

---

画面・帳票サポートシステム

# XMAP3 プログラミングガイド 画面編

解説・手引・操作書

3020-7-594-A0

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-262B-5344 XMAP3/NET Version 4 04-10 (適用 OS : Windows 95 , Windows 98 , Windows Me , Windows NT 4.0 , Windows 2000 , Windows XP , Windows Server 2003 , Windows Server 2003 x64 )

P-262B-534U XMAP3/NET Version 4 Upgrade 04-10 (適用 OS : Windows 95 , Windows 98 , Windows Me , Windows NT 4.0 , Windows 2000 , Windows XP , Windows Server 2003 , Windows Server 2003 x64 )

P-262B-5C44 XMAP3/Enterprise Edition Version 4 04-10 (適用 OS : Windows 95 , Windows 98 , Windows Me , Windows NT 4.0 , Windows 2000 , Windows XP , Windows Server 2003 , Windows Server 2003 x64 )

P-262B-5C4U XMAP3/Enterprise Edition Version 4 Upgrade 04-10 (適用 OS : Windows 95 , Windows 98 , Windows Me , Windows NT 4.0 , Windows 2000 , Windows XP , Windows Server 2003 , Windows Server 2003 x64 )

これらの製品は、ISO9001 および TickIT の認証を受けた品質マネジメントシステムで開発されました。

## 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

AIX は、米国における米国 International Business Machines Corp. の登録商標です。

AX は、AX 協会のシステム名称です。

BackOffice は、米国 Microsoft Corp. の登録商標です。

Delphi は、米国 Borland International,Inc. の登録商標です。

HP-UX は、米国 Hewlett-Packard Company のオペレーティングシステムの名称です。

Itanium は、アメリカ合衆国および他の国におけるインテル コーポレーションまたはその子会社の登録商標です。

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。

JFIF は、C-Cube Microsystems 社が規定したフォーマットの名称です。

Linux は、Linus Torvalds の米国およびその他の国における登録商標あるいは商標です。

Micro Focus COBOL は、Micro Focus International Limited の商標です。

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 Access 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。

MS-DOS 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。

OLE は、米国 Microsoft Corp. が開発したソフトウェア名称です。

PA-RISC は、米国 Hewlett-Packard Company の商標です。

PC-9800 は、日本電気 (株) の商品名称です。

Red Hat は、米国およびその他の国で Red Hat,Inc. の登録商標若しくは商標です。

Solaris 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。

TurboLinux の名称及びロゴは、TurboLinux,Inc. の商標です。

UNIX は、X/Open Company Limited が独占的にライセンスしている米国ならびに他の国における登録商標です。

Visual Basic 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。

Visual C++ 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 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 NT 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。

## 発行

2001 年 4 月 (第 1 版) 3020-7-594

2006 年 9 月 (第 11 版) 3020-7-594-A0

## 著作権

All Rights Reserved. Copyright (C) 2001, 2006, Hitachi, Ltd.

## 変更内容

### 変更内容 ( 3020-7-594-A0 ) XMAP3 Version 4 04-10

追加・変更内容	変更箇所
Windows Server 2003 R2 , Windows Server 2003 R2 x64 に対応した。	-
フィールドボックス内の複数行フィールドで、改行コードを認識する機能を追加した。	2.4.17 , 4.1.2 , 4.2.1 , 6.2.16 , 付録 A.2

単なる誤字・脱字などはお断りなく訂正しました。

### 変更内容 ( 3020-7-594-90 ) XMAP3 Version 4 04-09

追加・変更内容
使用目的が「日本語」のデータ型に「漢字 ( XX )」を追加した。

### 変更内容 ( 3020-7-594-80 ) XMAP3 Version 4 04-08

追加・変更機能
表示・印刷セットアップのコピー & ペーストモードが単体フィールドの場合に、タブまたは改行コードをペーストするかどうか設定時に選択できるようにした。
Windows Server 2003 x64 に対応した。

### 変更内容 ( 3020-7-594-70 ) XMAP3 Version 4 04-07

追加・変更内容
日立コード変換に対応した。

# 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機能および使用方法について説明したものです。

P-262B-5344 XMAP3/NET Version 4

P-262B-534U XMAP3/NET Version 4 Upgrade

P-262B-5C44 XMAP3/Enterprise Edition Version 4

P-262B-5C4U XMAP3/Enterprise Edition Version 4 Upgrade

このマニュアルでは、上記プログラムプロダクトを XMAP3 と表記しています。

## 対象読者

XMAP3 を使用して画面を設計し、AP を開発する方を対象としています。また、次のことを前提としています。

- Windows の基本操作を習得していること。
- COBOL, C 言語, または Java の知識を持っていること。

## 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 画面機能の概要

画面機能の概要, XMAP3 で作成できる画面について説明しています。

### 第 2 章 GUI 画面の作成

GUI 画面の設計, および GUI 画面のレイアウト定義について説明しています。

### 第 3 章 CUI 画面の作成

CUI 画面の設計, および CUI 画面のレイアウト定義について説明しています。

### 第 4 章 マウスとキー操作

操作キーの機能, およびマウスの操作について説明しています。

### 第 5 章 論理マップ生成規則とマッピング規則

画面のレイアウトを保存したときに生成される論理マップと, マッピング規則の関係について説明しています。

### 第 6 章 画面定義とコーディングの関連

AP と画面定義および画面の各オブジェクトの関連について説明しています。

### 第 7 章 AP のコーディング方法

XMAP3 実行支援の AP インタフェースについて説明しています。

### 第 8 章 標準パターンを使用した AP の作成

XMAP3 が用意している画面の定義サンプルと AP のパターンを組み合わせることで, COBOL ソースブ

はじめに

ログラムを作成する方法について説明しています。

## 第9章 チューニングとトラブルの対処方法

実行時の性能向上のポイントや、トラブル発生時の対処方法について説明しています。

付録A 属性情報一覧

付録B XMAP3 で提供している定義パターン

付録C 日立以外のコンパイラを使用するときの注意

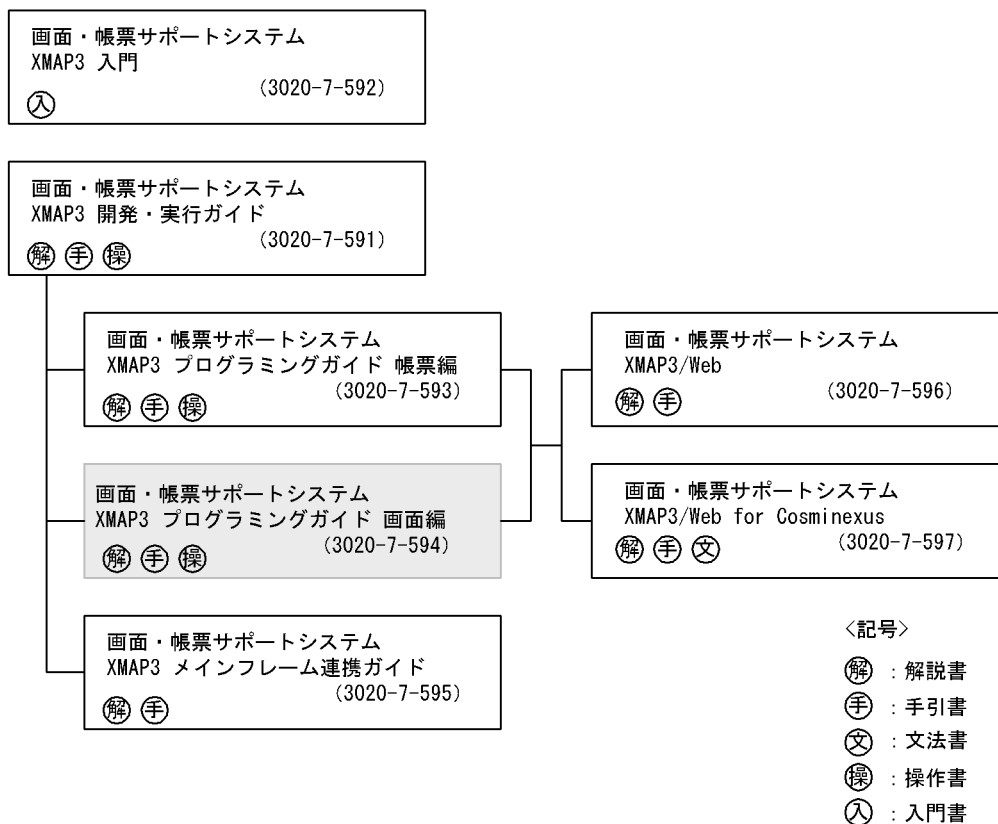
付録D 画面定義の制限値

付録E XMAP3 のリターンコード・詳細コード

付録F 用語解説

## マニュアル体系図

XMAP3 のマニュアル体系を次に示します。



## 関連マニュアル

このマニュアルに関連するマニュアルを次に示します。

画面・帳票サポートシステム XMAP3 開発・実行ガイド (3020-7-591)

画面・帳票サポートシステム XMAP3 入門 (3020-7-592)

画面・帳票サポートシステム XMAP3 プログラミングガイド 帳票編 (3020-7-593)

画面・帳票サポートシステム XMAP3 メインフレーム連携ガイド (3020-7-595)

画面・帳票サポートシステム XMAP3/Web (3020-7-596)

画面・帳票サポートシステム XMAP3/Web for Cosminexus (3020-7-597)

画面・帳票サポートシステム XMAP3 Server (3000-7-508)

COBOL85 言語 (3020-3-782)

COBOL85 操作ガイド (3020-3-851) <sup>1</sup>

COBOL85 ユーザーズガイド (3020-3-852)

COBOL85 操作ガイド (3020-3-873)

COBOL2002 操作ガイド (3020-3-D41)

COBOL2002 ユーザーズガイド (3020-3-D42)

COBOL2002 言語 標準仕様編 (3020-3-D44)

COBOL2002 言語 拡張仕様編 (3020-3-D45)

OpenTP1 Version 5 分散トランザクション処理機能 OpenTP1 プロトコル TP1/NET/XMAP3 編 (3000-3-375) <sup>2</sup>

OpenTP1 Version 6 分散トランザクション処理機能 OpenTP1 プロトコル TP1/NET/XMAP3 編 (3000-3-965) <sup>2</sup>

注 1

COBOL85 Version 5.0 の場合にお読みください。

注 2

このマニュアルの本文では、OpenTP1 関連マニュアルのタイトルにある「バージョン表記」を省略しています。省略した表記を次に示します。

マニュアル名	このマニュアルでの表記
「OpenTP1 Version 5 分散トランザクション処理機能 OpenTP1 プロトコル TP1/NET/XMAP3 編」 「OpenTP1 Version 6 分散トランザクション処理機能 OpenTP1 プロトコル TP1/NET/XMAP3 編」	「OpenTP1 プロトコル TP1/NET/XMAP3 編」

## このマニュアルでの表記

このマニュアルでは各製品を次のように表記しています。

製品名	略称
AIX 5L V5.1	AIX
AIX 5L V5.2	
AIX 5L V5.3	

はじめに

製品名	略称	
COBOL2002 Net Client Suite	COBOL2002	日立 COBOL
COBOL2002 Net Developer		
COBOL2002 Net Server Suite	COBOL85	
COBOL85 Server Version 6		
COBOL85 Server Version 7		
COBOL85 Version 5.0		
COBOL85 Version 6		
COBOL85 Version 7		
COBOL85 Version 7 for Server OS - Client Use		
HP-UX 10.20	HP-UX	
HP-UX 11		
HP-UX 11i		
HP-UX 11i V2 ( IPF )		
HP-UX 11i V2 ( PA-RISC )		
Java <sup>(TM)</sup>	Java	
Microsoft Access(R)	Access	
Microsoft(R) C	C	
Microsoft(R) Excel	Excel	
Microsoft(R) Visual Basic(R)	Visual Basic ( または VB )	
Microsoft(R) Visual C++(R)	Visual C++	
Microsoft(R) Visual C++(R) .NET 2002	Visual C++ .NET	
Microsoft(R) Visual C++(R) .NET 2003		
Microsoft(R) Windows(R) 95 Operating System	Windows 95	
Microsoft(R) Windows(R) 98 Operating System	Windows 98	
Microsoft(R) Windows(R) Millennium Edition Operating System	Windows Me	
Microsoft(R) Windows NT(R) Server Network Operating System Version 4.0	Windows NT Server 4.0	Windows NT ( または Windows NT 4.0 )
Microsoft(R) Windows NT(R) Server Network Operating System Version 4.0 Enterprise Edition		
Microsoft(R) BackOffice(R) Small Business Server Version 4.0		
Microsoft(R) Windows NT(R) Workstation Operating System Version 4.0	Windows NT Workstation 4.0	
Microsoft(R) Windows(R) 2000 Server Operating System	Windows 2000 Server	Windows 2000



製品名	略称	
Microsoft(R) Windows(R) 2000 Advanced Server Operating System	Windows 2000 Advanced Server	
Microsoft(R) Windows(R) 2000 Datacenter Server Operating System	Windows 2000 Datacenter Server	
Microsoft(R) Windows(R) 2000 Professional Operating System	Windows 2000 Professional	
Microsoft(R) Windows(R) XP Professional Operating System	Windows XP Professional	Windows XP
Microsoft(R) Windows(R) XP Home Edition Operating System	Windows XP Home Edition	
Microsoft(R) Windows Server(TM) 2003, Standard Edition 日本語版	Windows Server 2003	
Microsoft(R) Windows Server(TM) 2003, Enterprise Edition 日本語版		
Microsoft(R) Windows Server(TM) 2003 R2, Standard Edition 日本語版		
Microsoft(R) Windows Server(TM) 2003 R2, Enterprise Edition 日本語版		
Microsoft(R) Windows Server(TM) 2003, Standard x64 Edition 日本語版	Windows Server 2003 x64	
Microsoft(R) Windows Server(TM) 2003, Enterprise x64 Edition 日本語版		
Microsoft(R) Windows Server(TM) 2003 R2, Standard x64 Edition 日本語版		
Microsoft(R) Windows Server(TM) 2003 R2, Enterprise x64 Edition 日本語版		
MS-DOS(R) Version 5.0/V 以降	MS-DOS	
Solaris 7	Solaris	
Solaris 8		
Solaris 9		
TP1/Message Control	TP1/MCF	
TP1/NET/Library		
TP1/NET/XMAP3		
TurboLinux Server 日本語版 6.1	Linux	
Red Hat Linux 6.2J		
XMAP3/Enterprise Edition Version 4	XMAP3/Enterprise Edition	
XMAP3/NET Version 4	XMAP3/NET	

- Microsoft(R) Windows Server(TM) 2003 R2, Standard Edition 日本語版 , Microsoft(R) Windows Server(TM) 2003 R2, Enterprise Edition 日本語版を Windows Server 2003 R2 と表

記します。Microsoft(R) Windows Server(TM) 2003 R2, Standard x64 Edition 日本語版 , Microsoft(R) Windows Server(TM) 2003 R2, Enterprise x64 Edition 日本語版を , Windows Server 2003 R2 x64 と表記します。

- Windows 95 , Windows 98 , Windows Me , Windows NT , Windows 2000 , Windows XP , Windows Server 2003 , および Windows Server 2003 x64 に機能差がない場合 , Windows と表記します。
- AIX , HP-UX , Solaris , および Linux で機能差がない場合 , WS と表記します。
- COBOL85 Server Version 6 および COBOL85 Version 6 を合わせて , COBOL85 Version 6 と表記します。
- COBOL85 Server Version 7 , COBOL85 Version 7 , および COBOL85 Version 7 for Server OS - Client Use を合わせて , COBOL85 Version 7 と表記します。

## このマニュアルで使用する略語

このマニュアルでは、次に示す用語を略して表現しています。

略称	正式名称
ANK	Alphabet Numeric Kana
AP	Application Program
API	Application Programming Interface
C/S システム	Client / Server System
CSV	Comma Separated Value
CUI	Character User Interface
DCT	Digital Capture Technology
DLL	Dynamic Linking Library
DNS	Domain Name System
EUC	Extended UNIX Code
GUI	Graphical User Interface
IPF	Itanium(R) Processor Family
JFIF	JPEG File Interchange Format
JIS	Japanese Industrial Standard
JPEG	Joint Photographic Experts Group
LAN	Local Area Network
MCR	Magnetic Character Reader
OCR	Optical Character Reader
OLE	Object Linking and Embedding
OLTP	Online Transaction Processing
OS	Operating System

略称	正式名称
PC	Personal Computer
RAM	Random Access Memory
RPC	Remote Procedure Call
TCP/IP	Transmission Control Protocol/Internet Protocol
WS	Workstation

## このマニュアルで使用する記号

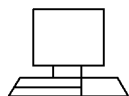
このマニュアルで使用する記号を次のように定義しています。

記号	意味
[ ]	メニュータイトル、メニュー項目、ボタン、キー、およびアイコンの名称を示す。 例：[ファイル]メニュー [OK] ボタン [Enter] キー など
[ ]+[ ]	+の前のキーを押したまま、後ろのキーを押すことを示す。 例：[Shift]+[A] キー [Shift] キーを押したまま [A] キーを押す。
[ ]-[ ]	例：[ファイル]-[開く] [ファイル]メニューから [開く] コマンドを選択することを示す。
	半角の空白を入れること、またはスペースキーを指定された数だけ押すことを示す。
[ ]	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。

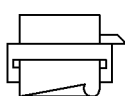
●WS、PC



●入出力の動作



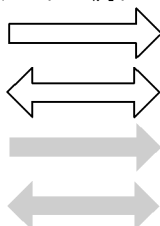
●プリンタ



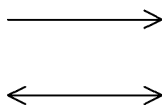
●ファイル



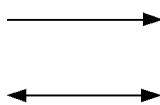
●データの流れ



●制御の流れ



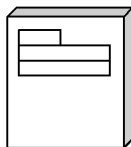
●その他の流れ



●プログラム



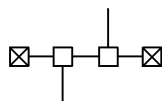
●論理マップを  
取り込んだAP



●画面の表示



●ネットワーク (LAN)



## オンラインヘルプの紹介

XMAP3 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。ヘルプでは、詳細な操作手順、トラブルの対処方法、用語、ダイアログやウィンドウの項目の説明などを参照できます。

## 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、次の用語については、常用漢字以外の漢字を使用しています。

個所（かしよ） 矩形（くけい） 桁（けた） 全て（すべて） 揃える（そろえる）  
貼り付け（はりつけ） 汎用（はんよう） 必須（ひつす） 頁（ページ）

## 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、1,024<sup>2</sup> バイト、1,024<sup>3</sup> バイト、1,024<sup>4</sup> バイトです。

