）、サーバでシステム管理者だけが利用できます。

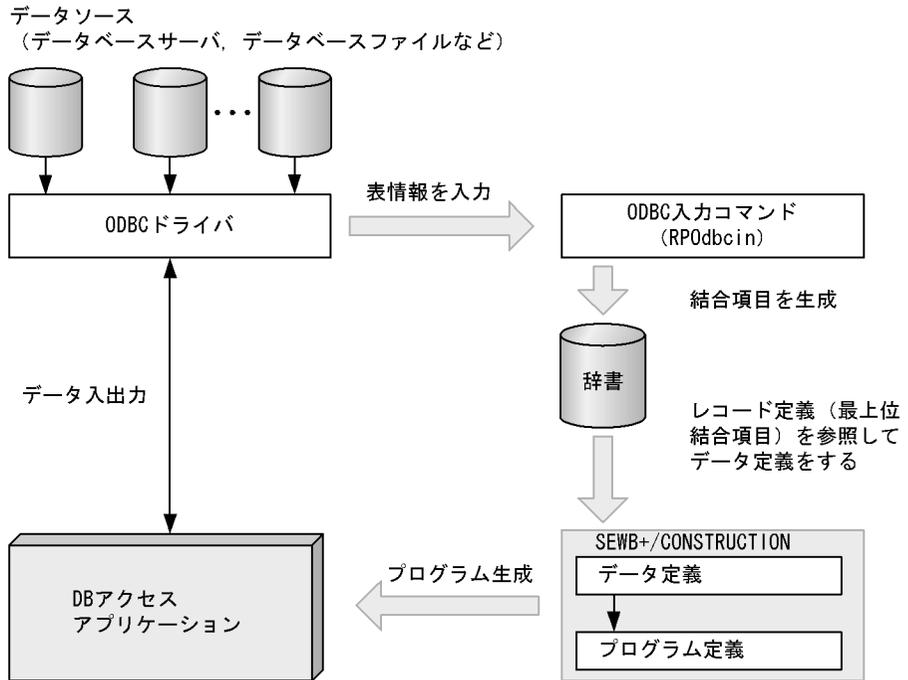
ODBC 入力コマンドを使って生成された結合項目は、既存データベースのレコード構造を反映しています。したがって、SEWB+/CONSTRUCTION のデータ定義で、辞書のレコード定義（最上位結合項目）を利用すれば、既存データベースにアクセスするアプリケーションの開発効率を向上できます。

4.2.1 データベースの表および列情報の入力

ODBC 入力コマンドは、各種データベースによって生成されたデータソースの中から表に関する情報を取得して、これを基に結合項目と構成項目を生成します。データ項目辞書に生成された結合項目のうち、最上位結合項目は、SEWB+/CONSTRUCTION から、辞書のレコード定義（最上位結合項目）として扱われます。したがって、SEWB+/CONSTRUCTION のデータ定義で、この最上位結合項目（つまり、生成元データベースのレコード定義）を利用すれば、このデータソースにアクセスするためのプログラムを生成できます。

ODBC 入力コマンドを使いデータベースから表情報を入力して、SEWB+/CONSTRUCTION でプログラムを生成するまでのデータの流れを、図 4-4 に示します。

図 4-4 データベースからのデータの流れ



- データソースとは
ユーザがアクセスするデータとそのデータを取得するために必要な情報です。例えば、ORACLE のデータベース、ORACLE データベースが保存されているサーバ、そのサーバへアクセスするために使われるネットワークがデータソースになります。
- ODBC 入力コマンドを使う前に
 - あらかじめ ODBC ドライバの組み込みと、データソースの追加およびセットアップが必要です。ODBC ドライバの組み込みと、データソースの追加およびセットアップには、ODBC コントロールパネルを使用します。
 - ODBC タイプを SEWB+/REPOSITORY の分類区分やタイプなどに変換するためのマッピング規則を、環境構築ユーティリティでカスタマイズします。ただし、標準で用意されている規則を適用する場合はカスタマイズは必要ありません。

4.2.2 結合項目と構成項目の生成規則

ここでは、データベースの表情情から、どのように結合項目と構成項目が生成されるのかについて説明します。

(1) 結合項目と構成項目の生成例

データソース「TESTDATA」を例に、使用するデータベースが ORACLE の場合、どのように結合項目と構成項目が生成されるのかについて説明します。データソース「TESTDATA」は表「NEWTABLE」を持ち、表「NEWTABLE」は「City」および

4. 辞書情報の一括登録と出力

「State」という列を持ちます。このデータソースの表から求められる列情報の例を、表 4-7 に示します。

表 4-7 データソースの表から求められる列情報の例

列情報項目	列情報 1	列情報 2
表名 (TABLE_NAME)	NEWTABLE	NEWTABLE
列名 (COLUMN_NAME)	City	State
データ型 (DATA_TYPE)	SQL_CHAR	SQL_CHAR
精度 (PRECISION)	10	2

この例から生成されるデータ項目の関係を図 4-5 に、データ項目の定義内容を表 4-8 に示します。

図 4-5 生成されるデータ項目間の関係

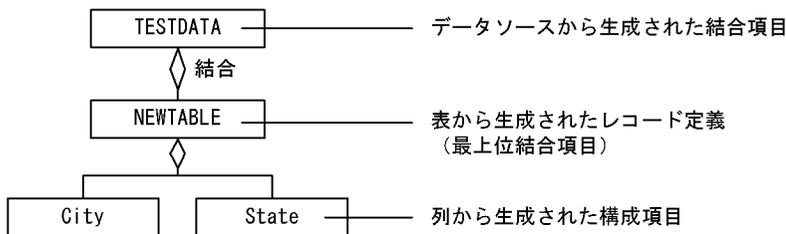


表 4-8 生成されるデータ項目の定義内容の例

定義項目	生成される定義内容			
	データソース	NEWTABLE (表)	City (列)	State (列)
データ項目名	TESTDATA	NEWTABLE	City	State
データ項目種別	結合項目	結合項目	単項目	単項目
分類	結合データ	結合データ	英数字文字列データ	英数字文字列データ
けた数	-	-	10	2
名前 (言語共通)	TESTDATA	NEWTABLE	City	State
タイプ	COBOL 又は OOCOBOL	-	-	X
	C 又は C++	-	-	char
	SQL	-	-	CHAR
	IDL(CORBA)	-	-	char
	帳票定義	-	-	C
	Java	-	-	char

定義項目	生成される定義内容			
	データソース	NEWTABLE (表)	City (列)	State (列)
言語 4 ~ 7	-	-	フリー定義	フリー定義

生成されるデータ項目の詳細な定義内容の例は、CSV 形式ファイルで提供されています。このサンプルの提供先を次に示します。

サーバのSEWB+ 基本開発環境の組み込み先パス名¥Repository¥sample¥odbcin.csv

なお、組み込み先パス名は、デフォルトでは次のように設定されています。システムドライブとは、Windows 2000 Server, Windows Server 2003 または Windows Server 2003 x64 が組み込まれているドライブです。

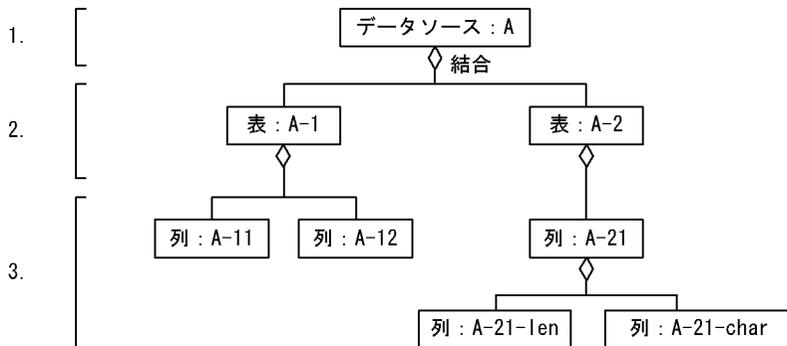
- Windows 2000 Server または Windows Server 2003 の場合
システムドライブ : ¥Program Files¥HITACHI¥Sewb
- Windows Server 2003 x64 の場合
システムドライブ : ¥Program Files(x86)¥HITACHI¥Sewb

(2) 生成規則

(a) 生成の概要

ODBC 入力コマンドを使用すると、生成元のデータソースから結合項目の構成が生成され、データ項目辞書に格納されます。生成されるデータ項目の関係の例を、図 4-6 に示します。

図 4-6 データソースから生成されるデータ項目の関係



データソースから生成されるデータ項目は、次の 3 種類に分類されます。

1. データソースに対応するデータ項目
結合項目となります。一つのデータソースに対し、一つだけ生成されます。2. のデータ項目が、この結合項目の構成項目になります。
2. 表に対応するデータ項目

4. 辞書情報の一括登録と出力

結合項目（最上位結合項目）となります。データソース中の表に対応して、複数生成されます。3.のデータ項目が、この結合項目の構成項目になります。

3. 列に対応するデータ項目

単項目¹または結合項目となります。データ項目種別（単項目または結合項目）は、列のSQLデータ型²によって異なります。

注 1

単項目とは、構成項目を持たないデータ項目のことです。

注 2

ODBC SQL データ型（SEWB+/REPOSITORY では「ODBC タイプ」と呼びます）とは、ODBC で規定される SQL データ型のことです。ODBC SQL データ型の詳細は、ODBC に接続するデータベースのマニュアルを参照してください。

(b) データソース，表，列から生成されるデータ項目の定義内容

(a) で説明した分類に従って、それぞれのデータ項目に生成される定義内容を表 4-9 に示します。

- データ項目名，標準名称，コメント，名前に設定される名称の文字数が，定義項目の上限値を超える場合，超えた部分が切り捨てられて生成されます。
- データソースの情報を使用するため，生成された言語別の名前が言語の規則（文字数や使用できる文字の種類など）に従っていない場合があります。生成後，リポジトリブラウザを使用して適切な名前に変更してください。
- 列に対応するデータ項目の種別，分類，けた数，小数部けた数，言語別のタイプは，列の ODBC タイプで決まります。詳細な対応については、「(c) ODBC タイプマッピングの規則」を参照してください。

表 4-9 データソースから生成されるデータ項目の定義内容

定義項目		データ項目の定義内容		
		データソースから生成	表から生成	列から生成
名称	データ項目名	データソース名	テーブル識別子	列識別子
	標準名称	データソース名 ¹	テーブル識別子 ¹	列識別子 ¹
	フリガナ	-	-	-
属性	分類	結合データ	結合データ	ODBC タイプに対応
	けた数	-	-	ODBC タイプに対応
	小数部けた数	-	-	ODBC タイプに対応
	反復回数	-	-	-
付加情報	フィールド 1 ~ 20	-	-	-
	コメント	-	テーブルの説明	列の説明

定義項目		データ項目の定義内容		
		データソースから生成	表から生成	列から生成
言語別 詳細情報	名前	データソース名 ²	テーブル識別子 ²	列識別子 ²
	タイプ	-	-	ODBC タイプに対応
	タイプ修飾情報 ³	-	-	-
	取りうる値	-	-	-
	初期値	-	-	-
	言語別フィールド	-	-	-
セキュ リティ 情報	所有者名	- ⁴	- ⁴	- ⁴
	グループ名	- ⁵	- ⁵	- ⁵
	アクセス権	- ⁶	- ⁶	- ⁶

(凡例)

- : 設定されない

注

データソース名は、コマンド実行時に、引数に指定したデータソース名です。
 テーブル識別子は、表の名前を示す文字列です。
 テーブルの説明は、表を説明する文字列です。
 列識別子は、列の名前を示す文字列です。
 列の説明は、列を説明する文字列です。

注 1

コマンド実行時に、引数に指定した国語区分に対して設定されます。指定していない場合、すべての国語区分に同じ内容が設定されます。

注 2

すべての言語区分に同じ内容が設定されます。

注 3

言語区分「COBOL 又は OOCOBOL」で定義する編集文字列、言語区分「IDL(CORBA)」で定義するシーケンスを指します。

注 4

コマンド実行時に、引数に指定した所有者名が設定されます。指定していない場合は、システム管理者が設定されます。

注 5

コマンド実行時に、引数に指定したグループ名が設定されます。指定していない場合は、システム管理者の所属するグループ名が設定されます。

注 6

4. 辞書情報の一括登録と出力

新規に登録したデータ項目にはアクセス権の初期値が設定されます。初期値については「3.2.1 アクセス権の設定」を参照してください。なお、既存のデータ項目を更新した場合には、設定されているアクセス権は変更されません。

(c) ODBC タイプマッピングの規則

データ項目種別

構造を持つ列は結合項目、構造を持たない列は単項目となります。

結合項目として生成される場合、構成項目も同時に生成され、そのデータ項目名には内容を表すサフィックスが付加されます。

ODBC タイプと付加されるサフィックスの有無の対応については、表 4-10 を参照してください。

分類、けた数、小数部けた数

タイプマッピングの規則に従って、ODBC タイプや条件に応じた分類、けた数、小数部けた数が設定されます。標準で用意されているタイプマッピングの規則を、表 4-10 に示します。

- 結合項目として生成されたデータ項目は分類が「結合データ」となるため、けた数と小数部けた数は設定されません。単項目または結合項目の構成項目として生成されたデータ項目には、ODBC タイプに応じた分類、けた数、小数部けた数が設定されます。
- 列の精度からけた数が設定されるデータ項目で、列の精度がけた数の上限値を超える場合、代わりに上限値が設定されます。

表 4-10 ODBC タイプと分類、けた数、小数部けた数の対応

ODBC タイプ	データ項目の定義内容			
	構成項目のサフィックス	分類	けた数	小数部けた数
SQL_CHAR	-	英数字文字列	長さ	-
SQL_VARCHAR	-length	整数	9	-
	-char	英数字文字列	長さ	-
SQL_LONGVARCHAR	-length	整数	9	-
	-char	英数字文字列	長さ	-
SQL_DECIMAL	-	整数	精度	-
		実数	精度	スケール
SQL_NUMERIC	-	整数	精度	-
		実数	精度	スケール
SQL_SMALLINT	-	整数	精度	-
SQL_INTEGER	-	整数	精度	-
SQL_REAL	-	実数	精度	-
SQL_FLOAT	-	実数	精度	-

ODBC タイプ	データ項目の定義内容			
	構成項目のサフィックス	分類	けた数	小数部けた数
SQL_DOUBLE	-	実数	精度	-
SQL_BIT	-filler	分類なし	7	-
	-bit	分類なし	1	-
SQL_TINYINT	-	整数	精度	-
SQL_BIGINT	-	整数	精度	-
SQL_BINARY	-	英数字文字列	長さ	-
SQL_VARBINARY	-length	整数	9	-
	-bit	英数字文字列	長さ	-
SQL_LONGVARBINARY	-length	整数	9	-
	-bit	英数字文字列	列最大長	-
SQL_DATE	-year	整数	4	-
	-month	整数	4	-
	-day	整数	4	-
SQL_TIME	-hour	整数	4	-
	-minute	整数	4	-
	-second	整数	4	-
SQL_TIMESTAMP	-year	整数	4	-
	-month	整数	4	-
	-day	整数	4	-
	-hour	整数	4	-
	-minute	整数	4	-
	-second	整数	4	-
	-fraction	整数	9	-
その他	-	分類なし	-	-

(凡例)

- : 設定されない

注

構成項目のサフィックス欄が「-」の ODBC タイプは、単項目として生成されるタイプです。「-」以外のタイプは、結合項目としてその構成項目が生成されるタイプです。

注

スケールに依存します。スケール=0 の場合は「整数データ」となります。その他の場合は「実数データ」となります。

言語別のタイプ

4. 辞書情報の一括登録と出力

タイプマッピングの規則に従って、ODBC タイプや条件に応じたタイプが設定されます。標準で用意されているタイプマッピングの規則を、表 4-11 から表 4-13 に示します。

- 結合項目として生成されたデータ項目には、言語別のタイプは設定されません。単項目または結合項目の構成項目として生成されたデータ項目には、ODBC タイプに応じた言語別のタイプが設定されます。
- 同じ ODBC タイプでも、列の符号の有無によって、設定されるタイプが異なる場合があります。
- 言語別 4 ~ 7 のタイプには「フリー定義」が設定されます。定義文字列は設定されません。

表 4-11 ODBC タイプと言語別タイプの対応 (COBOL 又は OOCOBOL, C 又は C++)

列の情報	データ項目の定義内容		
ODBC タイプ	構成項目の サフィックス	言語別のタイプ	
		COBOL 又は OOCOBOL	C 又は C++
SQL_CHAR	-	X	char
SQL_VARCHAR	-length	B	long
	-char	X	char *
SQL_LONGVARCHAR	-length	B	long
	-char	X	char *
SQL_DECIMAL	-	フリー定義 ¹	フリー定義 ⁵
SQL_NUMERIC	-	フリー定義 ¹	フリー定義 ⁵
SQL_SMALLINT	-	B ²	short ⁶
		BU ²	unsigned short ⁶
SQL_INTEGER	-	B ²	long ⁷
		BU ²	unsigned long ⁷
SQL_REAL	-	D	float
SQL_FLOAT	-	D	double
SQL_DOUBLE	-	D	double
SQL_BIT	-filler	1	フリー定義 ⁸
	-bit	1	フリー定義 ⁸
SQL_TINYINT	-	フリー定義 ³	int ⁹
			unsigned int ⁹
SQL_BIGINT	-	B ²	フリー定義 ⁵
		BU ²	
SQL_BINARY	-	X	フリー定義 ¹⁰
SQL_VARBINARY	-length	B	long

列の情報	データ項目の定義内容		
	構成項目の サフィックス	言語別のタイプ	
		COBOL 又は OOCOBOL	C 又は C++
ODBC タイプ	-bit	X	フリー定義 ¹⁰
SQL_LONGVARBINARY	-length	B	long
	-bit	X	フリー定義 ¹⁰
SQL_DATE	-year	B	short
	-month	BU	unsigned short
	-day	BU	unsigned short
SQL_TIME	-hour	BU	unsigned short
	-minute	BU	unsigned short
	-second	BU	unsigned short
SQL_TIMESTAMP	-year	B	short
	-month	BU	unsigned short
	-day	BU	unsigned short
	-hour	BU	unsigned short
	-minute	BU	unsigned short
	-second	BU	unsigned short
	-fraction	BU	unsigned long
その他	-	フリー定義 ⁴	フリー定義 ⁴

(凡例)

- : 設定されない

注 1

精度を \$p\$、スケールを \$s\$ とするとき、定義文字列は「PIC S9(\$p\$ - \$s\$)V9(\$s\$) SIGN LEADING SEPARATE」となります。

注 2

列符号がある場合は「B」、ない場合は「BU」となります。

注 3

定義文字列には「PIC X(1)」が設定されます。

注 4

定義文字列は設定されません。

注 5

定義文字列には「char *」が設定されます。

注 6

列符号がある場合は「short」、ない場合は「unsigned short」となります。

注 7

4. 辞書情報の一括登録と出力

列符号がある場合は「long」、ない場合は「unsigned long」となります。

注 8

定義文字列には「unsigned char」が設定されます。

注 9

列符号がある場合は「int」、ない場合は「unsigned int」となります。

注 10

定義文字列には「unsigned char *」が設定されます。

表 4-12 ODBC タイプと言語別タイプの対応 (SQL)

列の情報 ODBC タイプ	データ項目の定義内容		
	構成項目の サフィックス	言語別のタイプ (SQL)	
		データベースが ORACLE	データベースが HiRDB
SQL_CHAR	-	CHAR	CHAR
SQL_VARCHAR	-length	NUMBER	SMALLINT
	-char	VARCHAR2	VARCHAR
SQL_LONGVARCHAR	-length	NUMBER	DEC
	-char	LONG	フリー定義
SQL_DECIMAL	-	NUMBER	DEC
SQL_NUMERIC	-	NUMBER	DEC
SQL_SMALLINT	-	NUMBER	SMALLINT
SQL_INTEGER	-	NUMBER	INT
SQL_REAL	-	NUMBERf	SMALLFLT
SQL_FLOAT	-	NUMBERf	FLOAT
SQL_DOUBLE	-	NUMBERf	FLOAT
SQL_BIT	-filler	フリー定義	フリー定義
	-bit	フリー定義	フリー定義
SQL_TINYINT	-	NUMBER	SMALLINT
SQL_BIGINT	-	NUMBER	DEC
SQL_BINARY	-	フリー定義	フリー定義
SQL_VARBINARY	-length	NUMBER	SMALLINT
	-bit	フリー定義	フリー定義
SQL_LONGVARBINARY	-length	NUMBER	DEC
	-bit	フリー定義	フリー定義
SQL_DATE	-year	NUMBER	SMALLINT
	-month	NUMBER	SMALLINT
	-day	NUMBER	SMALLINT

列の情報	データ項目の定義内容		
ODBC タイプ	構成項目の	言語別のタイプ (SQL)	
	サフィックス	データベースが ORACLE	データベースが HiRDB
SQL_TIME	-hour	NUMBER	SMALLINT
	-minute	NUMBER	SMALLINT
	-second	NUMBER	SMALLINT
SQL_TIMESTAMP	-year	NUMBER	SMALLINT
	-month	NUMBER	SMALLINT
	-day	NUMBER	SMALLINT
	-hour	NUMBER	SMALLINT
	-minute	NUMBER	SMALLINT
	-second	NUMBER	SMALLINT
	-fraction	NUMBER	SMALLINT
その他	-	フリー定義	フリー定義

(凡例)

- : 該当しない

注

言語別 SQL に「ORACLE」または「HiRDB」のどちらのタイプで登録するかを、コマンドに指定してください。コマンドの詳細は、「4.2.3(3) ODBC 入力コマンド (RPOdbcin) の指定」の引数 /Q を参照してください。言語別 SQL で使用するタイプを、あらかじめ環境構築ユーティリティでカスタマイズしておく必要があります。標準に設定されている分類とタイプの対応については「表 2-4 分類区分とタイプの対応 (SQL)」、HiRDB のカスタマイズ例については「表 2-8 分類区分とタイプの対応のカスタマイズ例 (言語区分 SQL のタイプを HiRDB として使用する場合)」を参照してください。

注

定義文字列は設定されません。

表 4-13 ODBC タイプと言語別タイプの対応 (IDL(CORBA), 帳票定義, Java)

列の情報	データ項目の定義内容			
ODBC タイプ	構成項目の	言語別のタイプ		
	サフィックス	IDL(CORBA)	帳票定義	Java
SQL_CHAR	-	char	C	9
SQL_VARCHAR	-length	long	I	9
	-char	string	C	9
SQL_LONGVARCHAR	-length	long	I	9
	-char	string	C	9

4. 辞書情報の一括登録と出力

列の情報	データ項目の定義内容			
ODBC タイプ	構成項目の サフィックス	言語別のタイプ		
		IDL(CORBA)	帳票定義	Java
SQL_DECIMAL	-	フリー定義 ¹	I ⁶ フリー定義 ⁶	9
SQL_NUMERIC	-	フリー定義 ¹	I ⁷ B ⁷	9
SQL_SMALLINT	-	short ² unsigned short ²	I	9
SQL_INTEGER	-	long ³ unsigned long ³	I	9
SQL_REAL	-	float	I	9
SQL_FLOAT	-	double	B	9
SQL_DOUBLE	-	double	B	9
SQL_BIT	-filler	octet	フリー定義 ⁸	9
	-bit	octet	フリー定義 ⁸	9
SQL_TINYINT	-	long ³ unsigned long ³	I	9
SQL_BIGINT	-	フリー定義 ¹	I	9
SQL_BINARY	-	フリー定義 ⁴	フリー定義 ⁸	9
SQL_VARBINARY	-length	long	I	9
	-bit	フリー定義 ⁴	フリー定義 ⁸	9
SQL_LONGVARBINARY	-length	long	I	9
	-bit	フリー定義 ⁴	フリー定義 ⁸	9
SQL_DATE	-year	short	I	9
	-month	unsigned short	I	9
	-day	unsigned short	I	9
SQL_TIME	-hour	unsigned short	I	9
	-minute	unsigned short	I	9
	-second	unsigned short	I	9
SQL_TIMESTAMP	-year	short	I	9
	-month	unsigned short	I	9

列の情報	データ項目の定義内容			
ODBC タイプ	構成項目の	言語別のタイプ		
	サフィックス	IDL(CORBA)	帳票定義	Java
	-day	unsigned short	I	9
	-hour	unsigned short	I	9
	-minute	unsigned short	I	9
	-second	unsigned short	I	9
	-fraction	unsigned long	I	9
その他	-	フリー定義 ⁵	フリー定義 ⁵	フリー定義 ⁵

(凡例)

- : 設定されない

注 1

定義文字列には「string」が設定されます。

注 2

列符号がある場合は「short」、ない場合は「unsigned short」となります。

注 3

列符号がある場合は「long」、ない場合は「unsigned long」となります。

注 4

定義文字列には「octet *」が設定されます。

注 5

定義文字列は設定されません。

注 6

列の精度とスケールに依存します。スケール=0かつ精度 16の場合と、0<スケール 5かつ(精度-スケール) 16の場合は「I」となります。その他の場合はフリー定義で、内容は「C」となります。

注 7

列の精度とスケールに依存します。スケール=0かつ精度 10の場合は「I」となります。その他の場合は「B」となります。

注 8

定義文字列には「C」が設定されます。

注 9

環境構築ユーティリティで設定した分類区分に対応するタイプの初期値(デフォルトタイプ)が設定されます。

(3) データ項目名による更新入力

データベースから生成された定義情報で、すでに辞書に登録されているデータ項目を更

4. 辞書情報の一括登録と出力

新できます（更新入力）。この場合は、ODBC 入力コマンドで、引数「/N 辞書への登録方法」を指定します。

更新入力では、データ項目名によって、更新の対象となるデータ項目が特定されます。このため、既存の辞書フォルダ内および登録されるデータ項目には、一意のデータ項目名が定義されている必要があります。なお、対象となるデータ項目が辞書にない場合には、新しいデータ項目として作成、登録されます。

(4) 外部キーの扱い

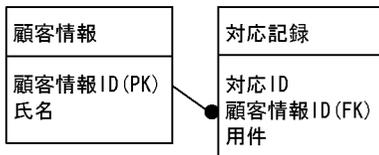
表に指定された外部キーの関係を、辞書に反映できます。ここでは、外部キーを含む表の生成規則について説明します。ODBC 入力コマンドの引数の指定によって、個別のデータ項目として生成するか、またはデータ項目に外部キーに基づいた関連を付けて生成するかを選択できます。

(a) 外部キーを個別のデータ項目として生成

ODBC 入力コマンドで、引数「/R 外部キー関連付けの有無」を省略、または引数「/R 外部キー関連付けの有無」で「nouse」を指定すると、外部キーは個別のデータ項目として生成されます。この場合の例を次に示します。

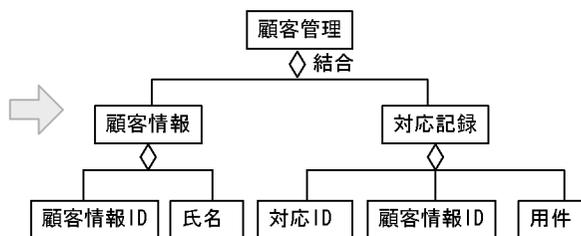
●生成元のデータソース

データソース：顧客管理



(凡例) PK：主キー
FK：外部キー

●生成されるデータ項目の関係



(b) 外部キーをデータ項目の関連として生成

ODBC 入力コマンドで、引数「/R 外部キー関連付けの有無」に「use」を指定すると、外部キーの関係をデータ項目間の関連として生成できます。この場合、外部キーとなるデータ項目は作成されず、主キーとなるデータ項目に関連が付けられます。

なお、外部キーの情報をデータソースから取得するためには、ODBC ドライバが API 適合レベル 2 に準拠している必要があります。ODBC ドライバの API 適合レベルについては、使用する ODBC ドライバまたはデータベースのマニュアルなどで確認してください。

外部キーの関連づけは、ODBC ドライバの API 適合レベルによって、生成規則が異なります。

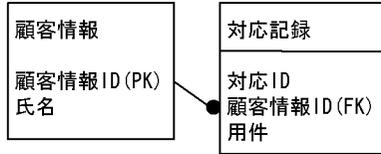
(i) API 適合レベル 2 の場合

- 関連の生成規則

外部キーとなるデータ項目は作成されず、主キーとなるデータ項目に結合の関連が付けられます。この場合の例を次に示します。

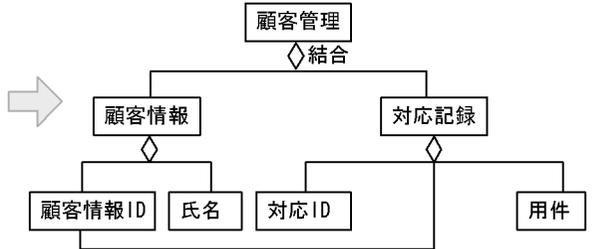
●生成元のデータソース

データソース：顧客管理



(凡例) PK：主キー
FK：外部キー

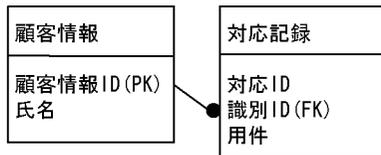
●生成されるデータ項目の関係



ただし、一度に入力する表に、主キーと外部キーの両方の指定が含まれていることが前提となります。主キーとなるデータ項目が一度に入力する表に含まれない場合には、外部キーのデータ項目だけが新規に作成されます。また、外部キーの名称を別名で指定している場合は、主キーのデータ項目を親として継承するデータ項目が作成されます。この場合の例を次に示します。