# 目次

<b>1</b>	<b>画面機能の概要</b>	<b>1</b>
1.1	XMAP3 で作成できる画面	2
1.2	XMAP3 の画面設計の概要	3
1.3	画面設計の流れ	4
1.3.1	作成する画面の種類を決定する	4
1.3.2	画面を設計する	4
1.3.3	画面のレイアウトを定義する	4
1.3.4	AP をコーディングする	5
1.4	画面で使用できる機能 ( GUI 画面 , CUI 画面 )	6
1.4.1	GUI 画面で使用できる機能	6
1.4.2	CUI 画面で使用できる機能	10
1.5	XMAP3 画面制御方式の解説	12
1.5.1	画面の制御方式	12
1.5.2	画面の描画方式	13
1.5.3	画面の表示モード	13
1.5.4	論理マップ	17
1.5.5	データ有無コード	17
1.5.6	表示属性の動的変更	18
1.5.7	オブジェクトの転送モード	19
1.5.8	入力単位	20
1.5.9	画面確定と AP への通知	25
1.5.10	表示先と表示環境の設定	28
1.6	XMAP3 関連ファイルの関係	29
1.7	画面を設計する前に	31
1.7.1	画面設計で使用する基本用語の説明	31
1.7.2	オブジェクト配置位置の基準	31
1.7.3	画面設計と定義	34
<b>2</b>	<b>GUI 画面の作成</b>	<b>37</b>
2.1	GUI 画面の設計	38
2.1.1	GUI 画面の構成	38
2.1.2	画面サイズ	43
2.1.3	画面の表示様式	46

2.1.4	レイアウト領域の背景色	49
2.1.5	一次ウィンドウと二次ウィンドウ	49
2.1.6	GUI 画面設計でこんなこともできます	51
2.1.7	GUI 画面設計のポイント	53
2.2	GUI 画面のレイアウト手順	56
2.2.1	マップ名の決定	57
2.2.2	定義パターンおよび部品の登録	58
2.3	GUI 画面属性の設定	60
2.3.1	画面属性の詳細	60
2.3.2	画面属性の動的変更	69
2.4	GUI 画面を構成するオブジェクト	70
2.4.1	タイトルバー	71
2.4.2	メニューバー	72
2.4.3	二次ウィンドウ	74
2.4.4	入出力テキストボックス	75
2.4.5	日付テキスト・フィールドと時刻テキスト・フィールド（入出力）	80
2.4.6	固定テキストボックス	83
2.4.7	出力テキストボックス	84
2.4.8	日付テキスト・フィールドと時刻テキスト・フィールド（出力）	85
2.4.9	コンボボックス	87
2.4.10	ポップアップテキスト	91
2.4.11	スピンボックス	100
2.4.12	リストボックス	102
2.4.13	ラジオボタンボックス	104
2.4.14	チェックボタンボックス	105
2.4.15	トグルフィールド	106
2.4.16	グループボックス	107
2.4.17	フィールドボックス	108
2.4.18	固定グラフィック	117
2.4.19	出力グラフィック	118
2.4.20	セバレータ	121
2.4.21	けい線	122
2.4.22	メッセージアイコン	124
2.4.23	塗りつぶしフィールド	125
2.4.24	プッシュボタンボックス（コマンドボタン）	125
2.4.25	MCR	129
2.4.26	フォーカスとカーソル	133

2.4.27	予約テキスト・フィールド	139
2.4.28	次画面	140
2.5	部品の利用	142
2.6	データ名の編集	143
2.6.1	データ名編集ダイアログの機能	143
2.6.2	データ名編集ダイアログでの操作と注意事項	144
2.7	テスト表示	147

# 3

CUI	画面の作成	149
3.1	CUI 画面の設計	150
3.1.1	CUI 画面の構成	150
3.1.2	画面サイズ	151
3.1.3	背景色および画面の表示フォント	151
3.2	CUI 画面のレイアウト手順	152
3.2.1	マップ名の決定	153
3.2.2	定義パターンおよび部品の登録	153
3.3	CUI 画面属性の設定	155
3.3.1	画面属性の詳細	155
3.3.2	画面属性の動的変更	160
3.4	CUI 画面を構成するオブジェクト	161
3.4.1	タイトルバー	161
3.4.2	AP への通知	162
3.4.3	入出力フィールド	162
3.4.4	固定フィールド	164
3.4.5	出力フィールド	165
3.4.6	フレーム（横／縦繰り返し）	167
3.4.7	けい線	169
3.4.8	カーソル	170
3.4.9	予約フィールド	171
3.4.10	次画面	172
3.5	部品の利用	174
3.6	データ名に関するコメントの編集	175
3.7	テスト表示	176

<b>4</b>	<b>マウスとキー操作</b>	<b>177</b>
4.1	有効となるキー	178
4.1.1	データ入力キー	178
4.1.2	制御キー	180
4.1.3	キーボードのカスタマイズ	185
4.2	各制御キーの機能	187
4.2.1	キーの機能	187
4.2.2	キーでのデータ入力と編集	203
4.3	マウス操作	208
4.3.1	有効となるボタン	208
4.3.2	シングルクリックとダブルクリック	209
4.3.3	マウスでのクリック	209
4.3.4	ホイールマウスでの操作	211
4.4	コピー＆ペースト操作	214
4.4.1	コピーの対象	214
4.4.2	ペーストの対象	216
4.4.3	クリップボードを使用したコピー＆ペースト操作	221
<b>5</b>	<b>論理マップ生成規則とマッピング規則</b>	<b>223</b>
5.1	論理マップとは	224
5.1.1	論理マップの構成	224
5.1.2	論理マップ生成規則で使用する用語	225
5.1.3	生成される標準のデータ名	226
5.1.4	ターゲットでの論理マップの違い	231
5.1.5	マッピングオプション	231
5.1.6	使用目的／詳細目的とデータ型	233
5.1.7	ドロースetAddressとの関係	236
5.1.8	下位項目を指定したときの扱い	236
5.1.9	フレームを使用したときの扱い	237
5.2	固定部の論理マップ生成規則とマッピング規則	238
5.2.1	固定部の出力論理マップ生成規則とマッピング規則	238
5.2.2	固定部の入力論理マップ生成規則とマッピング規則	241
5.3	可変部の論理マップ生成規則とマッピング規則	246
5.3.1	ウィンドウ表示	246
5.3.2	出力カーソル制御	247



5.3.3	入力カーソル制御	249
5.3.4	イベント通知コード	251
5.3.5	可変項目の論理マップ生成規則とマッピング規則	252
5.3.6	入力数字編集項目の論理マップ生成規則とマッピング規則	258
5.3.7	制御項目の論理マップ生成規則とマッピング規則	259
5.3.8	入力バイト数格納項目の論理マップ生成規則とマッピング規則	260
5.3.9	埋字と桁寄せに関するマッピング規則（出力フィールド）	262
5.3.10	フィールドキーを押したときのマッピング規則	262
5.3.11	データキーを押したときのマッピング規則	263
5.3.12	入力操作がなかったときのマッピング規則	264
5.3.13	コードエラーを検出したときのマッピング規則	264
5.3.14	埋字と桁寄せに関するマッピング規則（入力フィールド）	265
5.3.15	MCR 入力のマッピング規則	265
5.3.16	隠しフィールドの論理マップ生成規則とマッピング規則	266
5.3.17	確定キーの制御情報	267
5.3.18	下位項目の論理マップ生成規則とマッピング規則	268
5.4	可変部の論理マップ生成規則とマッピング規則（GUI 画面固有）	270
5.4.1	ポップアップメニューの論理マップ生成規則とマッピング規則	270
5.4.2	メニューバーの論理マップ生成規則	281
5.4.3	プッシュボタンボックスの論理マップ生成規則とマッピング規則	282
5.4.4	ラジオボタンボックスの論理マップ生成規則とマッピング規則	285
5.4.5	チェックボタンボックスの論理マップ生成規則とマッピング規則	289
5.4.6	リストボックスの論理マップ生成規則とマッピング規則	292
5.4.7	コンボボックスの論理マップ生成規則とマッピング規則	296
5.4.8	フォーカス・カーソル位置の論理マップ生成規則とマッピング規則	307
5.4.9	二次ウィンドウの表示に関するマッピング規則	314
5.4.10	出力グラフィックの論理マップ生成規則とマッピング規則	318
5.4.11	日付・時刻項目の論理マップ生成規則とマッピング規則	319
5.4.12	トグルフィールドの論理マップ生成規則とマッピング規則	322
5.4.13	スピンボックスの論理マップ生成規則とマッピング規則	324
5.4.14	数値・金額項目の論理マップ生成規則とマッピング規則	328
5.4.15	ウィンドウ表示位置の論理マップ生成規則とマッピング規則	329
5.5	定数部の論理マップ生成規則とマッピング規則	331
5.5.1	定数部の論理マップ生成規則	331
5.5.2	イベント定数の論理マップ生成規則	333
5.5.3	フォーカス定数の論理マップ生成規則	334
5.5.4	動的変更テーブル	336

5.6	予約テキスト・フィールドのマッピング規則	337
-----	----------------------	-----

# 6

画面定義とコーディングの関連	339
----------------	-----

6.1	画面属性と AP	340
6.1.1	ウィンドウの表示形態を変更する	340
6.1.2	ウィンドウ位置を変更する	341
6.1.3	PF キーなど確定キーの扱いを変更する	342
6.1.4	次画面を設定する	344
6.2	GUI 画面の各オブジェクトと AP	346
6.2.1	出力テキストの表示	346
6.2.2	出力数字編集項目の利用 (COBOL だけ有効)	348
6.2.3	出力テキストでの埋字と桁寄せ向きの利用	350
6.2.4	下位項目を利用した日付テキスト・フィールドと時刻テキスト・フィールド (出力)	352
6.2.5	入出力テキストを使用したデータの送受信	355
6.2.6	入出力テキストでの埋字と桁寄せ向きの利用	358
6.2.7	特定の入出力テキストだけを書き換える (データ有無コードの利用)	361
6.2.8	初期値の利用	362
6.2.9	入力できるデータ型の設定 (詳細目的の利用)	364
6.2.10	下位項目を利用した入出力テキスト	366
6.2.11	下位項目を利用した日付テキスト・フィールドと時刻テキスト・フィールド (入出力)	368
6.2.12	テキストの文字色の動的変更	372
6.2.13	パスワードの利用	375
6.2.14	MCR 入力専用のテキストの利用	377
6.2.15	フィールドボックスの入出力	380
6.2.16	複数行フィールドからのデータの抽出	380
6.2.17	可変ポップアップテキストを使用したデータの入力	384
6.2.18	ポップアップメニューファイルを使用したデータの入力	398
6.2.19	固定ポップアップテキストを使用したデータの入力	406
6.2.20	トグルフィールド	418
6.2.21	入出力・出力フィールドの一括反復 (フレームの使用)	420
6.2.22	メニュー選択コンボボックス (可変) を使用した候補の選択	423
6.2.23	メニュー選択コンボボックス (固定) を使用した候補の選択	427
6.2.24	キー入力コンボボックス (可変) を使用したデータの入力	430
6.2.25	キー入力コンボボックス (固定) を使用したデータの入力	434
6.2.26	可変ラジオボタンを使用した候補の選択	437

6.2.27	固定ラジオボタンを使用した候補の選択	440
6.2.28	可変チェックボタンを使用した候補の選択	443
6.2.29	固定チェックボタンを使用した候補の選択	446
6.2.30	チェックボタンの選択状態の動的変更	449
6.2.31	リストボックスを使用した項目の選択	452
6.2.32	コンボボックスの項目数の動的変更	455
6.2.33	プッシュボタンを使用した処理の選択	459
6.2.34	プッシュボタンの表示の動的変更	467
6.2.35	メニューバーを使用した処理の選択	474
6.2.36	出力グラフィックの利用	477
6.2.37	GUI 画面でのファンクションキーの利用	481
6.2.38	二次ウィンドウの表示	482
6.2.39	初期フォーカスの動的変更	485
6.2.40	隠しフィールドの設定	487

# 7

AP のコーディング方法	491
7.1 XMAP3 実行支援の AP インタフェースについて	492
7.1.1 XMAP3 での画面の入出力の基本	492
7.1.2 AP 間でオープンを引き継ぐ場合	494
7.1.3 AP 分割時の注意	495
7.1.4 AP での入力論理マップ上のデータチェック	498
7.1.5 AP 作成時の注意事項	500
7.2 COBOL の画面入出力命令	502
7.2.1 仮想端末の自動割当て	502
7.2.2 論理マップの取り込み方法 (COBOL)	502
7.2.3 画面表示命令 (COBOL)	503
7.2.4 CALL 文による方法	508
7.2.5 COBOL 開発マネージャを使用したコンパイルと実行のポイント	516
7.2.6 COBOL 開発マネージャでの XMAP3 の利用方法	520
7.3 C 言語の画面入出力命令	524
7.3.1 仮想端末の自動割当て	524
7.3.2 論理マップの取り込み方法 (C 言語)	525
7.3.3 画面表示命令 (C 言語)	525
7.3.4 C 言語固有のコーディング例	533
7.4 汎用 AP インタフェース	536
7.4.1 関数の機能概要	536

7.4.2	C 言語での表示方法	543
7.4.3	Visual Basic および Access での表示方法	543
7.4.4	Delphi での表示方法	544
7.4.5	文字編集 DLL	545
7.5	OLTP サーバ構成での画面表示	550
7.5.1	COBOL での画面表示命令	550
7.5.2	C 言語での画面表示命令	554
7.5.3	AP の実行	556

## 8

## 標準パターンを使用した AP の作成 557

8.1	標準パターン	558
8.1.1	レイアウトパターンとアプリケーションのパターン	558
8.1.2	COBOL エディタからの AP パターンの利用	560
8.1.3	C 言語用のサンプルと AP パターン	562
8.2	標準パターンを使用した機能別の作成例 (COBOL)	564
8.2.1	BTMENU01 パターンを使用したメニュー画面の表示	564
8.2.2	BTMENU01 の AP 実行前に表示するスプラッシュ画面の追加	566
8.2.3	GENDSP01 パターンを使用したメニュー画面からの次画面表示	569
8.2.4	GENDSP01 パターンを使用した CUI 画面でのフィールド単位のデータ入力	572
8.2.5	GENFLD01 パターンを使用した GUI 画面でのフィールド単位のデータ入力	574
8.2.6	GENEVN01 パターンを使用した GUI 画面でのイベント単位のデータ入力	577
8.2.7	GENDSP01 パターンを使用した行単位のデータ入力	581
8.2.8	GENDSP01 パターンを使用した入力フィールドのクリア	584
8.2.9	GENDSP01 パターンを使用した AP からのフィールド属性の変更	586
8.2.10	GENDSP01 パターンを使用した表形式の一覧表示	587
8.2.11	GENDSP01 パターンを使用した出力フィールドの再表示	589
8.2.12	GENDSP01 パターンを使用したヘルプ画面の表示	592
8.2.13	GENDSP01 パターンを使用したダイアログの表示	594
8.2.14	GENDSP01 パターンを使用した掲示板の表示	597
8.2.15	DSPPRT01 パターンを使用した画面表示と帳票印刷	600
8.2.16	MCFDP01 パターンを使用した画面表示と帳票印刷	602
8.3	各種例題プログラム	605
8.3.1	AP サンプルのコンパイルと実行手順	605
8.3.2	COBOL による GUI 画面を使用した画面単位入力例題	610
8.3.3	COBOL による GUI 画面を使用したフィールド単位入力例題	615
8.3.4	COBOL による GUI 画面を使用したイベント単位入力例題	622

8.3.5	COBOL による GUI 画面を使用した行単位入力例題	629
8.3.6	COBOL による CUI 画面を使用したフィールド単位入力例題	636
8.3.7	C 言語による GUI 画面を使用した GUI 入門例題	641
8.3.8	COBOL による OLTP クライアント AP 例 (OpenTP1 を使用した受注業務)	651
8.3.9	COBOL によるトランザクション型 AP 例 (OpenTP1 を使用した受発注業務)	670

## 9

チューニングとトラブルの対処方法	681
------------------	-----

9.1	性能向上のポイント	682
9.2	トラブルの対処方法	687
9.2.1	日立 COBOL 使用時のトラブル	687
9.2.2	旧バージョン提供ファイルのコンパイル時のトラブル	689
9.2.3	表示に関する対処方法	690
9.2.4	応答性能の対処方法	692

## 付録

付録		695
付録 A	属性情報一覧	696
付録 A.1	ウィンドウ制御情報一覧	696
付録 A.2	キーエントリ・ラベル系オブジェクトの表示属性一覧 (テキストとフィールド)	700
付録 A.3	キーエントリ・ラベル系オブジェクトの表示属性一覧 (日付テキスト・フィールドと時刻テキスト・フィールド)	702
付録 A.4	選択エントリ系オブジェクトの表示属性一覧	703
付録 A.5	候補選択コントロール系オブジェクトの表示属性一覧	705
付録 A.6	グループコントロール系オブジェクトの表示属性一覧	707
付録 A.7	グラフィックコントロール系オブジェクトの表示属性一覧	708
付録 A.8	コマンドコントロール系オブジェクトの表示属性一覧	708
付録 A.9	表示属性の解説	710
付録 A.10	修飾名と変更属性の標準値	733
付録 B	XMAP3 で提供している定義パターン	739
付録 B.1	GUI 画面の定義パターン	739
付録 B.2	CUI 画面の定義パターン	741
付録 B.3	XMAP3 で提供しているプログラムのパターン	741
付録 C	日立以外のコンパイラを使用するときの注意	746
付録 C.1	他社 COBOL との仕様比較	746
付録 C.2	C 言語を使用するときの注意	746
付録 C.3	Visual Basic , Delphi , Access を使用するときの注意	747

付録 D 画面定義の制限値	748
付録 D.1 オブジェクトの個数およびサイズ	748
付録 D.2 使用できるグラフィックデータ	751
付録 E XMAP3 のリターンコード・詳細コード	753
付録 F 用語解説	759

## 索引

# 1

## 画面機能の概要

この章では、XMAP3 で作成できる画面，および画面を作成する前に知っておかなければならない事項について説明します。

---

1.1 XMAP3 で作成できる画面

---

1.2 XMAP3 の画面設計の概要

---

1.3 画面設計の流れ

---

1.4 画面で使用できる機能（GUI 画面，CUI 画面）

---

1.5 XMAP3 画面制御方式の解説

---

1.6 XMAP3 関連ファイルの関係

---

1.7 画面を設計する前に

---

## 1.1 XMAP3 で作成できる画面

---

XMAP3 で作成できる画面には GUI 画面と CUI 画面があります。

GUI 画面は、キーボードからの文字入力とマウスによる操作ができる画面です。Windows の「ボタン」「スクロール」「プルダウン・カスケード」「ポップアップ」などが使用できます。画面の大きさは自由（最大 48 行× 160 列（半角文字））です。画面の背景色はユーザが指定できます。

CUI 画面は、キーボードからの文字入力を中心の画面です。メインフレーム型オンライン業務や、MS-DOS ベースの業務で使用していた画面です。画面の大きさは、24 行× 80 列（半角文字）で、画面の背景色は黒色で固定されています。

XMAP3 では、これらの画面の入出力を、マップを使用して実現しています。マップには、画面の各オブジェクトの位置などを格納した物理マップと、画面の変動情報を設定する論理マップ（AP 中のデータ領域）があります。この物理マップと論理マップを利用して画面の入出力を実現します。



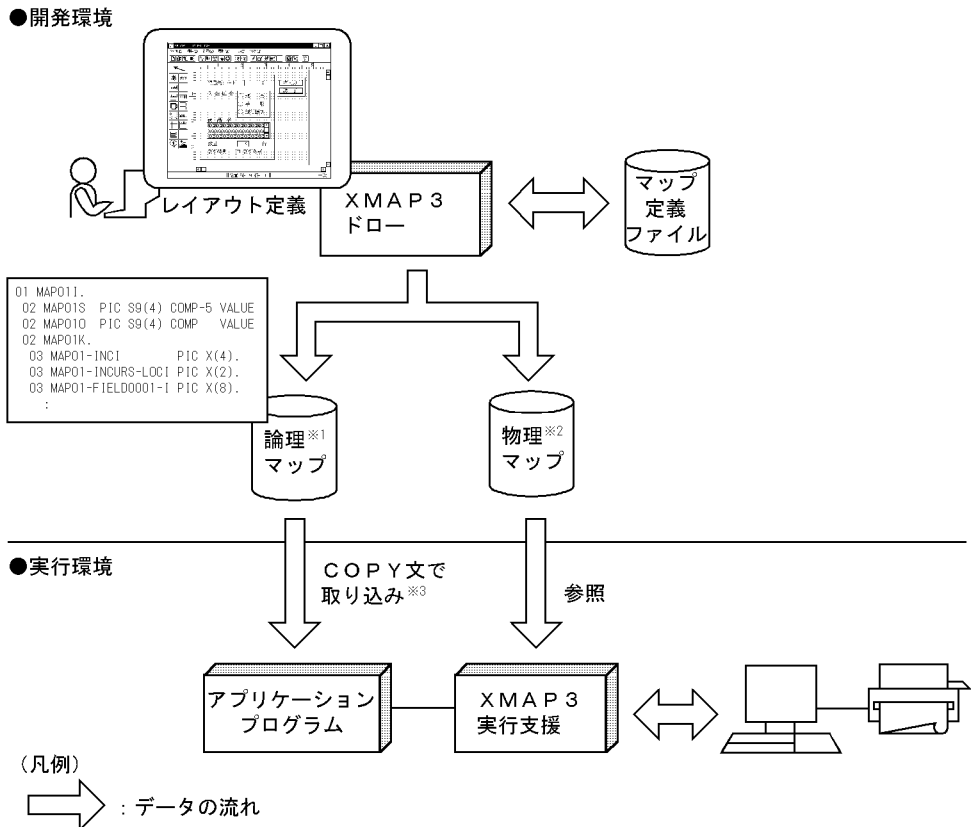
## 1.2 XMAP3 の画面設計の概要

XMAP3では、画面のレイアウト（定義）とAPの処理を分離したマッピング方法を採用しています。このため、画面設計では、画面のレイアウト定義作業とコーディング作業とを分離できます。XMAP3でAPを作成するときの利点を次に示します。

- データ名をあらかじめ決めておくことで、レイアウト定義で配置したオブジェクトの位置を意識することなく、APをコーディングできます。
- XMAP3で画面のレイアウトを定義すると、登録集原文が「論理マップ」として生成されます。生成された論理マップをAPでCOPY文で取り込んで使用すれば、効率良くAPを開発できます。

画面のレイアウト定義とAPの関係を次の図に示します。

図 1-1 画面のレイアウト定義とAPの関係



注※1 画面の可変情報を設定するAP中のデータ領域です。

注※2 画面の固定テキスト／フィールドや入出力テキスト／フィールドの位置などを格納したテーブルです。

注※3 C言語の場合、#include指示語になります。

## 1.3 画面設計の流れ

---

XMAP3 では、次のような流れで画面を設計します。

1. 作成する画面の種類を決定する
2. 画面を設計する
3. 画面のレイアウトを定義する
4. AP をコーディングする

それぞれの手順の概要を次に示します。

### 1.3.1 作成する画面の種類を決定する

キーボードからの文字入力とマウスによる操作ができる GUI 画面を作成するか、キーボードからの文字入力を中心の CUI 画面を作成するかを検討します。

GUI 画面と CUI 画面では使用できる機能が異なります。このため、GUI 画面、および CUI 画面で使用できる機能を確認してから、作成する画面の種類を決定します。使用できる機能については、「1.4 画面で使用できる機能 (GUI 画面, CUI 画面)」を参照してください。

### 1.3.2 画面を設計する

XMAP3 での画面の制御方式を理解し、画面のサイズや表示形式などを設計します。

XMAP3 での画面の制御方式については、「1.5 XMAP3 画面制御方式の解説」を参照してください。画面サイズや表示形式などの設計については、GUI 画面を作成する場合には「2.1 GUI 画面の設計」を、CUI 画面を作成する場合には「3.1 CUI 画面の設計」を参照してください。

### 1.3.3 画面のレイアウトを定義する

ドローを起動して表示される画面定義ウィンドウで、画面の属性を設定し、画面上にオブジェクトを配置してレイアウトを定義します。

#### (1) 画面属性を設定する

画面をレイアウトする前に、画面属性情報を設定します。画面属性情報とは、画面を制御する情報のことです。画面属性情報は、ドローの画面属性ダイアログで設定します。

画面属性の設定については、GUI 画面を作成する場合には「2.3 GUI 画面属性の設定」を、CUI 画面を作成する場合には「3.3 CUI 画面属性の設定」を参照してください。

## (2) オブジェクトを配置する

画面定義ウィンドウで、レイアウトを定義します。レイアウト定義後、定義情報を保存すると、画面の定義情報を格納したファイル（マップ定義ファイル）が生成されます。

画面のレイアウト定義の詳細については、GUI画面を作成する場合には「2.4 GUI画面を構成するオブジェクト」を、CUI画面を作成する場合には「3.4 CUI画面を構成するオブジェクト」を参照してください。

### 1.3.4 AP をコーディングする

レイアウト定義で作成された定義ファイルを使用して、APを作成します。APの作成には、COBOL、C言語、またはJavaを利用できます。また、XMAP3では言語に依存しないマッピング関数も用意しているので、Visual Basic や Delphi も利用できます。

なお、Javaを利用できるのは、XMAP3/Web for Cosminexus を使用して Cosminexus 上で実行する場合です。

APコーディングについては、「6. 画面定義とコーディングの関連」、「7. APのコーディング方法」、「8. 標準パターンを使用したAPの作成」を参照してください。ただし、XMAP3/Web for Cosminexus を使用して Cosminexus 上で実行する場合のAPコーディングについては、マニュアル「XMAP3/Web for Cosminexus」を参照してください。

XMAP3/Web for Cosminexus を使用して Cosminexus 上で実行する場合

XMAP3 で開発した画面・帳票は、XMAP3/Web for Cosminexus を使用して、Cosminexus 上で実行することもできます。この場合、AP開発言語には、Java または COBOL が使用できます。Java の場合、Java 言語用ツールを使用して動的変更テーブルおよび論理マップを XML ファイルに変換する必要があります。Java 言語用ツールの使用方法については、マニュアル「XMAP3 開発・実行ガイド」を参照してください。

## 1.4 画面で使用できる機能（GUI画面，CUI画面）

---

GUI画面およびCUI画面で使用できる機能について説明します。

### 1.4.1 GUI画面で使用できる機能

GUI画面の例を図1-2～図1-5に，GUI画面の機能内容を表1-1に示します。

図1-2 メニューバー，タイトルバーがない画面の例（スプラッシュ画面）

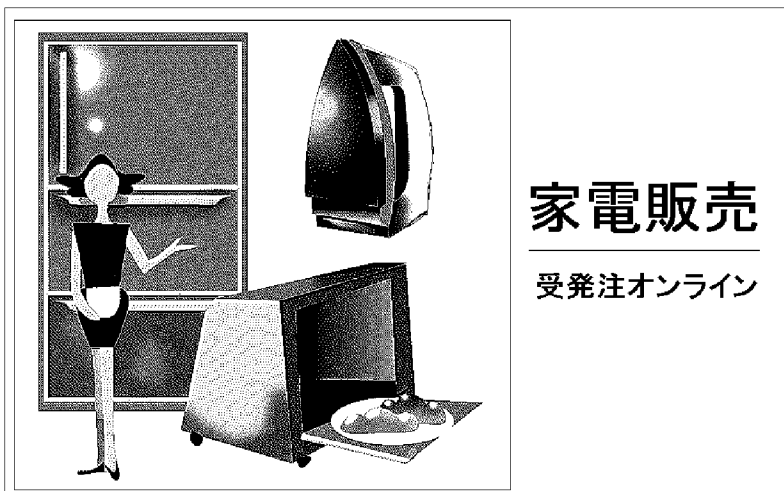
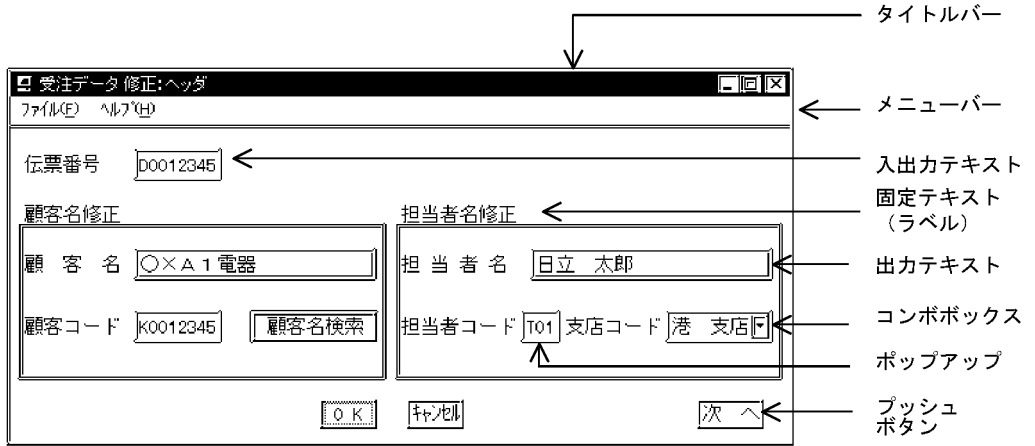


図 1-3 GUI 画面の例 1 (標準的な画面)



1. 画面機能の概要

図 1-4 GUI 画面の例 2 (ダイアログ表示時の画面)

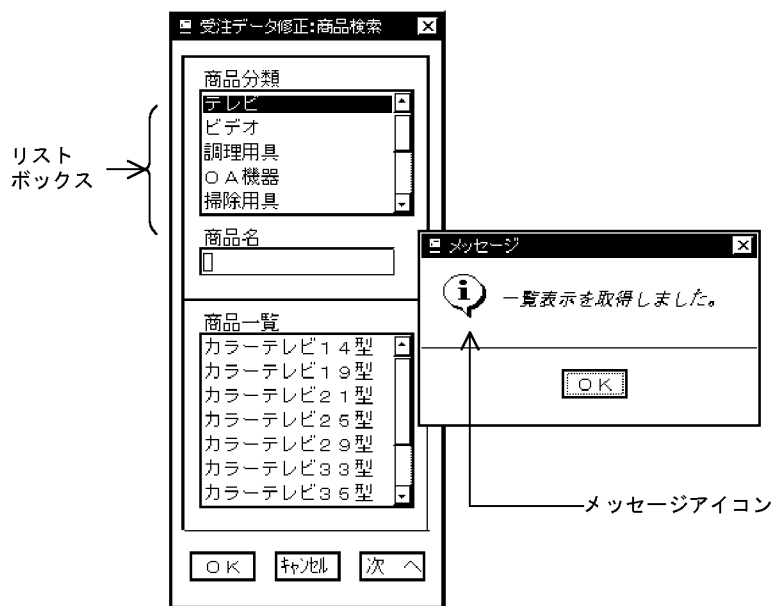


図 1-5 グラフィック利用画面の例

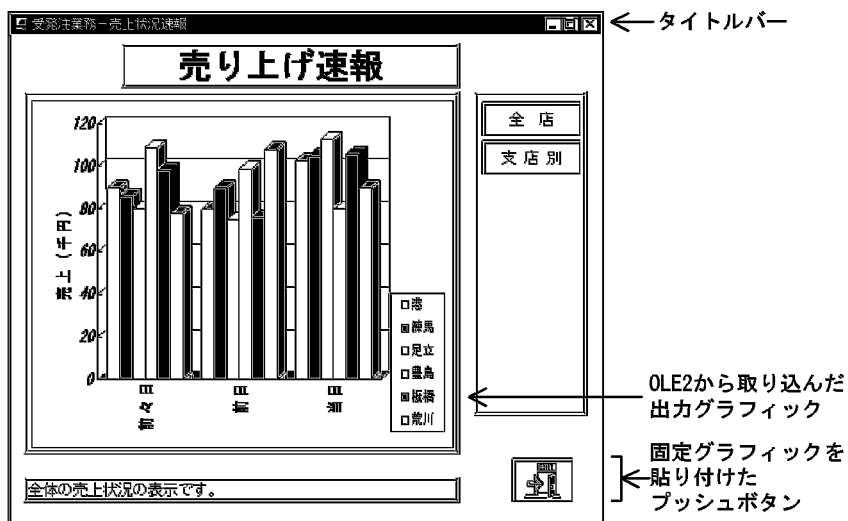


表 1-1 GUI 画面の機能内容

項目	機能内容		
画面サイズ	論理画面は最大 127 行 × 255 文字（フィールドボックスでスクロール使用時） ウィンドウサイズは最大 48 行 × 160 列（ディスプレイ装置の解像度による） ウィンドウサイズが画面より大きい場合は、オプションでスクロールバーを付ける 全画面表示（一次ウィンドウだけ有効）		
ウィンドウ	タイトルバーあり / なし オペレーティングキーあり / なし ウィンドウサイズ変更（リサイズ）あり / なし 背景色（15 色）、壁紙 ウィンドウ位置（XYZ 位置） / 一時非表示 フォーカス順序の設定		
同時に表示できる二次ウィンドウの数 <sup>1</sup>	一次ウィンドウ 1 個に対し、二次ウィンドウ 3 個		
GUI 機能 (使用できる オブジェクト (2D / 3D))	キーエントリ系	入出力テキスト・フィールド（数字、数値、金額、日本語、カナ、英数、MCR、パスワード） 日付テキスト・フィールド（入出力） 時刻テキスト・フィールド（入出力）	
	ラベル系	固定テキスト・フィールド 出力テキスト・フィールド（数字、日本語、英数） 日付テキスト・フィールド（出力） 時刻テキスト・フィールド（出力）	
	選択エントリ系	コンボボックス（キー入力コンボボックス（固定 / 可変）、メニュー選択コンボボックス（固定 / 可変） <sup>2</sup> 、ポップアップテキスト・フィールド（固定 / 可変） <sup>2</sup> 、スピンボックス	
	候補選択コントロール	リストボックス <sup>2</sup> 、ラジオボタン（固定 / 可変）、チェックボタン（固定 / 可変）、トグルフィールド	
	グループコントロール	グループボックス、フィールドボックス <sup>2</sup> 、フレーム	
	グラフィックコントロール	固定グラフィック <sup>3</sup> 、出力グラフィック <sup>2 3</sup> 、セパレータ、けい線、メッセージアイコン、塗りつぶしフィールド	
	コマンドコントロール	メニューバー、プルダウンメニュー、カスケードメニュー、プッシュボタン（グラフィック貼り付けボタンを含む）	
	予約項目	予約テキスト・フィールド <sup>4</sup>	
表示文字	画面単位	大、中、小	
	オブジェクト単位	文字サイズ	標準文字サイズ、大きい文字、小さい文字
		文字の書体	標準、明朝、ゴシック
		文字の強調	太字、斜体、太字 & 斜体
		文字の拡大	標準、長体（縦 2 倍）、平体（横 2 倍）、倍角（縦横 2 倍）
文字色	12 色		

# 1. 画面機能の概要

項目	機能内容
確定キー	送信キー，PF1 ~ PF84 キー，PA1 ~ PA3 キー，スクリーン消去キー，割込キー
その他	アラーム音，IDカード（MCR）の入力，フォーカス・カーソル位置設定

注

使用できるオブジェクトの制限値については、「付録 D 画面定義の制限値」またはオンラインヘルプを参照してください。また、各オブジェクトの詳細については「2. GUI画面の作成」を参照してください。

注 1

二次ウィンドウの表示は、一次ウィンドウが表示されていることが前提になります。

注 2

スクロールバーを付けられます。

注 3

フルカラー対応です。

注 4

予約テキスト・フィールドを定義した画面は OLTP サーバ構成の環境で使用できます。予約項目名は OpenTP1 で提供している範囲が使用できます。

## 1.4.2 CUI画面で使用できる機能

CUI画面の例を図 1-6 に、CUI画面の機能内容を表 1-2 に示します。

図 1-6 CUI画面の例

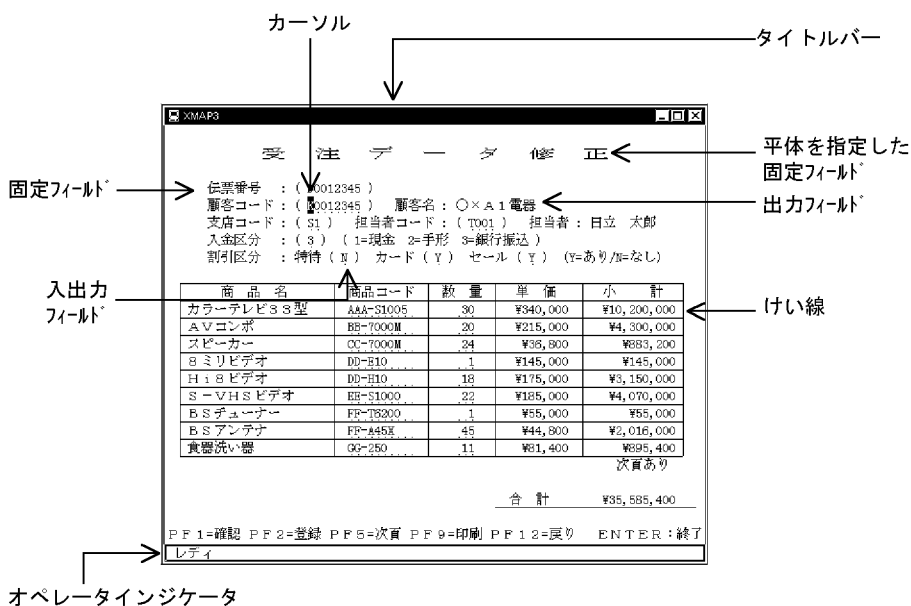




表 1-2 CUI 画面の機能内容

項目	機能内容	
画面サイズ	24 行 × 80 列	
CUI 機能 (使用できるオブ ジェクト)	ラベル系	固定フィールド, 出力フィールド (数字, 英数, 日本語)
	キーエント リ系	入出力フィールド (数字, カナ, 英数, 日本語, パスワード, MCR)
	グラフィッ クコント ロール	けい線 (縦線, 横線), 矩形, 表けい線の枠, 表けい線の横線 (縦線) 繰り返し
	グループコ ントロール	フレーム
	予約項目	予約フィールド
文字の拡大	標準, 長体 (縦 2 倍), 平体 (横 2 倍), 倍角 (縦横 2 倍)	
文字色	7 色	
確定キー	送信キー, PF1 ~ PF24 キー, スクリーン消去キー, 割込キー, PA1 ~ 3 キー	
その他	アラーム音, ID カード (MCR) の入力, カーソル位置設定	