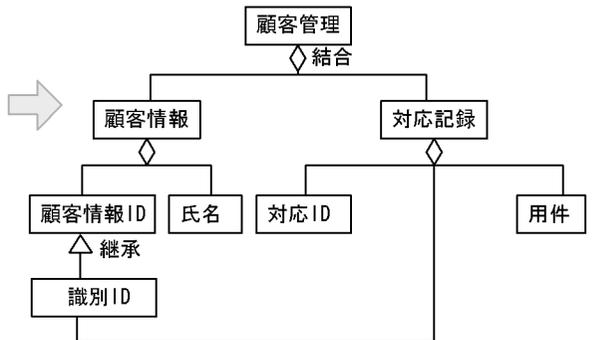
●生成元のデータソース

データソース：顧客管理



(凡例) PK：主キー
FK：外部キー

●生成されるデータ項目の関係



- 表から生成されたデータ項目と既存のデータ項目との関連
データ項目名による更新入力を併用することで、入力する外部キーのデータ項目と既存の辞書フォルダ内にある主キーのデータ項目とに関連が付けられます。既存の辞書フォルダ内にある主キーのデータ項目は、表の外部キーの属性で更新されます。また、外部キーが別名のデータ項目は、既存の辞書フォルダ内の主キーとなるデータ項目を親とする継承の関連が付けられます。ただし、継承の関連は、主キーとなるデータ項目が先に辞書フォルダに登録されていることが前提となります。

(ii) API 適合レベル 2 がサポートされていない場合

名称が同じ列に対するデータ項目が、一つのデータ項目に統合されます。この場合、属性が同じ列を除く列の名称は、入力する表全体で一意であることが前提となります。名称が同じ列が複数ある場合には、入力時に先に登録されたデータ項目に統合されます。

このように、列の名称によって一つのデータ項目に統合されるため、データ項目名

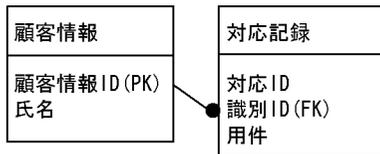
4. 辞書情報の一括登録と出力

による更新入力を行う場合に、入力するデータ項目の名称を一意に保ちやすくなります。

なお、列に別名が付けられている場合、主キーと外部キーで継承の関係が結べないため、独立したデータ項目として登録されます。この場合の例を次に示します。

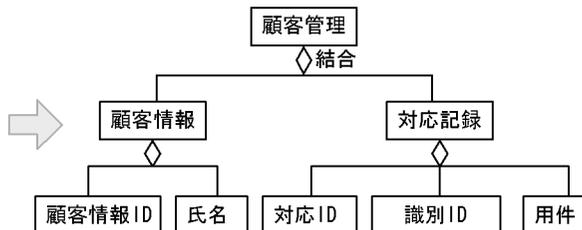
●生成元のデータソース

データソース：顧客管理



(凡例) PK：主キー
FK：外部キー

●生成されるデータ項目の関係



4.2.3 ODBC 入力コマンド (RPOdbcin) の使い方

(1) ODBC 入力コマンドが格納されているディレクトリ

コマンドは、次のディレクトリに格納されています。

サーバのSEWB+ 基本開発環境の組み込み先パス名¥Repository¥bin

SEWB+ 基本開発環境の組み込み先パス名は、デフォルトでは次のように設定されています。システムドライブとは、Windows 2000 Server, Windows Server 2003 または Windows Server 2003 x64 が組み込まれているドライブです。

- Windows 2000 Server または Windows Server 2003 の場合
システムドライブ：¥Program Files¥HITACHI¥Sewb
- Windows Server 2003 x64 の場合
システムドライブ：¥Program Files (x86)¥HITACHI¥Sewb

(2) コマンド実行時の注意事項

(a) コマンドを実行する前に

コマンドを実行するサーバでは、SEWB+/REPOSITORY からログアウトしておいてください。

(b) コマンド実行時にほかの作業をする場合

ODBC 入力コマンドでは、実行時に排他制御を行っていません。したがって、コマンドの実行中でもほかの作業をすることができます。ただし、次に示す点に注意してください。

- コマンドの実行中に対象となっている資源や関連情報を、更新または削除してはなりません。不整合が検出され、正常に終了しない場合があります。この場合には、ODBC 入力コマンドの再実行が必要です。

- コマンドの実行中に、更新の対象となっている資源を変更してはなりません。ODBC 入力コマンドが正常に終了しても、資源の内容が更新されていない場合があります。

(3) ODBC 入力コマンド (RPOdbcin) の指定

サーバのコマンドラインに、次に示す構文に従ってコマンドと引数を指定して実行します。実行中には、データ項目の登録件数が表示されます。また、詳細情報がログファイルに出力されます。

(a) 形式

```
RPOdbcin /S データソース名 [ [ /T 表名 ] ... ]
          [ /U ユーザ ID ] [ /P パスワード ]
          /D 辞書フォルダ名
          [ /O 所有者名 ] [ /G グループ名 ]
          [ /C 登録先国語別 ]
          [ /Q SQL タイプ ]
          [ /R 外部キー関連付けの有無 ]
          [ /N 辞書への登録方法 ]
          [ /L ログファイル名 ]
          [ /? ]
```

(b) 引数

/S データソース名

ODBC コントロールパネルを使って追加およびセットアップしたデータソースの名称を指定します。

/T 表名

データソースの中から結合項目を生成する表名を指定します。省略すると、データソース中にあるすべての表から結合項目が生成されます。

/U ユーザ ID

データソースとなるデータベースに接続するため、必要に応じてユーザ ID を指定します。

/P パスワード

データソースとなるデータベースに接続するため、必要に応じてパスワードを指定します。

/D 辞書フォルダ名

入力した情報の登録先となる辞書フォルダ名を指定します。

/O 所有者名 ¹

作成する辞書の所有者名を指定します。省略時はシステム管理者が仮定されます。

4. 辞書情報の一括登録と出力

/G グループ名 ¹

作成する辞書のグループ名を指定します。省略時はシステム管理者が所属するグループが仮定されます。

/C 登録先国語別

登録する国語区分を指定します。環境構築ユティリティで設定した国語区分の順序に対応して1～4の数字で指定します。省略時は、データソースから得られた値がすべての国語別に登録されます。

/Q SQL タイプ

言語別 SQL に登録するタイプを指定します。省略時は「ora」が仮定されます。

ora：言語別 SQL に、ORACLE のタイプを登録

hir：言語別 SQL に、HiRDB のタイプを登録

/R 外部キー関連付けの有無

外部キーの構成項目を、主キーとなるデータ項目に関連づけるかどうかを指定します。省略時は「nouse」が仮定されます。

nouse：外部キーでも新規にデータ項目を作成する

use：外部キーは主キーのデータ項目に関連づける

/N 辞書への登録方法

辞書フォルダへデータ項目を新規に登録するか、データ項目名で更新するかどうかを指定します。省略時は「new」が仮定されます。

new：すべてのデータ項目を新規に登録する

name：登録先の辞書フォルダに同じデータ項目名のデータ項目がある場合、定義情報を更新する

/L ログファイル名 ²

ログファイルの出力先を指定します。省略時は、次のファイルがログファイルとして作成されます。

パス：環境変数「TEMP」または「TMP」に代入されている値

ファイル名：rpodbcin.log

/?

コマンドの構文と引数の指定方法を表示します。

注 1

新しい辞書フォルダまたはデータ項目を登録するときに指定します。ここで指定した所有者名とグループ名は辞書フォルダ、データ項目に共通に設定されます。引数に指定する名称は、辞書登録するサーバに登録されているユーザ、グループの名称を使用してください。ユーザ、グループの登録についてはマニュアル「SEWB+/REPOSITORY 運用ガイド」を参照してください。なお、この指定では、既存の辞書情報に設定されている所有者名およびグループ名は変更できません。

注 2

ログファイルはコマンドの実行ごとに上書きされます。残しておきたい場合は異なるログファイル名を指定してください。

(c) リターン値

コマンドの実行が終了すると、次のリターン値が返されます。

- 0：正常終了した
- 1：その他のエラーが発生した
- 2：メモリが確保できない
- 3：引数が不正である
- 4：ログファイルが作成できない
- 5：辞書に対するアクセス権がない
- 6：データソースが開けない
- 7：表が開けない

4.3 SEWB3 資産を移行する

SEWB+/REPOSITORY では、WS の開発で蓄えられた SEWB3 の辞書情報を移行するためのコマンド（SEWB3 辞書移行コマンドといいます）が用意されています。この機能は、サーバでシステム管理者だけが利用できます。

SEWB+/REPOSITORY のデータ項目辞書には、SEWB3 のデータ項目の基本項目および詳細項目から、次の定義情報が移行されます。

- 基本項目...名称，属性，付加情報
- 詳細項目...入力項目，出力項目，チェック条件

また、業務ルール辞書には、詳細項目のフリー定義から定義情報が移行されます。

なお、それぞれの辞書の定義項目には相違がありますが、SEWB+/REPOSITORY が定めた規則に従って自動的に変換されます。

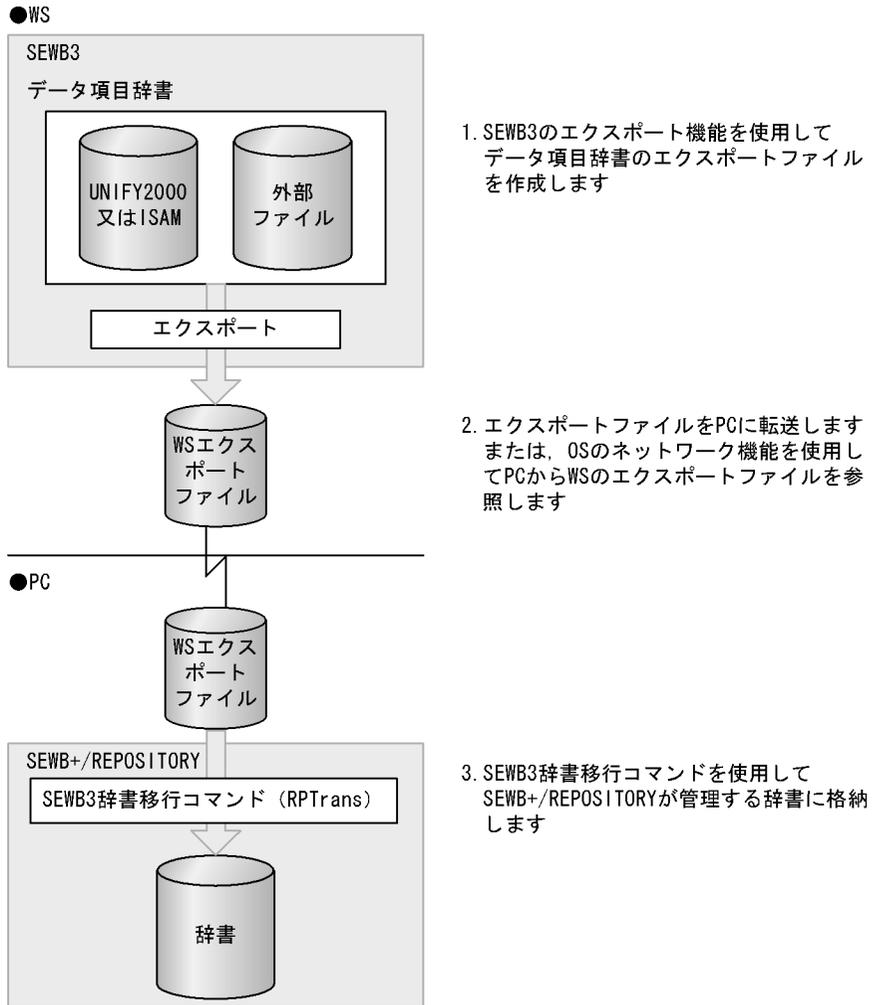
4.3.1 SEWB3 データ項目辞書の移行作業

ここでは、SEWB3 辞書移行コマンドを使用した移行作業について説明します。SEWB3 のエクスポート機能の使い方については、マニュアル「SEWB3 データ項目設計支援使用の手引」を参照してください。

（1）移行作業の流れ

移行作業の流れを、図 4-7 に示します。

図 4-7 移行作業の流れ



移行作業

SEWB3 のデータ項目辞書は UNIFY2000 リポジトリまたは ISAM リポジトリに格納されています。次に移行作業の手順を説明します。

注

SEWB3 辞書移行コマンドを使用して移行できるのは、シフト JIS のロケール環境で作成したデータ項目辞書だけです。したがって、シフト JIS 以外のロケール環境で作成したデータ項目辞書をこのコマンドで移行しても、SEWB+/REPOSITORY および SEWB+/REPOSITORY の機能を使用するクライアントプログラムの動作は保証されませんので注意してください。日本語 EUC 環境で作成したデータ項目辞書を移行したい場合には、SEWB3 でシフト JIS コードに変換したあと、そのファイルをエクスポートして、SEWB+ への移行に使

4. 辞書情報の一括登録と出力

用してください。変換の手順については、マニュアル「SEWB3 運用の手引」の異なる言語間のコード変換の説明を参照してください。

1. SEWB3 のエクスポート機能を使って、移行するデータ項目辞書をエクスポートファイルに取り出します。
2. エクスポートファイルを PC 環境に、バイナリモードで転送します。
または、OS (Windows 2000 Server , Windows Server 2003 もしくは Windows Server 2003 x64) のネットワーク機能を使用して、PC からエクスポートファイルを参照できるようにします。
3. エクスポートファイルを指定して SEWB3 辞書移行コマンドを実行します。

移行後の辞書情報の確認と変更作業

移行した辞書情報を SEWB+/REPOSITORY の辞書でどのように利用するか検討し、必要に応じて辞書情報を加工します。辞書情報の確認と変更や、データ項目の継承関係、結合関係を定義する作業にはリポジトリブラウザを使用します。また、移行した辞書情報を CSV 形式ファイル出力コマンドを使って CSV 形式ファイルに出力すれば、他のアプリケーション上で確認と変更作業をすることもできます。

(2) 辞書フォルダへの登録

辞書フォルダへの登録規則を次に説明します。

- 登録したい辞書フォルダがリポジトリにある場合
SEWB3 辞書移行コマンドで指定した辞書フォルダに、辞書情報が変換されて登録されます。データ項目と業務ルールが新規に追加されます。
- 新しい辞書フォルダに登録する場合
SEWB3 辞書移行コマンドで指定した辞書フォルダ名で、新しい辞書フォルダが作成されます。そこに、辞書情報が変換されて登録されます。

4.3.2 基本項目の変換規則

ここでは、SEWB3 データ項目の基本項目と、SEWB+/REPOSITORY データ項目との対応と変換規則について説明します。

(1) データ項目の名称、属性および付加情報の変換

SEWB3 データ項目の名称、属性、付加情報の対応関係と変換規則を表 4-14 に示します。

表 4-14 基本項目の変換

SEWB3 データ項目辞書	SEWB + 辞書 (データ項目)
データ項目名	データ項目名
日本語名	標準名称 (日本語)
振り仮名	フリガナ

SEWB3 データ項目辞書	SEWB + 辞書 (データ項目)
略称	フィールド 1 ¹
グループ	フィールド 2 ²
タイプ	分類, タイプ ³
けた数 (整数部), けた数 (小数部)	けた数 ⁴
けた数 (小数部)	小数部けた数
反復回数	反復回数
初期値	初期値 (COBOL 又は OOCOBOL)
編集文字列	タイプ修飾情報 (COBOL 又は OOCOBOL の編集文字列)
コメント	コメント (日本語)

注 1

「フィールド 1」の名称が「略称」に変更されて定義情報が移行されます。

注 2

「フィールド 2」の名称が「グループ」に変更されて定義情報が移行されます。

注 3

詳細は、「表 4-15 タイプから分類区分への変換」、「表 4-16 タイプから言語別タイプへの変換」を参照してください。

注 4

「けた数 (整数部)」と「けた数 (小数部)」の和が「けた数」に移行されます。

(2) タイプから分類区分への変換

SEWB3 データ項目に定義されたタイプは、その内容に応じて分類区分に変換されます。タイプから分類区分への対応関係と変換規則を表 4-15 に示します。

表 4-15 タイプから分類区分への変換

SEWB3 データ項目辞書		SEWB + 辞書 (データ項目)
タイプ	内容	分類区分
X	文字データ, 固定長文字列データ	英数字文字列データ
XR	可変長文字列データ	英数字文字列データ
XL	可変長長文字列データ	英数字文字列データ
N	固定長漢字文字列データ	日本語文字列データ
NR	可変長漢字文字列データ	日本語文字列データ
NL	可変長長漢字文字列データ	日本語文字列データ
M	固定長混在文字列データ	日本語文字列データ
MR	可変長混在文字列データ	日本語文字列データ

4. 辞書情報の一括登録と出力

SEWB3 データ項目辞書		SEWB + 辞書 (データ項目)
タイプ	内容	分類区分
ML	可変長混在文字列データ	日本語文字列データ
S	10進ゾーン形式データ (符号付き)	1
9	10進ゾーン形式データ (符号なし)	2
P	10進パック形式データ (符号付き)	1
U	10進パック形式データ (符号なし)	2
B	2進固定小数点データ (符号付き)	整数データ
BU	2進固定小数点データ (符号なし)	正整数データ
E	外部 10進浮動小数点データ	実数データ
D	内部 10進浮動小数点データ	実数データ
G	集団項目	分類なし
8	外部ブール項目	分類なし
1	内部ブール項目	分類なし
T	ポインタデータ	分類なし
Z	数字編集項目	英数字文字列データ
PL	拡張精度固定小数点データ (10進パック数形式)	1
XD	日付型データ	分類なし
XT	時刻型データ	分類なし
その他のタイプ		分類なし

注 1

小数部けた数が0または指定されていない場合は「整数データ」となります。これ以外の場合
は「実数データ」となります。

注 2

小数部けた数が0または指定されていない場合は「正整数データ」となります。これ以外の場合
は「実数データ」となります。

(3) タイプから言語別タイプへの変換

SEWB3 データ項目に定義されたタイプは、その内容に応じて言語別に「COBOL 又は
OOCOBOL」および「SQL」のタイプに変換されます。ただし、「C 又は C++」のタイ
プには変換されません。タイプから言語別のタイプへの対応関係と変換規則を表 4-16 に
示します。

表 4-16 タイプから言語別タイプへの変換

SEWB3 データ項目辞書		SEWB + 辞書 (データ項目) ¹	
タイプ	内容	言語区分が COBOL 又は OOCOBOL のタイプ	言語区分が SQL のタイプ ²
X	文字データまたは固定長文字列データ	X: 英数字項目	CHAR: 固定長文字列
XR	可変長文字列データ	X: 英数字項目	VARCHAR2: 可変長文字列
XL	可変長長文字列データ	X: 英数字項目	LONG: 可変長長文字列
N	固定長漢字文字列データ	N: 漢字項目	CHAR: 固定長文字列
NR	可変長漢字文字列データ	N: 漢字項目	VARCHAR2: 可変長文字列
NL	可変長長漢字文字列データ	N: 漢字項目	LONG: 可変長長文字列
M	固定長混在文字列データ	フリー定義	CHAR: 固定長文字列
MR	可変長混在文字列データ	フリー定義	VARCHAR2: 可変長文字列
ML	可変長長混在文字列データ	フリー定義	LONG: 可変長長文字列
S	10 進ゾーン形式データ (符号付き)	S: 符号付き外部 10 進項目	NUMBER: 固定小数点数
9	10 進ゾーン形式データ (符号なし)	9: 符号なし外部 10 進項目	NUMBER: 固定小数点数
P	10 進パック形式データ (符号付き)	P: 符号付き内部 10 進項目	NUMBER: 固定小数点数
U	10 進パック形式データ (符号なし)	U: 符号なし内部 10 進項目	NUMBER: 固定小数点数
B	2 進固定小数点データ (符号付き)	B: 符号付き 2 進項目	NUMBER: 固定小数点数
BU	2 進固定小数点データ (符号なし)	BU: 符号なし 2 進項目	NUMBER: 固定小数点数
E	外部 10 進浮動小数点データ	E: 外部浮動小数点項目	NUMBERf: 浮動小数点数
D	内部 10 進浮動小数点データ	D: 内部浮動小数点項目	NUMBERf: 浮動小数点数
G	集団項目	フリー定義	フリー定義
8	外部ブール項目	8: 外部ブール項目	フリー定義
1	内部ブール項目	1: 内部ブール項目	フリー定義
T	ポインタデータ	T: アドレスデータ項目	フリー定義
Z	数字編集項目	Z: 数字編集項目	フリー定義
PL	拡張精度固定小数点データ (10 進パック数形式)	P: 符号付き内部 10 進項目	NUMBER: 固定小数点数
XD	日付型データ	フリー定義	DATE: 日付
XT	時刻型データ	フリー定義	フリー定義
その他のタイプ		フリー定義	フリー定義

4. 辞書情報の一括登録と出力

注 1

言語区分「C 又は C++」、「IDL(CORBA)」、「帳票定義」および「Java」のタイプには、フリー定義が設定されます。

注 2

言語区分「SQL」には ORACLE の標準タイプが設定されます。ORACLE の標準タイプについては「表 2-4 分類区分とタイプの対応 (SQL)」を参照してください。

(4) 移行前のタイプの扱い

任意の利用のために用意されているフィールド「フィールド 3」の名称が、「移行前のタイプ」に変更されます。このフィールドの中に変換される前のタイプの定義情報が格納されます。

4.3.3 詳細項目の変換規則

ここでは、SEWB3 データ項目の詳細項目と、SEWB+/REPOSITORY のデータ項目および業務ルールとの対応と変換規則について説明します。移行の主な規則を次に示します。

- 詳細項目の入力項目、出力項目および条件名は、データ項目の定義情報に移行されます。このとき、「フィールド 4」の名称が「詳細項目種別」に変更されます。このフィールドには、移行元の詳細項目の種別に従って、「入力項目」、「出力項目」または「チェック条件」が格納されます。
- 業務ルール名は、基本項目のデータ項目名などから SEWB+/REPOSITORY が定めた規則に従って付けられます。
- 詳細項目のフリー定義情報は、業務ルールのルールスクリプト（「COBOL 又は OOCOBOL」のルールスクリプト）に移行されます。
- ワーク領域、条件キーワードおよびエラーコードの定義情報は、データ項目のコメントに移行されます。

(1) 詳細項目の変換

SEWB3 データ項目の詳細項目とデータ項目および業務ルールの定義項目との対応関係と変換規則を、表 4-17 に示します。

表 4-17 詳細項目からデータ項目および業務ルールへの変換

SEWB3 データ項目辞書		SEWB + 辞書	
詳細項目の種別	定義項目	辞書の種別	定義項目
入力項目	入力項目名	データ項目	データ項目名
	日本語名	データ項目	標準名称 (日本語)
	タイプ	データ項目	タイプ
	整数部けた数	データ項目	けた数
	小数部けた数	データ項目	小数部けた数

SEWB3 データ項目辞書		SEWB + 辞書	
詳細項目の種別	定義項目	辞書の種別	定義項目
	使用目的	データ項目	フィールド 5
	コメント, ワーク領域名, 編集キーワード	データ項目	コメント(日本語)
	フリー定義情報	業務ルール	ルールスクリプト
出力項目	出力項目名	データ項目	データ項目名
	日本語名	データ項目	標準名称(日本語)
	タイプ	データ項目	タイプ
	整数部けた数	データ項目	けた数
	小数部けた数	データ項目	小数部けた数
	使用目的	データ項目	フィールド 5
	コメント, ワーク領域名, 編集キーワード	データ項目	コメント(日本語)
	フリー定義情報	業務ルール	ルールスクリプト
チェック条件	条件項目名	データ項目	データ項目名
	日本語名	データ項目	標準名称(日本語)
	タイプ	データ項目	タイプ
	整数部けた数	データ項目	けた数
	小数部けた数	データ項目	小数部けた数
	使用目的	データ項目	フィールド 5
	コメント, ワーク領域名, 条件キーワード, エラーコード, メッセージ可変部	データ項目	コメント(日本語)
	フリー定義情報	業務ルール	ルールスクリプト
条件キーワード	条件キーワード, コメント, 条件式	データ項目	コメント(日本語)
エラーコード	エラーコード, エラーメッセージ	データ項目	コメント(日本語)

(2) 業務ルールの名称の付加規則

SEWB3 データ項目で、日本語名を定義している場合と定義していない場合とでは名称が付けられる規則が異なります。

4. 辞書情報の一括登録と出力

(a) 日本語名を定義していない場合

「データ項目名」と「詳細項目の種別」が順に組み合わせられて業務ルール名が付けられません。詳細項目の種別は、次の全角文字に変換されます。

入力項目...「入力」

出力項目...「出力」

チェック条件...「チェック」

(b) 日本語名を定義している場合

この場合は、業務ルール名と標準名称（日本語名）が付けられます。

- 業務ルール名

「データ項目名」、ハイフン（ - ）および「詳細項目の種別」が順に組み合わせられて業務ルール名が付けられます。詳細項目の種別は、次の半角文字に変換されます。

入力項目...「INPUT」

出力項目...「OUTPUT」

チェック条件...「CHECK」

- 標準名称（日本語名）

「データ項目名」と「詳細項目の種別」が順に組み合わせられて標準名称が付けられません。詳細項目の種別は、次の全角文字に変換されます。

入力項目...「入力」

出力項目...「出力」

チェック条件...「チェック」

(3) 業務ルールと関連するデータ項目の適用形態の設定

SEWB3 データ項目の詳細項目から、SEWB+ に移行した業務ルールとデータ項目に関連が付けられます。また、移行した業務ルールには、データ項目の適用形態、キーワードおよび適用条件が次のように設定されます。

- 適用形態：専用にする

- キーワード：「@」+ データ項目名

- 適用条件：

詳細項目の種別に応じて、入力項目は「入力」、出力項目は「出力」、チェック条件は「 - （指定しない）」が設定される

(4) 内部項目の変換

詳細項目の内部項目の変換規則を次に説明します。

なお、SEWB3/DATAITEM のバージョン 01-13 から、ワーク領域に日本語名が追加されました。01-13 より前のバージョンで作成したエクスポートファイルを移行する場合、日本語名は省略されます。

(a) 移行順序

データ項目の日本語のコメントに、次の順番で移行されます。

1. 内部構造：
レベル，従属項目名，再定義従属項目名
2. ワーク領域：
レベル，ワーク領域名，日本語名，タイプ，けた数，整数部，小数部，反復回数，再定義ワーク領域名
3. 詳細項目のワーク領域名
4. 詳細項目のキーワード
5. 詳細項目のエラーコード
6. 詳細項目のメッセージ可変部
7. リソース区分
8. 漢字タイトル
9. 仮名タイトル
10. 日本語説明文

注

1. ~ 10. の長さが 6,500 バイトを超える場合には、6,501 バイト目以降（2 バイトコードがあれば 6,500 バイト目以降）が切り捨てられます。

(b) ワーク領域の移行

詳細項目にワーク領域名がある場合には、ワーク領域定義情報（レベル，ワーク領域名，日本語名，タイプ，けた数，整数部，小数部，反復回数および再定義ワーク領域名）がデータ項目の日本語のコメントに移行されます。

(c) キーワードの移行

詳細項目にキーワードがある場合には、そのキーワード情報（キーワード，コメントおよび条件式）が、データ項目の日本語のコメントに移行されます。

(d) エラーコードの移行

詳細項目にエラーコードがある場合には、そのエラーコード情報（エラーコード，エラーメッセージ）がデータ項目の日本語のコメントに移行されます。

(e) コメントのフォーマット

コメントに移行される場合のフォーマット

- 内部構造：レベル，従属項目名，再定義従属項目名
- ワーク領域：レベル，日本語名，ワーク領域名，タイプ，けた数，整数部，小数部，反復回数，再定義ワーク領域名
- ワーク領域名：XXXXXX
- キーワード：XXXXXX，コメント，条件式
- エラーコード：XXXXXX，メッセージ可変部，エラーメッセージ
- リソース区分：XXXXXX

4. 辞書情報の一括登録と出力

- 漢字タイトル：XXXXXX
- 仮名タイトル：XXXXXX
- 日本語説明文：XXXXXX

注

XXXXXX は、定義情報を示します。

内部項目の移行の例

内部項目の移行の例を、図 4-8 に示します。

図 4-8 内部項目の移行の例

●SEWB3 データ項目辞書

データ項目	年月日	9(8)	...		ワーク領域	01	W_年月日	
入力項目	年	X(4)	W_年			03	W_年	9(4)
出力項目	年	X(8)	W_年	SUB(XX)		03	W_月	9(4)
チェック条件	年	X(8)	W_年	SUB(XX)	ERROR01		'年月日'	
エラーコード	ERROR01	@には実在する年月日を入力して下さい。						
条件キーワード	SUB	サブルーチンを呼び、リターンコードを判定。SUB(サブルーチン、引数、...)						



●SEWB+ データ項目辞書

データ項目	年月日	9(8)	...	●	データ項目が移行されます
データ項目	年	X(4)	...	●	詳細項目（入力項目）が移行されます
	ワーク領域： 01, W_年月日 03, W_年, 9, 4, 0 03, W_月, 9, 2, 0 03, W_日, 9, 2, 0 ワーク領域名：W_年				
データ項目	年月日	X(8)	...	●	詳細項目（出力項目）が移行されます
	キーワード SUB(XX), サブルーチンを呼び、リターンコードを判定。SUB(サブルーチン、引数、...)				
データ項目	年月日	X(8)	...	●	詳細項目（チェック項目）が移行されます
	キーワード SUB(XX), サブルーチンを呼ぶ。SUB(サブルーチン、引数、...) エラーコード ERROR01, @には実在する年月日を入力して下さい。,'年月日'				

注 「ワーク領域」、「キーワード」および「エラーコード」はタイトルとして付けられます。

4.3.4 SEWB3 辞書移行コマンド (RPTrans) の使い方

(1) SEWB3 辞書移行コマンドが格納されているディレクトリ

コマンドは、次のディレクトリに格納されています。

サーバのSEWB+ 基本開発環境の組み込み先パス名¥Repository¥bin

4. 辞書情報の一括登録と出力

SEWB+ 基本開発環境の組み込み先パス名は、デフォルトでは次のように設定されています。システムドライブとは、Windows 2000 Server、Windows Server 2003 または Windows Server 2003 x64 が組み込まれているドライブです。

- Windows 2000 Server または Windows Server 2003 の場合
システムドライブ : %Program Files%HITACHI%Sewb
- Windows Server 2003 x64 の場合
システムドライブ : %Program Files(x86)%HITACHI%Sewb

(2) コマンド実行時の注意事項

(a) コマンドを実行する前に

コマンドを実行するサーバでは、SEWB+/REPOSITORY からログアウトしておいてください。

(b) コマンド実行中にほかの作業をする場合

SEWB3 辞書移行コマンドでは、実行中に排他制御を行っていません。したがって、コマンドの実行中でも並行してほかの作業をすることができます。ただし、次に示す点に注意してください。

- コマンドの実行中に、対象となっている資源や関連情報を更新または削除してはなりません。不整合が検出され、正常に終了しない場合があります。この場合には、SEWB3 辞書移行コマンドの再実行が必要です。
- コマンドの実行中に、移行先で更新の対象となっている資源を更新してはなりません。SEWB3 辞書移行コマンドが正常に終了しても、資源の内容が更新されていない場合があります。

(3) SEWB3 辞書移行コマンド (RPTrans) の指定

サーバのコマンドラインに、次に示す構文に従ってコマンドと引数を指定して実行します。実行中には、データ項目および業務ルールの登録件数が表示されます。また、詳細情報がログファイルに出力されます。

(a) 形式

```
RPTrans /F エクスポートファイル名 /D 辞書フォルダ名  
      [ /O 所有者名 ] [ /G グループ名 ]  
      [ /L ログファイル名 ]  
      [ /? ]
```

(b) 引数

/F エクスポートファイル名
入力するエクスポートファイルを指定します。

/D 辞書フォルダ名

辞書情報を登録する辞書フォルダ名を指定します。

/O 所有者名 ¹

作成する辞書の所有者名を指定します。省略時はシステム管理者が仮定されます。

/G グループ名 ¹

作成する辞書のグループ名を指定します。省略時はシステム管理者が所属するグループが仮定されます。

/L ログファイル名 ²

ログファイルの出力先を指定します。省略時は次のファイルがログファイルとして作成されます。

パス：環境変数「TEMP」または「TMP」に代入されている値

ファイル名：rptrans.log

/?

コマンドの構文と引数の指定方法を表示します。

注 1

新しい辞書フォルダ、データ項目または業務ルールを登録するときに指定します。ここで指定した所有者名とグループ名は、辞書フォルダ、データ項目および業務ルールに共通に設定されます。引数に指定する名称は、辞書登録するサーバに登録されているユーザ、グループの名称を使用してください。ユーザ、グループの登録についてはマニュアル「SEWB+/REPOSITORY 運用ガイド」を参照してください。なお、この指定では既存の辞書情報に設定されている所有者名およびグループ名は変更できません。

注 2

ログファイルはコマンドの実行ごとに上書きされます。残しておきたい場合は異なるログファイル名を指定してください。

(c) リターン値

コマンドの実行が終了すると、次のリターン値が返されます。

- 0：正常終了した
- 1：その他のエラーが発生した
- 2：メモリが確保できない
- 3：引数が不正である
- 4：ログファイルが作成できない
- 5：辞書に対するアクセス権がない
- 6：入力ファイルがない
- 7：入力ファイルのフォーマットが不正である

5

EUR Professional Edition の レポート設計

帳票設計ツールに EUR Professional Edition (バージョン 04-01 以降) を使用すれば、辞書のレコード定義情報を基にしてレポートを効率良く設計できます。さらに、SEWB+/CONSTRUCTION で作成したレポート用のデータファイルを出力するアプリケーションと組み合わせ、効率良くレポート出力アプリケーションを作成できます。この章では、あらかじめ辞書の作成者に理解しておいてほしい辞書のレコード定義 (最上位結合項目) とアプリケーションの関係、EUR Professional Edition を使ったレポートの設計手順や、フィールド定義情報の生成規則について説明します。

5.1 辞書のレコード定義 (最上位結合項目) を基にレポートを設計する

5.1 辞書のレコード定義（最上位結合項目）を基にレポートを設計する

SEWB+/REPOSITORY では、クライアントの EUR Professional Edition および SEWB+/CONSTRUCTION と連携して、効率良くレポート出力アプリケーションを作成する機能を提供しています。この機能では、辞書のレコード定義（最上位結合項目）を構成するデータ項目の定義情報から、EUR Professional Edition でレポートを設計する際に必要なデータのけた数、データ種別などの情報（フィールド定義情報）を自動的に生成できます。生成元となったレコード定義は SEWB+/CONSTRUCTION でのアプリケーション作成時に共通に利用されるため、レポートの設計内容とアプリケーションで出力するデータファイルのレコードの構造を一致させることができます。

フィールド定義情報を生成したあとの基本的なレポート設計および印刷方法を知りたい方は、マニュアル「EUR 帳票設計」および「EUR 帳票出力」を参考にしてください。

5.1.1 辞書のレコード定義（最上位結合項目）とレポート出力アプリケーションの関係

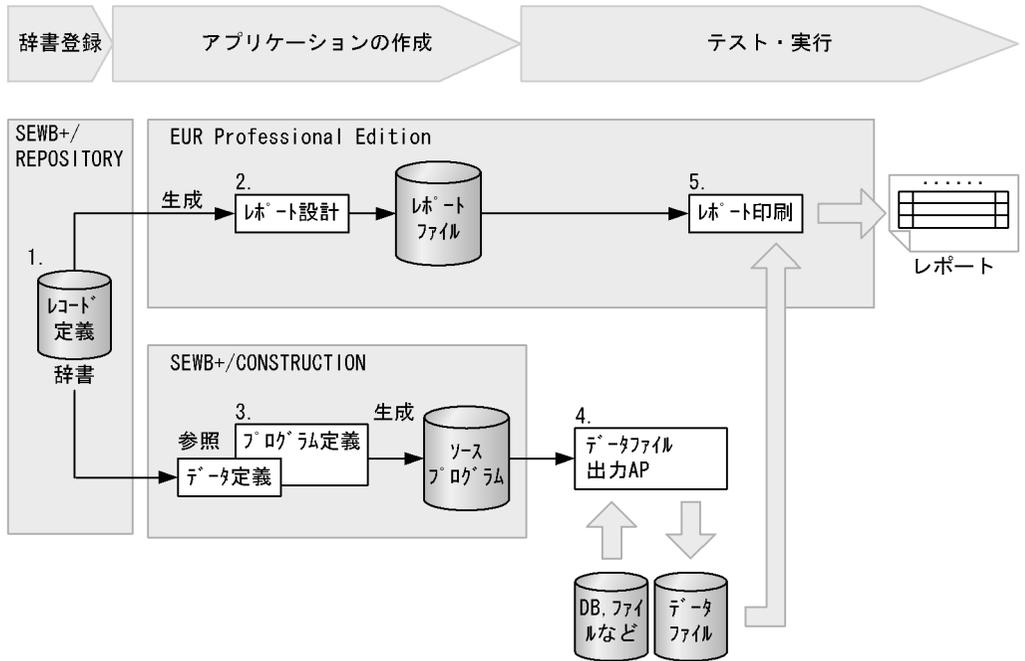
EUR Professional Edition と連携して効率良く、また、効果的に利用できる辞書のレコード定義（最上位結合項目）を用意するために、辞書のレコード定義（最上位結合項目）とレポート出力アプリケーションの関係について理解しておきましょう。辞書に登録したレコード定義の情報がどのようにレポート設計や、レポート出力アプリケーションに反映されていくのかを、図 5-1 に示します。SEWB+/REPOSITORY では、図 5-1 の流れに沿ったサンプルを提供しています。このサンプルの使用方法については、サンプルディレクトリ中の Readme.txt を参照してください。サンプルは次に示すディレクトリに格納されています。

SEWB+ 基本開発環境の組み込み先パス名¥Repository¥sample¥EUR_Sample

SEWB+ 基本開発環境の組み込み先パス名は、デフォルトでは次のように設定されています。システムドライブとは、Windows 2000 Server、Windows Server 2003 または Windows Server 2003 x64 が組み込まれているドライブです。

- Windows 2000 Server または Windows Server 2003 の場合
システムドライブ：¥Program Files¥HITACHI¥Sewb
- Windows Server 2003 x64 の場合
システムドライブ：¥Program Files (x86)¥HITACHI¥Sewb

図 5-1 辞書のレコード定義（最上位結合項目）がレポート出力アプリケーションに反映されるまで



辞書登録（1.）

辞書の作成者が、レコード定義を用意します。

アプリケーションの作成（2. ～ 3.）

アプリケーションの開発者が、辞書のレコード定義（最上位結合項目）を利用してレポート出力アプリケーションを作成します。

テスト・実行（4. ～ 5.）

作成したアプリケーションをテスト・実行して、データファイルの出力やレポートの印刷をします。

1. レコード構造を登録する

レポート出力に使うレコードの構造（レコード定義）を辞書に登録します。このレコード定義は、EUR Professional Edition でレポートを設計するとき、および SEWB+/CONSTRUCTION でデータ定義をするときに共通に利用することを目的とします。レポート出力のための辞書のレコード定義（最上位結合項目）については、「2.5.2(2) レポート出力に使用するレコード定義（最上位結合項目）を登録する」を参照してください。また、辞書のレコード定義（最上位結合項目）を登録する際に考慮する生成規則については、「5.1.3 フィールドの生成規則」を参照してください。

2. レポートを設計する

EUR Professional Edition のレポートエディタで、レポートに使用する辞書のレコー

5. EUR Professional Edition のレポート設計

ド定義（最上位結合項目）を選択し、レポートを構成するデータのフィールド定義情報を生成します。生成された情報を基にレイアウトを編集し、レポートを設計していきます。完成したレイアウトなどのレポート設計情報は、レポートファイル に保存します。設計手順については、「5.1.2 レポート設計の手順」を参照してください。

注

アプリケーションを WS 上で実行するレポートの場合は、フォームシートファイルに保存します。レポートファイル (.agr) およびフォームシートファイル (.fms) については、EUR Professional Edition のヘルプを参照してください。

3. SEWB+/CONSTRUCTION を使ってデータファイルを出力するアプリケーションを作成する

アプリケーションでは、EUR Professional Edition を使ってレポートを印刷するとき使用する CSV 形式や FIX 形式¹のデータ²を、日立 COBOL2002 の表計算プログラム入出力機能やテキストファイル機能などを使用して出力します。このアプリケーションで使用するデータファイルのデータ定義では、2. で利用したものと同一辞書のレコード定義（最上位結合項目）を参照して、データを定義します。そして、プログラム定義からソースプログラムを生成し、使用するコンパイラでビルドして実行可能ファイルを作成します。

注 1

FIX 形式データを扱う場合には、作成したデータファイル出力アプリケーションの仕様に合わせて、レポートファイルのデータ幅を編集する必要があります。レポートエディタを使ったデータ幅の編集方法については、EUR Professional Edition のヘルプを参照してください。

注 2

EUR Professional Edition では、非表示形式（バイナリ）データを印刷時の入力データとして扱えません。このため、非表示形式データは表示形式に変換するように、データファイル出力アプリケーションを作成する必要があります。表示形式への変換方法については、各プログラミング言語のマニュアルを参照してください。

4. データファイルを出力する

3. で作成したアプリケーションを実行すると、CSV 形式または FIX 形式のデータファイルが出力されます。

5. EUR Professional Edition を使ってレポートを印刷する

EUR Professional Edition をコマンドを使って起動するか、またはアプリケーションから呼び出して、レポートを印刷します。このとき、EUR Professional Edition が、2. で作成したレポートファイル（WS で実行する場合はフォームシートファイル）と 4. で出力したデータファイルを読み込み、レポートとして印刷します。印刷方法については、EUR Professional Edition のヘルプを参照してください。

5.1.2 レポート設計の手順

辞書のレコード定義情報を利用した、EUR Professional Edition のレポート設計の手順を説明します。詳細な使い方については、EUR Professional Edition のヘルプを参照してください。

(1) レポートの設計

設計手順を次に説明します。なお、EUR Professional Edition をクライアントにインストールしたあと、リポジトリブラウザを起動すると、ドキュメント種別の情報（ドキュメント種別の名称、テンプレートファイル、ファイルエクステンション）が自動的にサーバの運用環境に設定されます。

1. リポジトリブラウザでドキュメントを作成する場所を選びます。
2. [新規作成] - [ドキュメント]を選び、レポートファイル（またはフォームシートファイル）作成の指定をします。
このとき、レポート出力アプリケーションを実行する環境（PCまたはWS）に合わせて、ドキュメント種別とファイルエクステンションを選択してください。
 - 実行環境が PC の場合
ドキュメント種別は、EUR レポートファイル（PC 版）または EUR フォームシートファイル（PC，WS 版）
ファイルエクステンションは、.agr または .fms
 - 実行環境が WS の場合
ドキュメント種別は、EUR フォームシートファイル（WS 版）
ファイルエクステンションは、.fms

新しいレポートファイル（またはフォームシートファイル）が作成されて自動的にリポジトリに登録されます。
同時に、クライアント側の指定した場所にチェックアウトされ、EUR Professional Edition のウィンドウが表示されます。
3. EUR Professional Edition のウィンドウから [データ] - [最上位結合項目]を選びます。
辞書のレコードを選択するためのダイアログが表示されます。ここで目的のレコードを検索するために [検索開始]を選びます。
4. 表示されたレコード定義の一覧から目的のレコードを選択し、[OK] ボタンを選びます。
辞書のレコード定義情報から、レポートのフィールド定義情報が自動的に生成されます。この生成規則については、「5.1.3 フィールドの生成規則」を参照してください。
5. 自動的に生成された情報を基にして目的のレポートを設計します。
完成後、EUR Professional Edition を終了すると、レポートの情報がレポートファイル（またはフォームシートファイル）に保存されます。
6. レポートファイル（またはフォームシートファイル）をリポジトリにチェックインし

ます。

チェックインすると、レポート設計に利用した辞書のレコード定義（最上位結合項目）とそのファイルの間に自動的に関連が付けられます。

（２）レポート設計内容の変更

リポジトリブラウザで目的のレポートファイル（またはフォームシートファイル）を選び、クライアント側にチェックアウトします。そのファイルを開いて EUR Professional Edition を起動し、レポートの設計内容を変更します。

なお、前回 EUR Professional Edition でレポートを設計した時点から、生成元となった辞書のレコード定義（最上位結合項目）が変更されている場合には、最新のレコード定義情報を読み込むかどうかを確認するメッセージが表示されます。レコード定義情報を再度読み込むと、レイアウトが変更される場合があります。この場合には、レイアウトを見直して再編集してください。

5.1.3 フィールドの生成規則

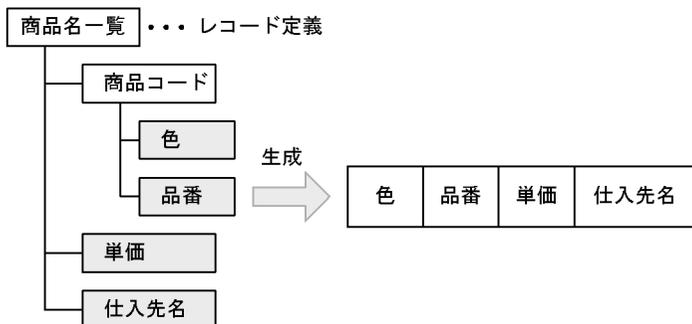
辞書のレコード定義（最上位結合項目）を構成するデータ項目の定義情報から、レポートの設計に必要なフィールド定義情報が生成されます。EUR Professional Edition を利用したレポート出力アプリケーション用の辞書のレコード定義（最上位結合項目）を登録する際には、ここで説明するフィールドの生成規則を考慮することをお勧めします。

（１）フィールドの生成

辞書のレコード定義（最上位結合項目）の結合関係で構成項目に相当するデータ項目から、フィールドが生成されます。結合項目からは、フィールドは生成されません。構成項目に結合項目が定義されている場合には、さらにその結合項目の構成項目が生成対象となります。フィールドが生成される順番は、構成項目が定義された順番に対応します。

（例）

レコード定義「商品名一覧」からは、次のようにフィールドが生成されます。



（２）フィールド定義情報の生成規則

フィールド定義情報として、フィールド名称、データ種別およびサンプルデータが生成

されます。この生成規則を、表 5-1 に示します。

また、言語区分「帳票定義」に定義したタイプからデータ種別およびサンプルデータが生成される規則を、表 5-2 に示します。

表 5-1 フィールド定義情報の生成規則

フィールド定義情報	データ項目の定義
フィールド名称	言語区分「帳票定義」の言語別の名前から生成される。言語別の名前が定義されていない場合には、データ項目名から生成される。
データ種別	言語区分「帳票定義」のタイプから変換される（表 5-2 を参照）
サンプルデータ	変換されたデータ種別に対応するサンプルデータが生成される（表 5-2 を参照）

表 5-2 データ種別とサンプルデータの生成規則

データ項目の定義項目		生成される情報	備考
分類区分	タイプ	データ種別記号 ¹ : サンプルデータ	
英数字文字列データ	文字データ	C: A ~ Z をけた数分繰り返し	
	画像データ	M: sample.bmp	2
	フリー定義	ユーザ指定: ユーザが指定したデータ	3
日本語文字列データ	文字データ	C: あ ~ んまでけた数分繰り返し	
	画像データ	M: サンプル.bmp	2
	フリー定義	ユーザ指定: ユーザが指定したデータ	3
整数データ	数値データ	I: 1 ~ 0 をけた数分繰り返し	
	集計対象外数値データ	B: 1 ~ 0 をけた数分繰り返し	
	特殊編集レベルデータ	SP: 1	2
	フリー定義	ユーザ指定: ユーザが指定したデータ	3
正整数データ	数値データ	I: 1 ~ 0 をけた数分繰り返し	
	集計対象外数値データ	B: 1 ~ 0 をけた数分繰り返し	
	特殊編集レベルデータ	SP: 1	2
	フリー定義	ユーザ指定: ユーザが指定したデータ	3
実数データ	数値データ	I: 1 ~ 0 をけた数分繰り返し	4
	集計対象外数値データ	B: 1 ~ 0 をけた数分繰り返し	4
	フリー定義	ユーザ指定: ユーザが指定したデータ	3
日付データ	年次データ	CY: 1996	2, 5

5. EUR Professional Edition のレポート設計

データ項目の定義項目		生成される情報	備考
分類区分	タイプ	データ種別記号 ¹ : サンプルデータ	
	月次データ	MO : 199603	2, 6
	日次データ	DA : 19960331	2, 7
	日付・時刻データ	HS : 1996/03/31 12:42:07	2, 8
	文字データ	C : 19960331	2
	フリー定義	ユーザ指定ユーザが指定したデータ	3
時刻データ	文字データ	C : 12 : 42 : 07	2
	フリー定義	ユーザ指定 : ユーザが指定したデータ	3
分類なし	文字データ	C : A ~ Z をけた数分繰り返し	
	数値データ	I : 1 ~ 0 をけた数分繰り返し	4
	集計対象外数値データ	B : 1 ~ 0 をけた数分繰り返し	4
	年次データ	CY : 1996	2, 5
	月次データ	MO : 199603	2, 6
	日次データ	DA : 19960331	2, 7
	日付・時刻データ	HS : 1996/03/31 12:42:07	2, 8
	画像データ	M : sample.bmp	2
	特殊編集レベルデータ	SP : 1	2
	フリー定義	ユーザ指定 : ユーザが指定したデータ	3

注 1

データ種別記号とデータ種別の対応は、表 5-3 を参照してください。

注 2

データ項目に定義されたけた数に関係なく、固定のサンプルデータが生成されます。

注 3

タイプに「フリー定義」を指定した場合は、表 5-3 に示したデータ種別記号か、または EUR で新たに追加されたデータ種別記号をフリー定義に指定します。例えば、バーコードデータにしたい場合は「R」を指定すると、フィールド定期情報の生成時にバーコードとして生成されます。

注 4

小数部のけた数が 1 以上の場合、'!' を付けた形式のサンプルデータが生成されます。

注 5

年次データの場合、けた数は 4 を指定します。

注 6

月次データの場合、けた数は 6 を指定します。

注 7

日次データの場合、けた数は 8 を指定します。

注 8

時刻・日付データの場合、けた数は 4 ~ 19 を指定します。

表 5-3 データ種別記号とデータ種別

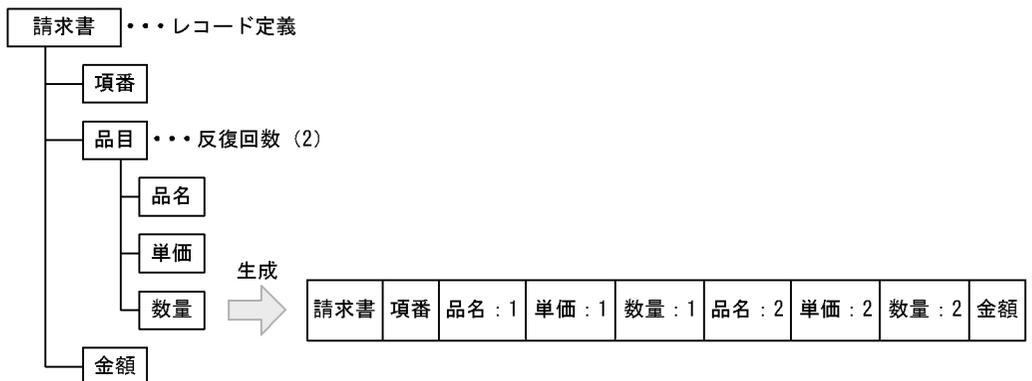
データ種別記号	データ種別	データの例
C	文字列	ABCD, あいう, 12:42:07, 19960331
I	数値	1234
B	集計対象外数値	1234
DA	日付(日次)	19960331
MO	日付(月次)	199603
CY	日付(年次)	1996
HS	日付(日付・時刻)	1996/03/31 12:42:07
M	画像	sample.bmp, サンプル.bmp
SP	特殊編集レベル	1

(3) フィールドの反復生成

フィールドの生成対象に当たるデータ項目の反復回数に 2 以上を定義している場合には、反復回数分、フィールドが繰り返して生成されます。

(例)

データ項目「品目」の反復回数に「2」が定義されている場合、次のようにフィールドが生成されます。



- フィールド名称の生成規則

フィールド名称には、「データ項目名:n」が生成されます (n は反復回数を示します)。なお、データ項目の反復が入れ子になっている場合 (例えば、上記の例で考えてみる

と、品名にも反復回数に「2」が定義されている場合)のフィールド名称には、「データ項目名:n1:n2」が生成されます(n1は1次元目の反復回数、n2は2次元目の反復回数を示します)。

また、反復回数の情報を付加するとフィールド名称が重複する場合には、一意の名称にするために、データ項目名に「m |」が付加されます(mは同一名称の出現回数を示します)。

6

APPGALLERY Enterprise の クライアントアプリケーション作成

SEWB+/REPOSITORY では、アプリケーション開発環境である APPGALLERY Enterprise (バージョン 05-00 以降) と連携し、クライアントアプリケーションの作成を支援しています。APPGALLERY Enterprise から、OpenTP1 通信用電文の定義に辞書のレコード定義情報を利用したり、GUI (グラフィカルユーザインタフェース) 作成にデータ項目の定義情報や辞書の環境設定を利用したりできます。SEWB+/REPOSITORY と APPGALLERY Enterprise とを連携すれば、C/S (クライアントサーバ) システムのような大規模なシステムを効率良く開発できます。

この章では、辞書の作成者に必要な、連携のための基礎知識について説明します。

6.1 辞書のレコード定義 (最上位結合項目) を基に OpenTP1 クライアントアプリケーションを作成する

6.2 データ項目と辞書の環境設定を基に GUI を作成する

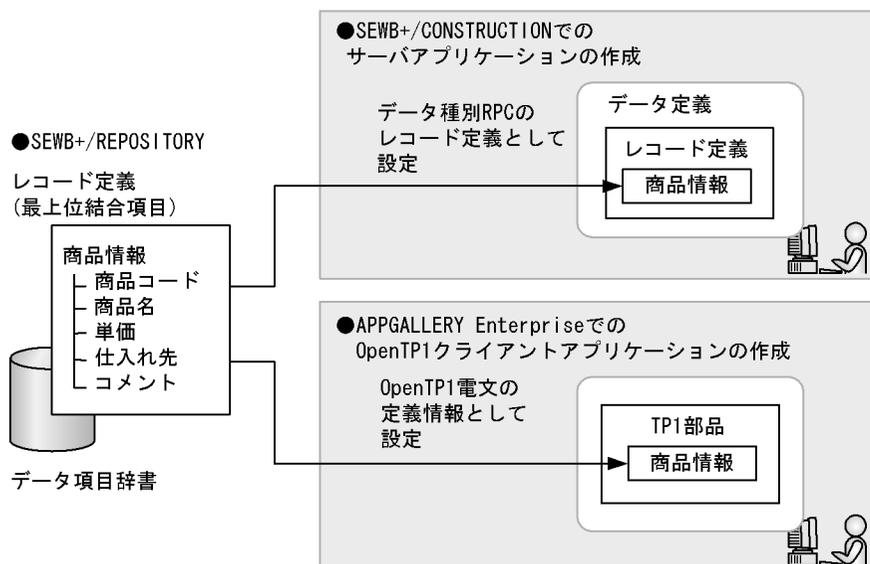
6.1 辞書のレコード定義（最上位結合項目）を基に OpenTP1 クライアントアプリケーションを作成する

SEWB+/REPOSITORY では、APPGALLERY Enterprise と連携して、C/S システムの OpenTP1 クライアントアプリケーションの作成を支援する機能を提供しています。この機能では、APPGALLERY Enterprise から、SEWB+/REPOSITORY で管理している辞書のレコード定義（最上位結合項目）の定義情報が利用されます。

ここでは、連携の概要と、SEWB+/REPOSITORY 側でどのような準備をすればよいかについて説明します。利用方法や、OpenTP1 のサーバアプリケーションとクライアントアプリケーションの関係については、マニュアル「APPGALLERY プログラマーズガイド」を参照してください。

利用の概要を、図 6-1 に示します。

図 6-1 辞書のレコード定義（最上位結合項目）の利用の概要



注

APPGALLERY Enterprise でこの機能を利用するには、TP1/Client Objects 02-00以降、および TP1/Client/P 03-04 以降がインストールされていて、さらに TP1/Client Objects (TP1 部品) が APPGALLERY にインポートされている必要があります。

辞書登録

辞書の作成者がレコード定義を用意します。OpenTP1 を利用した C/S システムで、通信に使用するレコード構造（レコード定義）を辞書に登録します。このレコード

定義は、APPGALLERY Enterprise でクライアントアプリケーションを作成するとき、および SEWB+/CONSTRUCTION でサーバアプリケーションを作成するときに共通に利用することを目的とします。

OpenTP1 クライアントアプリケーションの作成

APPGALLERY Enterprise でクライアントアプリケーションを作成します。TP1 部品に設定する電文の定義情報に、辞書のレコード定義（最上位結合項目）を利用します。

OpenTP1 サーバアプリケーションの作成

SEWB+/CONSTRUCTION でサーバアプリケーションを作成します。データ種別「RPC」のデータ定義に辞書のレコード定義（最上位結合項目）を利用します。

このように、サーバ側とクライアント側のアプリケーションの両方で、共通のレコード定義を利用するため、通信するデータの整合性を保てます。

6.2 データ項目と辞書の環境設定を基に GUI を作成する

SEWB+/REPOSITORY では、APPGALLERY Enterprise と連携して、C/S システムのクライアントアプリケーションでの GUI (グラフィカルユーザインタフェース) の作成を支援する機能を提供しています。この機能では、APPGALLERY Enterprise から、SEWB+/REPOSITORY で管理しているデータ項目と環境構築ユティリティの設定情報が利用されます。

ここでは、連携の概要と、SEWB+/REPOSITORY 側でどのような準備をすればよいかについて説明します。詳細な利用方法については、マニュアル「APPGALLERY プログラマーズガイド」を参照してください。

6.2.1 SEWB+/REPOSITORY と APPGALLERY Enterprise で作成する GUI の関係

APPGALLERY Enterprise は、さまざまな部品を組み合わせてアプリケーションを開発する、アプリケーション開発環境です。APPGALLERY Enterprise では、部品を組み合わせて視覚的にプログラミングするためのキャンパスという作業領域を使用します。このキャンパス上に部品を配置したり、部品を編集したりすることでアプリケーションを作成していきます。また、ウィンドウとして使用する部品およびウィンドウ上に配置された部品を編集するには、フォームエディタというツールを使用します。