## 注

予約テキスト・フィールドを定義した画面は OLTP サーバ構成の環境で使用できます。

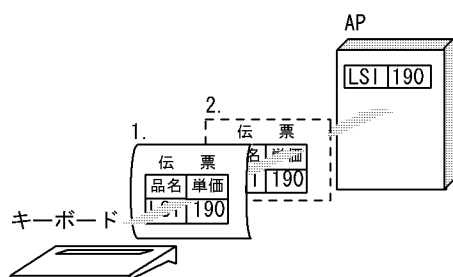
## 1.5 XMAP3 画面制御方式の解説

XMAP3 での、画面と AP 間の制御方式や、画面設計時の描画方式について説明します。

### 1.5.1 画面の制御方式

XMAP3 では、GUI 画面であっても、オンライン端末と同じように、画面単位の入出力（手続き型）で AP を作成できます。このような画面制御を実現させるために、XMAP3 では画面バッファという論理的な画面を設けてウィンドウシステムと AP とのやり取りをします。なお、GUI 画面特有の機能を実現させるために、オプションで、フィールド単位の入出力の AP も作成できます。

図 1-7 XMAP3 での制御方式



#### 1. 実際のウィンドウ表示

ウィンドウには、AP から返されたデータが表示されています。ユーザは、このウィンドウに対し、データを入力します。

#### 2. 画面バッファ

画面ごとのデータを AP とやり取りするために、XMAP3 がメモリ中に持っている論理的な画面です。AP から返されたデータは、画面バッファに展開されてから、実際のウィンドウに表示されます。ユーザによってウィンドウに入力されたデータも、画面バッファを介して AP に返されます。

#### 画面表示

AP から、データの表示が要求されます。XMAP3 は、画面バッファに表示内容を展開し、実際のウィンドウ上に表示します。

OLTP サーバ構成の環境では、TP1/NET/XMAP3 の XMAP3 共通定義 (mcftpx) や論理端末定義 (mcftalcle) での設定が優先されます。XMAP3 共通定義、および論理端末定義については、マニュアル「OpenTP1 プロトコル TP1/NET/XMAP3 編」を参照してください。

#### 画面操作

マウスやキーボードを使用した操作は、「イベント」としてウィンドウシステムから XMAP3 に返されます。XMAP3 では、このイベントを画面バッファに反映していき

ます。この時点では、まだ AP へ制御は返されません。

#### AP へ制御を返すタイミング

AP へ制御を返すタイミングは、画面確定のときです。画面が確定すると、画面バッファの内容は AP が受け取れる型に編集され、AP へ送られます。

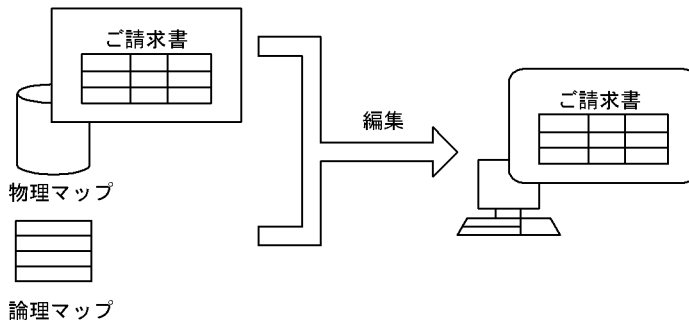
画面確定は、一般的に送信キー、ファンクションキー、およびボタン（プッシュボタン）などを押した場合に成立します。どのような操作によって画面が確定したかは、イベント通知コードで AP に通知されます。また、フォーカス・カーソルがオブジェクト（またはフィールド）を離れたときに、強制的に画面確定を成立させるオプションもあります。

なお、OLTP サーバ構成の環境の場合、TP1/NET/XMAP3 上で、「次メッセージ要求キー」として割り当てられているキーのイベントは AP に渡りません。したがって、AP にイベントを通知したいキーは、TP1/NET/XMAP3 上で次メッセージ要求キーに割り当てないよう設定してください。

## 1.5.2 画面の描画方式

XMAP3 では、マッピング方式という画面描画方式を利用します。マッピング方式とは、定義されたレイアウトの情報（物理マップ）と、AP が出力するデータ（論理マップ）を編集して、画面を生成する方式です。

図 1-8 マッピング方式



## 1.5.3 画面の表示モード

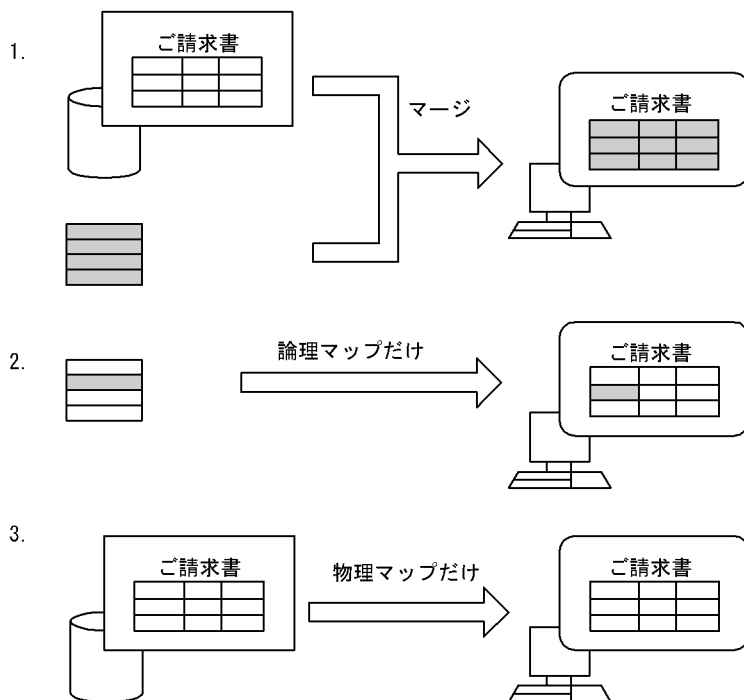
### (1) マッピングオプション

画面の表示モードには、「マージ」「論理マップだけ」「物理マップだけ」の三つの指定があります。これらは、日立 COBOL で SEND 文および TRANSCIVE 文を使用したときは、MAPPING MODE 句で指定します。日立 COBOL で CALL 文を使用したとき、および C 言語を使用したときは、マッピングインタフェースで指定します。

なお、OLTP サーバ構成で使用する場合、「物理マップだけ」は指定できません。画面の表示モードは「マージ」、または「論理マップだけ」を指定してください。

## 1. 画面機能の概要

図 1-9 画面の表示モード



### 1. マージ：

論理マップに代入したデータと物理マップをマージして画面を表示します。同一画面を書き換えるときに使用します。

表示属性の変更要求をしないオブジェクト属性については、画面定義時に指定した表示属性になります。

同じ画面に対し、2回目以降の表示では固定部分（タイトルやけい線）は再描画しません。

### 2. 論理マップだけ：

論理マップに代入したデータを使用して画面を表示します。一般的に、2回目以降の表示で、入力した状態をそのままにして部分的に書き換えるときに使用します。表示属性の変更指定をしないオブジェクト属性については、直前の表示属性になります。

### 3. 物理マップだけ：

論理項目、または制御項目に指定したデータを無視して、画面定義時の情報（物理マップ）だけで画面を表示します。論理マップをすべてデータ有無コードでクリアしたときと同じで、メニューなどの初期表示に使用します。

### 注

画面属性ダイアログで「入力・選択状態の扱い」の「初期状態」を選んでいる場合、入力データは消去されます。入力データを残したい場合は、「初期状態」以

外を選んでください。

## (2) 表示形態

表示形態には、「全面書換」、「一部上書」、および「自動」の三つの指定があります。なお、表示形態とマッピングオプション（マッピングモード）の関係については、「(3) マッピングオプションと表示形態の組み合わせ」および「2.3 GUI 画面属性の設定」を参照してください。

### 全面書換

表示中の画面を消去し、次の画面を全画面描画します。

### 一部上書

直前に表示した画面の、一部のオブジェクトだけを変更して表示します。「全面書換」に比べて表示時間を短縮できます。この場合、マッピングオプションの「マージ」か「論理マップだけ」とあわせて使用する必要があります。

なお、直前の画面と異なるマップ名のときは「全面書換」の動作となります。

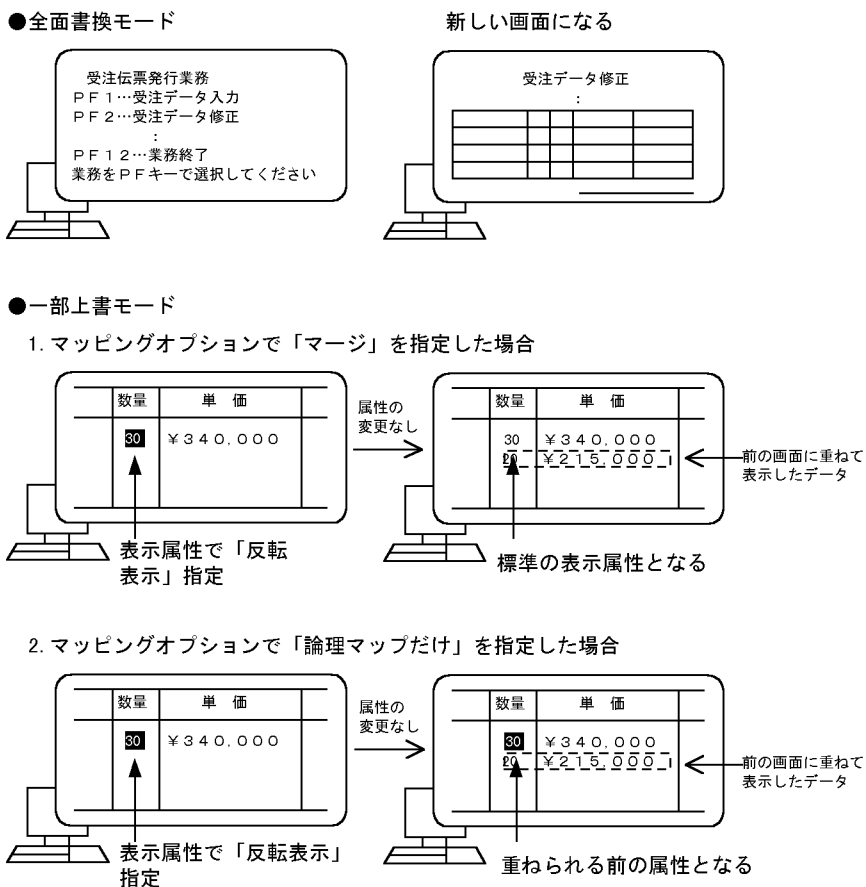
### 自動

「一部上書」と「全面書換」を XMAP3 に任せて、特に AP では意識しません。直前の画面と同じマップ名のときは「一部上書」を仮定します。

標準は「自動」です。

# 1. 画面機能の概要

図 1-10 表示形態



## (3) マッピングオプションと表示形態の組み合わせ

指定したマッピングオプションによって、指定できる表示形態が変わります。

マッピングオプションと表示形態の組み合わせを次の表に示します。

表 1-3 マッピングオプションと表示形態の組み合わせ（1回目：新しいマップ名で表示する場合）

表示形態	マッピングオプション		
	マージ	論理マップだけ	物理マップだけ
全面書換		×	
一部上書	×	×	×
自動		×	×

(凡例)

- : 指定できる。
- x : 指定できない。

表 1-4 マッピングオプションと表示形態の組み合わせ (2 回目以降 : 同じマップ名で表示する場合)

表示形態	マッピングオプション		
	マージ	論理マップだけ	物理マップだけ
全面書換		x	x
一部上書			x
自動			x

(凡例)

- : 指定できる。
- : 指定できるが、すべて表示し直すため効率が悪い。
- x : 指定できない。

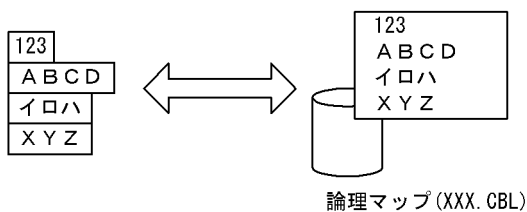
また、これらの表示形態と、Z 位置指定 (ウィンドウの重なり指定) をあわせて設定できます。詳細は「2.1.3(2) ウィンドウ表示位置と重なり」を参照してください。

## 1.5.4 論理マップ

XMAP3 と AP との間でやり取りされるデータを、論理データといいます。論理データを AP で扱うための領域定義文を論理マップといいます。領域定義文は、COBOL 言語では登録集原文 (COPY)、C 言語の場合はヘッダファイルになります。

論理マップは XMAP3 が自動生成します。XMAP3 へ渡すデータを格納する領域を出力論理マップ、XMAP3 から受け取るデータを格納する領域を入力論理マップといいます。

図 1-11 論理マップ



## 1.5.5 データ有無コード

画面にデータを表示させる場合は、論理マップのデータ項目に、表示させたいデータを代入して XMAP3 に返します。ただし、データ項目にデータを代入したくない場合には、そのデータ項目が無効であることを明示的に XMAP3 に知らせる必要があります。デー

## 1. 画面機能の概要

データ項目の内容が無効であることを XMAP3 に知らせるためには、データ有無コード（標準は  $(1F)_{16}$ ）という特別なコードを使用します。

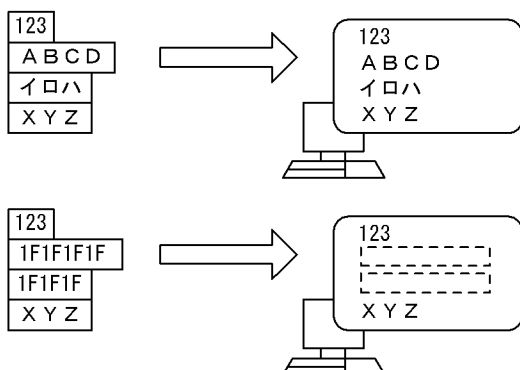
同じ画面に対してデータの上書き（一部上書）をする場合、書き換えたくないデータにデータ有無コードを指定して使用します。

また、制御項目のデータ項目にデータ有無コードを代入すると、データ書き換え時に表示属性を変更しないようにしたり、変更した表示属性を元に戻したりする設定を AP から制御できます。

なお、データ有無コードの値は必要に応じて AP から変更できます。ただし、すべての AP にかかわるコードなので、できるだけ標準値を利用することをお勧めします。

OLTP サーバ構成で使用する場合、AP で使用するデータ有無コードは TP1/NET/XMAP3 のマッピングサービス属性定義ファイルで指定してください。マッピングサービス属性定義ファイルについては、マニュアル「OpenTP1 プロトコル TP1/NET/XMAP3 編」を参照してください。

図 1-12 データ有無コード



### 1.5.6 表示属性の動的変更

AP でオブジェクトの表示属性を実行時に変更できます。これを「動的変更」といいます。オブジェクトとは XMAP3 で定義できるテキスト・フィールドやボタンなどの総称です。

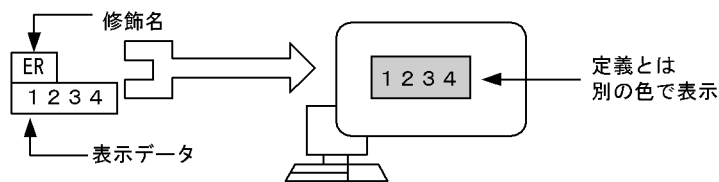
表示属性の動的変更を指定すると、マップ生成後には、表示属性変更の情報を受け取る領域（接尾語「-A」が付くデータ項目（これを制御項目といいます））が自動的に生成されます。AP では、表示属性を変更するデータ項目に XMAP3 が用意する修飾名をセットすると、実行時に表示属性を変更して表示します。なお、XMAP3 が用意している修飾名以外に、ユーザが表示属性を設定して用意しておく修飾名も利用できます。各修飾名で設定されている表示属性の詳細については、「付録 A 属性情報一覧」を参照してください。



表示属性はドロースेटアップで修飾名ごとに設定します。ドロースेटアップでの設定については、マニュアル「XMAP3 開発・実行ガイド」を参照してください。

なお、マッピングオプションに「空白または0(マージ)」を指定している場合、表示属性を変更するデータ項目にヌル( $(00)_{16}$ )またはデータ有無コード(データがないことを示す1バイトのコード)をセットすると、標準の表示属性で表示します。「2(論理マップだけ)」を指定している場合、前回の画面表示時に指定した表示属性がそのまま表示されます。

図 1-13 動的変更



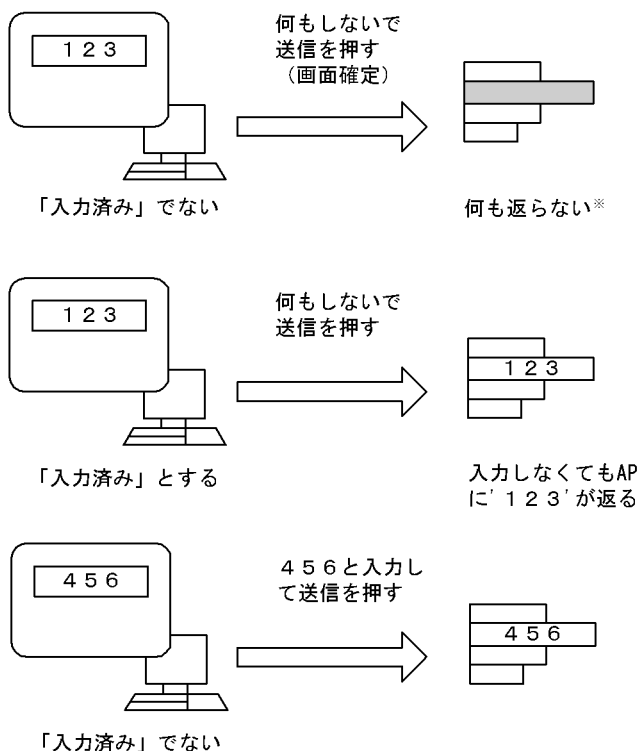
## 1.5.7 オブジェクトの転送モード

転送モードでは、入出力テキストや入出力フィールドおよびボタンを選ばれた状態(例えばデータが入力されている状態や、ボタンが選択されている状態)にしておけます。

転送モードには、転送状態(入力済み、または選択済みの状態)と、非転送状態(未入力、未選択)があります。転送状態にしておくと、ユーザがデータを入力しないで画面確定を成立させた場合でも、画面に表示されているデータが無条件にAPへ返されます。ただし、オブジェクトに初期値が設定されていると、未入力または未選択(非転送状態)のとき、この初期値がAPに返されます。