SEWB+/REPOSITORY では、クライアント画面の GUI 作成に使用する部品に対し、そのプロパティ (部品の動作や表示などに関する属性) の設定に関する情報を一元管理します。部品を使用する際に SEWB+/REPOSITORY で管理する設定値を利用することで、複数の開発者によるチーム開発や大規模なシステムの開発でも、GUI の統一を図れます。また、効率良く GUI を作成、変更できます。

次の二通りの方法があります。

- データ項目とマッピング情報を利用する
- プロパティセットを利用する

なお、APPGALLERY Enterprise で作成したファイルをドキュメントとしてリポジトリに登録すれば、バージョンや関連を設定して管理・運用できます。辞書のデータ項目と、そのデータ項目を利用した APPGALLERY Enterprise のドキュメントには、自動的に関連 (関連種別「辞書参照」) が設定されます。

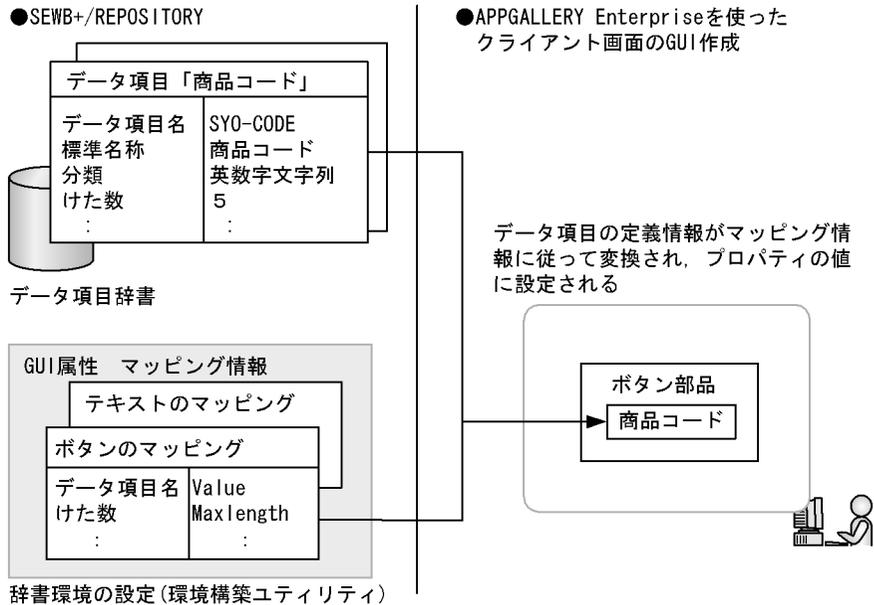
6.2.2 データ項目とマッピング情報を利用した GUI の統一

データ項目の名称やけた数などの定義情報を、部品のプロパティ値として利用できます。

データ項目を C/S システムのクライアント側のアプリケーション作成に利用することで、システム全体でデータの整合性を保てます。データ項目の定義情報を APPGALLERY Enterprise のどの部品に対応させ、プロパティ値に変換するかといった規則は、環境構築ユティリティの GUI 属性で設定しておきます。SEWB+/REPOSITORY で一元管理する変換規則を「マッピング情報」といいます。

利用の概要を、図 6-2 に示します。

図 6-2 データ項目とマッピング情報の利用の概要



辞書登録

辞書の作成者がデータ項目を登録します。

辞書の環境設定

辞書の作成者、またはシステム管理者が、環境構築ユティリティを使ってマッピング情報を設定します。マッピング情報は GUI 作成に使う部品ごとに用意します。詳細は、「3.1.1(12) GUI 属性を設定する」を参照してください。

クライアントアプリケーションの GUI 作成

アプリケーションの開発者が、APPGALLERY Enterprise を使って GUI を作成します。

1. データ項目をドラッグ&ドロップ
データ項目種別に「単項目」を指定したデータ項目を利用できます。リポジットリブラウザから GUI にしたいデータ項目をドラッグし、APPGALLERY Enterprise のキャンバス上にドロップします。
2. マッピング情報を選択

6. APPGALLERY Enterprise のクライアントアプリケーション作成

データ項目をキャンバスにドロップすると、[マッピング情報の選択]ダイアログが表示されます。適用したい部品に応じたマッピング情報を選択してください。

3. GUI 作成

選択したマッピング情報に対応する GUI 部品が、APPGALLERY Enterprise 上に配置されます。この部品には、データ項目の定義情報がマッピング情報に従って変換され、自動的にプロパティに設定されます。

この部品を使用して目的の GUI を作成していきます。

6.2.3 プロパティセットを利用した GUI の統一

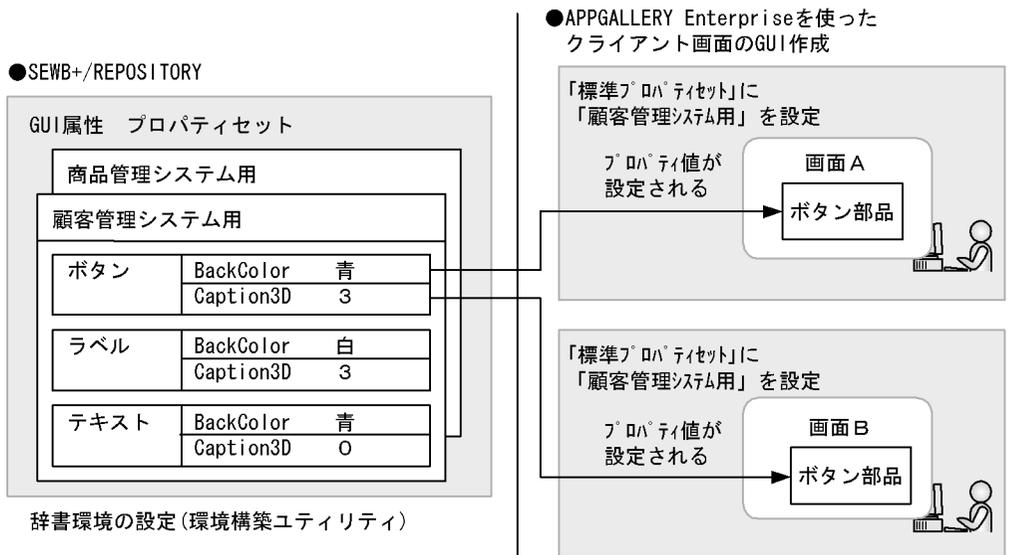
APPGALLERY Enterprise では、アプリケーションのウィンドウを、GUI で使用する部品を使って作成します。このとき、複数の部品に同じプロパティを設定するには、複数の部品を選択して一括してプロパティを設定するか、または、すでに配置された部品を複製して同じプロパティを持つ部品を作成します。しかし、このような方法では、部品の選択を忘れるなどでプロパティの統一に漏れが生じ、結果として GUI を統一できない場合があります。

SEWB+/REPOSITORY では、各部品ごとに統一したいプロパティをまとめ、プロパティセットとして登録し、一元管理できます。APPGALLERY Enterprise での GUI 作成で部品を配置する際、プロパティセットを利用することで部品のプロパティを統一できます。

複数のプロパティセットを登録できるため、異なるシステムや業務を並行開発している場合でも、対象ごとにプロパティセットを用意しておけます。

利用の概要を、図 6-3 に示します。

図 6-3 プロパティセットの利用の概要



辞書の環境設定

辞書の作成者またはシステム管理者が、環境構築ユーティリティを使ってプロパティセットを登録します。GUIを統一する対象ごとにプロパティセットを用意します。詳細は、「3.1.1(12) GUI属性を設定する」を参照してください。

また、APPGALLERY Enterprise から、部品パレットの部品情報をプロパティセットとして登録できます。マニュアル「APPGALLERY プログラマーズガイド」を参照してください。

クライアントアプリケーションのGUI作成

アプリケーションの開発者が、APPGALLERY Enterprise を使って GUI を作成します。

1. プロパティセットを選択

リポジトリブラウザの [オプション] ダイアログを表示し、[標準プロパティセット] タブで利用するプロパティセットを選択します。

2. GUI作成

APPGALLERY Enterprise の部品パレットから GUI 作成に使用する部品を選択して、キャンバス上に配置します。この部品には、1. で選択したプロパティセットの情報が自動的に設定されます。この部品を使用して目的の GUI を作成していきます。

付録

付録 A	標準データ項目辞書の紹介
付録 B	ERwin/ERX からの辞書登録
付録 C	命名ルールインタフェース
付録 D	検査項目インタフェース
付録 E	標準提供の同一項目用業務ルール
付録 F	用語解説

付録 A 標準データ項目辞書の紹介

SEWB+/STANDARD-DICTIONARY および SEWB+/STANDARD-SUBROUTINE を利用すれば、効率良く辞書の設計や登録の作業ができ、また、辞書の品質の向上を図れます。ここでは、これらのプログラムプロダクトの概要を紹介します。

付録 A.1 SEWB+/STANDARD-DICTIONARY (標準データ項目辞書)

SEWB+/STANDARD-DICTIONARY は、「標準データ項目辞書」と「データ項目辞書生成機能」を提供しています。標準データ項目辞書には、開発するシステムの業種、業務を問わず、汎用的に使用できる標準データ項目と、このデータ項目に対応する標準業務ルールが提供されています。データ項目辞書生成機能を使うことで、この標準データ項目辞書と、プロジェクト独自に使用するデータ項目が自動的に組み合わせられ、効率良く辞書を構築できます。

詳細は、マニュアル「SEWB+/STANDARD-DICTIONARY 標準データ項目辞書 使用の手引」を参照してください。

- 標準データ項目
一貫した命名基準に従ったデータ項目名が使用されているため、データ項目名の標準化、一元管理を促進できます。標準データ項目は、SEWB+/REPOSITORY の辞書に登録したとき、ドメインとして位置づけられます。
- 標準業務ルール
標準データ項目に対応する標準的な処理が、業務ルールとして提供されています。処理に使用している部品は、SEWB+/STANDARD-SUBROUTINE で提供されている COBOL 外部サブルーチンを利用しています。

付録 A.2 SEWB+/STANDARD-SUBROUTINE (標準データ項目辞書サブルーチン)

SEWB+/STANDARD-SUBROUTINE は、標準的なデータ項目に対応する処理（文字列の字種チェック、日付・時間のチェックや編集処理など）を実行する部品を COBOL 外部サブルーチンとして提供しています。SEWB+ ツールと日立 COBOL を連携したアプリケーション開発に利用できます。データ項目単位に処理できるよう設計されているため、データ中心アプローチに基づく部品化、再利用を促進できます。

詳細は、マニュアル「SEWB+/STANDARD-SUBROUTINE 標準サブルーチン 使用の手引」を参照してください。

付録 B ERwin/ERX からの辞書登録

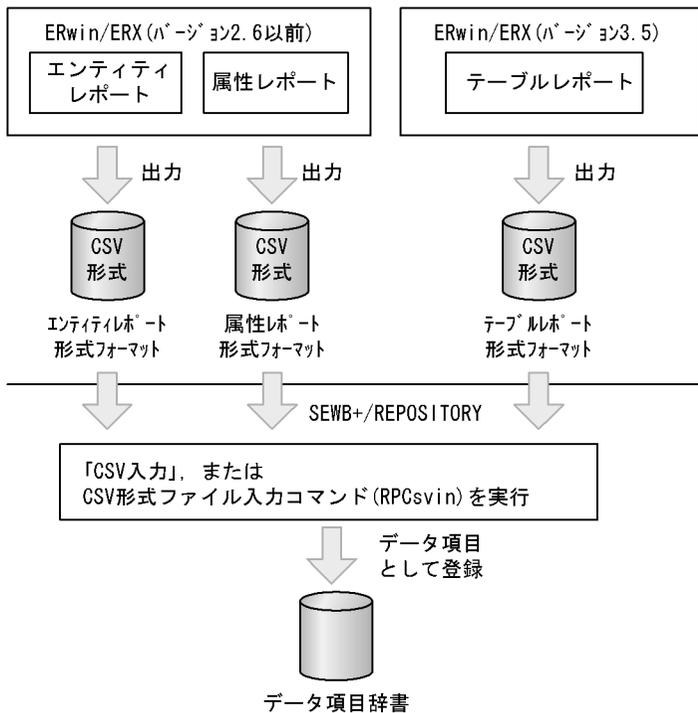
付録 B.1 ERwin/ERX からデータ分析情報を連携する

SEWB+/REPOSITORY では、ERwin/ERX からデータ項目辞書への連携をサポートしています。ここでは、CSV 形式ファイルを経由して行う、ERwin/ERX で作成したデータ分析情報の連携について説明します。

ERwin/ERX からは、3 通りのレポート形式から情報を辞書に登録できます。辞書に登録できるレポート形式は ERwin/ERX のバージョンによって異なります。

辞書登録の概要を、図 B-1 に示します。

図 B-1 ERwin/ERX からの辞書登録



1. ERwin/ERX のレポート（エンティティレポート、属性レポートまたはテーブルレポート）から、CSV 形式ファイルを出力する

SEWB+/REPOSITORY の辞書へ連携するには、各レポート形式に専用のフォーマットの CSV 形式ファイルを使用します。ERwin/ERX で、このマニュアルの説明に従って出力項目を設定し、CSV 形式ファイルを出力してください。

注

ERwin/ERX のフィルタ機能を使って一部のエンティティの内容だけを出力した

レポートを辞書登録したとき、FK 属性（外部キー属性）に対応する PK 属性（主キー属性）の定義が同一レポート内に存在しないと、FK 属性のデータ項目は対応する PK 属性を継承できない場合があります。FK 属性を含むエンティティをフィルタ出力するときは、対応する PK 属性を含むエンティティも同時にフィルタ出力対象にしてください。

2. 辞書登録の準備をする

ここからの SEWB+/REPOSITORY の作業は、システム管理者が行ってください。必要に応じて、次の準備をします。

- 辞書フォルダを準備する
- ERwin/ERX タイプを SEWB+/REPOSITORY の分類区分やタイプなどに変換するためのタイプマッピング規則を、環境構築ユティリティでカスタマイズする（ただし、標準で用意されている規則を適用する場合はカスタマイズは不要）

3. CSV 形式ファイルの入力を実行する

登録する CSV 形式ファイル、登録先の辞書フォルダなどを指定して実行します。[スタート]メニューの[プログラム]または[すべてプログラム]から [Sewb+] - [Repository] - [CSV 入力] を選ぶと表示されるダイアログから実行する場合は、[CSV 入力オプション]ダイアログで必ず次のオプションを指定してください。

- 「CSV 入力フォーマット種別」で「ERwin/ERX」を選ぶ
- 「レポート形式選択」で該当するレポート形式を選ぶ

CSV 形式入力コマンドを使ってコマンドラインから実行する場合は、必ず引数「/M ファイルフォーマット種別」を指定してください。この引数には、ERwin/ERX で出力したレポート形式に対応した値を指定します。

- エンティティレポート形式：/M ewe
- 属性レポート形式：/M ewa
- テーブルレポート形式：/M ewt

ERwin/ERX でエンティティの属性のカラム名を一意に定義していれば、エンティティレポートと属性レポートの情報をデータ項目辞書で統合できます。統合するには、先にエンティティレポートをデータ項目辞書に入力し、次に属性レポートをデータ項目辞書に入力します。統合する場合には、CSV 形式ファイル入力コマンドの引数「/S 辞書の登録方法」に「name」を指定、または [CSV オプション] ダイアログの「辞書の登録方法」に「データ項目名/業務ルール名で更新する」を指定してください。なお、統合するとき、エンティティレポートから変換されたデータ項目に、属性レポートから変換されたデータ項目の情報が上書きされる場合があります。

CSV 形式の記述規則や CSV 形式ファイルの入力の詳細は、「4.1 CSV 形式ファイルを使って辞書登録と出力をする」を参照してください。

付録 B.2 エンティティレポートの連携

ここでは、ERwin/ERX（バージョン 2.6 以前）で作成したエンティティレポートの辞書

登録について説明します。

(1) ERwin/ERX でのエンティティレポートの出力設定

ERwin/ERX で次に示す設定をして CSV 形式ファイルを出力してください。

- サーバの対象 SQL DBMS の設定
 - 使用するデータベースが ORACLE の場合
対象データベース (SQL ベース): ORACLE
 - 使用するデータベースが HiRDB の場合
対象データベース (SQL ベース): DB2
- CSV 形式ファイルへの出力項目の設定
エンティティレポートから, SEWB+/REPOSITORY の辞書に連携するために必要な出力項目を設定します。出力項目と指定順序を, 表 B-1 に示します。指定順序が列の並び順になります。このため, 必ず指定順序に従って設定してください。

表 B-1 エンティティレポートの出力項目と指定順序

指定順	出力項目 (オプション名)	
	ERwin/ERX 日本語版	ERwin/ERX 英語版
1	テーブル名 (テーブルオプション)	Table Name(Table Option)
2	エンティティ名 (エンティティオプション)	Entity Name(Entity Option)
3	属性名 (属性オプション)	Attribute Names(Attribute Option)
4	カラム / タイプ (属性オプション)	Columns/Types(Attribute Option)
5 ¹	コメント (エンティティオプション)	Note(Entity Option)
5 ¹	定義 (エンティティオプション)	Definition(Entity Option)
6 ²	外部キー (リレーションシップオプション)	Foreign Keys(Relationship Option)
7 ³	カンマ区切り (レポート形式)	Comma Delimited(Report Format)
8 ³	ヘッダー (表示形式)	Header(Multi-Valued Format)
9 ³	マージ (表示形式)	Merge(Multi-Valued Format)
10 ³	改行コードの除去 ⁴	Remove Newline

注 1
どちらかを選択してください。両方の同時指定はできません。

注 2
属性レポートを同時入力する場合は, 必ず指定してください。

注 3
指定順序どおりでなくてもかまいません。

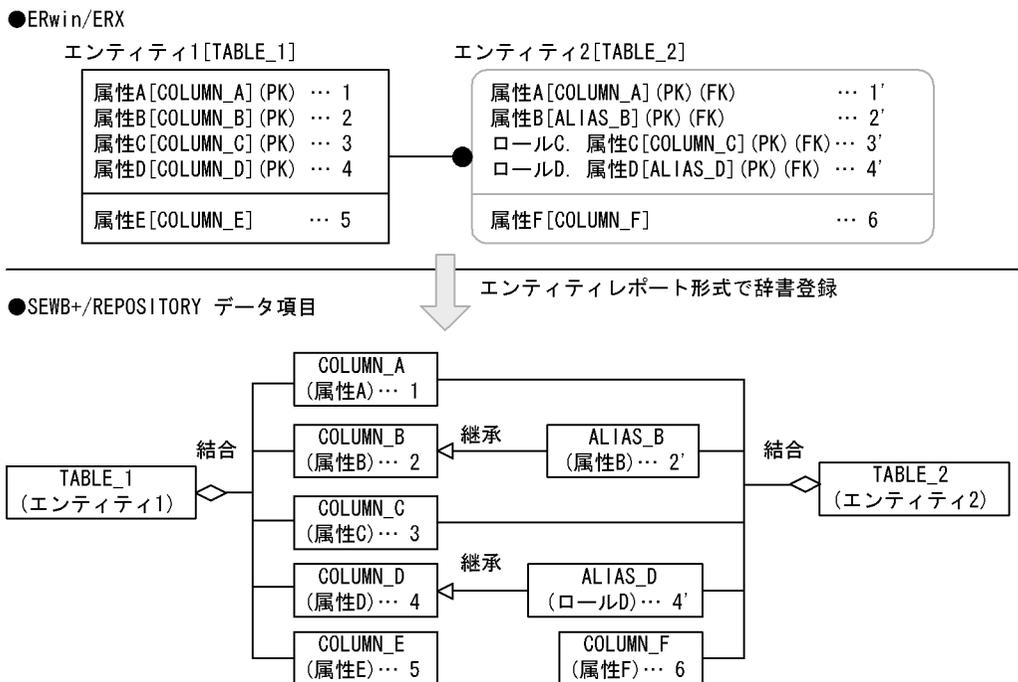
注 4

ERwin/ERX 2.5 以前のバージョンでは「新規行の除去」となります。

(2) エンティティレポートからデータ項目辞書への変換規則

依存関係のあるエンティティレポートを辞書登録する場合の変換規則について説明します。エンティティレポート形式の辞書登録の例を、図 B-2 に示します。

図 B-2 エンティティレポート形式の辞書登録の例



- 注
- ERwin/ERXのエンティティ内の[]は、テーブル名またはカラム名を示します。
 - リレーションシップでロール名が定義された属性は「ロール名. カラム名」の形式で示します。
 - SEWB+/REPOSITORY側の()は、データ項目の標準名称を示します。

エンティティレポート形式のCSV形式ファイルを辞書に登録するときの指定によって、登録されるデータ項目およびデータ項目の関係が異なります。図 B-2 は、属性レポートの同時入力の指定と、物理名の異なる外部キーを識別する指定をした場合の登録例です。

(a) PK 属性と FK 属性の定義を、データ項目間の継承関係として登録する

- エンティティレポートと属性レポートを同時に入力
PK 属性と FK 属性の定義をデータ項目間の継承関係として登録するには、エンティティレポートと属性レポートを同時に入力します。CSV形式ファイル入力コマンドで引数「/A 同時に入力する属性レポートファイル名」を指定するか、または[CSV入力]ダイアログで「同時に入力する ERwin/ERX 属性レポートファイル名」を指定します。
- エンティティレポートだけを単独で入力
属性レポートを同時に入力しない場合には、図 B-2 で示した「2」と「2'」、「4」と「4'」

のデータ項目間の継承関係は登録されません。

(b) 物理名の異なる外部キーを識別して登録する

物理名の異なる外部キーを識別して登録するには、CSV形式ファイル入力コマンドで引数「/Z 物理名の異なる外部キーを識別する」を省略するか、[CSV入力オプション]ダイアログの「物理名の異なる外部キーを識別する」のチェックをオンにします。

この指定をしない場合には、図 B-2 で示した「2」と「4」のデータ項目は登録されません。

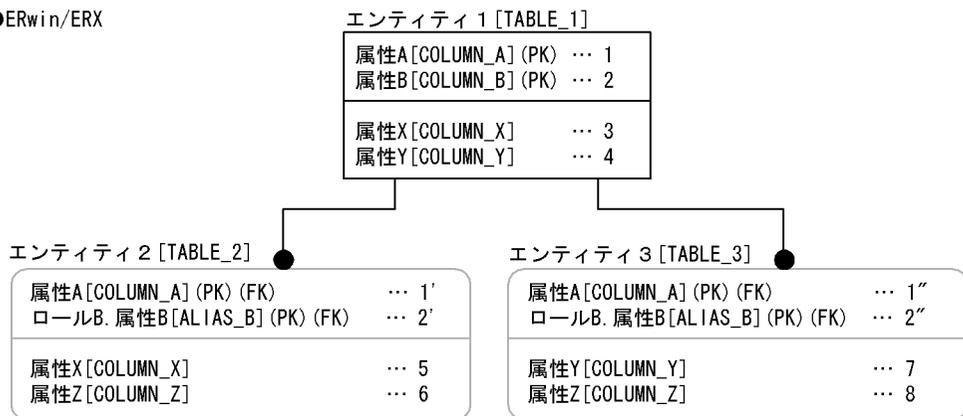
(3) エンティティレポートからデータ項目辞書への変換規則（同一カラム名をマージする場合）

エンティティレポート中に、同一のカラム名が定義された属性情報が複数ある場合、その情報をマージしてデータ項目辞書に登録できます。

この場合のエンティティレポートの変換規則を、図 B-3 に示します。

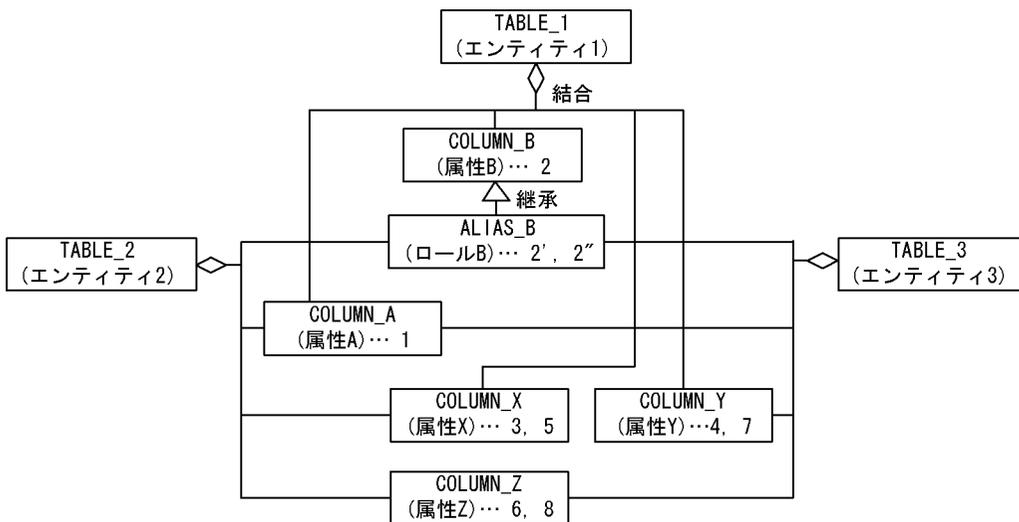
図 B-3 エンティティレポート形式の辞書登録の例（同一カラム名をマージする場合）

●ERwin/ERX



エンティティレポート形式で辞書登録

●SEWB+/REPOSITORY データ項目



- 注 ・ ERwin/ERXのエンティティ内の[]は、テーブル名またははカラム名を示します。
- ・ リレーションシップでロール名が定義された属性は「ロール名. 属性名」の形式で示します。
- ・ SEWB+/REPOSITORY側の()は、データ項目の標準名称を示します。

エンティティレポート形式のCSV形式ファイルを辞書に登録するときの指定によって、登録されるデータ項目およびデータ項目の関係が異なります。図 B-3 は、属性レポートの同時入力の指定と、物理名の異なる外部キーを識別する指定をした場合の登録例です。

(a) 同一カラム名をマージして登録する

同一カラム名をマージして辞書に登録するには、CSV形式ファイル入力コマンドで引数「/R 同一カラム名のマージ有無」に「use」を指定するか、または[CSV入力オプション]ダイアログの「同一カラム名の属性をマージする」のチェックをオンにします。

なお、FK属性のカラム名と、その移行先のPK属性のカラム名が同じ場合は、この指定に関係なく、必ずマージされて登録されます。詳細は、「(2) エンティティレポートからデータ項目辞書への変換規則」を参照してください。

(b) マージされる属性の条件

同一カラム名の属性のマージでは、マージされる属性間で、属性の値が一致している必要があります。

マージされる属性間の条件を、表B-2に示します。条件は、エンティティレポートを単独で入力する場合と、エンティティレポートと属性レポートを同時に入力する場合とで異なります。

表B-2 マージされる属性間の条件

属性情報		マージする属性間の条件	
		エンティティレポートを単独で入力	エンティティレポートと属性レポートを同時に入力
カラム名			
属性名			
カラムデータタイプ (精度、スケールを含む)			
デフォルト値		-	
移行先を示す情報	親テーブル名	×	
	ベース名	-	
主キー		×	×
外部キー		×	×
コメントまたは定義		-	×

(凡例)

：一致しなければならない

：マージする属性の双方が移行元の情報を持っている場合、移行元の情報が一致している場合はマージできる。また、マージする属性の一方だけ移行元の情報を持っている場合、またはマージする属性の双方が移行元の情報を持っていない場合はマージできる(マージする属性の双方が同じ移行元の情報を持っている場合、およびマージする属性の一方だけ移行元を示す情報を持っている場合は、移行元を示す情報を基に、マージしたデータ項目に継承の関連が付けられる)

×：一致しなくてもよい

-：該当しない

注

コメントまたは定義の情報が、マージ対象の複数の項目に設定されている場合は、その中で最も長い情報が、データ項目のコメントに設定されます。

(4) データ項目辞書とエンティティレポート形式の対応

データ項目辞書の定義項目とエンティティレポート形式の対応を、表 B-3 に示します。

表 B-3 データ項目辞書とエンティティレポートの対応

データ項目辞書の定義項目		エンティティレポート形式		
		結合項目 ¹	構成項目	
			単項目	結合項目とその構成項目 結：結合項目 構：構成項目
名称	データ項目名	テーブル名	カラム名 ²	結：カラム名 ² 構：カラム名+サフィックス ²
	標準名称	エンティティ名	属性名 ³	結：属性名 ³ 構：属性名+サフィックス ^{2, 3}
	フリガナ	×	×	結：× 構：×
属性	分類	結合データ	カラム名 ^{4, 5}	結：結合データ 構：カラム名 ^{4, 5}
	けた数	-	カラム名 ^{4, 5}	結：- 構：カラム名 ^{4, 5}
	小数部けた数	-	カラム名 ^{4, 5}	結：- 構：カラム名 ^{4, 5}
	反復回数	-	×	結：- 構：×
付加情報	フィールド 1 ~ n	×	×	結：× 構：×
	コメント	エンティティコメント/定義	* 属性コメント/定義	結：* 属性コメント/定義 構：×
言語別詳細情報	名前	テーブル名	カラム名 ²	結：カラム名 ² 構：カラム名+サフィックス ²
	タイプ	-	カラム名 ^{4, 5}	結：- 構：カラム名 ^{4, 5}
	タイプ修飾情報 ⁶	-	×	結：- 構：×
	取りうる値	-	×	結：- 構：×
	初期値	-	* デフォルト値	結：- 構：* ⁷
	言語別フィールド	×	×	結：× 構：×

(凡例)

- × : ERwin/ERX 側にデータ項目辞書に対応する情報がない項目
- : データ項目辞書に値が設定されない項目
- * : 属性レポートから取得される情報。属性レポートを同時入力しない場合は「×」

注 1

エンティティから変換されるデータ項目は、最上位結合項目となります。

注 2

属性名の末尾に出力される (PK) などのキー情報は削除されます。

注 3

カラム名の空白で区切られた情報の一つ目から取得されます。

注 4

カラム名の空白で区切られた情報の二つ目から取得されます。

注 5

変換される分類とタイプは、「(5) ERwin/ERX タイプマッピングの規則」を参照してください。

注 6

「COBOL 又は OOCOBOL」で定義する編集文字列、「IDL(CORBA)」で定義するシーケンスを指します。

注 7

構成項目のサフィックスが `-char` の場合はデフォルト値、`-length` の場合は デフォルト値の文字列長が設定されます。

規則

- データ項目は通常、継承関係の最上位にあるデータ項目として登録されます。
- 属性レポートとの同時入力で、CSV 形式ファイル入力コマンドの引数「/Z 物理名の異なる外部キーを識別する」、または [CSV 入力オプション] ダイアログの「物理名の異なる外部キーを識別する」を指定すると、データ項目の継承関係も登録されます。この場合、PK 属性とカラム名が異なる FK 属性は、PK 属性のベース名と同じ属性名の PK 属性データ項目を継承します。
- 継承関係の下位にあるデータ項目として登録されたとき、表 B-2 で「×」または「-」以外が記された定義項目には、上位のデータ項目の情報は継承されません。また、「×」が記された定義項目の継承モードは、環境構築ユティリティのデータ項目の継承モードの設定に従います。継承モードについては「3.1.1(3)(b) データ項目の継承モードの設定」を参照してください。
- CSV 形式ファイル入力コマンドの引数「/W 更新対象属性」、または [CSV 入力オプション] ダイアログの「CSV 入力フォーマット種別」で、ERwin/ERX 属性だけを更新する指定をすると、表 B-2 で「×」または「-」以外が記された定義情報のうち言語別名前を除く情報だけを更新できます。

(5) ERwin/ERX タイプマッピングの規則

ERwin/ERX で指定したタイプから、SEWB+/REPOSITORY のデータ項目の分類とタイプへ変換されます。変換にはタイプマッピングの規則が適用され、ERwin/ERX タイプや条件に応じた分類、タイプ、けた数、小数部けた数が設定されます。

標準で用意されているタイプマッピングの規則については、使用するデータベースの種類に応じて次から選択してください。

ORACLE を使用する場合

- 変換規則 A (表 B-4, B-5 参照) SEWB+ リポジトリセットの SEWB+/REPOSITORY 02-01 以降
- 変換規則 B (表 B-6, B-7 参照) SEWB+ リポジトリセットの SEWB+/REPOSITORY 02-00 との互換用

HiRDB を使用する場合

- 変換規則 C (表 B-8, B-9 参照)

適用する変換規則は、CSV 形式ファイル入力コマンドの引数「/E ファイルフォーマット種別 (02-01 互換用)」または「/B データベース種別」で指定するか、[CSV 入力オプション] ダイアログの「対象データベース選択」で指定します。

なお、これらの変換をするには、あらかじめ環境構築ユーティリティで言語別タイプをそれぞれの分類で指定できるタイプとして登録しておく必要があります。標準に設定されている分類とタイプの対応については「2.4.1(3)(a) 分類」を参照してください。

変換規則を表 B-4 から表 B-9 に示します。

表 B-4 ERwin/ERX のタイプの変換規則 A-1 (ORACLE)

ERwin/ERX			SEWB + /REPOSITORY		
データベースタイプ	精度・長さ	スケール	分類	けた数	小数部けた数
CHAR	[n] (255)	×	英数字文字列	n	-
CHARACTER	[n] (255)	×	英数字文字列	n	-
CHARZ	[n] (255)	×	英数字文字列	n	-
CURSOR	×	×	なし	-	-
DATE	×	×	日付	7	-
DECIMAL	p (38)	[s] (0)	実数 ¹	p	s ³
DISPLAY	p (38)	[s] (0)	なし	p	s ⁴
FLOAT	[p] (126)	×	実数	p	-
INTEGER	×	×	整数	38	-
LONG	[n] (16777215)	×	英数字文字列	n	-
LONG RAW	[n] (16777215)	×	なし	n	-

ERwin/ERX			SEWB + /REPOSITORY		
データベースタイプ	精度・長さ	スケール	分類	けた数	小数部けた数
LONG VARCHAR	[n] (16777215)	x	英数字文字列	n	-
LONG VARRAW	n (16777215)	x	なし	n	-
MLSLABEL	[n] (4)	x	なし	n	-
NUMBER	[p] (38)	[s] (0)	なし	p+2 ²	s ³
RAW	[n] (2000)	x	なし	n	-
RAW MLSLABEL	[n] (255)	x	なし	n	-
ROWID	x	x	なし	2000	-
STRING	[n] (2000)	x	英数字文字列	n	-
UNSIGNED	[p] (10)	x	なし	p	-
VARCHAR	[n] (65533)	x	英数字文字列	n	-
VARCHAR2	[n] (2000)	x	英数字文字列	n	-
VARNUM	x	x	なし	22	-
VARRAW	n (65533)	x	なし	n	-
その他	[p]	[s]	なし	-	-

(凡例)

x : 指定できない

- : 設定されない

n : 長さ

p : 精度

s : スケール

[] : 括弧内は省略できる

() : 括弧内は、ERwin/ERX での記述が省略または数値以外の文字列の場合の仮定値

注

データ項目種別はすべて「単項目」となります。

注 1

s=0 の場合と、s が省略された場合は「整数」となります。

注 2

s < 0 の場合は「p+(sの絶対値)+1」となります。p < s の場合は「s+2」となります。

注 3

s=0 の場合と、s が省略された場合は「-」となります。

注 4

s が省略された場合は「-」となります。

表 B-5 ERwin/ERX のタイプの変換規則 A-2 (ORACLE)

ERwin/ERX のデータベース タイプ	SEWB + /REPOSITORY の言語別タイプ			
	COBOL 又は OOCOBOL	C 又は C++	SQL	その他
CHAR	X	char	CHAR	8
CHARACTER	X	char	CHAR	8
CHARZ	X	char	フリー定義 ⁶	8
CURSOR	フリー定義 ¹	フリー定義 ⁴	フリー定義 ⁶	8
DATE	X	char	DATE	8
DECIMAL	P	float	フリー定義 ⁶	8
DISPLAY	フリー定義 ²	char	フリー定義 ⁶	8
FLOAT	D	float	フリー定義 ⁶	8
INTEGER	フリー定義 ³	int	フリー定義 ⁶	8
LONG	X	char	LONG	8
LONG RAW	X	char	フリー定義 ⁶	8
LONG VARCHAR	X	char	フリー定義 ⁶	8
LONG VARRAW	X	char	フリー定義 ⁶	8
MLSLABEL	X	char	フリー定義 ⁶	8
NUMBER	X	char	7	8
RAW	X	char	フリー定義 ⁶	8
RAW MLSLABEL	X	char	フリー定義 ⁶	8
ROWID	X	char	フリー定義 ⁶	8
STRING	X	char	フリー定義 ⁶	8
UNSIGNED	フリー定義 ⁴	unsigned int	フリー定義 ⁶	8
VARCHAR	フリー定義 ⁵	char	フリー定義 ⁶	8
VARCHAR2	X	char	VARCHAR2	8
VARNUM	X	char	フリー定義 ⁶	8
VARRAW	X	char	フリー定義 ⁶	8
その他	フリー定義 ⁴	フリー定義 ⁴	フリー定義 ⁶	8

注 1

定義文字列には「SQL-CURSOR」が設定されます。

注 2

定義文字列には、「PIC S9(\$p)V9(\$s) DISPLAY SIGN LEADING SEPARATE」が設定され
ます。(\$p: 精度 (長さ), \$s: スケール)

注 3

定義文字列には「PIC S9(\$p) COMP-5」が設定されます。(\$p: 精度 (長さ))

注 4

定義文字列は設定されません。

注 5

定義文字列には「PIC X(\$p) VARYING」が設定されます。(\$p: 精度 (長さ))

注 6

定義文字列には ERwin/ERX タイプが設定されます。

注 7

p と s が省略された場合は「NUMBERf」が設定されます。s=0 の場合と、p が指定されて s が省略された場合は「NUMBER」が設定されます。その他の場合はフリー定義となり、定義文字列には ERwin/ERX タイプが設定されます。

注 8

言語区分「IDL(CORBA)」、「帳票定義」、「Java」のタイプには、環境構築ユーティリティで設定した分類区分に対応するタイプの初期値 (デフォルトタイプ) が設定されます。言語別 4 ~ 7 のタイプはフリー定義となり、定義文字列は設定されません。

表 B-6 ERwin/ERX のタイプの変換規則 B-1 (ORACLE)

ERwin/ERX			SEWB + /REPOSITORY		
データベースタイプ	精度・長さ	スケール	分類	けた数	小数部けた数
CHAR	[n] (18)	x	英数字文字列	n	-
CHARACTER	[n] (18)	x	英数字文字列	n	-
DATE	x	x	なし	18	-
DECIMAL	[p] (18)	[s] (0)	実数 ¹	p	s ²
FLOAT	x	x	実数	18	-
INTEGER	x	x	整数	18	-
LONG	x	x	英数字文字列	18	-
LONG RAW	x	x	英数字文字列	18	-
LONG VARCHAR	[n] (18)	x	英数字文字列	n	-
MLSLABEL	x	x	なし	18	-
NUMBER	[p] (18 (* のとき 38))	[s] (0)	実数 ¹	p	s ²

ERwin/ERX			SEWB + /REPOSITORY		
データベースタイプ	精度・長さ	スケール	分類	けた数	小数部けた数
RAW	[n] (18)	x	英数字文字列	n	-
RAW MLSLABEL	x	x	なし	18	-
ROWID	x	x	なし	18	-
SMALLINT	x	x	整数	18	-
VARCHAR	[n] (18)	x	英数字文字列	n	-
VARCHAR2	[n] (18)	x	英数字文字列	n	-
その他	[p] (18)	[s]	なし	p	-

(凡例)

x : 指定できない

- : 設定されない

n : 長さ

p : 精度

s : スケール

[] : 括弧内は省略できる

(): 括弧内は、ERwin/ERX での記述が省略または数値以外の文字列の場合の仮定値

注

データ項目種別はすべて「単項目」となります。

注 1

s=0 の場合と、p が指定されて s が省略された場合は「整数」となります。

注 2

s=0 の場合と、s が省略された場合は「-」となります。

表 B-7 ERwin/ERX のタイプの変換規則 B-2 (ORACLE)

ERwin/ERX のデータベースタイプ	SEWB + /REPOSITORY の言語別タイプ	
	SQL	その他
CHAR	CHAR	3
CHARACTER	フリー定義 ¹	3
DATE	フリー定義 ¹	3
DECIMAL	フリー定義 ¹	3
FLOAT	フリー定義 ¹	3
INTEGER	フリー定義 ¹	3

ERwin/ERX のデータベース スタイル	SEWB + /REPOSITORY の言語別タイプ	
	SQL	その他
LONG	LONG	3
LONG RAW	フリー定義 ¹	3
LONG VARCHAR	フリー定義 ¹	3
MLSLABEL	フリー定義 ¹	3
NUMBER	NUMBERf ²	3
RAW	フリー定義 ¹	3
RAW MLSLABEL	フリー定義 ¹	3
ROWID	フリー定義 ¹	3
SMALLINT	フリー定義 ¹	3
VARCHAR	フリー定義 ¹	3
VARCHAR2	VARCHAR2	3
その他	フリー定義 ¹	3

注 1

定義文字列には ERwin/ERX タイプが設定されます。

注 2

s=0 の場合と、p が指定されて s が省略された場合は「NUMBER」が設定されます。

注 3

言語区分「COBOL 又は OOCOBOL」、「C 又は C++」、「IDL(CORBA)」、「帳票定義」、「Java」のタイプには、環境構築ユーティリティで設定した分類区分に対応するタイプの初期値（デフォルトタイプ）が設定されます。言語別 4～7 のタイプはフリー定義となり、定義文字列には ERwin/ERX タイプが設定されます。

表 B-8 ERwin/ERX のタイプの変換規則 C-1 (HiRDB)

ERwin/ERX			SEWB + /REPOSITORY			
データベースタイプ	精度・長さ	スケール	構成項目サ フィックス	分類	けた数	小数部 けた数
CHAR	[n] (1)	x	-	英数字文 字列	n	-
CHARACTER	[n] (1)	x	-	英数字文 字列	n	-
DATE	x	x	-	日付	10	-
DEC	[p] (15)	[s] (0)	-	実数 ¹	p	s ²
DECIMAL	[p] (15)	[s] (0)	-	実数 ¹	p	s ²

ERwin/ERX			SEWB + /REPOSITORY			
データベースタイプ	精度・長さ	スケール	構成項目サフィックス	分類	けた数	小数部けた数
DOUBLE PRECISION	x	x	-	実数	16	-
FLOAT	x	x	-	実数	16	-
INT	x	x	-	整数	9	-
INTEGER	x	x	-	整数	9	-
INTERVAL YEAR TO DAY	x	x	-	なし	8	-
INTERVAL HOUR TO SECOND	x	x	-	なし	6	-
MCHAR	[n] (1)	x	-	英数字文字列	n	-
MVARCHAR	n (32000)	x	-length	整数	4	-
			-char	英数字文字列	n	-
NCHAR	[n] (1)	x	-	日本語文字列	n	-
NVARCHAR	n (16000)	x	-length	整数	4	-
			-char	日本語文字列	n	-
REAL	x	x	-	実数	7	-
SMALLFLT	x	x	-	実数	7	-
SMALLINT	x	x	-	整数	4	-
TIME	x	x	-	時刻	8	-
VARCHAR	n (32000)	x	-length	整数	4	-
			-char	英数字文字列	n	-
その他	[p]	[s]	-	なし	-	-

(凡例)

x : 指定できない

- : 設定されない

n : 長さ

p : 精度

s : スケール

[] : 括弧内は省略できる

(): 括弧内は, ERwin/ERX での記述が省略または数値以外の文字列の場合の仮定値

注

構成項目のサフィックス欄が「-」のタイプは, 単項目として生成されるタイプです。「-」以外のタイプは, 結合項目としてその構成項目が生成されるタイプです。

注 1

s=0の場合と、pが指定されてsが省略された場合は「整数」となります。

注 2

s=0の場合と、sが省略された場合は「-」となります。

表 B-9 ERwin/ERX のタイプの変換規則 C-2 (HiRDB)

ERwin/ERX のデータ ベースタイプ	SEWB + /REPOSITORY の言語別タイプ					
	構成項目 サフィックス	COBOL 又 は OOCOBOL	C 又は C++	SQL	その他	備考
CHAR	-	X	char	CHAR	3	
CHARACTER	-	X	char	CHAR	3	
DATE	-	X	char	DATE	3	
DEC	-	P	フリー定義 1	DEC	3	
DECIMAL	-	P	フリー定義 1	DEC	3	4
DOUBLE PRECISION	-	D	double	フリー定義 ²	3	5
FLOAT	-	D	double	FLOAT	3	
INT	-	B	long	INT	3	
INTEGER	-	B	long	INT	3	
INTERVAL YEAR TO DAY	-	P	フリー定義 1	フリー定義 ²	3	
INTERVAL HOUR TO SECOND	-	P	フリー定義 1	フリー定義 ²	3	
MCHAR	-	X	char	MCHAR	3	
MVARCHAR	-length ⁶	B	short	SMALLINT	3	4
	-char ⁶	X	char	MVARCHAR	3	5
NCHAR	-	N	char	NCHAR	3	
NVARCHAR	-length ⁶	B	short	SMALLINT	3	4
	-char ⁶	N	char	NVARCHAR	3	5
REAL	-	D	float	フリー定義 ²	3	
SMALLFLT	-	D	float	SMALLFLT	3	
SMALLINT	-	B	short	SMALLINT	3	
TIME	-	X	char	TIME	3	

ERwin/ERX のデータ ベースタイプ	SEWB + /REPOSITORY の言語別タイプ					
	構成項目 サフィックス	COBOL 又 は OOCOBOL	C 又は C++	SQL	その他	備考
VARCHAR	-length ⁶	B	short	SMALLINT	3	4
	-char ⁶	X	char	VARCHAR	3	5
その他	-	フリー定義 ¹	フリー定義 ¹	フリー定義 ²	3	

注 1

定義文字列は設定されません。

注 2

定義文字列には ERwin/ERX タイプが設定されます。

注 3

言語区分「IDL(CORBA)」、「帳票定義」、「Java」のタイプには、環境構築ユティリティで設定した分類区分に対応するタイプの初期値（デフォルトタイプ）が設定されます。言語別 4 ~ 7 のタイプはフリー定義となり、定義文字列は設定されません。

注 4

初期値に、カラムのデフォルト値の文字列長が設定されます。

注 5

初期値に、カラムのデフォルト値が設定されます。

注 6

環境構築ユティリティのタイプマッピングの設定で、サフィックスをカスタマイズできます。

(6) エンティティレポート形式フォーマット

ERwin/ERX のエンティティレポートに対応する CSV 形式ファイルフォーマットについて説明します。

(a) ファイル内部の構成

エンティティレポート形式のファイル内部の構成を次に示します。

ヘッダ	一つの結合項目と、それを直接構成する構成項目だけで記述します
データ項目（結合項目）定義情報1	
データ項目（構成項目）定義情報1-1	
⋮	
データ項目（構成項目）定義情報1-n	
⋮	
データ項目（結合項目）定義情報N	
データ項目（構成項目）定義情報N-1	
⋮	
データ項目（構成項目）定義情報N-m	

1 行目にはヘッダを、2 行目からは個々のデータ項目の定義情報を記述します。エンティティレポート形式では、結合関係を記述できます。ただし、結合項目を階層化して記述することはできません。

結合項目を 1 行で記述します。そのあとに続けて、その結合項目を構成する個々の構成項目を 1 行ずつ連続して記述します。構成項目は、次の結合項目を記述するまで、同じ結合項目と関連する構成項目とみなされます。結合項目と構成項目の区別は、テーブル名があるかどうかで判断されます。

(b) ヘッダ

ヘッダは ERwin/ERX でレポートを出力した場合に付けられます。ヘッダの内容は任意ですが、ヘッダ行のカンマで区切られた列数は、定義情報と同じでなくてはなりません。

(c) 定義情報

定義情報のフォーマットとエンティティレポートで定義する必要がある値と省略できる値を、表 B-10 に示します。エンティティレポート形式では、文字列の値はすべてクォーテーション（" "）または（' '）で囲まないで記述します。

表 B-10 データ項目の定義情報の記述内容（エンティティレポート形式）

記述順	記述する定義情報	結合項目	構成項目
1	テーブル名		-
2	エンティティ名		-
3	属性名	-	
4	カラム名	-	
5 ¹	エンティティコメント		-
5 ¹	エンティティ定義		-
6 ²	外部キー		

(凡例)

- : 必要な値
- : 省略できる値
- : 該当しない

注 1

どちらか一方を記述します。

注 2

属性レポートを同時入力する場合は、必ず記述します。

付録 B.3 属性レポートの連携

ここでは、ERwin/ERX (バージョン 2.6 以前) で作成した属性レポートの辞書登録について説明します。

(1) ERwin/ERX での属性レポートの出力設定

ERwin/ERX で次に示す設定をして、CSV 形式ファイルを出力してください。

- サーバの対象 SQL DBMS の設定

使用するデータベースが ORACLE の場合

対象データベース (SQL ベース): ORACLE

使用するデータベースが HiRDB の場合

対象データベース (SQL ベース): DB2

- CSV 形式ファイルへの出力項目の設定

属性レポートから、SEWB+/REPOSITORY の辞書に連携するために必要な出力項目を設定します。出力項目と指定順序を、表 B-11 に示します。指定順序が列の並び順になります。このため、必ず指定順序に従って設定してください。

表 B-11 属性レポートの出力項目と指定順序

指定順	出力項目 (オプション名)	
	ERwin/ERX 日本語版	ERwin/ERX 英語版
1	カラム名 (カラムオプション)	Column Name (Column Option)
2	属性名 (属性オプション)	Attribute Name (Attribute Option)
3	FK 指定 (属性オプション)	FK Designator (Attribute Option)
4	カラムデータタイプ (カラムオプション)	Column Datatype (Column Option)
5	デフォルト値 (制約オプション)	Default Values (Constraint Option)
6 ¹	属性コメント (属性オプション)	Note (Attribute Option)
6 ¹	属性定義 (属性オプション)	Definition (Attribute Option)
7 ²	PK 指定 (属性オプション)	PK Designator (Attribute Option)

指定順	出力項目 (オプション名)	
	ERwin/ERX 日本語版	ERwin/ERX 英語版
8 ²	ベース名 (属性オプション)	Basename (Attribute Option)
9 ³	カンマ区切り (レポート形式)	Comma Delimited (Report Format)
10 ³	ヘッダー (表示形式)	Header (Multi-Valued Format)
11 ³	マージ (表示形式)	Merge (Multi-Valued Format)
12 ³	改行コードの除去 ⁴	Remove Newline

注 1

どちらか一方を選択してください。両方の同時指定はできません。

注 2

属性レポートを同時入力する場合、または物理名の異なる外部キーを識別する場合には、必ず指定してください。

注 3

指定順序どおりでなくてもかまいません。

注 4

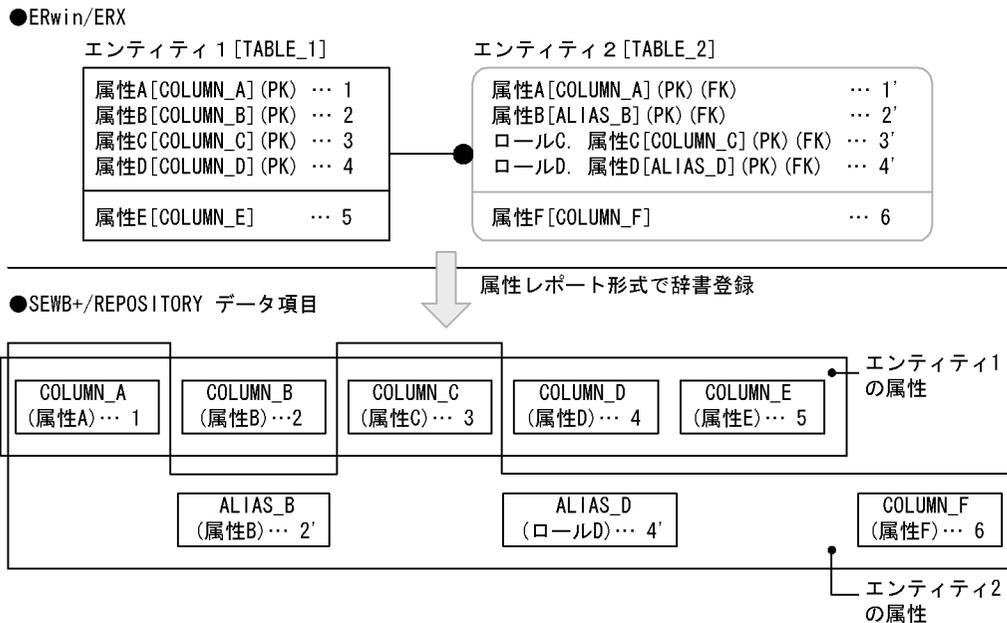
ERwin/ERX 2.5 以前のバージョンでは「新規行の除去」となります。

(2) 属性レポートからデータ項目辞書への変換規則

依存関係のある属性レポートを単独で辞書登録する場合の変換規則について説明します。エンティティレポートと同時に入力する場合には、「付録 B.2(2) エンティティレポートからデータ項目辞書への変換規則」を参照してください。

属性レポート形式の辞書登録の例を、図 B-4 に示します。

図 B-4 属性レポート形式の辞書登録の例



- 注
- ・ERwin/ERXのエンティティ内の[]は、テーブル名またはカラム名を示します。
 - ・リレーションシップでルール名が定義された属性は「ルール名. カラム名」の形式で示します。
 - ・SEWB+/REPOSITORY側の()は、データ項目の標準名称を示します。

属性レポート形式のCSV形式ファイルを辞書に登録するとき、物理名の異なる外部キーの識別を指定するかどうかによって、登録されるデータ項目が異なります。図B-4は、物理名の異なる外部キーを識別する指定をした場合の登録例です。

物理名の異なる外部キーを識別して登録するには、CSV形式ファイル入力コマンドで引数「/Z 物理名の異なる外部キーを識別する」を省略するか、[CSV入力オプション]ダイアログの「物理名の異なる外部キーを識別する」のチェックをオンにします。

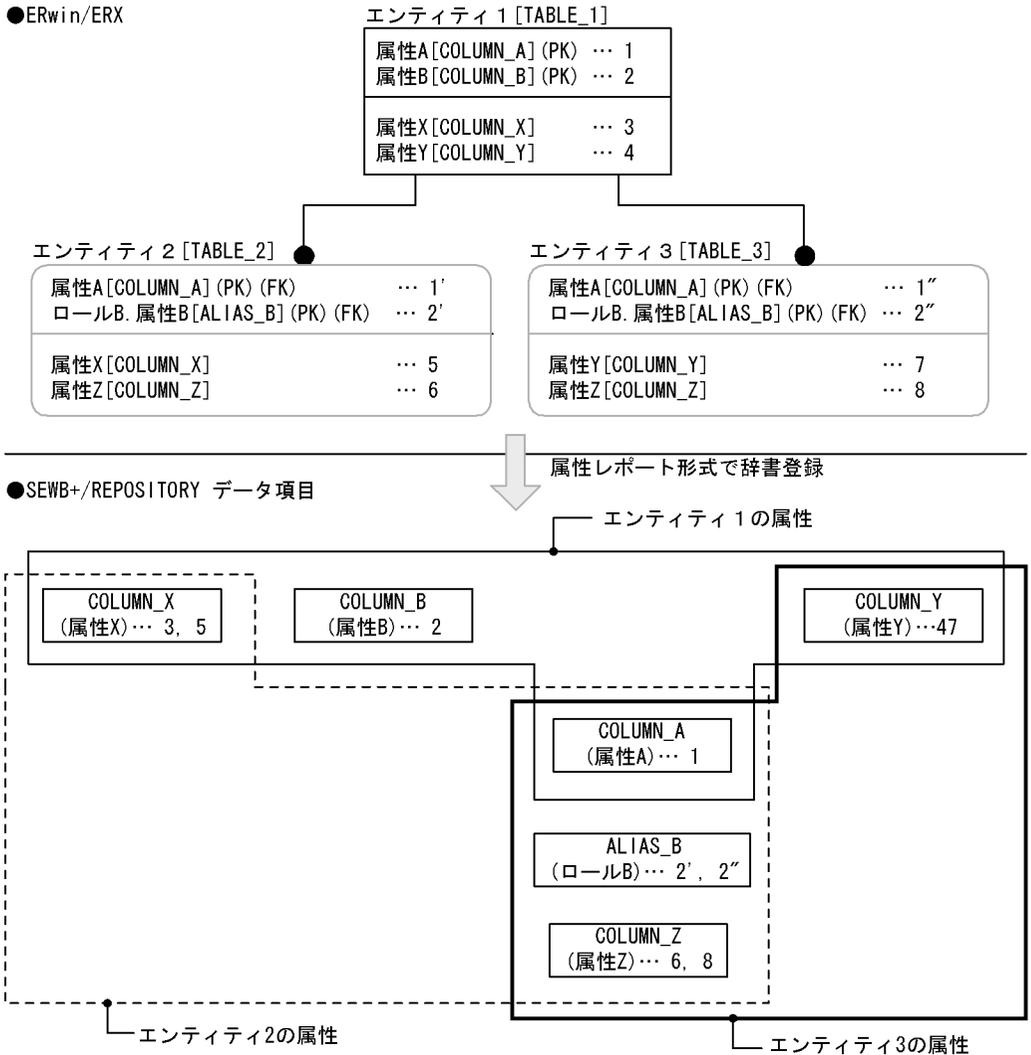
この指定をしない場合には、図B-4の「2'」と「4'」のデータ項目は登録されません。

(3) 属性レポートからデータ項目辞書への変換規則（同一カラム名をマージする場合）

属性レポート中に、同一のカラム名が定義された属性情報が複数ある場合、その情報をマージしてデータ項目辞書に登録できます。エンティティレポートと同時に入力する場合については、「付録B.2(3) エンティティレポートからデータ項目辞書への変換規則（同一カラム名をマージする場合）」を参照してください。

同一カラム名の属性をマージする場合の属性レポートの変換規則を、図B-5に示します。

図 B-5 属性レポート形式の辞書登録の例（同一カラム名をマージする場合）



- 注
- ・ERwin/ERXのエンティティ内の[]は、テーブル名またはカラム名を示します。
 - ・リレーションシップでロール名が定義された属性は「ロール名. 属性名」の形式で示します。
 - ・SEWB+/REPOSITORY側の()は、データ項目の標準名称を示します。

属性レポート形式のCSV形式ファイルを辞書に登録するとき、物理名の異なる外部キーの識別を指定するかどうかによって、登録されるデータ項目が異なります。図B-5は、物理名の異なる外部キーを識別する指定をした場合の登録例です。

(a) 同一カラム名をマージして登録する

同一カラム名をマージして辞書に登録するには、CSV形式ファイル入力コマンドで引数「/R 同一カラム名のマージ有無」に「use」を指定するか、または[CSV入力オプション]ダイアログの「同一カラム名の属性をマージする」のチェックをオンにします。