## 1. 画面機能の概要

図 1-14 転送モード

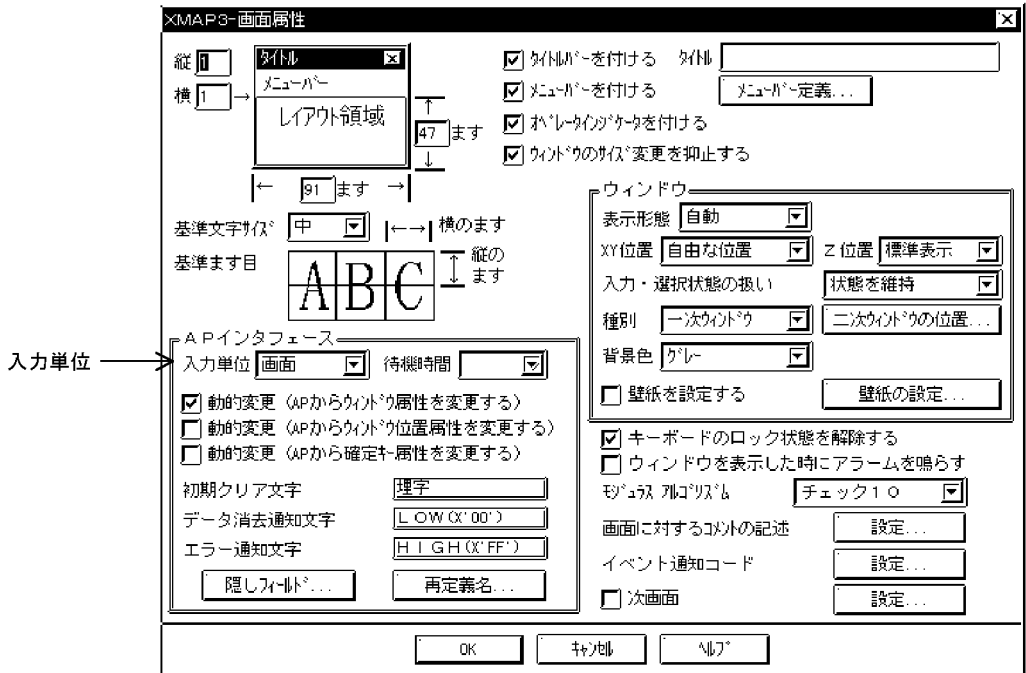


注※ オプションで、何もしないで画面を確定させたことを示すデータを返せます。

### 1.5.8 入力単位

入力は、基本的に画面単位です。ただしオプションとして、フィールド単位およびイベント単位の入力を選べます。ただし、入力単位がイベント単位またはフィールド単位の場合は、オブジェクトごとにAPとの通信が発生します。そのためC/Sシステム環境では、画面単位の場合と同じようなスムーズなオブジェクト間の遷移はできません。応答性を要求するようなシステムでは注意が必要です。

なお、OLTPサーバ構成で使用する場合には画面単位を指定してください。画面単位以外を指定した場合、APを継続問い合わせ応答型のプログラムとして作成する必要があります。継続問い合わせ応答型のプログラムは複雑になりやすく、お勧めできません。



## (1) 画面単位

### (a) 画面確定のタイミング

画面属性ダイアログの入力単位で「画面」を指定すると、AP から画面を表示するときに、入出力の制限（不活性）を指定しているオブジェクト以外は入力・選択できる状態で表示されます。これらのオブジェクトを選択したり、データを入力したりしたあと、画面確定を設定しているメニューバー、プッシュボタンまたは画面確定キーを押すと、AP に制御が渡ります。なお、オブジェクトに「自動送信」を設定すると、そのオブジェクトからフォーカス・カーソルが離脱するタイミングで「送信」の画面確定が仮定されます。

## (2) フィールド単位

### (a) 画面確定のタイミング

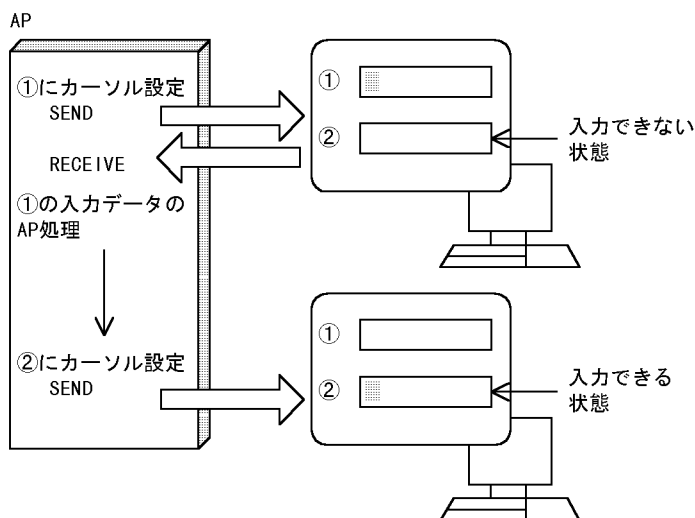
画面属性ダイアログの入力単位で「フィールド」を指定すると、AP から画面を表示するときに、AP で設定されているフォーカス（カーソル）があるオブジェクトだけが入力・選択可能になります。ただし、確定キー（送信キーや PF キー）、プッシュボタンおよびメニューバーは除かれます。

これらのオブジェクトを選択したり、データを入力したりしたあと、フォーカス（カーソル）が次のオブジェクトに移動するタイミングで AP に制御が渡ります。なお、各オブジェクトでの「自動送信」の指定は無効になります。

## 1. 画面機能の概要

### (b) AP での処理

AP では、画面上の該当するオブジェクトに対応する項目、イベント通知コード、隠しフィールドおよびフォーカス・カーソルにだけ値が格納されます。AP ではこの項目のデータチェック処理などをします。AP でのデータチェック処理などが終わると、AP が次に操作するオブジェクトにフォーカス・カーソルを設定し、SEND 要求で画面を表示します。すると XMAP3 では、今回 AP 側でフォーカスが設定してあるオブジェクトだけを入力できる状態にして画面を表示します。なお、PF キーやプッシュボタン、タイトルバーなどでの画面確定は常に有効です。



### 注

入力単位が「フィールド」の場合、「自動送信」の指定は無効になります。

### (3) イベント単位

#### (a) 画面確定のタイミング

画面属性ダイアログの入力単位で「イベント」を指定すると、AP から画面を表示するときに、入出力の制限（不活性）を指定しているオブジェクト以外は入力・選択できる状態で表示されます。なお、各オブジェクトでの「自動送信」の指定は無効になります。

これらのオブジェクトの一つを選択したり、データを入力したあと、フォーカス・カーソルが次のオブジェクトに移動するタイミングで AP に制御が渡ります。このとき、個々のオブジェクトを識別する通知コードが AP に渡ります。通知コードはイベント通知コードという定数です。なお、イベントが発生したとき、AP に通知するかどうかは画面属性ダイアログで指定できます。イベントの定義に関しては「2.3.1(6) イベント通知コード」を参照してください。また AP にイベントが渡るタイミングの詳細については「1.5.9 画面確定と AP への通知」を参照してください。

## (b) AP での処理

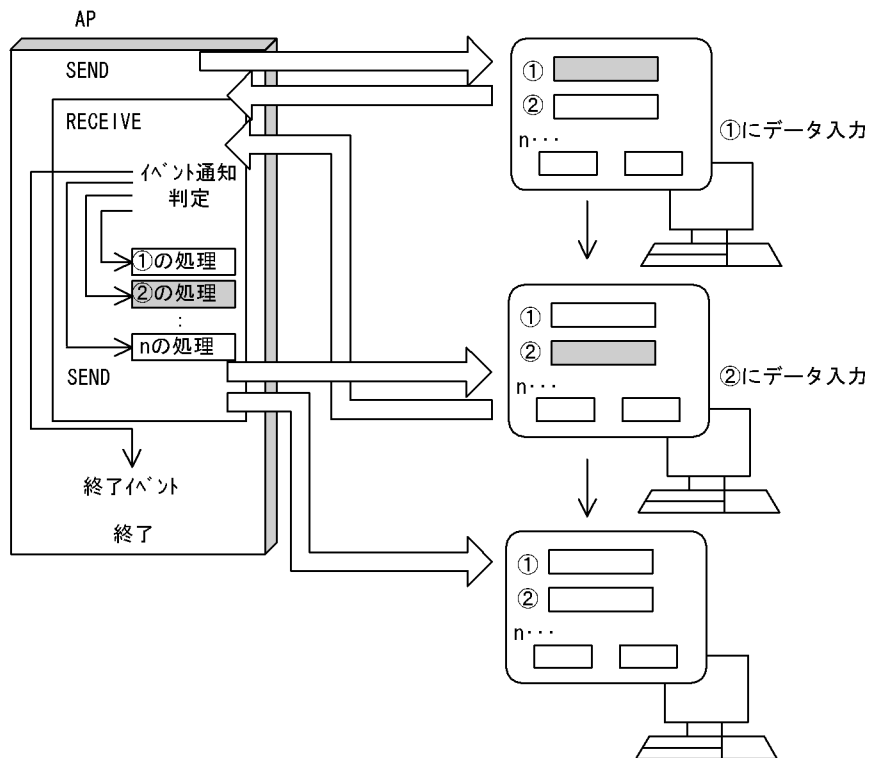
AP では、論理マップ中のイベント通知項目に格納されているイベント通知コードによって、渡されたデータがどのオブジェクトから渡されたものを判断し、そのオブジェクトに該当する処理のサブルーチンなどを呼び出します。

そのため、オブジェクトにはユニークなイベント通知コードが自動的に割り当てられます。ただし、反復（繰り返し）オブジェクトに対しては、同じイベントを返します。

イベント通知コードは、論理マップと一緒にオブジェクトに対応したデータ名に「-V」を付けてイベント定数として生成されます。オブジェクトに対応するデータ名を使用しているため、AP 中で、オブジェクトとイベントの対応がわかりやすくなっています。

## (例)

データ名が FIELD1- の場合  
 出力データ項目：FIELD1-O  
 入力データ項目：FIELD1-I  
 イベント定数：FIELD1-V



## 注 1

入力単位が「イベント」の場合、「自動送信」の指定は無効になります。入力単位が「イベント」のときは常に、活性状態のオブジェクトの一つを選択したり、データを

## 1. 画面機能の概要

入力したあと、フォーカス・カーソルが次のオブジェクトに移動するタイミングで AP に制御が渡ります。

### 注 2

繰り返しの指定があるテキスト・フィールドのイベント通知コードは、先頭のテキスト・フィールドのイベント定数が AP に返ります。

### 注 3

ドロースेटアップのイベント通知コードダイアログで「通知コードの長さ」を 4 以外で設定すると、入力単位「イベント」は利用できません。

## (4) 表示直後（強制送信）

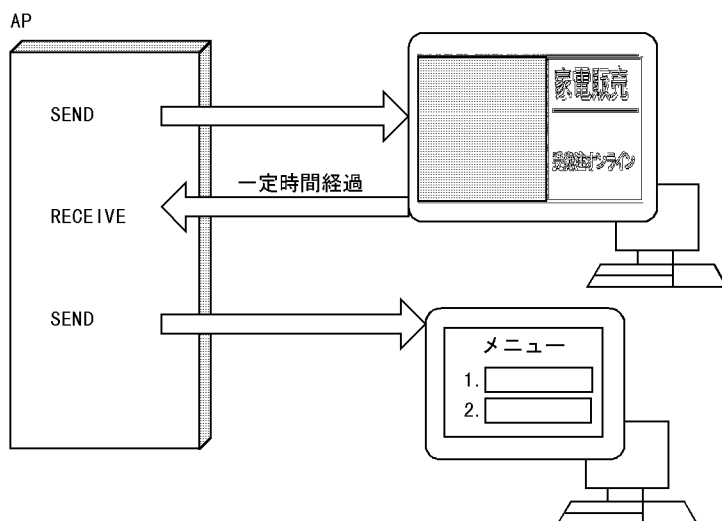
### (a) 画面確定のタイミング

画面属性ダイアログの入力単位で「表示直後」を指定すると、画面に対して何の操作がされなくても、一定の時間が過ぎると自動的に AP に制御を渡します。その場合のイベント通知コードは、送信キーを押したときと同じ値を返します。また、AP に制御が渡るまでの時間を設定しないで、表示直後に AP に制御を渡すような設定もできます。なお、設定した時間が過ぎる前に、ボタンを押すなど、画面に対する操作をした場合は、表示直後の指定が解除され、画面単位と同じ動作になります。

スプラッシュ画面や、表示専用の掲示板のような画面を表示するときに利用できます。

スプラッシュ画面や掲示板のような画面については「1.4.1 GUI 画面で使用できる機能」および「2.1.6 GUI 画面設計でこんなこともできます」を参照してください。

### (b) AP での処理



## (c) 表示直後を使用するときの注意

画面属性で「表示直後」を指定した画面の注意事項について説明します。

- 画面には、入力必須属性を設定したオブジェクトを定義しないでください。入力必須属性を設定したオブジェクトを定義した画面を表示した場合、待機終了時に AP に対してエラーが通知されます。
- 表示する画面にグラフィックオブジェクトがあると、画面表示に時間がかかることがあります。このため、指定した「待機時間」よりも長く待機することがあります。

## 1.5.9 画面確定と AP への通知

### (1) AP にイベントが渡るタイミング

次に AP にイベントが渡るタイミングを示します。

#### GUI 画面 / CUI 画面共通

- PF キーなど、確定キーを押したとき。
- メニューバーで AP 通知となっているメニューを選択したとき。
- プッシュボタンで AP 通知となっているメニューを選択したとき。
- 「自動送信」が指定してあるオブジェクトからフォーカス・カーソルが離脱したとき。
- タイトルバーの [ 閉じる ] メニュー / ボタンを選んだとき。

#### GUI 画面固有

- リストボックスのリスト項目をマウスでダブルクリックしたとき。
- MCR カードの読み込みが終了したとき。
- 画面属性ダイアログの「入力単位」で「表示直後」が指定してある画面で、設定した待機時間に達したとき。
- 画面属性ダイアログの「入力単位」で「フィールド」または「イベント」が指定されている画面で、フォーカス・カーソルがオブジェクトから離脱したとき。

#### 注

手動ポップアップの表示は、画面確定の対象とはなりません。

### (2) 各入力単位での画面確定操作

各入力単位が指定されているウィンドウで画面確定の操作した場合の結果を次に示します。

画面確定の操作	入力単位			
	画面	フィールド	イベント	表示直後
画面確定キー				
プッシュボタン				

## 1. 画面機能の概要

画面確定の操作	入力単位			
	画面	フィールド	イベント	表示直後
メニューバー				
[閉じる]メニュー				
[閉じる]ボタン				
リストボックスのダブルクリック				
自動送信		-	-	×
MCR 入力後		-		
待機時間	-	-	-	
オブジェクトからのフォーカス・カーソルの離脱	-			-

(凡例)

- : 画面確定する。
- × : 画面確定しない。
- : 操作は無効になる。

### (3) AP へ通知するための設定

各画面確定の操作で、AP へ通知するための必要な設定内容を示します。

画面確定の操作	表示・印刷セットアップ	ドロースセットアップ	ドロ	AP
画面確定キー	キーの割り当て	有効となるキーの特定	AP から動的変更するかどうかの指定	修飾名の指定 (動的変更時)
プッシュボタン	-	-	通知コードの割り当て	-
メニューバー	-	-	通知コードの割り当て	-
[閉じる]メニュー	使用するかどうか	-	-	-
[閉じる]ボタン	使用するかどうか	-	-	-
リストボックスのダブルクリック	使用するかどうか	-	-	-
自動送信	-	-	各オブジェクトの定義ダイアログで指定	-



画面確定の操作	表示・印刷セットアップ	ドローセットアップ	ドロー	AP
入力単位	-	-	画面属性ダイアログ	-
MCR 入力	-	-	各オブジェクトの定義ダイアログで指定	-

(凡例)

- : 操作は無効になる。

注

表示・印刷セットアップの「キー操作 1」タブの「入力直後 AP へ制御を渡す (自動送信) 動作」の「データ変更時だけ制御を渡す」を設定すると、入力操作をしたときだけ自動送信の通知をします。

#### (4) イベントとイベント通知コード

画面確定の操作とイベント種別、および通知コードの関係を次に示します。

画面確定の操作		イベント種別	標準の通知コード (ドローセットアップで変更できる)
画面確定キー	PF1 ~ PF24	確定キー	PF01 ~ PF24
	PF25 ~ PF84		A001 ~ A060
	送信 (実行) キー		ENTR
	割込みキー		BREK
	スクリーン消去キー		SCRN
	PA1 ~ PA3		PA1 ~ PA3
プッシュボタン / メニューバー	ボタン・メニュー	A061 ~ A071	
閉じる (ボタン / メニュー)	割込み	BREK	
ダブルクリック	送信	ENTR	
自動送信	送信	ENTR	
MCR	MCR	標準なし (初期クリア文字を通知)	
設定した待機時間に達したとき	送信	ENTR	
フィールド単位で戻ったとき	送信	ENTR	
イベント単位で戻ったとき	イベント	標準なし (論理マップ中のイベント定数を通知)	

## 1.5.10 表示先と表示環境の設定

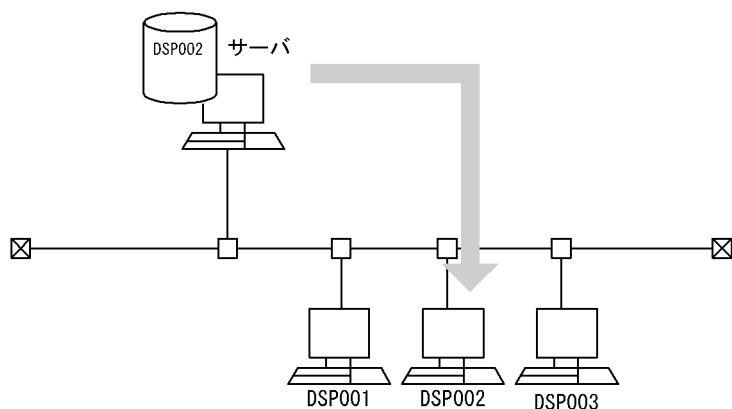
### (1) 表示先の指定

AP から画面に表示するときは、表示先を指定します。この表示先の名前を「仮想端末名」といいます。標準値は「DSP001」です。

スタンドアロンの場合は、自分自身の画面のウィンドウが表示先になります。

C/S システム構成の場合は、サーバから見た表示先の PC を指定します。C/S システム構成では、この物理的な LAN アドレスと、論理的なアドレスである「仮想端末名」をあらかじめ定義しておきます。なお、C/S システム構成の場合、仮想端末の自動割当て機能も使用できます。仮想端末設定および仮想端末自動割当ての詳細については、マニュアル「XMAP3 開発・実行ガイド」を参照してください。

図 1-15 表示先



### (2) 表示環境の設定

(1) で説明した仮想端末名の定義のほかに、表示先として使用する PC の属性を設定しておくことができます。設定できる主な属性を次に示します。

- 使用するキーボードやキーボードの機能の設定
- 画面のデザインの設定
- 日本語入力の設定
- 表示色の設定
- 表示文字の設定

これらの属性設定を「表示・印刷セットアップ」といいます。表示・印刷セットアップの詳細については、マニュアル「XMAP3 開発・実行ガイド」を参照してください。

## 1.6 XMAP3 関連ファイルの関係

ドローで使用するファイルと XMAP3 の機能の関連を次の図に示します。また、ドローで使用するファイルとその説明も次の表に示します。また、XMAP3 で使用するファイルと機能の関連については、マニュアル「XMAP3 開発・実行ガイド」を参照してください。

図 1-16 XMAP3 で使用するファイルと XMAP3 の機能の関連

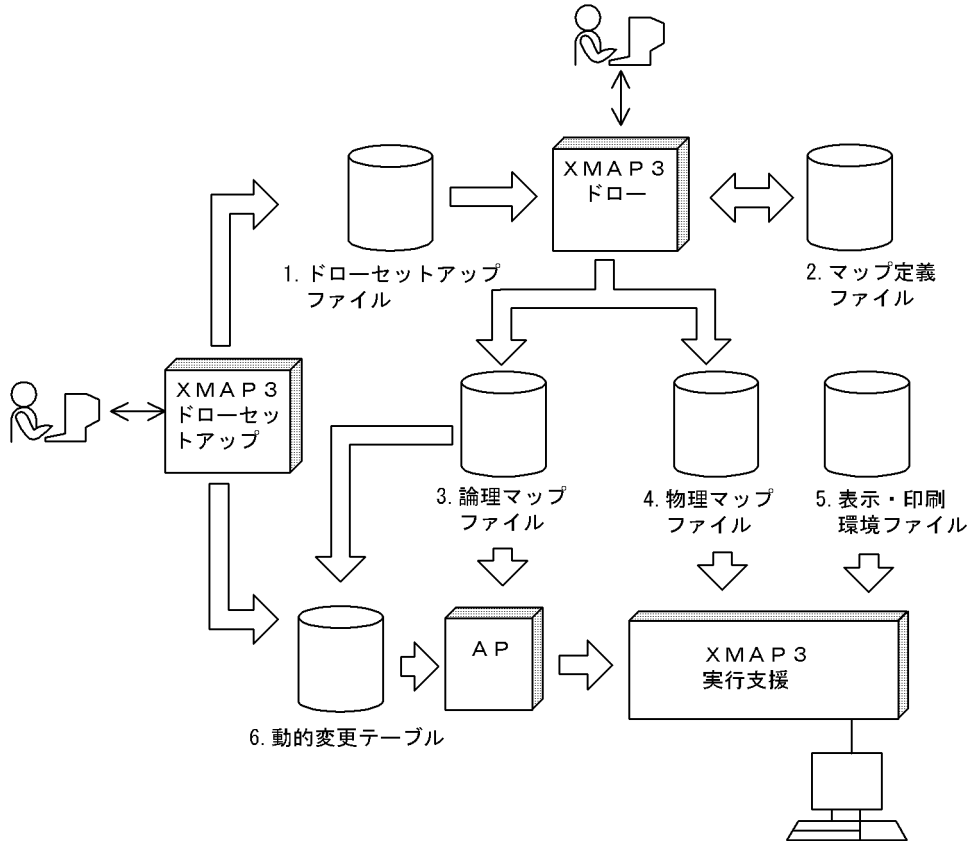


表 1-5 XMAP3 で使用するファイルとその説明

ファイル名	内容	区分	拡張子
1. ドローセットアップファイル	ドロースセットアップで設定した結果を格納	入出力	-
2. マップ定義ファイル	画面定義のソースを格納	入出力	.IMP
3. 論理マップファイル	入力論理マップファイル 出力論理マップファイル	入出力	.CBL .H

## 1. 画面機能の概要

ファイル名	内容	区分	拡張子
4. 物理マップファイル	物理マップファイル	出力	.PMP
5. 表示・印刷環境ファイル	印刷モードやプリンタ種別などの情報を格納	入力	-
6. 動的変更テーブル	ドローセットアップで設定した修飾名のテーブルを格納	入出力	.CBL .H

(凡例)

- : 該当しない。

注

これらのファイルは、修正する前にバックアップを取っておくことをお勧めします。  
「XMAP3Backup」などの別フォルダ(またはディレクトリ)を作成し、その中に対象のファイルをコピーしてください。インストール時の状態に戻りたいときは、フォルダ中の対象ファイルを XMAP3 インストール先に上書きコピーしてください。

## 1.7 画面を設計する前に

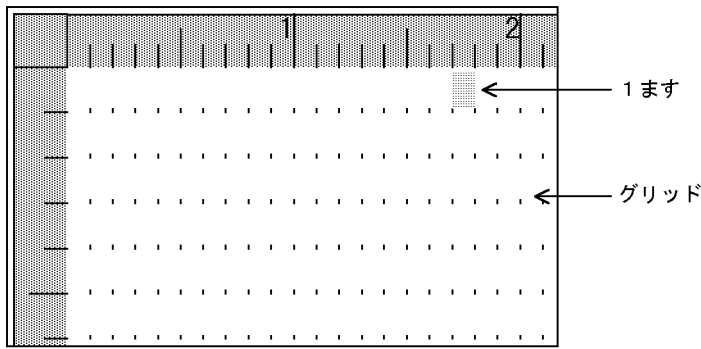
### 1.7.1 画面設計で使用する基本用語の説明

XMAP3の画面設計でよく使用する用語について説明します。ここで説明していない用語については、「付録 F 用語解説」を参照してください。

#### (1) レイアウト領域の設定に関する用語

##### グリッド

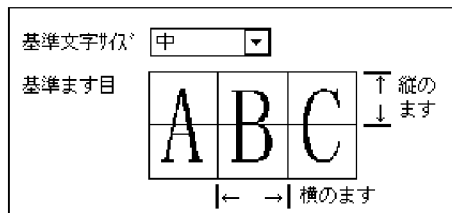
レイアウト領域にオブジェクトを配置する位置を決める基準になります。



##### 基準文字サイズ

画面のレイアウト領域の基準となる文字サイズです。標準サイズと、標準よりも大きいまたは小さいサイズが選択できます。

画面のレイアウト領域でオブジェクトを配置するときの単位となるグリッド（ます目）は、この基準文字サイズが基準になります。行方向は基準文字サイズの半分、列方向は半角文字の幅でそれぞれグリッドの縦幅／横幅が決まります。各オブジェクトでは、この基準文字サイズが適用されます。基準文字サイズを定義する画面属性ダイアログの一部を次に示します。



### 1.7.2 オブジェクト配置位置の基準

ドローでオブジェクトを配置する位置は、オブジェクトの属性定義によって異なります。

## 1. 画面機能の概要

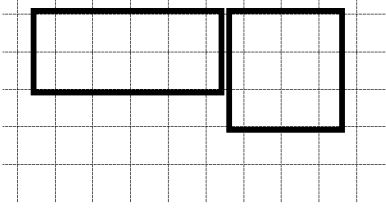
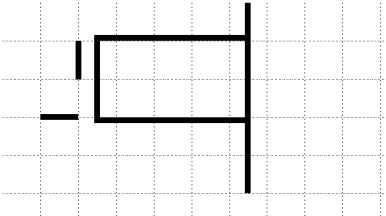
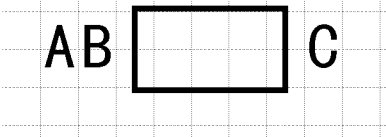
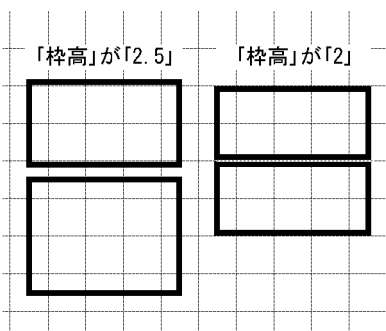
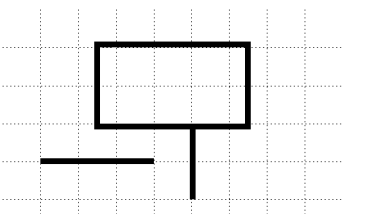
### (1) ます目とオブジェクト

ます目に対してオブジェクトがどのように配置されるかを示します。

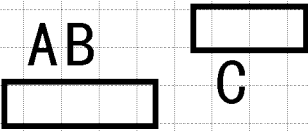
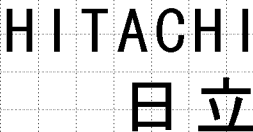
オブジェクト	ます目との関係	表示される状態
セパレータ, けい線系	<ul style="list-style-type: none"> <li>横けい線の場合, ます目の上, 中央, 下に配置される。</li> <li>縦けい線の場合, ます目の左, 中央, 右に配置される。</li> </ul>	
テキスト系	<p>半角文字は, ます目に対して横 1 ます, 縦 2 ます。全角は横 2 ます, 縦 2 ます。倍角は全角の 2 倍になる。文字はこの領域の中央に表示される。</p>	
枠付きのオブジェクト	<p>ボックス枠があるオブジェクトの配置は, オブジェクトの「枠高」での指定に依存する。</p> <ul style="list-style-type: none"> <li>「2.5」や「3」のときは, ボックス枠の横線はます目の外側に, 縦線はます目の中央に位置づけられる。</li> <li>「2」のときは, ボックス枠の横線はます目の内側に, 縦線はます目の中央に位置づけられる。</li> </ul> <p>スクロールバーが付く場合は, ボックス枠の外側にバー用のボックスが付く。バー用のボックスの大きさは横 1 ます, 縦 2 ます。</p>	<p>「枠高」が「2.5ます」「枠高」が「2ます」</p> <p>スクロールバー付き</p>

### (2) オブジェクト同士の位置関係

オブジェクト同士を並べて配置したり, 隣接して配置するときの規則を示します。ただし, セパレータ以外のオブジェクト同士を重ねて配置することはできません。

横 / 縦	オブジェクト	配置位置	表示状態
横	枠付きボックス同士	同じますに配置する。	
	枠付きボックスとセパレータ	同じますに配置する。	
	枠付きボックスと文字列	次のますに配置する。	
縦	枠付きボックス同士 <ul style="list-style-type: none"> <li>「枠高」が「2ます」の枠付きボックス同士</li> <li>「枠高」が「2.5ます」または「3ます」の枠付きボックス同士</li> </ul>	<ul style="list-style-type: none"> <li>1ます空けて配置する。</li> <li>直後のますに配置する。</li> <li>1ます空ける。</li> </ul>	
	枠付きボックスとセパレータ	<ul style="list-style-type: none"> <li>縦セパレータは枠付きボックスの配置されているますの直前、直後のますに配置する。</li> <li>横セパレータは枠付きボックスの配置されているますから1ます空けて配置する。</li> </ul>	

## 1. 画面機能の概要

横 / 縦	オブジェクト	配置位置	表示状態
	枠付きボックスと文字列	直後のますに配置する。	
	文字列同士	直後のますに配置する。	

### 1.7.3 画面設計と定義

画面を定義する前に確認する項目について説明します。

#### (1) ハードウェア依存およびドライバ依存の確認

ハードウェアやドライバ依存に関する設定は、実際の定義を始める前に確認しておくことをお勧めします。また、C/S システム構成の場合には、各クライアントでの設定を合わせておくことをお勧めします。

次に確認する項目を示します。

- キーボードの種類別
- 画面の解像度
- 画面単位での基準文字サイズ（大 / 中 / 小）
- オブジェクトの外観（2D（平面的に表示） / 3D（立体的に表示））
- 表示色（256色以上 / 16色）

16色の場合には背景色が使用できず、また、3D表示ができないため、文字色の工夫が必要になります。なるべく256色以上のディスプレイドライバを用意してください。

注

画面属性ダイアログで指定する基準文字サイズです。ドロー上で文字を編集するときの基準のサイズになります。

#### (2) レイアウト定義前の決定事項

レイアウトを定義する前に次のことを決めておいてください。

- ディスプレイの物理サイズに見合った画面サイズ



使用する PC の中で、「画面解像度のいちばん低いものの行数×列数」の値を物理サイズの最大値とし、画面サイズはこの値を超えないように設計する。

- 画面全体の文字サイズ（大きい文字 / 標準文字サイズ / 小さい文字）
- オブジェクトの概観（2D（平面的に表示） / 3D（立体的に表示））
- 一次ウィンドウと二次ウィンドウの区別  
二次ウィンドウは、一次ウィンドウに対しての補助的な表示をする場合に使用する。  
例えば、データエントリ画面やメッセージ画面、ヘルプ画面などを使用する場合には、二次ウィンドウを使用する。
- 使用するオブジェクトおよびオブジェクト配置のデザイン
- AP から変更できる表示属性
- PF キーやボタンの使用基準



# 2

## GUI 画面の作成

GUI 画面のデザインは，XMAP3 のドローを使用して設計します。この章では，GUI 画面の設計で使用できる機能について説明します。

---

2.1 GUI 画面の設計

---

2.2 GUI 画面のレイアウト手順

---

2.3 GUI 画面属性の設定

---

2.4 GUI 画面を構成するオブジェクト

---

2.5 部品の利用

---

2.6 データ名の編集

---

2.7 テスト表示

---

## 2.1 GUI 画面の設計

基幹業務では、大量の画面を扱わなければなりません。そのような場合、画面を目的別や性質別にまとめたり、操作方式や画面形式を統一したりすることで、作業の煩雑さや誤操作を防げます。また、見た目にも画面の違いがわかるようバリエーションに富んだ画面を設計することで、作業の効率化が図れます。

XMAP3 での GUI 画面設計ではボタンやリスト、グラフィックなどのオブジェクトを利用して、多彩な表現を実現できます。ここでは、GUI 画面のウィンドウ構成や GUI 画面を構成するオブジェクトについて説明します。

### 2.1.1 GUI 画面の構成

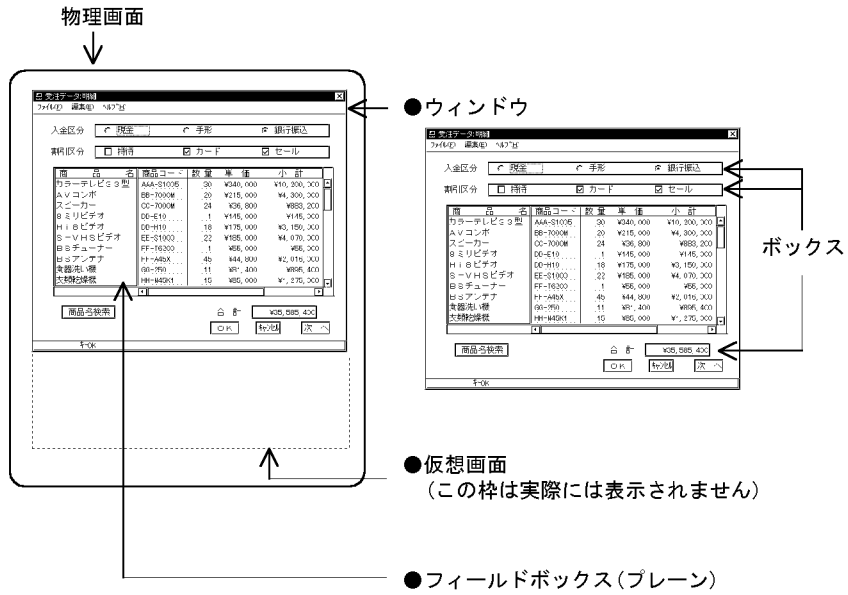
GUI 画面は、次に示す 2 種類の画面、および 2 種類の平面で構成されています。GUI 画面を構成する画面と平面の種類を次の表に示します。

表 2-1 GUI 画面を構成する画面と平面の種類

画面, 平面		意味
画面	物理画面	ディスプレイ画面上に表示される画面
	仮想画面	ウィンドウを配置する論理上の表示画面
平面	ウィンドウ	実際のレイアウト領域に表示する、各オブジェクトを配置する平面
	プレーン領域	ウィンドウ上のボックスに表示する各種のオブジェクトを配置する平面

各画面の関係を次の図に示します。

図 2-1 物理画面，仮想画面，ウィンドウおよびブレンの関係



●ウィンドウ

●仮想画面  
(この枠は実際には表示されません)

●フィールドボックス (ブレン)

商 品 名	商品コード	数量	単 価	小 計
カラーテレビ33型	AAA-S1005	20	¥340,000	¥10,200,000
A/Vコンボ	BB-7000M	20	¥215,000	¥4,300,000
スピーカー	CC-7000M	24	¥36,800	¥883,200
8ミリビデオ	DD-E10	.1	¥145,000	¥145,000
H i 8ビデオ	DD-H10	.18	¥175,000	¥3,150,000
S-VHSビデオ	EE-S1000	22	¥185,000	¥4,070,000
B Sチューナー	FF-I6200	.1	¥55,000	¥55,000
B Sアンテナ	FF-A45X	45	¥44,800	¥2,016,000
食器洗い機	GG-250	11	¥81,400	¥895,400
衣類乾燥機	HH-N45K1	15	¥85,000	¥1,275,000

各画面，および平面についての説明を次に示します。

(1) 物理画面

物理画面は，実際のディスプレイ画面のことで。各ウィンドウは，最終的にこの物理画面上に表示されます。

(2) 仮想画面

仮想画面は，物理画面から独立した，論理上の表示画面のことで。

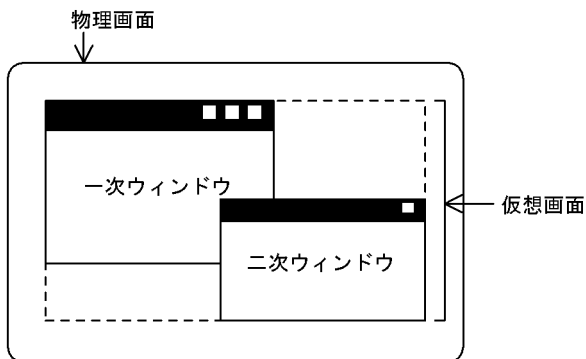
実際の画面表示では，仮想画面から物理画面上に表示できる範囲を PC 上のウィンドウ表示システムが切り取って表示します。ただし，仮想上の枠線などの境界は表示されません。

仮想画面のサイズは，指定がない場合，画面サイズの最大値 (48 行 × 160 列) を仮定します。このため，画面レイアウト上，仮想画面は特に意識する必要はありません。なお，GUI 画面は半行単位で定義するため，指定値は最大 96 行となります。

ただし，物理画面の範囲をはみ出すようなウィンドウは配置できないので，ウィンドウのサイズおよび位置は，物理画面内に収まるように配置してください。

## 2. GUI 画面の作成

画面レイアウト時には、この仮想画面上に一次ウィンドウ、および二次ウィンドウを配置します。

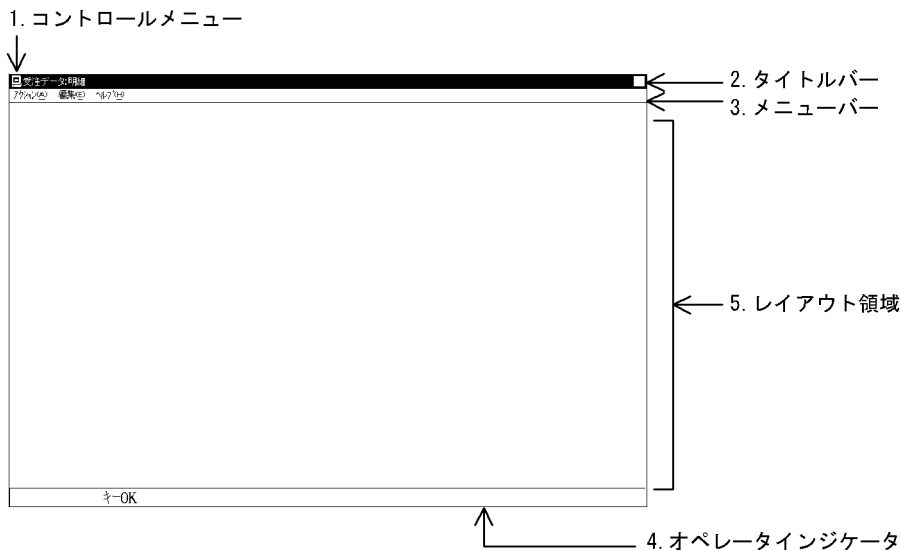


### (3) ウィンドウ

ウィンドウは、仮想画面上の領域内に各種のオブジェクトを配置するための平面です。

実際の画面レイアウト時には、このウィンドウ内へのオブジェクトの配置を考慮することが主となります。オブジェクトは、種類によって配置する平面が異なります。エントリ系、ラベル系オブジェクトは直接ウィンドウ上に配置します。リストボックスとフィールドボックスなどは、ウィンドウ上には表示位置や大きさだけを配置し、内容はプレーン領域に配置します。