なお、FK 属性のカラム名と、その移行先の PK 属性のカラム名が同じ場合は、この指定に関係なく、必ずマージされて登録されます。詳細は、「(2) 属性レポートからデータ項目辞書への変換規則」を参照してください。

(b) マージされる属性の条件

同一カラム名の属性のマージでは、マージされる属性間で、属性の値が一致している必要があります。マージされる属性間の条件を、表 B-12 に示します。

表 B-12 マージされる属性間の条件

属性情報		マージする属性間の条件
カラム名		
属性名		
カラムデータタイプ (精度, スケールを含む)		
デフォルト値		
移行先を示す情報	ベース名	×
PK 指定		×
FK 指定		×
属性コメントまたは定義		×

(凡例)

: 一致しなければならない

x : 一致しなくてもよい

注

コメントまたは定義の情報が、マージ対象の複数の項目に設定されている場合は、その中で最も長い情報が、データ項目のコメントに設定されます。

(4) データ項目辞書と属性レポート形式の対応

データ項目辞書の定義項目と属性レポート形式の対応を、表 B-13 に示します。

表 B-13 データ項目辞書と属性レポート形式の対応

データ項目辞書の定義項目		属性レポート形式	
		構成項目	
		単項目	結合項目とその構成項目 結: 結合項目 構: 構成項目
名称	データ項目名	カラム名	結: カラム名 構: カラム名 + サフィックス ¹
	標準名称	属性名	結: 属性名 構: 属性名 + サフィックス ¹

データ項目辞書の定義項目		属性レポート形式	
		構成項目	
		単項目	結合項目とその構成項目 結：結合項目 構：構成項目
	フリガナ	×	結：×
			構：×
属性	分類	カラムデータタイプから分類へ変換 ¹	結：結合データ 構：カラムデータタイプから分類へ変換 ¹
	けた数	カラムデータタイプから抽出 ¹	結：- 構：カラムデータタイプから抽出 ¹
	小数部けた数	カラムデータタイプから抽出 ¹	結：- 構：カラムデータタイプから抽出 ¹
	反復回数	×	結：- 構：×
付加情報	フィールド 1 ~ n	×	結：×
			構：×
	コメント	属性コメント / 定義	結：属性コメント / 定義 構：×
言語別詳細情報	名前	カラム名	結：カラム名 構：カラム名 + サフィックス ¹
	タイプ	カラムデータタイプから分類へ変換 ¹	結：- 構：カラムデータタイプから分類へ変換 ¹
	タイプ修飾情報 ³	×	結：- 構：×
	取りうる値	×	結：- 構：×
	初期値	デフォルト値	結：- 構： ²
	言語別フィールド	×	結：×
			構：×

(凡例)

- ×：データ項目辞書に対応する ERwin/ERX 側の情報がない項目
- ：データ項目辞書に値がない項目

注 1

変換される分類とタイプの変換規則は、エンティティレポート形式と同じです。「付録 B.2(5) ERwin/ERX タイプマッピングの規則」を参照してください。

注 2

サフィックス "-char" の構成項目にはデフォルト値を、サフィックス "-length" の構成項目にはデフォルト値の文字列長を初期値として設定します。

注 3

「COBOL 又は OOCOBOL」で定義する編集文字列、「IDL(CORBA)」で定義するシーケンスを指します。

- データ項目は通常、継承関係の最上位にあるデータ項目として登録されます。
- 表 B-13 で「x」または「-」以外が記された定義項目には、継承関係の上位のデータ項目の情報は継承されません。また、「x」が記された定義項目の継承モードは、環境構築ユーティリティのデータ項目の継承モードの設定に従います。継承モードについては「3.1.1(3)(b) データ項目の継承モードの設定」を参照してください。
- CSV 形式ファイル入力コマンドの引数「/W 更新対象属性」、または [CSV 入力オプション] ダイアログの「CSV 入力フォーマット種別」で、ERwin/ERX 属性だけを更新する指定をすると、表 B-13 で「x」または「-」以外が記された定義情報のうち言語別名前を除く情報だけを更新できます。

(5) 属性レポート形式フォーマット

ERwin/ERX の属性レポートに対応する CSV 形式ファイルフォーマットについて説明します。

(a) ファイル内部の構成

属性レポート形式のファイル内部の構成を次に示します。

ヘッダ	
データ項目	定義情報1
	:
データ項目n	定義情報

1 行目にはヘッダを、2 行目からは個々のデータ項目の定義情報を記述します。

(b) ヘッダ

ヘッダは ERwin/ERX でレポートを出力した場合に付加されます。ヘッダの内容は任意ですが、ヘッダ行のカンマで区切られた列数は、定義情報と同じでなくてはなりません。

(c) 定義情報

定義情報のフォーマットと属性レポートで定義する必要のある値と省略できる値を、表 B-14 に示します。属性レポート形式では、文字列の値はすべてクォーテーション（"）または（'）で囲まないで記述します。

表 B-14 データ項目の定義情報の記述内容（属性レポート形式）

記述順	記述する定義情報	構成項目
1	カラム名	
2	属性名	
3	FK 指定	

記述順	記述する定義情報	構成項目
4	カラムデータタイプ	
5	デフォルト値	
6 ¹	属性コメント	
6 ¹	属性定義	
7 ²	PK 指定	
8 ²	ベース名	

(凡例)

: 必要な値

: 省略できる値

注 1

どちらか一方を記述します。

注 2

物理名の異なる外部キーを識別する場合は、必ず記述します。

付録 B.4 テーブルレポートの連携

ここでは、ERwin/ERX (バージョン 3.5) で作成したテーブルレポートの辞書登録について説明します。

(1) ERwin/ERX でのテーブルレポートの出力設定

ERwin/ERX で次の設定をして、CSV 形式ファイルを出力してください。

(a) サーバの対象 SQL DBMS の設定

データベースに ORACLE を使用する場合

- SQL DBMS の設定値 : ORACLE
- DB のバージョン : 任意

データベースに HiRDB を使用する場合

- SQL DBMS の設定値 : HiRDB
- DB のバージョン : 任意

(b) レポート作成時の設定

ERwin/ERX のでレポートを作成するときには、次のオプションを設定してください。

論理 / 物理 : 物理

カテゴリ : テーブル

(c) レポート形式に設定する項目

テーブルレポートから辞書に連携するために必要なカラムオプションと、その並び順を次に示します。

順序	カラム
1	テーブル 名前
2	テーブル カラム 名前
3	テーブル カラム データ型
4	テーブル カラム 主キーである
5	テーブル カラム 外部キーである
6	テーブル カラム デフォルト デフォルト
7	テーブル カラム 属性 名前
8	テーブル カラム 属性 ノート
8	テーブル カラム 属性 定義
9	テーブル 親リレーションシップ 親テーブル 名前
10	テーブル 親リレーションシップ 移行カラム 名前
11	テーブル 親リレーションシップ 移行カラム 属性 ベース名
12	テーブル エンティティ 名前
13 1	テーブル エンティティ ノート
13 1	テーブル エンティティ 定義
- 2	テーブル カラム フィルタ 実カラム (非 FK) を含む
- 2	テーブル カラム フィルタ FK カラムを含む
- 2	テーブル カラム ソート 物理順序

注 1

どちらか一方だけを指定してください。

注 2

必ず指定してください。順序は該当しません。

(d) レポートのエクスポート時の設定

レポートをファイルにエクスポートするときには、次のオプションを設定してください。

エクスポート形式：CSV

表示形式：表形式 (重複)

エクスポート先：ファイル

(2) テーブルレポートからデータ項目辞書への変換規則

依存関係のあるテーブルレポートを辞書登録する場合の変換規則は、エンティティレポートと属性レポートを同時入力する場合と同様です。「付録 B.2(2) エンティティレポートからデータ項目辞書への変換規則」を参照してください。

(3) テーブルレポートからデータ項目辞書への変換規則（同一カラム名をマージする場合）

テーブルレポート中に、同一のカラム名が定義された属性情報が複数ある場合、その情報をマージしてデータ項目辞書に登録できます。この変換規則は、エンティティレポートと属性レポートを同時入力する場合と同様です。「付録 B.2(3) エンティティレポートからデータ項目辞書への変換規則（同一カラム名をマージする場合）」を参照してください。

(4) データ項目辞書とテーブルレポート形式の対応

データ項目辞書の定義項目とテーブルレポート形式の対応を、表 B-15 に示します。

表 B-15 データ項目辞書とテーブルレポート形式の対応

データ項目辞書の定義項目		テーブルレポート形式		
		結合項目 ¹	構成項目	
			単項目	結合項目とその構成項目 結：結合項目 構：構成項目
名称	データ項目名	テーブル 名前	カラム 名前	結：カラム 名前 構：カラム 名前 + サフィックス ²
	標準名称	エンティティ 名前	属性 名前	結：属性 名前 構：属性 名前 + サフィックス ²
	フリガナ	×	×	結：×
属性	分類	結合データ	カラム データ型を 変換 ²	結：結合データ 構：カラム データ型を変換 ²
	けた数	-	カラム データ型か ら抽出 ²	結：- 構：カラム データ型から抽出 ²
	小数部 けた数	-	カラム データ型か ら抽出 ²	結：- 構：カラム データ型から抽出 ²
	反復回数	-	×	結：- 構：×

データ項目辞書の定義項目		テーブルレポート形式		
		結合項目 ¹	構成項目	
			単項目	結合項目とその構成項目 結：結合項目 構：構成項目
付加情報	フィールド 1 ~ n	×	×	結：×
	コメント	エンティティ ノート / 定義	属性 ノート / 定義	結：属性 ノート / 定義 構：×
言語別詳細情報	名前	テーブル 名前	カラム 名前	結：カラム 名前 構：カラム 名前 + サフィックス ²
	タイプ	-	カラム データ型を 変換 ²	結：- 構：カラム データ型を変換 ²
	タイプ修飾情報 ³	-	×	結：- 構：×
	取りうる値	×	×	結：×
	初期値	-	カラム デフォルト デフォルト	結：- 構： ⁴
	言語別フィールド	×	×	結：×

(凡例)

- × : ERwin/ERX 側にデータ項目辞書に対応する情報がない項目
- : データ項目辞書側に値が設定されない項目

注 1

エンティティから変換されるデータ項目は、最上位結合項目となります。

注 2

変換される分類とタイプは、「付録 B.2(5) ERwin/ERX タイプマッピングの規則」を参照してください。

注 3

「COBOL 又は OOCOBOL」で定義する編集文字列、「IDL(CORBA)」で定義するシーケンスを指します。

注 4

初期値にはタイプマッピングで定義した値が設定されます。

- データ項目は通常、継承関係の最上位にあるデータ項目として登録されます。
- 「カラム 名前」が PK 属性と異なる FK 属性は、CSV 形式ファイル入力コマンドの引数「/Z 物理名の異なる外部キーを識別する」、または [CSV 入力オプション] ダイアログの「物理名の異なる外部キーを識別する」に物理名の異なる外部キーを識別す

る指定をすると、FK 属性の「ベース名」と同じ「属性 名前」の PK 属性のデータ項目を継承します。

- 継承関係の下位にあるデータ項目として登録されたとき、表 B-15 で「×」または「-」以外が記された定義項目には、上位のデータ項目の情報は継承されません。また、「×」が記された定義項目には初期値が設定されます。データ項目の初期値については、「4.1.6(4) 定義項目の初期値」を参照してください。
- CSV 形式ファイル入力コマンドの引数「/W 更新対象属性」、または [CSV 入力オプション] ダイアログの「CSV 入力フォーマット種別」で、ERwin/ERX 属性だけを更新する指定をすると、表 B-15 で「×」または「-」以外が記された定義情報のうち言語別名前を除く情報だけを更新できます。

(5) テーブルレポート形式フォーマット

ERwin/ERX のテーブルレポートに対応する CSV 形式ファイルフォーマットについて説明します。

(a) ファイル内部の構成

テーブルレポート形式のファイル内部の構成を次に示します。

ヘッダ	
定義情報 (エンティティと属性)	L-1
定義情報 (属性)	L-2
:	
定義情報 (属性)	L-m
:	
:	
定義情報 (エンティティと属性)	K-1
定義情報 (属性)	K-2
:	
定義情報 (属性)	K-n

一つのエンティティと、その属性だけを記述する

1 行目からはヘッダを、2 行目からはこのデータ項目の定義情報を記述します (ヘッダは ERwin/ERX のバージョンによっては出力されない場合があります)。定義情報の行では、前の行と異なるテーブル名の記述がある行をエンティティの開始行とし、結合項目に対応するエンティティの情報と構成項目に対応する属性のうち 1 番目の情報を合わせて 1 行に記述します。そのあとの行で、2 番目以降の属性の情報を一行ずつ連続して記述します。次のエンティティ情報が現れるまでをエンティティに属する属性情報とみなします。

(b) ヘッダ

ヘッダは ERwin/ERX (バージョン 3.5) でレポートを出力した場合に付加されます。ヘッダの内容は任意ですが、ヘッダ行のカンマで区切られた列数は、定義情報と同じでなくてはなりません。

(c) 定義情報

定義情報のフォーマットとテーブルレポートで定義する必要のある値と省略できる値を、表 B-16 に示します。文字列の値はすべてクォーテーション（「"」または「'」）で囲まな
いで記述します。

表 B-16 データ項目の定義情報の記述内容（テーブルレポート形式）

記述順	記述内容	設定区分
1	テーブル名	
2	カラム名	
3	カラムデータタイプ	
4	PK 指定	
5	FK 指定	
6	デフォルト値	
7	属性名	
8	属性コメント	
8	属性定義	
9	親テーブル名	
10	移行カラム名	
11	移行属性ベース名	
12	エンティティ名	
13	エンティティコメント	
13	エンティティ定義	

(凡例)

- : 必要な値
- : 省略できる値

注

どちらか一方を記述します。

付録 B.5 辞書登録の例

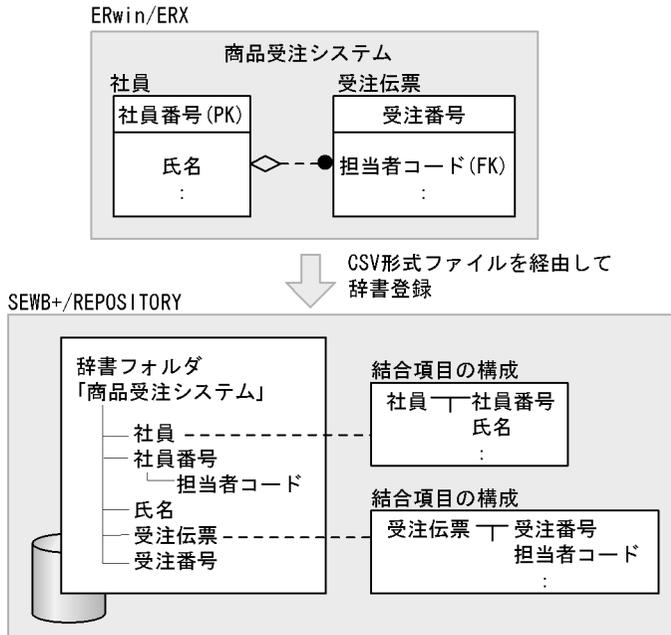
次のケースを例に、ERwin/ERX から辞書登録するときに SEWB+/REPOSITORY でどのような設定をするとよいかを説明します。

- ERwin/ERX の分析情報を、新規に辞書登録する
- 辞書のデータ項目を、ERwin/ERX の分析情報で更新する

(1) ERwin/ERX の分析情報を、新規に辞書登録する

(a) 辞書登録の概要

ERwin/ERX で設計した分析情報を CSV 形式ファイルに出力し、指定した辞書フォルダに新しいデータ項目として登録します。この例を次に示します。



(b) SEWB+/REPOSITORY での CSV 形式ファイル入力時の設定

ダイアログから実行する場合

[スタート]メニューの[プログラム]または[すべてプログラム]から[Sewb+] - [Repository] - [CSV 入力]を選びます。表示されるダイアログで次の設定をします。

表 B-17 [CSV 入力]ダイアログ

設定項目	設定値
入力 CSV ファイル名	ERwin/ERX から出力したレポートファイル名を指定
登録先辞書フォルダ名	データ項目を登録する辞書フォルダ名を指定
同時に入力する ERwin/ERX 属性レポートファイル名	エンティティレポートと同時に入力する属性レポートファイル名を指定
入力する辞書種別	「データ項目」を選択

注

PK 属性のデータ項目と FK 属性のデータ項目との間に継承関係を登録したい場合、指定してく

ださい。

表 B-18 [CSV 入力オプション] ダイアログ

設定項目	設定値
辞書への登録方法	「新規に登録する」を選択 (ただし属性レポートでは「データ名項目 / 業務ルール名で更新する」を選択)
CSV 入力フォーマット種別	<ul style="list-style-type: none"> • 「ERwin/ERX」を選択 • 「対象データベース選択」でデータベースを選択 • 「レポート形式選択」で ERwin/ERX のバージョンに対応した形式を選択 • 物理名の異なる外部キーを識別する：オン
CSV ファイルのヘッダを識別する	オン (レポート出力時にヘッダを付加した場合)

コマンドラインから実行する場合

CSV 形式ファイル入力コマンドを使用してコマンドラインから実行する場合、次のように指定します。

- ERwin/ERX (バージョン 2.6 以前) からのエンティティレポートの登録 (属性レポートの同時入力)
RPCsvin /F エンティティレポートファイル名 /D 登録先辞書フォルダ名 /M ewe /A 属性レポートファイル名 /B データベース種別 /Z use

注

属性レポートをあとに登録するときは「/S name」を必ず指定してください。

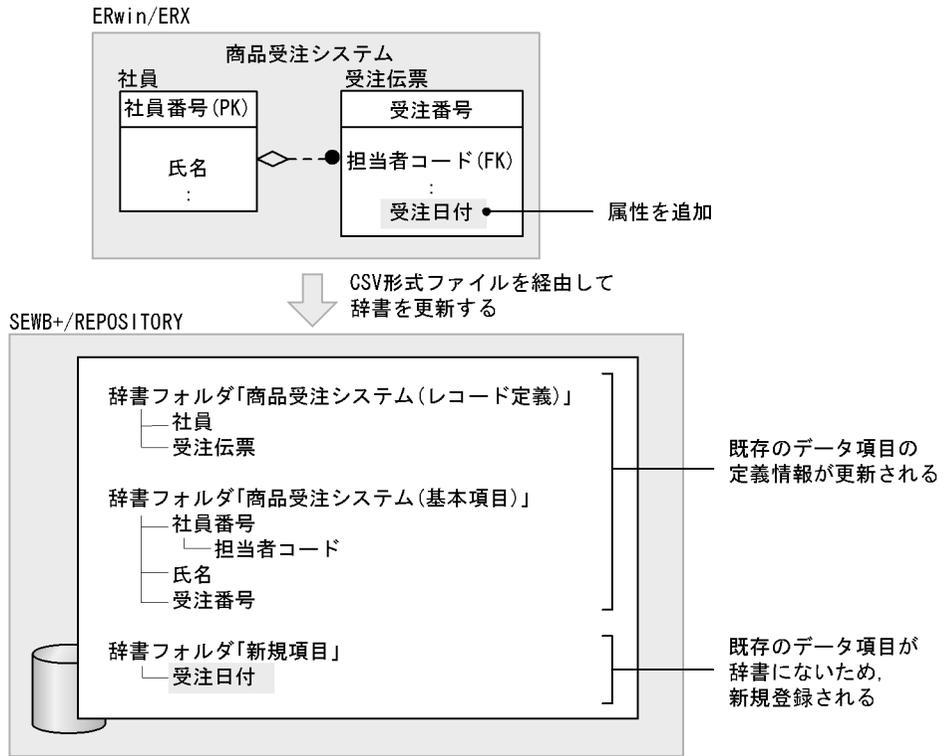
- ERwin/ERX (バージョン 3.5) からのテーブルレポートの登録
RPCsvin /F テーブルレポートファイル名 /D 登録先辞書フォルダ名 /M ewt /B データベース種別 /Z use

(2) 辞書のデータ項目を、ERwin/ERX の分析情報で更新する

(a) 辞書登録の概要

一度、辞書に登録したデータ項目を、そのデータ項目に対応する ERwin/ERX の分析情報で更新します。データ項目の対応は、データ項目名で判断されます。

この例を次に示します。(1) で説明した新規登録のあと、用途ごとにフォルダに分けて管理している例です。この辞書に ERwin/ERX で変更した分析情報を再登録すると、既存のデータ項目の定義情報が更新され、追加した属性が新しいデータ項目として登録されます。



(b) SEWB+/REPOSITORY での CSV 形式ファイル入力時の設定

ダイアログから実行する場合

[スタート]メニューの[プログラム]から[Sewb+] - [Repository] - [CSV入力]を選択します。表示されるダイアログで次の設定をします。

表 B-19 [CSV入力]ダイアログ

設定項目	設定値
入力 CSV ファイル名	ERwin/ERX から出力したレポートファイル名を指定
登録先辞書フォルダ名	データ項目を登録する辞書フォルダ名を指定
同時に入力する ERwin/ERX 属性レポートファイル名	エンティティレポートと同時に入力する属性レポートファイル名を指定
入力する辞書種別	「データ項目」を選択

注

PK 属性のデータ項目と FK 属性のデータ項目との間に継承関係を登録したい場合、指定してください。

表 B-20 [CSV 入力オプション] ダイアログ

設定項目	設定値
辞書への登録方法	<ul style="list-style-type: none"> • 「データ項目名 / 業務ルール名で更新する」を選択 • 全辞書フォルダを更新対象にする：オン • 継承関係も更新する：オフ¹
CSV 入力フォーマット 種別	<ul style="list-style-type: none"> • 「ERwin/ERX」を選択 • 「対象データベース選択」でデータベースを選択 • 「レポート形式選択」で ERwin/ERX のバージョンに対応した形式を選択 • 物理名の異なる外部キーを識別する：オン • Erwin/ERX 属性のみを更新する：オン²
CSV ファイルのヘッダ を識別する	オン (レポート出力時にヘッダを付加した場合)

注 1

新規登録したあとにデータ項目の継承関係を変更していて、その関係を保ったまま ERwin/ERX 属性だけを更新したい場合に、必ず指定してください。

注 2

フリガナなどの ERwin/ERX 属性以外の定義情報を保ったまま、ERwin/ERX 属性だけでデータ項目を更新したい場合に、必ず指定してください。

コマンドラインから実行する場合

CSV 形式ファイル入力コマンドを使用して、コマンドラインから実行する場合は、次のように指定します。

- ERwin/ERX (バージョン 2.6 以前) からのエンティティレポートの登録 (属性レポートの同時入力)
RPCsvin /F エンティティレポートファイル名 /D 登録先辞書フォルダ名 /S name /P nouse /M ewe /A 属性レポートファイル名 /B データベース種別 /Z use /W erwin /V all
- ERwin/ERX (バージョン 3.5) からのテーブルレポートの登録
RPCsvin /F テーブルレポートファイル名 /D 登録先辞書フォルダ名 /S name /P nouse /M ewt /B データベース種別 /Z use /W erwin /V all

付録 C 命名ルールインタフェース

SEWB+/REPOSITORY では、独自の条件 DLL を作成するためのインタフェースを提供しています。

ここでは、条件 DLL の作成の概要と、インタフェース（ヘッダファイルおよび関数）の詳細について説明します。なお、このインタフェースの利用には、C 言語の基本文法をある程度理解されていることを前提としています。

付録 C.1 条件 DLL の作成

(1) インタフェースの構成

命名ルールインタフェースは、次に示すヘッダファイル、関数で構成します。

(a) ヘッダファイル

ヘッダファイル (Nrl_interface.h) には、関数の戻り値などの宣言が記述されています。

(b) 関数

(i) 条件情報取得関数 (NrlGetConditionInfo)

追加する条件名、条件の値の型および条件の値の初期値を SEWB+/REPOSITORY に返す関数です。この関数で返した値が、環境構築ユーティリティの [データ項目] ダイアログの [命名ルール] の条件一覧に表示されます。条件の値の初期値は、「データ項目名」や「データ項目名標準名称」などのすべての名称の種別に対して設定します。

(例)

条件名として「最大長 (バイト数)」, 条件の値の型として
NRL_VALUE_TYPE_INT, 初期値として名称の種別ごとの最大長を格納した
NRL_values_struct_t 構造体を返す関数

(ii) チェック関数 (NrlCheckName)

検査項目「名称と命名ルールのマッチングチェック」の検査が実行されたときに、定義された名称が条件に従っているかどうかを判定する関数です。比較する値には、環境構築ユーティリティで設定した値が渡されます。

(例) 名称の長さを調べ、16 以下なら NRL_OK, 17 以上なら NRL_ERROR を返す関数

(2) インタフェースが格納されているディレクトリ

ヘッダファイルが格納されているディレクトリを次に示します。

サーバのSEWB+ 基本開発環境の組み込み先パス名¥Repository¥include

次のディレクトリには、サンプルソース「Nrl_Check_Byte.c」が提供されています。条

件 DLL を作成するときの参考にしたたり、必要に応じてサンプルをコピーしたりして利用してください。

サーバのSEWB+ 基本開発環境の組み込み先パス名`¥Repository¥sample`

SEWB+ 基本開発環境の組み込み先パス名は、デフォルトでは次のように設定されています。システムドライブとは、Windows 2000 Server, Windows Server 2003 または Windows Server 2003 x64 が組み込まれているドライブです。

- Windows 2000 Server または Windows Server 2003 の場合
システムドライブ : `¥Program Files¥HITACHI¥Sewb`
- Windows Server 2003 x64 の場合
システムドライブ : `¥Program Files (x86)¥HITACHI¥Sewb`

(3) 条件 DLL の登録

環境構築ユーティリティを使用して、独自に作成した条件 DLL を登録すると、SEWB+/REPOSITORY で利用できるようになります。最大 10 種類の条件 DLL を登録できます。条件 DLL を登録すると、登録時に指定したパスとファイル名を基に、条件 DLL から「条件名」、「条件の値の型」および「条件の値の初期値」の情報が引き継がれ、[データ項目] ダイアログの [命名ルール] タブの条件一覧に表示されます。

付録 C.2 ヘッドファイル (Nrl_interface.h)

(1) 関数の戻り値

```
typedef enum {
    NRL_ERROR = 0,           //異常終了
    NRL_OK           //正常終了
} Nrl_status_t;
```

(2) 値の型

```
typedef enum {
    NRL_VALUE_TYPE_CHAR = 0, //文字型
    NRL_VALUE_TYPE_STRING,  //文字列型
    NRL_VALUE_TYPE_INT,     //整数型
    NRL_VALUE_TYPE_BOOLEAN  //論理値型
} Nrl_value_type_t;
```

(3) 初期値の構造体

```
typedef struct Nrl_value_struct_t {
    const void* item_name;           //データ項目名
    const void* item_std_name;      //データ項目標準名称(国語別共通)
    const void* item_furigana;     //データ項目フリガナ(国語別共通)
    const void* item_var_name;     //データ項目名前(言語別共通)
    const void* rule_name;         //業務ルール名
    const void* rule_std_name;     //業務ルール標準名称(国
```

```

    const void* rule_furigana; // 語別共通
                                // 業務ルールフリガナ(国
                                // 語別共通)
} Nrl_init_value_struct_t;

```

(4) チェック結果

```

typedef enum {
    NRL_CHECK_NG = 0, // 条件を満たさない
    NRL_CHECK_OK // 条件を満たす
} Nrl_result_t;

```

(5) 論理値型の値

```

typedef enum {
    NRL_OFF = 0, // 不可
    NRL_ON // 可
} Nrl_boolean_t;

```

付録 C.3 関数の記述形式

(1) 条件情報取得関数

条件名, 条件の値の型および値の初期値を返します。

(a) 関数名

NrlGetConditionInfo

(b) 形式

```

Nrl_status_t NrlGetConditionInfo(
    Nrl_char_t** ConditionName, // /* 条件名 */
    Nrl_value_type_t* ValueType, // /* 値の型 */
    Nrl_init_value_struct_t* InitValueList // /* 条件の値の初期値*/
);

```

(c) 引数

ConditionName : 条件名 (出力引数)

ValueType : 条件の値の型 (出力引数)

InitValueList : 条件の値の初期値 (出力引数)

(d) 戻り値

NRL_OK : 正常終了

NRL_ERROR : 異常終了

(2) チェック関数

定義されている名称 (文字列) が, 条件にあっているかどうかをチェックします。

(a) 関数名

NrlCheckName

(b) 形式

```
Nrl_status_t NrlCheckName(  
    const Nrl_char_t*      Name,  
                           /* チェック対象の文字列 */  
    const void*           Value,  
                           /* 値 */  
    Nrl_result_t*        Result  
                           /* チェック結果 */  
);
```

(c) 引数

Name : チェック対象の文字列 (入力引数)

Value : 値 (入力引数)

Result : チェック結果 (出力引数)

(d) 戻り値

NRL_OK : 正常終了

NRL_ERROR : 異常終了

付録 D 検査項目インタフェース

SEWB+/REPOSITORY では、独自の検査 DLL を作成するためのインタフェースを提供しています。

ここでは、検査 DLL の作成の概要と、インタフェース（ヘッダファイルおよび関数）の詳細について説明します。なお、このインタフェースの利用には、C 言語の基本文法をある程度理解されていることを前提としています。