図 2-2 ウィンドウ各部の名称



#### 1. コントロールメニュー

コントロールメニューとは、画面の左上にある枠をマウスでクリックすると表示され

るシステムメニューです。タイトルバー左端にカーソルを位置づけ、マウスの左ボタンを押すと表示されます。コントロールメニューをクリックすると、Windows のシステムメニューが表示されます。システムメニューには、[元のサイズに戻す][移動][サイズ変更][最小化][最大化]および[閉じる]があります。[閉じる]メニューでAPに割込キーのイベントを返すかどうかは、表示・印刷セットアップで指定できます。設定方法の詳細についてはマニュアル「XMAP3 開発・実行ガイド」を参照してください。また、イベントについては「1.5.9 画面確定とAPへの通知」を参照してください。

## 2. タイトルバー（オプション）

そのウィンドウに対応する業務の名称を表示します。一つの業務が二次ウィンドウを含め複数の画面から構成される場合は、画面での処理や、画面遷移を考慮してタイトルを付けます。タイトルは左側に表示されます。

タイトルは、画面属性ダイアログで設定します。タイトルの長さは、最大で78文字（半角の場合）です。タイトルを省略した場合は、「XMAP3」が仮定されます。APからは変更できません。

なお、スプラッシュ画面など、タイトルバーやメニューバーを表示しない画面の場合は、画面属性ダイアログで「タイトルバーを付ける」の項目をチェックしません。

（例）

「受注処理業務：データ入力」、「受注処理業務：データ検索」

タイトルはドロウ定義時に指定しますが、タイトルの表示は画面属性ダイアログで指定します。

## 3. メニューバー

画面の業務に対して、各種操作をするメニューを表示します。表示する文字列、表示属性はドロウで指定します。メニューバーは必要に応じて使用します。なお、スプラッシュ画面など、タイトルバーやメニューバーを表示しない画面の場合は、画面属性ダイアログで「メニューバーを付ける」の項目をチェックしません。

## 4. オペレータインジケータ（オプション）

キーボード状態、入力データのチェック結果などのメッセージをXMAP3が表示します。APからのメッセージの表示はできません。オペレータインジケータを表示するかしないかは画面属性ダイアログで指定します。表示位置は、XMAP3の表示・印刷セットアップで変更します。表示は、「上に表示」「下に表示」「表示しない」および「拡張形式」が選択できます。なお、OLTPサーバ構成の環境で使用する場合には、「上に表示」または「下に表示」のどちらかを選択します。

注

拡張形式では、オペレータインジケータは常に下に表示されます。

## 5. レイアウト領域

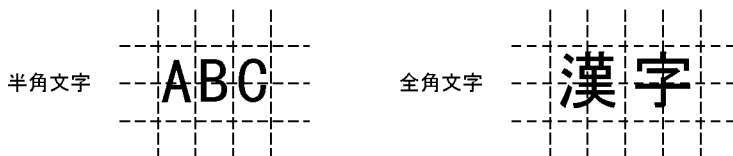
実際の業務処理に使用するオブジェクトを配置します。

これ以降、表示レイアウトなどの説明での「作業領域」は、断りがないかぎりレイアウト領域を示します。

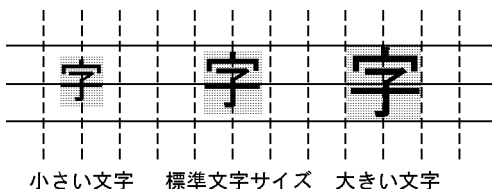
## 2. GUI 画面の作成

### 《レイアウト領域のグリッド》

グリッドのます目の最小単位は、画面属性ダイアログで指定した文字サイズ「標準文字サイズ」を基準にして決まります。横が半角 1 文字分です。GUI 画面での文字とます目の関係について次に示します。



なお、レイアウト領域で表示されている文字に対し、さらに「大きい文字」「小さい文字」「標準文字サイズ」を指定できます。それぞれの指定をしたときの表示を次に示します。なお、ここでは全角文字の場合を例にします。



■ : 文字が占める範囲を示します。

#### • ますの高さ

GUI 画面の場合、ますの高さは標準文字の高さの半分になります。ただし、フィールドボックス中では、ますの高さは文字の高さと同じになります。このように、オブジェクトをどこに配置するかによって、レイアウトに必要な領域長が異なります。

オブジェクトを配置する場所とますの高さの関係を次の表に示します。

表 2-2 オブジェクトを配置する場所とますの高さの関係

オブジェクトを配置する場所	ますの高さ
フィールドボックス以外	半角文字 1 / 2 文字分
フィールドボックス内	半角文字 1 文字分

### (4) プレーン領域

プレーン領域は、ウィンドウ上に配置したボックス内にけい線やフィールドを配置するための平面です。

プレーンサイズは、対応するボックスのボックスサイズと同じか、大きいサイズを指定できます。画面表示上は、ボックスに表示できる範囲を PC の表示システムが切り取っ



て表示します。ボックスサイズの範囲外の部分は、スクロールすることで表示されます。

プレーンサイズがボックスサイズよりも大きくなる場合、プレーン上に配置するオブジェクトは、別画面としてレイアウトします。このとき、切り取られてボックス内に表示される部分が、ウィンドウ上のほかのオブジェクトと関連する場合はボックスとオブジェクトの重なり範囲を意識してボックス内のオブジェクトを配置する必要があります。

## 2.1.2 画面サイズ

ディスプレイ画面の解像度（ドット数）、または表示する文字サイズの大きさによって表示できる行数、および列数が異なります。また、各ウィンドウに分けて表示できるため、実質的に表示する画面サイズは一定ではありません。

代表的な画面解像度ごとの文字サイズを表 2-3 に、画面サイズを表 2-4 に示します。

表 2-3 画面解像度ごとの文字サイズ

《2Dの場合》単位：ピクセル					
XMAP3の画面属性での基準文字サイズ	各オブジェクトでの文字サイズ	640 × 480	800 × 600	1,024 × 768	1,280 × 1,024
大	大	28	28	38	38
	標準	24	24	32	32
	小	18	18	24	24
中	大	18	18	28	28
	標準	16	16	24	24
	小	12	12	18	18
小	大	14	14	18	18
	標準	12	12	16	16
	小	8	8	12	12

《3Dの場合》単位：ピクセル					
XMAP3の画面属性での基準文字サイズ	各オブジェクトでの文字サイズ	640 × 480	800 × 600	1,024 × 768	1,280 × 1,024
大	大	26	26	26	26
	標準	22	22	22	22
	小	16	16	16	16
中	大	16	16	16	16
	標準	14	14	14	14
	小	10	10	10	10
小	大	14	14	14	14

## 2. GUI 画面の作成

《3D の場合》単位：ピクセル

XMAP3 の画面属性での基準文字サイズ	各オブジェクトでの文字サイズ	640 × 480		800 × 600		1,024 × 768		1,280 × 1,024	
		列	行	列	行	列	行	列	行
	標準	12		12		12		12	
	小	8		8		8		8	

注

10 ピクセル以下の文字は、小さ過ぎて見にくい場合があります。あらかじめご確認ください。

表 2-4 画面解像度ごとの画面サイズ

《2D の場合》単位：ドット

XMAP3 の画面属性での基準文字サイズ	各オブジェクトでの文字サイズ	640 × 480		800 × 600		1,024 × 768		1,280 × 1,024	
		列	行	列	行	列	行	列	行
大	大	45	17	57	22	53	21	67	29
	標準	53	17	66	22	64	21	80	29
	小	71	17	88	22	85	21	106	29
中	大	71	25	88	32	73	28	91	38
	標準	80	25	100	32	85	28	106	38
	小	106	25	133	32	113	28	142	38
小	大	91	32	114	41	113	41	142	48
	標準	106	32	133	41	128	41	160	48
	小	160	32	160	41	160	41	160	48

《3D の場合》単位：ドット

XMAP3 の画面属性での基準文字サイズ	各オブジェクトでの文字サイズ	640 × 480		800 × 600		1,024 × 768		1,280 × 1,024	
		列	行	列	行	列	行	列	行
大	大	49	17	61	22	78	28	98	38
	標準	58	17	72	22	93	28	116	38
	小	80	17	100	22	128	28	160	38
中	大	80	25	100	32	128	41	160	48
	標準	91	25	114	32	146	41	160	48
	小	128	25	160	32	160	41	160	48
小	大	91	29	114	36	146	47	160	48

《3Dの場合》単位：ドット

XMAP3の画面属性 での基準文字サイズ	各オブジェクト での文字サイズ	640 × 480		800 × 600		1,024 × 768		1,280 × 1,024	
		列	行	列	行	列	行	列	行
	標準	106	29	133	36	160	47	160	48
	小	160	29	160	36	160	47	160	48

## 注 1

この表の値の単位は、XMAP3の標準フォントサイズ（単位：ドット）を使用したレイアウト領域で表示できる列数×行数です。行数は、オペレーティングータ分を含みません。この表の行列サイズは最大値であり、画面定義上は、この値では制限されません。ウィンドウシステムの設定（システムフォントサイズなど）によってこの値より小さな値になることがあります。

## 注 2

画面上にオペレーティングータを表示する場合、およびメニューバーを表示する場合は、画面のサイズは変わりませんが、ユーザが使用できる領域はそれぞれの場合、1行ずつ減ります。例えば、オペレーティングータ、メニューバーをどちらも表示していないときの行数が「17」の場合、オペレーティングータまたはメニューバーのどちらかを表示すると「16」に、どちらも表示させると「15」になります。

表示できる画面行列数の計算方法を次に示します。

## 《行数の計算方法》

画面行ドット数を求める計算式

画面行ドット数 =

$$(\text{フォントサイズ(ドット)} + 2(\text{または}4)) \times (\text{ウィンドウ行数} + 1(\text{1行単位})) + \text{タイトルバーの高さドット数} + \text{メニューバードット数}$$

## 計算式の説明

- ・フォントサイズに加える値が4となるのは、表示・印刷セットアップで3D表示を指定し、ディスプレイが256色以上の表示の場合だけである。
- ・タイトルバーの高さドット数とメニューバードット数は、Windowsが表示をするため、使用する標準フォントやディスプレイドライバによってPCごとに異なる。
- ・メニューバーは、1行に収まらない場合、複数行で表示する。また、メニューバーを指定しない場合は、行ドット数に含まれない。

必要な画面ドット数を求める場合、通常はそれぞれ1.5行分のドット数が必要なものとして計算することをお勧めします。

画面ドット数から表示できる行数を求める場合、この計算式を逆算します。この場合、行数は整数値なので、端数は切り捨てます。

## 《列数の計算方法》

画面列ドット数を求める計算式

画面列ドット数 =

$$(\text{フォントサイズ(ドット)} / 2) \times \text{ウィンドウ列数(半角文字数)}$$

画面ドット数から表示できる文字数（半角）を求める場合、この計算式を逆算します。この場合、文字数は整数値なので端数は切り捨てます。また、表示できる全角文字数を求めるには、「表示できる半角文字数 / 2」の値を偶数値となるように値を丸めます。

### 2.1.3 画面の表示様式

XMAP3 で表示する画面の表示環境は、仮想端末ごとに設定できます。AP では、仮想端末名で表示環境を区別しているため、表示環境を動的に変更することはできません。XMAP3 では、物理画面上の複数の仮想端末に対して、いちいちオープン・クローズをしなくても、一時的に非表示にしたり、ウィンドウの位置を設定したりできます。

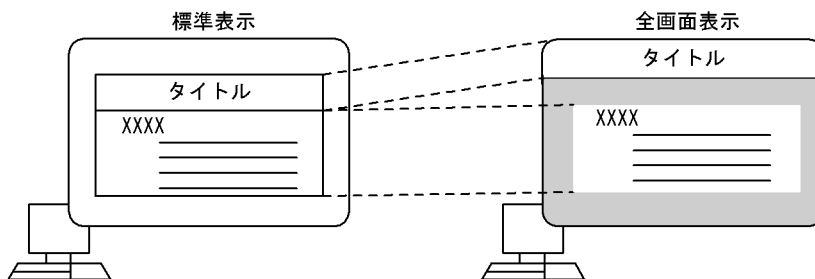
#### (1) 表示様式

「全画面表示」および「ウィンドウ表示」は表示・印刷セットアップで設定します。

##### 全画面表示

全画面表示は、一次ウィンドウにだけ設定できます。全画面表示を設定すると、タスクバーを除くディスプレイの全領域にウィンドウが表示されます。全画面表示時には次の点に注意してください。

- タスクバーの「自動的に隠す」を指定している場合に全画面表示を設定すると、タスクバーを表示できなくなる。「自動的に隠す」を指定しないか、またはマウス移動や [Ctrl] + [Esc] の押下でスタートメニューを表示させ、「自動的に隠す」の指定を解除すると、タスクバーは表示される。
- 画面属性ダイアログで「ウィンドウのサイズ変更を抑止する」をチェックしていても、画面のリサイズはできない。また、画面属性ダイアログの [ウィンドウサイズを変更する] のチェックが外れている状態で、表示・印刷セットアップの「デザイン 1」タブの [全画面表示する] をチェックしていても、画面のリサイズはできない。
- ウィンドウの移動はできない。
- 二次ウィンドウに対しては、全画面表示の設定はできない。
- 表示・印刷セットアップで設定する。
- 動的変更はできない。
- ドロー上での定義は、標準表示と変わらない。
- 壁紙は、ドローで定義したウィンドウの大きさまで表示される。
- 物理画面（ディスプレイ）の中央位置を基準に物理画面がウィンドウ枠より大きいときだけ、背景を引き伸ばし全画面化する。また、タイトルは画面の上部に表示する。



網掛けの部分は伸ばされた背景を示します。

### ウィンドウ表示

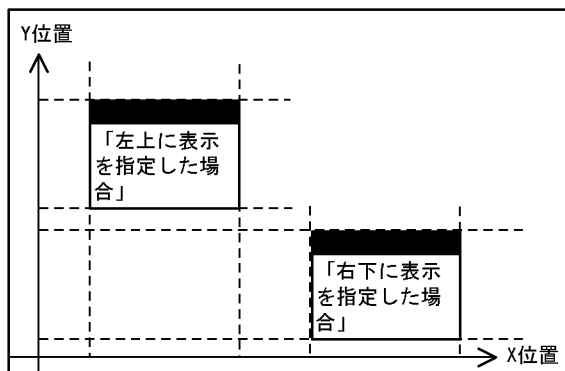
標準的な使い方です。ウィンドウサイズに従って表示されます。ウィンドウサイズは変更でき、小さく変更する場合は必要に応じてスクロールバーが付きます。それぞれのウィンドウの表示位置を指定しておきます。位置の指定には縦横（XY位置）を指定して物理画面上での平面的な位置を指定する方法と、奥行き（Z位置）を指定してウィンドウ同士の重なりを指定する方法があります。これらの指定はAPから動的に変更できます。

## (2) ウィンドウ表示位置と重なり

ウィンドウ表示位置およびウィンドウ同士の重なりは「XY位置」および「Z位置」で設定します。ただし、APでの「XY位置」および「Z位置」の設定が有効になるのは、表示形態で「全面書換」を指定している場合だけです。

### XY位置

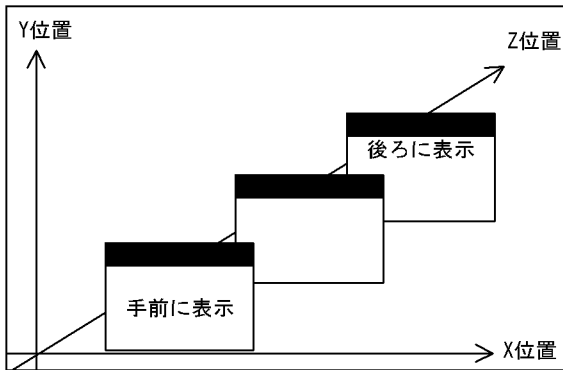
XY位置は「左上」「左下」「中央」「右上」「右下」「自由な位置」から選べます。



### Z位置

Z位置は「標準表示」「手前に表示」「後ろに表示」「一時非表示」から選べます。「一時非表示」は奥行きの指定に関係なく、一時的にウィンドウを隠します。なお、二次ウィンドウの場合は、標準表示だけとなります。

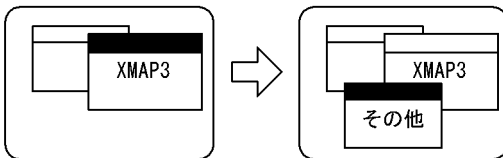
## 2. GUI 画面の作成



複数のウィンドウを表示するときのウィンドウ同士の重なりを、奥行きを示す Z 位置で指定できます (Z オーダー)。

- 標準表示

表示するといちばん手前に表示されます。ほかの AP のウィンドウが表示されると、後ろへ移動されます。

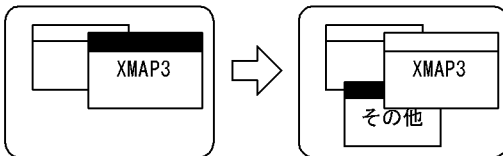


XMAP3のウィンドウを表示したとき

ほかのAPのウィンドウを表示したとき

- 手前に表示

活性 / 不活性に関係なく、常に手前に表示されます。

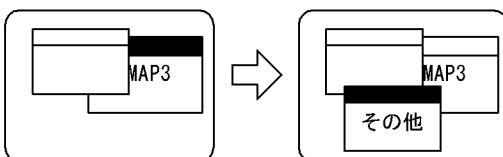


XMAP3のウィンドウを表示したとき

ほかのAPのウィンドウを表示したとき

- 後ろに表示

活性 / 不活性に関係なく、常に後ろに表示されます。

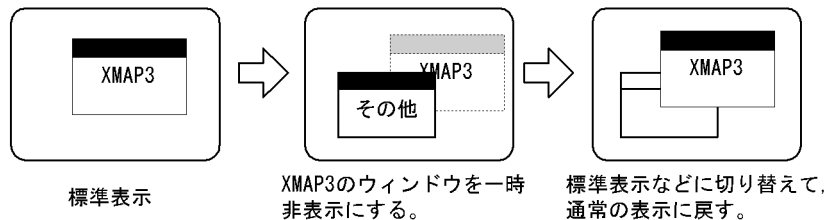


XMAP3のウィンドウを表示したとき

ほかのAPのウィンドウを表示したとき

- 一時非表示

ウィンドウを見えない状態にします。通常の見え表示にするには、AP から動的変更をして「標準表示」などに切り替えます。一時非表示を利用すると、一時的にウィンドウをクローズしたような状態を作れます。なお、一時非表示の場合、送信キーを押したときと同じイベント通知コードが、自動的に AP に渡されます。



### (3) 表示様式の関係

画面の表示様式とウィンドウの関係を示します。

表示様式	ウィンドウの定義			
	一次ウィンドウ	二次ウィンドウ	XY 位置	Z 位置
全画面表示	様式の定義は有効	様式の定義は無効	×	
ウィンドウ表示	様式の定義は有効	様式の定義は有効		