付録 D.1 検査 DLL の作成

(1) インタフェースの構成

検査項目インタフェースは、次に示すヘッダファイル、関数で構成します。

(a) ヘッダファイル

ヘッダファイル (Dsc_interface.h) には、関数の戻り値などの宣言が記述されています。

(b) 関数

(i) データ項目の検査関数 (DscCheckItem)

データ項目の定義情報をチェックし、その結果を Result に返す関数です。
この関数はデータ項目の検査 DLL に必ず定義します。

(ii) 業務ルールの検査関数 (DscCheckRule)

業務ルールの定義情報をチェックし、その結果を Result に返す関数です。
この関数は業務ルールの検査 DLL に必ず定義します。

(iii) エラー ID からのメッセージ取得関数 (DscGetMessageFromErrID)

エラー ID からエラーメッセージを取得する関数です。指定されたエラー ID に対応するメッセージがない場合は、戻り値に DSC_ERROR を返します。
この関数は検査 DLL に必ず定義します。

(iv) エラー ID 領域の解放関数 (DscFreeErrID)

検査関数で返されたエラー ID 領域を破棄する関数です。
この関数は検査 DLL に必ず定義します。

(2) インタフェースが格納されているディレクトリ

ヘッダファイルが格納されているディレクトリを次に示します。

サーバのSEWB+ 基本開発環境の組み込み先パス名¥Repository¥include

(3) 検査 DLL の登録

環境構築ユーティリティを使用して、独自に作成した検査 DLL を登録すると、SEWB+/

REPOSITORY で利用できるようになります。最大 10 種類の検査 DLL を登録できます。なお、登録時には、DLL 中に定義されているユーザ定義の関数の有無がチェックされません。

検査 DLL を登録すると、登録時に指定したパスとファイル名を基に、検査 DLL から情報が引き継がれ、[データ項目] ダイアログの [検査項目] タブの検査項目一覧に表示されます。

付録 D.2 ヘッドファイル (Dsc_interface.h)

(1) 構成項目情報構造体 (Dsc_member_item_struct_t)

```
typedef struct Dsc_member_item_struct_t {
    Dsc_int_t      item_kind;           //データ項目種別
    Dsc_char_t    *item_name;          //データ項目名
    Dsc_int_t      type_group;         //分類
    Dsc_int_t      total;              //けた数
    Dsc_int_t      dec_part;           //小数部けた数
    Dsc_int_t      repeat;             //反復回数
    Dsc_char_t    *var_name[11];       //名前(言語別)
    Dsc_int_t      type[11];           //タイプ(言語別)
    Dsc_char_t    *type_name[11];     //タイプキーワード名
                                        (言語別)
    Dsc_char_t    *type_comment[11];  //タイプキーワードの
                                        説明(言語別)
    Dsc_char_t    *edit_str[11];      //タイプ修飾情報
                                        (言語別)
    void          *reserve;            //予備
} Dsc_member_item_struct_t;
```

(2) データ項目情報構造体 (Dsc_item_struct_t)

```
typedef struct Dsc_item_struct_t {
    Dsc_int_t      item_kind;           //データ項目種別
    Dsc_char_t    *item_name;          //データ項目名
    Dsc_char_t    *parent_name;        //親データ項目名
    Dsc_char_t    *dict_name;          //辞書フォルダ名
    Dsc_char_t    *std_name[5];        //標準名称(国語別)
    Dsc_char_t    *furigana[5];        //フリガナ(国語別)
    Dsc_char_t    *comment[5];         //コメント(国語別)
    Dsc_char_t    *field[21];          //フィールド1~20
    Dsc_int_t      type_group;         //分類
    Dsc_int_t      total;              //けた数
    Dsc_int_t      dec_part;           //小数部けた数
    Dsc_int_t      repeat;             //反復回数
    Dsc_char_t    *var_name[11];       //名前(言語別)
    Dsc_int_t      type[11];           //タイプ(言語別)
    Dsc_char_t    *type_name[11];     //タイプキーワード名
                                        (言語別)
    Dsc_char_t    *type_comment[11];  //タイプキーワードの
                                        説明(言語別)
    Dsc_char_t    *type_free[11];     //タイプフリー定義
                                        (言語別)
    Dsc_char_t    *edit_str[11];      //タイプ修飾情報
                                        (言語別)
}
```

```

Dsc_char_t  *init_val[11];      //初期値(言語別)
Dsc_char_t  *lang_field[11];   //言語別フィールド
                                   (言語別)
Dsc_int_t   exam_result;       //検査状態
time_t      update_time;      //更新日時
Dsc_member_item_struct_t *member_array;
                                   //構成項目情報
Dsc_int_t   member_cnt;       //構成項目数
void        *reserve;         //予備
} Dsc_item_struct_t;

```

(3) 関連データ項目情報構造体 (Dsc_link_item_struct_t)

```

typedef struct Dsc_link_item_struct_t {
Dsc_int_t   item_kind;        //データ項目種別
Dsc_char_t  *item_name;      //データ項目名
Dsc_char_t  *keyword;       //キーワード
Dsc_trigger_t trigger;      //適用条件
Dsc_int_t   type_group;     //分類
Dsc_int_t   total;          //けた数
Dsc_int_t   dec_part;       //小数部けた数
Dsc_int_t   repeat;         //反復回数
Dsc_char_t  *var_name[11];   //名前(言語別)
Dsc_int_t   type[11];       //タイプ(言語別)
Dsc_char_t  *type_name[11];  //タイプキーワード名
                                   (言語別)
Dsc_char_t  *type_comment[11]; //タイプキーワードの説明(言語別)
Dsc_char_t  *type_free[11];  //タイプフリー定義(言語別)
                                   (言語別)
Dsc_char_t  *edit_str[11];   //タイプ修飾情報(言語別)
Dsc_char_t  *init_val[11];   //初期値(言語別)
Dsc_char_t  *lang_field[11]; //言語別フィールド(言語別)
                                   (言語別)
void        *reserve;         //予備
} Dsc_link_item_struct_t;

```

(4) 業務ルール情報構造体 (Dsc_rule_struct_t)

```

typedef struct Dsc_rule_struct_t {
Dsc_int_t   rule_kind;       //業務ルール種別
Dsc_char_t  *rule_name;     //業務ルール名
Dsc_char_t  *dict_name;     //辞書フォルダ名
Dsc_char_t  *std_name[5];   //標準名称(国語別)
Dsc_char_t  *furigana[5];  //フリガナ(国語別)
Dsc_char_t  *comment[5];   //コメント(国語別)
Dsc_char_t  *field[4];     //フィールド1~3
Dsc_char_t  *src_info[11];  //ルースクリプト
                                   (言語別)
Dsc_int_t   exam_result;    //検査状態
time_t      update_time;    //更新日時
Dsc_link_item_struct_t *item_array;
                                   //関連データ項目情報
Dsc_int_t   item_cnt;       //関連データ項目数
void        *reserve;       //予備
} Dsc_rule_struct_t;

```

(5) 関数の戻り値 (Dsc_status_t)

```
typedef enum {
    DSC_ERROR = 0,           //エラー発生
    DSC_OK             //正常終了
} Dsc_status_t;
```

(6) 検査結果 (Dsc_result_t)

```
typedef enum {
    DSC_CHECK_NG = 0,       //エラーあり
    DSC_CHECK_OK          //エラーなし
} Dsc_result_t;
```

(7) エラー ID (Dsc_err_id_t)

```
typedef enum {
    DSC_ERRID_ITEM_ERR_1 = 10000, //データ項目検査用
    DSC_ERRID_ITEM_ERR_2,
    DSC_ERRID_ITEM_ERR_3,
    DSC_ERRID_ITEM_ERR_4,
    DSC_ERRID_ITEM_ERR_5,
    :
    DSC_ERRID_ITEM_ERR_100,
    DSC_ERRID_RULE_ERR_1 = 20000, //業務ルール検査用
    DSC_ERRID_RULE_ERR_2,
    DSC_ERRID_RULE_ERR_3,
    DSC_ERRID_RULE_ERR_4,
    DSC_ERRID_RULE_ERR_5,
    :
    DSC_ERRID_RULE_ERR_100
} Dsc_err_id_t;
```

付録 D.3 関数の記述形式

(1) データ項目の検査関数

データ項目の定義情報をチェックし、その結果を Result に返します。

(a) 関数名

DscCheckItem - データ項目の定義情報チェック

(b) 形式

```
Dsc_status_t DscCheckItem(
    const Dsc_item_struct_t *Item,
    /* データ項目情報構造体 */
    Dsc_err_id_t **ErrIDList,
    /* エラーID配列 */
    Dsc_int_t *ErrCount,
    /* エラーの数 */
    Dsc_result_t *Result
    /* 検査結果 */
);
```

(c) 引数

Item

データ項目情報構造体を格納するポインタを指定します。

- データ項目種別 (item_kind) には、対象データ項目の種別が返ります。
DSD_ITM_KIND_SIMPLE : 単項目
DSD_ITM_KIND_COMPOSITE : 結合項目
- 親データ項目名 (parent_name) には、対象データ項目の親データ項目名が返ります。ただし、親データ項目なし (辞書フォルダ直下のデータ項目) の場合は NULL が返ります。
- 標準名称 (std_name), フリガナ (furigana), コメント (comment) は、配列の添字に国語区分番号を指定することで値を取得できます。指定できる値を次に示します。
DSD_NATIONAL_NO_JAPANESE : 国語区分「日本語」
DSD_NATIONAL_NO_ENGLISH : 国語区分「英語」
DSD_NATIONAL_NO_RESERVED_1 : 国語区分「未設定 1」
DSD_NATIONAL_NO_RESERVED_2 : 国語区分「未設定 2」
- フィールド (field) は、配列の添字にフィールド番号を指定することで値を取得できます。指定できる値は、「DSD_ITEM_FIELD_NO_x」です。x は 1 ~ 20 の数字で、それぞれフィールド 1 ~ 20 に対応します。
- 分類 (type_group) には、次に示す DS_type_group_t 列挙型 の値が返ります。
DSD_GRP_ALPHABETIC : 英数字文字列データ
DSD_GRP_JAPANESE : 日本語文字列データ
DSD_GRP_INTEGRAL : 整数データ
DSD_GRP_POSITIVE : 正整数データ
DSD_GRP_REAL : 実数データ
DSD_GRP_DATE : 日付データ
DSD_GRP_TIME : 時刻データ
DSD_GRP_COMPOSITE : 結合データ
DSD_NO_GRP : 分類なし
- 名前 (var_name), タイプ (type), タイプキーワード名 (type_name), タイプキーワードの説明 (type_comment), タイプフリー定義 (type_free), タイプ修飾情報 (edit_str), 初期値 (init_val), 言語別フィールド (lang_field) は、配列の添字に言語区分番号を指定することで値を取得できます。指定できる値を次に示します。
DSD_PROG_NO_COBOL : 言語区分「COBOL 又は OOCOBOL」
DSD_PROG_NO_C : 言語区分「C 又は C++」
DSD_PROG_NO_SQL : 言語区分「SQL」
DSD_PROG_NO_CORBA : 言語区分「IDL(CORBA)」
DSD_PROG_NO_EUR : 言語区分「帳票定義」
DSD_PROG_NO_JAVA : 言語区分「Java」
DSD_PROG_NO_RESERVED_4 : 言語区分「言語 4」

DSD_PROG_NO_RESERVED_5 : 言語区分「言語 5」

DSD_PROG_NO_RESERVED_6 : 言語区分「言語 6」

DSD_PROG_NO_RESERVED_7 : 言語区分「言語 7」

- 言語区分「COBOL 又は OOCOBOL」のメンバ edit_str には、編集文字列の定義情報、言語区分「IDL(CORBA)」のメンバ edit_str には、シーケンスの定義情報が返ります。
- 言語区分「COBOL 又は OOCOBOL」、「C 又は C++」、「IDL(CORBA)」で、タイプが「フリー定義」以外の場合、メンバ type_free には、文字列表現された取りうる値の定義情報が返ります。

例えば、取りうる値リストの記号と値の組み合わせが次に示す表の場合、

"TOKYO=,NAGOYA,OOSAKA= ¥,¥=¥¥ ¥¥ " という文字列が返ります。

記号	値
"TOKYO"	" "
"NAGOYA"	NULL
"OOSAKA"	" ,=¥ ¥ "

- タイプ (type) には、DS_type_id_t 列挙型 の値が返ります。
- タイプキーワード名 (type_name) には、DS_type_id_t 列挙型 の値に対応したタイプキーワード名が返ります。
- タイプキーワードの説明 (type_comment) には、DS_type_id_t 列挙型 の値に対応したタイプキーワードの説明が返ります。
- 検査状態 (exam_result) には、次に示す DS_exam_result_t 列挙型 の値が返ります。

DSD_EXM_OK : エラーなし

DSD_EXM_NG : エラーあり

DSD_NO_EXM : 未検査

注

列挙型の値は、ヘッダファイル (Dsc_interface.h) 中のコメントで確認できます。

ErrIDList

エラー ID 配列へのポインタを受け取るエリアのポインタを指定します。

ErrCount

エラーの数 (ErrIDList に指定した配列の要素数) を受け取るエリアのポインタを指定します。

Result

検査結果を受け取るエリアのポインタを指定します。

(d) 戻り値

DSC_OK : 正常終了

DSC_ERROR : エラーが発生した

(e) 注意事項

検査 DLL を複数登録する場合は、検査関数で返すエラー ID (Dsc_err_id_t 列挙型) は、登録した検査 DLL 内でユニークにする必要があります。

次に該当する場合、データ項目情報構造体または構成項目情報構造体のメンバには次の未定義値が返ります。

- データ項目種別 (item_kind) が DSD_ITM_KIND_SIMPLE (単項目) でけた数, 小数部けた数, 反復回数が未定義の場合
- データ項目種別 (item_kind) が DSD_ITM_KIND_COMPOSITE (結合項目) で反復回数が未定義の場合

各メンバの未定義値を次に示します。

- けた数 (total): DSD_ITM_TOTAL_NULL (-1)
- 小数部けた数 (dec_part): DSD_ITM_DEC_PART_NULL (-1)
- 反復回数 (repeat): DSD_ITM_REPEAT_NULL (-1)

データ項目種別 (item_kind) が DSD_ITM_KIND_COMPOSITE (結合項目) の場合, けた数 (total), 小数部けた数 (dec_part), タイプ (type) の値は保証されません。

(2) 業務ルールの検査関数

業務ルールの定義情報をチェックし, その結果を Result に返します。

(a) 関数名

DscCheckRule - 業務ルールの定義情報チェック

(b) 形式

```
Dsc_status_t DscCheckRule(
    const Dsc_Rule_struct_t    *Rule,
                                /* 業務ルール情報構造体 */
    Dsc_err_id_t               **ErrIDList,
                                /* エラーID配列 */
    Dsc_int_t                   *ErrCount,
                                /* エラーの数 */
    Dsc_result_t               *Result,
                                /* 検査結果 */
);
```

(c) 引数

Rule

業務ルール情報構造体を格納するポインタを指定します。

- 業務ルール種別 (rule_kind) には、対象業務ルールの種別が返ります。
DSD_BRL_KIND_SINGLE : 単項目関連業務ルール (共有する)
DSD_BRL_KIND_MULTI : 複項目関連業務ルール (専用にする)
DSD_BRL_KIND_MULTI_SAME_ANY : 同一項目用業務ルール (結合, 単項目)
DSD_BRL_KIND_MULTI_SAME_SIMPLE : 同一項目用業務ルール (単項目)
- 標準名称 (std_name), フリガナ (furigana), コメント (comment) は、配列の添字に国語区分番号を指定することで値を取得できます。指定できる値を次に示します。
DSD_NATIONAL_NO_JAPANESE : 国語区分「日本語」
DSD_NATIONAL_NO_ENGLISH : 国語区分「英語」
DSD_NATIONAL_NO_RESERVED_1 : 国語区分「未設定 1」
DSD_NATIONAL_NO_RESERVED_2 : 国語区分「未設定 2」
- フィールド (field) は、配列の添字にフィールド番号を指定することで値を取得できます。指定できる値を次に示します。
DSD_RULE_FIELD_NO_1 : フィールド 1
DSD_RULE_FIELD_NO_2 : フィールド 2
DSD_RULE_FIELD_NO_3 : フィールド 3
- ルールスクリプト (src_info) は、配列の添字に言語区分番号を指定することで値を取得できます。指定できる値を次に示します。
DSD_PROG_NO_COBOL : 言語区分「COBOL 又は OOCOBOL」
DSD_PROG_NO_C : 言語区分「C 又は C++」
DSD_PROG_NO_SQL : 言語区分「SQL」
DSD_PROG_NO_CORBA : 言語区分「IDL(CORBA)」
DSD_PROG_NO_EUR : 言語区分「帳票定義」
DSD_PROG_NO_JAVA : 言語区分「Java」
DSD_PROG_NO_RESERVED_4 : 言語区分「言語 4」
DSD_PROG_NO_RESERVED_5 : 言語区分「言語 5」
DSD_PROG_NO_RESERVED_6 : 言語区分「言語 6」
DSD_PROG_NO_RESERVED_7 : 言語区分「言語 7」
- 検査状態 (exam_result) には、次に示す DS_exam_result_t 列挙型 の値が返ります。
DSD_EXM_OK : エラーなし
DSD_EXM_NG : エラーあり
DSD_NO_EXM : 未検査
- 関連データ項目情報の適用条件 (trigger) には、次に示す DS_trigger_t 列挙型 の値が返ります。
DSD_NO_TRIGGER : 未指定
DSD_TRIGGER_INPUT : 入力
DSD_TRIGGER_OUTPUT : 出力
関連データ項目情報のほかのメンバについては、「(1) データ項目の検査関数」の引数 Item の説明を参照してください。

注

列挙型の値は、ヘッダファイル (Dsc_interface.h) 中のコメントで確認できます。

ErrIDList

エラー ID 配列へのポインタを受け取るエリアのポインタを指定します。

ErrCount

エラーの数 (ErrIDList に指定した配列の要素数) を受け取るエリアのポインタを指定します。

Result

検査結果を受け取るエリアのポインタを指定します。

(d) 戻り値

DSC_OK : 正常終了

DSC_ERROR : エラーが発生した

(e) 注意事項

複数の検査 DLL を登録する場合は、検査関数で返すエラー ID (Dsc_err_id_t 列挙型) は、登録した検査 DLL 内でユニークにする必要があります。

業務ルール種別が DSD_BRL_KIND_SINGLE (単項目関連業務ルール (共有する)) または DSD_BRL_KIND_MULTI (複項目関連業務ルール (専用にする)) のとき、次に該当する場合は関連データ項目情報構造体のメンバには、未定義値が返ります。

- 関連データ項目情報構造体のデータ項目種別 (item_kind) が DSD_ITM_KIND_SIMPLE (単項目) で、けた数、小数部けた数、反復回数が未定義の場合
- 関連データ項目情報構造体のデータ項目種別 (item_kind) が DSD_ITM_KIND_COMPOSITE (結合項目) で、反復回数が未定義の場合

各メンバの未定義値を次に示します。

- けた数 (total): DSD_ITM_TOTAL_NULL (-1)
- 小数部けた数 (dec_part): DSD_ITM_DEC_PART_NULL (-1)
- 反復回数 (repeat): DSD_ITM_REPEAT_NULL (-1)

業務ルール種別が DSD_BRL_KIND_SINGLE (単項目関連業務ルール (共有する)) または DSD_BRL_KIND_MULTI (複項目関連業務ルール (専用にする)) のとき、関連データ項目情報構造体のデータ項目種別 (item_kind) が DSD_ITM_KIND_COMPOSITE (結合項目) の場合、けた数 (total)、小数部けた数 (dec_part)、タイプ (type) の値は保証されません。

業務ルール種別が DSD_BRL_KIND_MULTI_SAME_ANY または DSD_BRL_KIND_MULTI_SAME_SIMPLE (同一項目用業務ルール) の場合、関連

データ項目情報構造体のキーワード (keyword) と適用条件 (trigger) 以外は保証されません。

(3) エラー ID からメッセージの取得関数

エラー ID からエラーメッセージを取得します。

指定されたエラー ID に対応するメッセージがない場合は、戻り値に DSC_ERROR を返します。

(a) 関数名

DscGetMessageFromErrID - エラー ID からメッセージを取得

(b) 形式

```
Dsc_status_t DscGetMessageFromErrID(
    Dsc_err_id_t ErrID, /* エラーID */
    Dsc_char_t **Message /* エラーメッセージ */
);
```

(c) 引数

ErrID

エラー ID を指定します。

Message

エラーメッセージへのポインタを受け取るエリアのポインタを指定します。

- エラー ID (Dsc_err_id_t) に対応したメッセージを返します。
エラー ID とメッセージの対応例を次に示します。

DSC_ERRID_ITEM_ERR_1

コメント (COBOL 又は OOCOBOL) が指定されていません。

DSC_ERRID_ITEM_ERR_2

コメント (C 又は C++) が指定されていません。

DSC_ERRID_RULE_ERR_1

ルールスクリプト (COBOL 又は OOCOBOL) に誤りがあります。

DSC_ERRID_RULE_ERR_2

ルールスクリプト (C 又は C++) に誤りがあります。

(d) 戻り値

DSC_OK : 正常終了

DSC_ERROR : エラー ID に対応するメッセージがない

(4) エラー ID 領域の解放関数

引数 ErrIDList で指定されたエラー ID 配列を破棄します。

(a) 関数名

DscFreeErrID - エラー ID 配列の破棄

(b) 形式

```
void DscFreeErrID(
    Dsc_err_id_t    *ErrIDList    /* エラーID配列 */
);
```

(c) 引数

ErrIDList

削除するエラー ID (検査関数で返されたエラー ID) のポインタを指定します。

(d) 戻り値

戻り値はありません。

付録 D.4 エラーメッセージ

検査関数でエラーの場合

DscCheckItem, DscCheckRule の戻り値として DSC_ERROR が返された場合, 次のメッセージが表示されます。

「検査 DLL の検査関数でエラーが発生しました。DLL : MMMM, 関数 : NNNN」

MMMM : 検査 DLL 名称 (パス名を含む)

NNNN : 検査関数名称 (DscCheckItem または DscCheckRule)

注

P 2651 1524 SEWB+/REPOSITORY-BROWSER のバージョン 02-02 以前で, 検査関数でエラー ID が返された場合は, 次のメッセージとなります。

「検査でエラーが発生しました。エラーコード : XXXXX」

XXXXX : 検査関数で返されたエラー ID

エラー ID からメッセージ取得関数で取得できた場合

DscGetMessageFromErrID の戻り値で DSC_OK が返された場合, 取得したメッセージが表示されます。

エラー ID からメッセージ取得関数で取得できなかった場合

DscGetMessageFromErrID の戻り値で DSC_ERROR が返された場合, 次のメッセージが表示されます。

「定義内容に誤りがあります。エラーコード : XXXXX」

XXXXX : 検査関数で返されたエラー ID

付録 E 標準提供の同一項目用業務ルール

SEWB+/REPOSITORY では、実際の業務に使用することを想定した同一項目用業務ルールを提供しています。この同一項目用業務ルールは、辞書に登録すれば、そのまま利用できます。独自の同一項目用業務ルールを作成する場合は、サンプルとしてお役立てください。

付録 E.1 同一項目用業務ルールの概要

業務ルール名：「同一項目間転記」

処理概要

同一のデータ項目間で、入力側のデータ項目から出力側のデータ項目へデータを転記します。一般に、アプリケーションの入力側でそのデータ項目が含まれるレコード定義の構成と、出力側でそのデータ項目が含まれるレコード定義の構成は、異なる場合があると考えられます。この同一項目用業務ルールでは、それぞれのレコード定義を構成する結合項目に異なる反復回数が定義されている場合の転記処理にも対応しています。

データ項目の対象範囲

適用するデータ項目には、「すべての単項目」が指定されています。リポジトリ内の結合項目ではないデータ項目（単項目）に対して適用できます。

付録 E.2 ルールスクリプト

ルールスクリプトを次に示します。なお、このルールスクリプトには、テンプレート記述言語が使用されており、ルールスクリプトとテンプレートで共通に使用する可変記号が定義されています。このため、どのようにこの同一項目用業務ルールを利用するかについてを、あらかじめテンプレート作成者と検討しておくことをお勧めします。

```

@@*****
@@** MOVE 文生成用ルールスクリプト **
@@** 注意:SEWB+/CONSTRUCTION テンプレートに「@@global @RULE_FLAG;」と **
@@** 作業領域を展開するための「@@merge @作業領域_MERGE;」の指定が必要 **
@@*****
@@set @ilevel = @OCCURS_DIMENSION(@IN_ITEM);
@@set @ioccurs = {@OCCURS1(@IN_ITEM), @OCCURS2(@IN_ITEM), @OCCURS3(@IN_ITEM),
                 @OCCURS4(@IN_ITEM), @OCCURS5(@IN_ITEM), @OCCURS6(@IN_ITEM),
                 @OCCURS7(@IN_ITEM)};
@@set @olevel = @OCCURS_DIMENSION(@OUT_ITEM);
@@set @ooccurs = {@OCCURS1(@OUT_ITEM), @OCCURS2(@OUT_ITEM), @OCCURS3(@OUT_ITEM),
                 @OCCURS4(@OUT_ITEM), @OCCURS5(@OUT_ITEM), @OCCURS6(@OUT_ITEM),
                 @OCCURS7(@OUT_ITEM)};

@@set @iitem = '@IN_ITEM';
@@set @oitem = '@OUT_ITEM';
@@*****
@@** これ以降は, SEWB+/CONSTRUCTION 部品 とすることもできる。 **
@@** 部品への引数は @iitem, @oitem, @ilevel, @olevel, @ioccurs, @ooccurs **
@@*****
@@if (@ilevel == 0 && @olevel == 0) @@* 反復回数がない場合
    MOVE @iitem TO @oitem
@@elseif (@olevel == 0) @@* 入力側のみ反復回数がある場合
    @@set @wcnt = 1;
    @@set @ary = "1";
    @@while (@wcnt < @ilevel)
        @@set @ary = "@ary,1";
        @@set @wcnt = @wcnt + 1;
    @@end;
    MOVE @iitem(@ary) TO @oitem
@@elseif (@ilevel == 0) @@* 出力側のみ反復回数がある場合
    @@set @wcnt = 1;
    @@set @ary = "1";
    @@while (@wcnt < @olevel)
        @@set @ary = "@ary,1";
        @@set @wcnt = @wcnt + 1;
    @@end;
    MOVE @iitem TO @oitem(@ary)
@@elseif (@ioccurs[1] == @ooccurs[1] && @@* 次元数と反復回数が等しい場合
        @ioccurs[2] == @ooccurs[2] &&
        @ioccurs[3] == @ooccurs[3] &&
        @ioccurs[4] == @ooccurs[4] &&
        @ioccurs[5] == @ooccurs[5] &&
        @ioccurs[6] == @ooccurs[6] &&
        @ioccurs[7] == @ooccurs[7] )
    @@set @wcnt = 1;
    @@while (@wcnt <= @ilevel)
        @@set @wname = "CNT@wcnt";
        @@if (!(@defined(@RULE_FLAG[@wname])))
            @@put @作業領域_MERGE <<
01 @wname PIC S9(9) COMP.
            @@end;
            @@set @RULE_FLAG[@wname] = 'Y';
        @@end;
    PERFORM VARYING @wname FROM 1 BY 1
        UNTIL @wname > @ioccurs[@wcnt]

```

```

@@set @wcnt = @wcnt + 1;
@@end;
@@set @wcnt = 2;
@@set @ary = 'CNT1';
@@while (@wcnt <= @ilevel)
    @@set @ary = "@ary.CNT@wcnt";
@@set @wcnt = @wcnt + 1;
@@end;
    MOVE @iitem(@ary) TO @oitem(@ary)
@@set @wcnt = 1;
@@while (@wcnt <= @ilevel)
    END-PERFORM
@@set @wcnt = @wcnt + 1;
@@end;
@@else
@@* 次元数と反復回数が異なる場合
@@if (!(@defined(@RULE_FLAG[LOOP_END])))
    @@put @作業領域_MERGE <<
    01 LOOP-END PIC X(1).
    @@end;
    @@set @RULE_FLAG[LOOP_END] = 'Y';
@@end;
@@set @wcnt = 1;
    MOVE SPACE TO LOOP-END
@@while (@wcnt <= @ilevel)
    @@set @wname = "CNT1@wcnt";
    @@if (!(@defined(@RULE_FLAG[@wname])))
        @@put @作業領域_MERGE <<
    01 @wname PIC S9(9) COMP.
        @@end;
        @@set @RULE_FLAG[@wname] = 'Y';
    @@end;
    MOVE 1 TO @wname
    @@if (@wcnt == 1)
        @@set @iary = "@wname";
    @@else
        @@set @iary = "@iary.@wname";
    @@end;
    @@set @wcnt = @wcnt + 1;
@@end;
@@set @wcnt = 1;
@@while (@wcnt <= @olevel)
    @@set @wname = "CNT0@wcnt";
    @@if (!(@defined(@RULE_FLAG[@wname])))
        @@put @作業領域_MERGE <<
    01 @wname PIC S9(9) COMP.
        @@end;
        @@set @RULE_FLAG[@wname] = 'Y';
    @@end;
    PERFORM VARYING @wname FROM 1 BY 1
    UNTIL @wname > @ooccurs[@wcnt]
    OR LOOP-END = 'E'
    @@if (@wcnt == 1)
        @@set @oary = "@wname";
    @@else
        @@set @oary = "@oary.@wname";
    @@end;

```

```
    @@set @wcnt = @wcnt + 1;
@@end;
    MOVE @iitem(@iary) TO @oitem(@oary)
@@set @wcnt = @ilevel;
@@while (@wcnt >= 1)
    ADD 1 TO CNT1@wcnt
    IF CNT1@wcnt > @ioccurs[@wcnt]
        MOVE 1 TO CNT1@wcnt
    @@set @wcnt = @wcnt - 1;
@@end;
    MOVE 'E' TO LOOP-END
@@set @wcnt = @ilevel;
@@while (@wcnt >= 1)
    ELSE
    CONTINUE
    END-IF
    @@set @wcnt = @wcnt - 1;
@@end;
@@set @wcnt = 1;
@@while (@wcnt <= @olevel)
    END-PERFORM
    @@set @wcnt = @wcnt + 1;
@@end;
@@end;
```

付録 F 用語解説

(英字)

APPGALLERY Enterprise

さまざまな部品（再利用性を高めるために機能単位に分割されたソフトウェア）を組み合わせるアプリケーションを開発する、アプリケーション開発環境です。画面上で部品を組み合わせるビジュアルプログラミングを採用しています。特に、GUI（グラフィカル ユーザインタフェース）を中心としたプログラムの作成に適しています。大規模な C/S システムの開発では、クライアント側のプログラムの開発に適しています。

APPGALLERY Enterprise では、SEWB+/REPOSITORY と連携して、大規模システムのアプリケーション開発を支援する機能を備えています。

CORBA (Common Object Request Broker Architecture)

OMG (Object Management Group) が標準化を進めている、ネットワークに分散したオブジェクト間のメッセージング機構の規約です。ORB (オブジェクト リクエスト ブローカ。クライアントとサーバに分散されたアプリケーション間の通信を制御し、要求を伝える役割を持つ分散オブジェクト実行環境) の標準仕様がまとめられています。

CSV 形式ファイル (Comma Separated Values 形式ファイル)

CSV 形式でデータが記述されたファイルのことです。CSV 形式とは、表計算プログラムやリレーショナルデータベースでデータを扱えるテキストデータの形式をいいます。データの区切りをコンマ(,)、レコードの区切りを改行で表します。レコードは可変長形式になります。辞書情報を一括して登録したり出力したりするには、CSV 形式ファイルを使います。

EUR Professional Edition

SEWB+/REPOSITORY と連携し、レポートの設計から印刷までを支援するツールです。EUR Professional Edition では、ワープロを操作する感覚で、自由な形式のレポートを効率良く設計できます。設計時には、SEWB+/REPOSITORY の辞書のレコード定義（最上位結合項目）から、レポートの設計に必要なフィールド定義情報を自動的に生成して利用できます。また、設計したレポートの印刷にはプログラミングの必要がなく、レポート出力アプリケーションを容易に作成できます。

GUI 属性

GUI 属性は、APPGALLERY Enterprise を使ったアプリケーション開発で使用する部品（再利用性を高めるために機能単位に分割されたソフトウェア）のプロパティに関する設定です。APPGALLERY Enterprise でリポジトリの資源を利用したアプリケーション開発をする場合に、GUI 属性のマッピング情報とプロパティセットの設定情報が参照されます。

IDL (Interface Definition Language)

ORB (分散オブジェクト実行環境) を利用する C/S システムで、オブジェクト（個々のアプリケーションのこと）のインタフェースを定義する言語です。SEWB+ では、CORBA の IDL をサポートしています。したがって、SEWB+/CS-DESIGN および SEWB+/REPOSITORY を連携して作成した IDL は、CORBA をサポートした ORB（日立製品では、TPBroker があります）が提供する IDL

コンパイラでコンパイルし、利用することができます。

SEWB+/CONSTRUCTION

SEWB+ 基本開発環境のプログラム構築支援機能のことです。

特に、TP モニタやジョブ運用管理ツールなどを利用する大規模な C/S システム上で動作するサーバ側のプログラムおよびバッチシステムの開発に適しています。テンプレート、データ定義、プログラム定義の機能を使い、C/S システムのサーバプログラムと、クライアントプログラムに必要なサーバプログラムとのリンクのインタフェースを効率良く作成できます。

SEWB+/CS-DESIGN

ORB (分散オブジェクト実行環境) を利用した C/S システムのアプリケーション (オブジェクト) の論理構成設計を支援するツールです。特に、3 層 - 画面入出力などのユーザインタフェースを制御する「プレゼンテーション層」、中核的な業務処理を実行する「ファンクション層」、データベースの管理およびデータベースへのアクセスを実行する「データ層」 - で構成される C/S システムの設計に適しています。SEWB+/CS-DESIGN では、C/S システムのアプリケーション分割や関連の定義を、論理設計図を描きながらビジュアルに設計できます。論理設計図には、各アプリケーションの外部インタフェースとして、アプリケーションの操作関数、属性や例外を定義できます。また、定義したインタフェースの情報からは、CORBA の IDL を生成できます。なお、インタフェース情報の定義では、SEWB+/REPOSITORY の辞書で一元管理されたデータ項目を利用するため、複数の開発者によるチーム開発にも適しています。

SEWB+/RECORD DEFINER

SEWB+ 基本開発環境のレコード設計支援機能のことです。

SEWB+/REPOSITORY の辞書情報を使用して、レコード情報を定義できます。

SEWB+/REPORT MANAGER

SEWB+ で定義した情報の印刷を支援するツールです。SEWB+/CONSTRUCTION と連携してプログラム処理概要図を、SEWB+/REPOSITORY と連携してインパクトレポートやデータ項目または業務ルールの定義内容などを印刷できます。これらのドキュメントを定義内容の確認や、保守作業に役立てられます。

SEWB+ 基本開発環境

次に示す機能を一つに統合したツールです。

- リポジトリ管理機能 (SEWB+/REPOSITORY)
- リポジトリブラウザ機能 (SEWB+/REPOSITORY-BROWSER)
- プログラム構築支援機能 (SEWB+/CONSTRUCTION)
- レコード設計支援機能 (SEWB+/RECORD DEFINER)

SEWB+ 基本開発環境セット

SEWB+ 基本開発環境と Groupmax ObjectServer から構成されるプログラムプロダクトです。

SEWB+ ツール

PC を使った分散開発環境で、システム開発を支援する SEWB+ シリーズのツールの総称です。

(ア行)

アクセス権

リポジトリに格納された辞書やドキュメントを、参照したり更新したりする権利のことです。所有者（辞書やドキュメントを作成するユーザ）、グループおよびすべてのユーザに対して、それぞれのアクセス権を設定できます。

インポート

エクスポートで取り出したシステム開発情報をリポジトリに取り込むことをいいます。各サーバに分散された資源を収集する場合などに利用します。

エクスポート

リポジトリのシステム開発情報を、ハードディスク上のファイル、フロッピーディスク、またはカートリッジ磁気テープなどの外部媒体に取り出すことをいいます。ルートドキュメントフォルダ単位または辞書フォルダ単位でエクスポートできます。各サーバへ資源を配布または複製する場合に利用します。

(カ行)

関連

データ項目とドキュメント、データ項目と業務ルールというように、資源間に付けられた依存関係をいいます。関連は、リポジトリブラウザを使用して任意に設定できる「ユーザ関連」と、SEWB+ツールによって自動的に設定される「関連」に大別されます。リポジトリブラウザの関連ブラウザ機能を使用すると、関連づけられた資源をブラウジングでき、変更波及の解析などに利用できます。

キーワード

業務ルールに使用するデータ項目の仮の名称です。ルールスクリプトでは、データ項目の名称をキーワードで記述します。このキーワードは、SEWB+/CONSTRUCTIONでソースプログラム中にルールスクリプトが展開されたときに、データ項目の言語別の名前に置換されます。

業務ルール

データ項目に着目して、データ項目特有の処理を部品化したものです。SEWB+/CONSTRUCTIONを使ったアプリケーション開発で共通に利用することを目的としています。業務ルールは必ずデータ項目と関連づけて登録し、業務ルール辞書で管理します。

（データ項目と関連づけられない特殊な業務ルールもあります。用語「同一項目用業務ルール」の説明を参照してください）

実際にソースプログラムに展開する処理の内容は、ルールスクリプトに記述します。業務ルールに適した処理としては、データ項目の値や形式の妥当性をチェックする処理、形式を変換する処理、編集処理、複数のデータ項目で成り立つ計算処理などが挙げられます。

業務ルール辞書

アプリケーション開発で共有する業務ルールを格納する辞書です。

グループ

複数のユーザ（SEWB+/REPOSITORYの利用者）をまとめる単位です。ユーザが所属するグループは、システム管理者によってグループ登録ファイルに登録されます。辞書およびオブジェクトに対するアクセス権は、グループ単位で設定することもできます。

継承

データ項目間で共通の性質を持つデータ項目（上位データ項目）から、その定義情報を複数のデータ項目（下位データ項目）に引き継ぐことです。継承によって、データ項目を階層化できます。上位データ項目の定義情報を変更すれば、継承関係にある下位データ項目の定義情報も同時に変更されます。

結合項目

複数のデータ項目が連結して定義されるデータ項目です。COBOL の集団項目、C 言語の構造体の考え方に相当します。リポジトリブラウザの結合項目ブラウザ機能を使用すると、データ項目を結合関係に基づいてブラウジングしたり編集したりできます。

構成項目

結合項目の構成要素となるデータ項目です。

(サ行)

辞書

リポジトリで管理されるデータ項目辞書および業務ルール辞書の総称です。一つのリポジトリ上に、管理内容や用途に応じて複数の辞書を作成できます。

辞書フォルダ

リポジトリで辞書を管理する最上位の単位です。辞書フォルダは、データ項目辞書および業務ルール辞書を管理します。

システム管理者

Windows 2000 Server、Windows Server 2003 または Windows Server 2003 x64 の Administrators 権限を持ち、SEWB+/REPOSITORY を利用したシステム開発環境を管理する人です。ユーザ登録ファイルに、ユーザ名 [system] で登録されているユーザを指します。

(タ行)

データ項目

ファイル設計、レコード設計、リレーショナルデータベース設計などのシステム設計およびプログラム設計と作成で使用する情報の基本単位のことです。

データ項目辞書

プロジェクト間で共有するデータ項目を格納する辞書です。データ中心アプローチに基づいてデータ分析し、標準化したデータ項目を蓄積します。SEWB + ツールを使ったシステム開発で作成するドキュメント間で共通に利用することを目的とします。

データ中心アプローチ

データを企業全体の重要な財産であると考え、共有資源として扱うシステム開発の方法論です。企業情報システムの基盤整備のために必要な方法論といえます。データ中心アプローチを採用したシステム開発では、先にデータを共有資源として分析して標準化したあと、システムやプログラムを設計、開発します。

データ定義

SEWB+/CONSTRUCTION での AP 作成に使用するファイルや DB を設計します。ファイルや DB の設計には、辞書のレコード定義（最上位結合項目）や SEWB+/RECORD DEFINER で作成したレコード定義を利用します。

適用条件

アプリケーションに業務ルールを適用する条件をいいます。適用条件は、業務ルールに使用するデータ項目ごとに指定します。一般に、業務ルールの処理の入力となるデータ項目には「入力」を、業務ルールの処理結果を出力するデータ項目には「出力」を指定します。どちらにも役割を限定しない場合や、役割を明確にしたい場合には、指定しなくてもかまいません。

SEWB+/CONSTRUCTION のプログラム定義では、プログラムで使用するデータ項目を基にして、その処理に使用できる業務ルールの候補が表示されます。あらかじめ、SEWB+/CONSTRUCTION のテンプレート上に抽出条件を指定しておくことで、表示される業務ルールの候補を絞り込みます。プログラマは、データ項目の役割に適合した業務ルールから、目的の業務ルールを容易に選択できます。

テンプレート

SEWB+/CONSTRUCTION が提供する機能の一つです。テンプレートとは、ソースプログラムの生成時にアプリケーションの枠組みとなる処理構成を記述したものです。テンプレートは、テンプレート記述言語とアプリケーション開発に使用するプログラミング言語で記述します。

同一項目用業務ルール

同じデータ項目間の処理を記述するための業務ルールです。主に、転記処理の記述に使用します。この同一項目用業務ルールは、リポジトリ内のすべてのデータ項目に適用できます。同一項目用業務ルールでは、データ項目との関連は付けません。しかし、業務ルールを利用する SEWB+/CONSTRUCTION からは、アプリケーションに使用するデータ項目と同一項目用業務ルールの間に関連が付けられているとみなされ、通常の業務ルールと同様に扱えます。同一項目用業務ルールの作成時には、この仮想的な関連を、リポジトリ内の結合項目を含めたすべてのデータ項目に対して付けるのか、またはリポジトリ内の結合項目ではないデータ項目（単項目）だけに付けるのかを選択できます。

ドキュメント

リポジトリで管理されるシステム分析・設計情報およびプログラム開発情報を保管するファイルです。ドキュメントの例としては、SEWB+ ツールで作成したシステム分析図やデータ定義情報、ソースプログラム、または Microsoft Word や Microsoft Excel などで作成した企画文書、設計文書類があります。

ドメイン

データ中心アプローチの用語で、共通の意味と属性を持つデータ要素の集合を指します。SEWB+/REPOSITORY の辞書の仕組みを効果的にシステム開発に利用するには、ドメインを分析することが重要な作業になります。業務に使用するデータ項目を調査し、その中から共通のデータの値域（タイプ、意味、けた数、構成や範囲など）を持つデータをグルーピングします。データ項目辞書には、この共通のデータの値域を定義したデータ項目を、継承関係の上位に位置づけます。

(八行)

プロパティ

APPGALLERY Enterprise では、アプリケーション開発に使用する部品に設定された、動作や表示などに関する属性を指します。ボタンなどの GUI 作成で使用する部品の色、フォントやサイズなどは、プロパティに値として設定されています。これをプロパティ値といいます。

プロパティセット

APPGALLERY Enterprise での GUI (グラフィカルユーザインタフェース) 作成のために、部品ごとに標準とする色やフォントなどの設定値をまとめたものです。プロパティセットを SEWB+/REPOSITORY で一元管理し、アプリケーションの GUI 作成に利用すれば、GUI を効率良く作成でき、変更にも柔軟に対応できます。また、GUI の統一を図れます。

アプリケーション開発を始める前に、SEWB+/REPOSITORY で、GUI を統一する対象 (システム、業務、開発環境など) ごとに、プロパティセットを準備しておきます。プロパティセットは、環境構築ユティリティの GUI 属性で設定します。APPGALLERY Enterprise でのアプリケーション開発では、開発対象に応じたプロパティセットを選択します。GUI 作成で部品を配置した時点で自動的に、プロパティ値が設定されます。

APPGALLERY Enterprise から部品パレットの部品情報を SEWB+/REPOSITORY に転送し、プロパティセットとして登録することもできます。標準提供されている部品のほかに市販ソフトウェアを利用した部品やユーザが独自に作成した部品についてもプロパティセットを準備できるため、さらに、作成効率の向上と GUI の統一化を促進できます。

(マ行)

マッピング情報

APPGALLERY Enterprise と連携し、辞書のデータ項目から GUI の部品が生成されるときに使用される情報です。マッピング情報には、データ項目の定義情報をどのように部品のプロパティ値に置き換えるかが設定されています。

アプリケーション開発を始める前に、SEWB+/REPOSITORY で、マッピング情報を準備しておきます。マッピング情報は、環境構築ユティリティの GUI 属性で設定します。ボタン部品とテキストボックス部品に対応するマッピング情報が標準に設定されています。必要に応じて、マッピング情報を編集したり、ほかの部品に関するマッピング情報を追加したりすることができます。

命名ルール

データ項目の名称 (データ項目名、標準名称、フリガナおよび言語別の名前) および業務ルールの名称 (業務ルール名、標準名称およびフリガナ) の規則をいいます。プロジェクトの名称基準を命名ルールに定義して、検査機能でチェックできます。

(ヤ行)

ユーザ

SEWB+/REPOSITORY を使い、リポジトリのシステム開発資源を操作できる人です。システム管理者によってユーザ登録ファイルに登録されます。それぞれのユーザは、固有に与えられたユーザ ID とパスワードを持ちます。

(ラ行)

リポジトリ

システム開発の各工程で発生する情報のデータベースです。このシステム開発資源（ドキュメントと辞書）と、その資源間の関連をサーバで統合管理します。リポジトリによって、PCクライアントで動作する SEWB+ ツールやそのほかのアプリケーション間で、システム開発資源を共有したり連携したりできます。

リポジトリブラウザ

リポジトリのシステム開発資源を、クライアントでビジュアルに操作できる機能です。リポジトリに格納されたシステム開発資源がツリー形式で表示されます。リポジトリブラウザでは、リポジトリへ辞書やドキュメントを登録したり、目的の資源を検索したり、資源間の関連を手がかりにブラウジングしたりできます。

ルールスクリプト

業務ルールの処理を、プログラミング言語を使ってコーディングしたものです。この内容は、SEWB+/CONSTRUCTION のプログラム生成でソースプログラムに展開されます。汎用的に使える業務ルールを作成するために、ルールスクリプトには独自の文法が用意されています。ソースプログラムへの展開位置を指定する @@section 文や、データ項目の定義情報を取り出す @type などの予約キーワードが用意されています。

レコード生成キーワード

データ項目の定義情報を、どのようにソースコードとして生成するかを指定するものです。言語別のタイプごとに指定します。レコード生成キーワードの内容は、プログラミング言語で規定されたデータの定義と、データ項目の定義情報を置換するための可変記号で記述されます。辞書に定義したレコード構造は SEWB+/CONSTRUCTION のデータ定義で参照され、さらに、プログラム生成またはレコード生成機能でレコード生成キーワードに従ってソースプログラム中に展開されます。

レコード定義

レコードの構造を定義した最上位結合項目またはレコード定義ファイルのことです。レコード定義は、SEWB+/RECORD DEFINER を利用すれば、効率良く登録できます。レコード定義は、次のアプリケーション開発の場面で利用されます。

- SEWB+/CONSTRUCTION でのデータ定義
- EUR Professional Edition でのレポート設計
- APPGALLERY Enterprise での OpenTP1 クライアントアプリケーション

索引

記号

@@merge 文 82
@@rule 81
@@rule 文 78
@@rule 文で抽出される業務ルール 97
@@section 81
@len 87
@modify 93
@occurs 91
@occurs_dimension 91
@occurs1 ~ @occurs9 92
@type 87

A

APPGALLERY Enterprise 14, 129, 302

C

CORBA 10, 302
CSV 形式の記述規則 179
CSV 形式ファイル 154, 302
CSV 形式ファイル出力コマンド 175
CSV 形式ファイル入力コマンド 165
CSV 形式ファイルの出力 17, 18
CSV 形式ファイルの入力 17, 18
CSV 形式ファイルフォーマット 183
CSV 出力 17, 18, 154
CSV 入力 17, 18, 154

E

ERwin/ERX 20, 247
EUR Professional Edition 13, 302

G

GUI 属性 129, 302
GUI 属性を設定 129

I

IDL 10, 302

O

ODBC 入力コマンド 17, 208

R

RPCsvin 165
RPCsvout 175
RPOdbcin 208
RPTrans 223

S

SEWB+/CONSTRUCTION 12, 303
SEWB+/CS-DESIGN 10, 303
SEWB+/RECORD DEFINER 303
SEWB+/REPORT MANAGER 22, 117, 303
SEWB+/STANDARD-DICTIONARY 246
SEWB+/STANDARD-SUBROUTINE 246
SEWB+ 基本開発環境 303
SEWB+ 基本開発環境セット 303
SEWB+ ツール 303
SEWB3 辞書移行コマンド 17, 223
SEWB3 データ項目辞書の移行 212

あ

アクセス権 304
アクセス権の設定 136

い

一意性チェック範囲を設定する 131
インポート 139, 148, 304
インポート運用 140, 148

う

上書きインポート 148

え

英数字文字列データ 38
 エクスポート 139, 143, 304
 エクスポート・インポートの利用 139
 エクスポート・インポートユティリティ
 17, 139
 エクスポート運用 140, 143
 エラー ID からのメッセージ取得関数 287
 エラー ID 領域の解放関数 287
 エンティティレポート 248

か

カナ文字以外の使用 128
 仮システム内部 ID 158
 環境構築情報のエクスポート 147
 環境構築ユティリティ 17, 120
 環境設定 21, 120
 関連 117, 304

き

キーワード 71, 80, 304
 業務ルール 8, 67, 304
 業務ルール辞書 5, 8, 67, 304
 業務ルール辞書の標準フォーマット 188
 業務ルールに設定するアクセス権 137
 業務ルールに適用するデータ項目 69
 業務ルールのエクスポート 147
 業務ルールの検査関数 287
 業務ルールの検査項目 127
 業務ルールの設計 67
 業務ルールの定義項目 67
 業務ルール名 68
 業務ルール名の一意性チェック 127
 共用する 70

く

グループ 304

け

継承 7, 31, 51, 305
 継承関係 51

継承モード 53, 121
 けた数 43
 けた数とタイプの整合性チェック 126
 結合 7, 53
 結合関係 53
 結合関係の継承 55
 結合項目 7, 53, 305
 結合項目の定義項目 54
 言語区分を設定 120
 言語別フィールド 51
 検査 114
 検査 DLL 128, 287
 検査 DLL の作成 287
 検査 DLL の登録 287
 検査項目 114, 126, 127
 検査項目インタフェース 287
 検査項目を定義 126
 検査状態 115

こ

構成項目 7, 53, 305
 構成項目の名前の重複チェック 127
 国語区分を設定 120
 コメント 48, 69, 133

さ

最上位結合項目 54, 61, 228, 238
 最大長 (バイト数) 128

し

シーケンス 49
 時刻データ 38
 辞書 305
 辞書情報のセキュリティ 136
 辞書登録 22
 辞書登録の確認 114
 辞書と関連情報のインポート規則 149
 辞書のインポート 148
 辞書のエクスポート 143
 辞書の環境設定 21
 辞書の基本構成 5
 辞書の準備 21, 120

辞書の配布,複製 139
 辞書フォルダ 5,132,305
 辞書フォルダに設定するアクセス権 136
 辞書フォルダの定義項目 132
 辞書フォルダ名 132
 辞書フォルダを作成 132
 システム開発の概要 3
 システム管理者 305
 システム内部 ID 158,163
 実数データ 38
 条件 DLL 129,283
 条件 DLL の作成 283
 条件 DLL の登録 284
 条件情報取得関数 283
 小数部けた数 47
 初期値 51
 初期値の記述チェック 126

せ

整数データ 38
 正整数データ 38
 制約条件 30
 全体エクスポート 143
 専用にする 70

そ

挿入インポート 148
 増分エクスポート 143
 属性レポート 266

た

タイプ 48
 タイプ修飾情報 49
 タイプに対応するけた数 (COBOL 又は OOCOBOL) 44
 タイプに対応するけた数 (C 又は C++) 44
 タイプに対応するけた数 (IDL(CORBA)) 45
 タイプに対応するけた数 (Java) 47
 タイプに対応するけた数 (SQL) 45
 タイプに対応するけた数 (帳票定義) 46
 タイプの組み合わせチェック 126

タイプの組み合わせを変更 122
 タイプのけた数の範囲を変更 122
 タイプマッピング 198

ち

値域 30
 チェック関数 283
 チェック処理を記述する場合には 94

て

データ項目 7,35,305
 データ項目から別のデータ項目の値を導き出す処理 9
 データ項目辞書 5,305
 データ項目辞書の標準フォーマット 183
 データ項目種別 37
 データ項目特有の処理とは 9
 データ項目に設定するアクセス権 137
 データ項目の継承 7,51
 データ項目の継承モードの設定 121
 データ項目の結合 7,53
 データ項目の検査関数 287
 データ項目の設計 35
 データ項目の値の妥当性をチェックする処理 9
 データ項目の定義項目 35
 データ項目への定義情報の継承 52
 データ項目名 36
 データ項目名の一意性チェック 126
 データ中心アプローチ 2,26,305
 データ中心アプローチの基盤となる辞書 2
 データ定義 12,63,306
 データ分析 26
 テーブルレポート 273
 適用条件 71,78,95,306
 テンプレート 13,76,306

と

同一項目用業務ルール 8,72,134,298,306
 ドキュメント 306
 ドメイン 27,30,56,306
 ドメインの分析 30

取りうる値 50

な

名前 48

に

日本語文字列データ 38

は

排他制御 138

配布 139

反復回数 47

ひ

日付データ 38

標準データ項目辞書 246

標準フォーマット 183

標準名称 37,69

ふ

フィールド 47,69

フィールドの属性を設定 121

フィールドの名称を設定 121

複製 139

フリー定義 48

フリガナ 37,69

プリフィックス 128

プロパティ 129,307

プロパティセット 130,307

分類 38

分類区分 38

分類区分とタイプの対応 (COBOL 又は
OOCOBOL) 38

分類区分とタイプの対応 (C 又は C++) 39

分類区分とタイプの対応 (IDL(CORBA))

40

分類区分とタイプの対応 (Java) 42

分類区分とタイプの対応 (SQL) 40

分類区分とタイプの対応 (帳票定義) 41

分類区分とタイプの対応のカスタマイズ例
43

分類区分とタイプの対応を変更 122

分類なし 38

へ

ヘッダ 184,189

編集文字列 49

編集文字列の記述チェック 127

ま

マッピング情報 129,307

マルチ辞書 6,133

め

名称基準 33

名称と命名ルールのマッチングチェック
126,128

名称の標準化 33

命名ルール 115,307

命名ルールインタフェース 283

命名ルールの条件 128

命名ルールを定義 128

も

文字数の上限 (フィールド長) の設定 121

ゆ

ユーザ 307

ユーザ定義のタイプを追加 121

よ

予約キーワード 79,87

予約キーワードの留意事項 95

り

リポジトリ 2,308

リポジトリブラウザ 308

る

ルールスクリプト 75,308

ルールスクリプトのキーワード使用チェック
128
ルールスクリプトの記述 79

れ

レコード構造の設計 61
レコード生成キーワード 308
レコード生成キーワードとは 64
レコード生成キーワードを編集 123
レコード定義 61, 228, 238, 308

ソフトウェアマニュアルのサービス ご案内

ソフトウェアマニュアルについて、3種類のサービスをご案内します。ご活用ください。

1. マニュアル情報ホームページ

ソフトウェアマニュアルの情報をインターネットで公開しております。

URL <http://www.hitachi.co.jp/soft/manual/>

ホームページのメニューは次のとおりです。

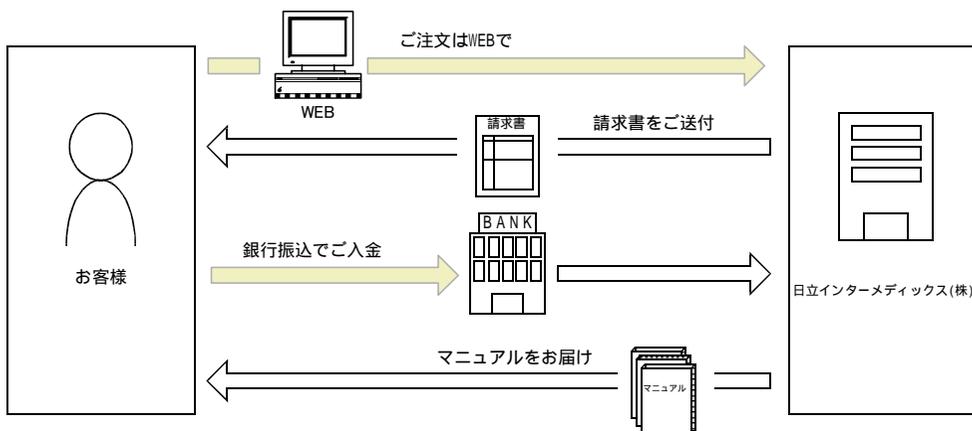
マニュアル一覧	日立コンピュータ製品マニュアルを製品カテゴリ、マニュアル名称、資料番号のいずれかから検索できます。
CD-ROMマニュアル情報	複数マニュアルを格納したCD-ROMマニュアルを提供しています。どの製品に対応したCD-ROMマニュアルがあるか、を参照できます。
マニュアルのご購入	日立インターメディックス(株)の「日立コンピュータ製品マニュアルサイト」からお申し込みできます。 (詳細は「3. マニュアルのご注文」を参照してください。)
Web提供マニュアル一覧	インターネットで参照できるマニュアルの一覧を提供しています。 (詳細は「2. インターネットからのマニュアル参照」を参照してください。)
ご意見・お問い合わせ	マニュアルに関するご意見、ご要望をお寄せください。

2. インターネットからのマニュアル参照(ソフトウェアサポートサービス)

ソフトウェアサポートサービスの契約をしていただくと、インターネットでマニュアルを参照できます。本サービスの対象となる契約の種別、及び参照できるマニュアルは、マニュアル情報ホームページでご確認ください。なお、ソフトウェアサポートサービスは、マニュアル参照だけでなく、対象製品に対するご質問への回答、問題解決支援、バージョン更新版の提供など、お客様のシステムの安定的な稼働のためのサービスをご提供しています。まだご契約いただいていない場合は、ぜひご契約いただくことをお勧めします。

3. マニュアルのご注文

日立インターメディックス(株)の「日立コンピュータ製品マニュアルサイト」からご注文ください。



下記 URL にアクセスして必要事項を入力してください。

URL http://www2.himdx.net/manual/privacy.asp?purchase_flag=1

ご注文いただいたマニュアルについて、請求書をお送りします。

請求書の金額を指定銀行へ振り込んでください。なお、送料は弊社で負担します。

入金確認後、7日以内にお届けします。在庫切れの場合は、納期を別途ご案内いたします。