(凡例)

○ : 定義できる。

× : 定義できない。

## 2.1.4 レイアウト領域の背景色

レイアウト領域には背景色が指定できます。背景色には、標準色として「グレー」が仮定されています。また、背景として壁紙を選ぶこともできます。壁紙の詳細については、「2.1.6(2) 壁紙の利用」を参照してください。

## 2.1.5 一次ウィンドウと二次ウィンドウ

一次ウィンドウは、データ入力のために使用する画面です。これに対し二次ウィンドウは、ヘルプやメッセージを表示するために使用する画面です。二次ウィンドウは同時に3個まで、一つの一次ウィンドウ上に重ねて表示できます。なお、二次ウィンドウ表示は、一次ウィンドウが表示されていることが前提になります。

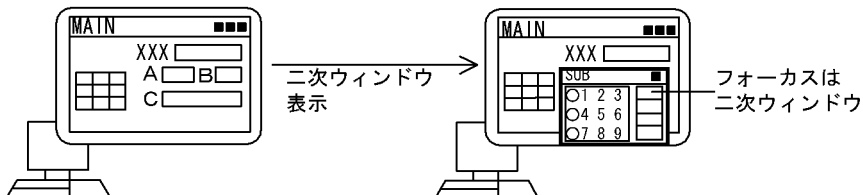
一次ウィンドウおよび二次ウィンドウの表示位置や重なって表示される場合の前後関係は、画面属性ダイアログで設定します。

複数のウィンドウを表示しているとき、オペレータが画面操作できるのは、活性状態の

## 2. GUI画面の作成

ウィンドウだけです。操作できる状態にあることをフォーカスがあるといいます。  
ウィンドウの代表的な表示規則を次に示します。

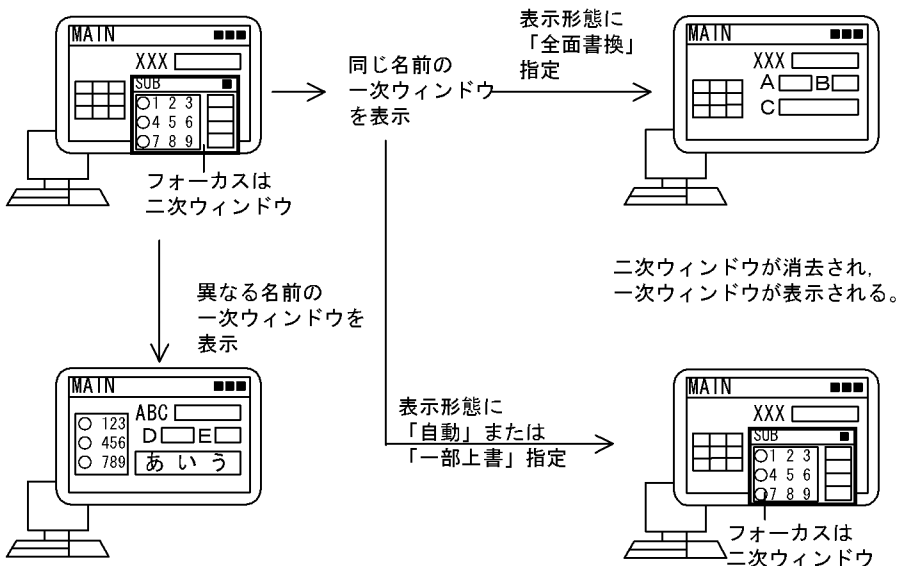
### (1) 一次ウィンドウから二次ウィンドウの表示



＜ 二次ウィンドウの表示位置を仮定した場合の二次ウィンドウ表示 ＞

- 一次ウィンドウ定義中に同じウィンドウIDの二次ウィンドウ定義がある場合 → 対応するウィンドウIDを持つ一次ウィンドウ定義での二次ウィンドウの指定位置に指定した大きさで表示
- 一次ウィンドウ定義中に二次ウィンドウ定義なし、または同じウィンドウIDの二次ウィンドウの定義なしの場合 → 二次ウィンドウ定義の表示位置に従う

### (2) 複数ウィンドウ表示後に一次ウィンドウ表示

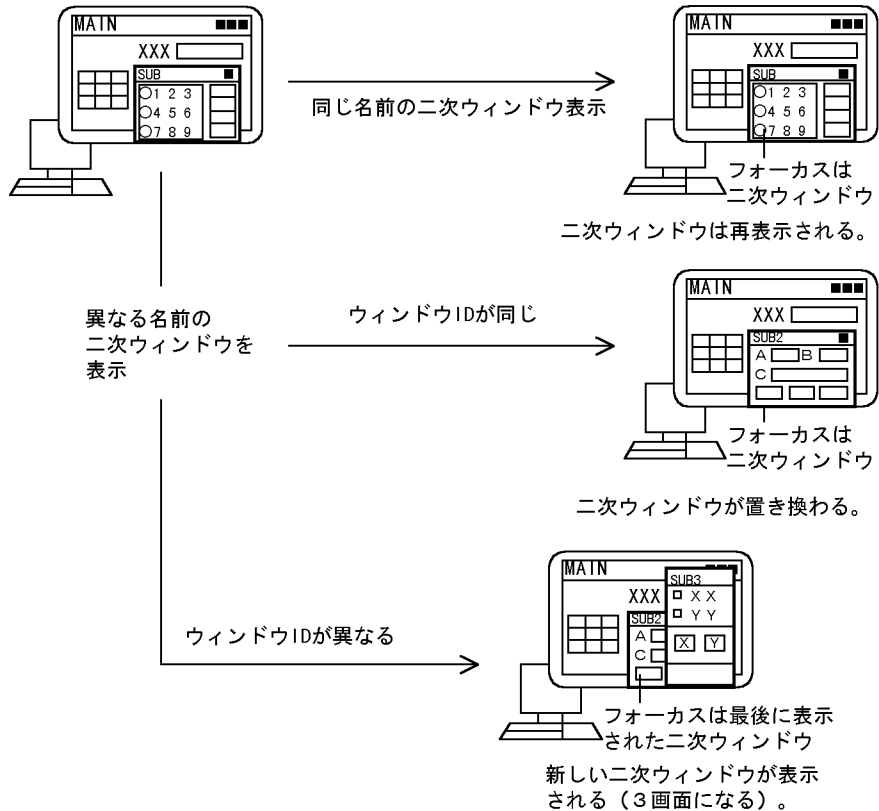


新しい一次ウィンドウが、指定した大きさで、前回の一次ウィンドウと置き換わる。

二次ウィンドウを残したまま、一次ウィンドウは再表示される。ただし、APでデータ受信処理の整合性が取れなくなるため、通常では使用しない。



## (3) 複数ウィンドウ表示後に二次ウィンドウ表示



## (4) 画面の表示フォント

- XMAP3 で表示に使用するフォントは、フォント構成ファイル (X3PFONT) の定義情報に従います。
- 画面の外観を 3D (立体的な表示) にした場合、標準では画面解像度に関係なく、MS ゴシックの 14 ドットフォントとなります。これ以外のフォントおよびサイズを使用したい場合は、表示・印刷セットアップの「表示文字」タブで設定してください。

## 2.1.6 GUI画面設計でこんなこともできます

## (1) スプラッシュ画面の作成

スプラッシュ画面では、メニューバーやオペレーティングデータを表示しません。スプラッシュ画面の場合、画面に対して何も操作がされなくても、あらかじめ設定した時間が過ぎた場合、または表示直後に自動的に AP に制御を渡せます。

スプラッシュ画面の詳細、および例題については「8.2.2 BTMENU01 の AP 実行前に表示するスプラッシュ画面の追加」を参照してください。

## 2. GUI 画面の作成

なお、OLTP サーバ構成用のプログラムでは、スプラッシュ画面を適用できません。

### (2) 壁紙の利用

定義するウィンドウごとに、壁紙を設定できます。壁紙はビットマップファイル (.bmp) です。壁紙とレイアウト領域の関係はを次に示します。

- 壁紙のサイズがレイアウト領域より小さい場合  
レイアウト領域の左上を原点にして、並べて表示します。
- 壁紙のサイズがレイアウト領域より大きい場合  
レイアウト領域の左上を原点にして表示します。レイアウト領域からはみ出た部分は切り捨てます。

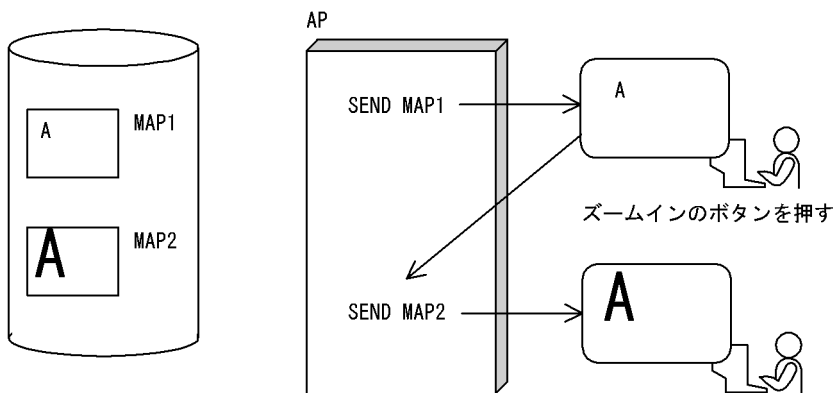
なお、ウィンドウ内に壁紙を使用する設定をしている場合に、全画面表示を設定しているときは、壁紙はドローで定義したウィンドウの大きさまでしか表示されません。

### (3) 掲示板の作成

ウィンドウ上にずっと表示しておく画面です。一定間隔で AP から表示内容を書き換えます。ユーザが画面確定の操作をしなくても、AP に制御を渡します。掲示板の詳細、および例題については「8.2.14 GENDSP01 パターンを使用した掲示板の表示」を参照してください。

### (4) 画面のズームイン、ズームアウトの作成

画面属性ダイアログで指定する基準文字サイズ(大/中/小)に対し、ドロー上でさらに3種類の文字サイズを指定できます。つまり、ドロー上では9種類の文字サイズを使用して画面を作成できます。同じレイアウトで文字サイズだけを変更した数種類の画面を作成しておき、AP で切り替えると、画面をズームイン、またはズームアウトできます。



2種類の文字サイズのマップを用意しておく。

論理マップはMAP1とMAP2で同じなので共用する。

## 2.1.7 GUI画面設計のポイント

GUI画面の設計ではさまざまな機能が利用できることで、大量の画面を必要とする基幹業務などへの適応ができます。しかし、利用できる機能が多ければ多いほど、ポイントをしばって設計しなければ煩雑な画面になってしまいます。

ここでは、より快適に業務を行うための画面の設計のポイントについて説明します。

### (1) 使いやすいGUI画面の特長

使いやすいGUI画面の特長を次に示します。

- 操作方法が統一されている。
- 画面のサイズ、形式が画面の種類ごとに統一されている。
- 入力必須、選択必須の項目は、スクロールしなくても必ず表示される位置にある。
- 操作を画面上だけに集中させず、マウス操作やキー操作にも割り振るなどして、最も簡潔な方法で操作できるようになっている。
- 画面上で使用している色数が適当である。また、長時間作業していても目に負担をかけない色使いである。

### (2) GUI画面を設計するときの検討課題

GUI画面を設計するときには、次の点をよく検討します。

#### (a) 画面中の機能の絞り込み

一つのGUI画面には、まとまりのある機能を持たせます。最初に表示する画面にはなるべく汎用的な機能を持たせ、そこから二次ウィンドウやダイアログに制御を移していく方が、画面を効率良く使えます。

#### (b) 操作方式、画面形式の統一

操作方式や画面形式を統一すると、初めて操作する画面でもそれまでの経験を基に操作できるので、効率良く作業ができます。また、決まった機能を持つボタンの位置が画面間で統一されていれば、誤入力や誤操作を防げます。

#### (例)

画面を終了するときに表示されるダイアログのボタンの配置は、常に「はい」「いいえ」「キャンセル」の順である。

#### (c) 使用している色の考慮

GUI画面の特長である多彩な色を使いながら、見やすい画面にするには、次の点に注意してください。

- 目立たせたい項目を決め、背景色とは対立する色を指定する。それ以外は同系色でまとめる。
- 使用する色は統一する。  
(タイトルバーの色、背景色の色、エラーダイアログの色など)

## 2. GUI画面の作成

- 目が疲れないように、広範囲の色には彩度の低い色を使う。
- 文字色には必ず見やすい色を使う。
- モノクロの画面や液晶画面で表示されることも考慮する。
- 色弱者を考慮する（赤と緑の組み合わせを避けるなど）

具体的な画面での色使いについては、サンプル画面を参考にしてください。

### (d) ユーザレベルの考慮

熟練者でも初心者でも操作しやすいように配慮してください。

### (例)

初心者用のガイドを用意する。ただし、熟練者には必要ないので画面上に常駐させない。

### (e) PF キー、テンキー入力の利用

操作をすべて画面に集中させずに、今までのオペレーションとの連続性や手間などを考慮して機能を割り振ってください。

### (f) ハードウェアデバイスの影響

ブラウン管や液晶画面などのハードウェアデバイスによって、表示結果やサイズが異なる点を考慮してください。

### (g) 基幹業務で利用するということ（AP 生産性、保守性の配慮）

基幹業務は、大量の画面、AP から構成されています。また、常に変更・更新されます。これらの点を考慮して、生産性、保守性、信頼性について十分、検討してください。

### (h) CUI 画面での操作に対応する GUI 画面の機能

これまで CUI 画面上で操作していたものを GUI 画面に移行させる場合、各フィールドに対応する GUI の機能は次のようになっています。

#### キー入力するフィールド

CUI 画面でキー入力していた各フィールドは、目的に応じて次の分類のオブジェクトに割り振れます。各オブジェクトの詳細は「2.4 GUI 画面を構成するオブジェクト」を参照してください。

#### キーエントリしていたフィールドに対応する GUI 画面の機能

- 直接、入出力するもの  
「入出力テキスト・フィールド」「入出力日付テキスト・フィールド」「入出力時刻テキスト・フィールド」
- 選択して入出力するもの  
「コンボボックス」「ポップアップテキスト・フィールド」「スピンボックス」

#### 固定（ラベル）フィールドに対応する GUI 画面の機能

「固定テキスト・フィールド」「出力テキスト・フィールド」「出力日付テキスト・フィールド」「出力時刻テキスト・フィールド」

一覧を表示するフィールドに対応する GUI 画面の機能

「フィールドボックス」「リストボックス」「テキスト・フィールドの反復」

PF キーなど画面を確定し、AP に制御を渡して処理をコントロールするものに対応する GUI 画面の機能

- CUI でのファンクションキー操作をそのまま使いたい場合
  - 「GUI のボタンに PF キーや送信キーを割り振って使う」
  - 「GUI のボタンと PF キーを組み合わせる」「プッシュボタン」
- GUI 画面の機能を利用したい場合
  - 「メニューバー」「テンキー（自動送信と組み合わせる）」「ラジオボタン」
  - 「チェックボタン」「トグルフィールド」

## 2.2 GUI画面のレイアウト手順

---

実際にドローを起動し、画面上にオブジェクトを配置してレイアウト定義をします。ここでは、GUI画面を新規に作成するときのレイアウト手順について説明します。

GUI画面を新規に作成するときの手順を次に示します。

1. スタートメニューから [XMAP3] - [ドロー] を選び、ドローを起動する。  
新規・更新選択ダイアログが表示されます。
2. 新規・更新選択ダイアログで [対象] から「マップ」<sup>1</sup> を選択し、「画面・帳票・書式を新規作成する」を選択する。  
選択後、[OK] ボタンをクリックすると新規作成ダイアログが表示されます。
3. 新規作成ダイアログでマップ名を指定する。  
マップ名の決定の詳細については、「2.2.1 マップ名の決定」を参照してください。
4. 新規作成ダイアログで「定義種別」を選択する。  
「定義対象の選択」から定義種別を選択します。  
一次ウィンドウを作成する場合  
「GUI画面(一次)」を選択します。  
二次ウィンドウを作成する場合  
「GUI画面(二次)」を選択します。
5. 新規作成ダイアログで定義パターンを選択する。  
定義パターンを選択します。選択できる定義パターンには、XMAP3 が用意しているパターンまたはユーザが登録したパターンがあります。XMAP3 で提供している定義パターンについては、「付録B XMAP3 で提供している定義パターン」を参照してください。  
サイズを自由にするときには、「フリー」を選んで画面属性ダイアログで設定します。
6. 新規作成ダイアログで言語種別を選択する。  
COBOL または C のどちらかを選択します。  
なお、XMAP3/Web for Cosminexus を使用して Cosminexus 上で実行する GUI 画面を開発する場合で、AP 開発言語が Java のときは、C を選択してください。
7. [OK] ボタンをクリックする。  
画面定義ウィンドウが表示されます。
8. 画面定義ウィンドウで、画面属性を設定する。  
ウィンドウの表示形態やイベント通知コードなど、画面を制御する情報(画面属性情報)を定義します。[属性] - [画面属性] を選択して表示される画面属性ダイアログで値を設定します。
9. 画面レイアウトを定義する。  
画面定義ウィンドウのツールボックスからオブジェクトのボタンを選んで、レイアウト

ト領域にオブジェクトを配置し、画面をレイアウトします。オブジェクトについては、「2.4 GUI画面を構成するオブジェクト」を参照してください。

10. 定義内容を保存し、終了する。

画面定義ウィンドウのツールバーから [ ドローの終了 ] を選ぶと定義内容が保存され、画面定義が終了します。このとき、定義ファイルに加え、ドローセットアップで設定したターゲット用<sup>2</sup>の論理マップと物理マップが生成されます。

注 1

[ 対象 ] から [ パターン ] を選択した場合には定義パターンを、[ 部品 ] を選択した場合には部品を作成できます。定義パターンや部品の作成については、「2.2.2 定義パターンおよび部品の登録」を参照してください。

注 2

XMAP3/Enterprise Edition の場合、ドローセットアップのターゲットの設定で、次の指定をした場合に、論理マップと物理マップが出力されます。

- ・メインフレーム - PC 分散用の画面・帳票開発
- ・AIX (文字コード: シフト JIS) 用の画面・帳票開発
- ・AIX (文字コード: EUC) 用の画面・帳票開発
- ・HP-UX (文字コード: シフト JIS) 用の画面・帳票開発
- ・Solaris (文字コード: シフト JIS) 用の画面・帳票開発
- ・Solaris (文字コード: EUC) 用の画面・帳票開発
- ・Linux (文字コード: EUC) 用の画面・帳票開発
- ・Windows 用の画面・帳票開発

## 2.2.1 マップ名の決定

画面の定義情報と生成される物理マップや論理マップなどの名称の整合性を図るために、マップ名を設定します。このマップ名は、各ファイル名称の先頭の文字列（拡張子を除く文字列）などになります。

マップ名に MAP001 という名称を付けた場合のマップ定義ファイル名称  
MAP001.imp になります。

マップ名を MAP001 とした場合に生成される物理マップファイル名称  
マップ名が 6 文字 の場合、MAP001ND.pmp になります。マップ名が 7 文字 の場合は、MAP001O.pmp になります。

注

マップ名の 6 文字または 7 文字については、ドローセットアップで設定します。ドローセットアップについては、マニュアル「XMAP3 開発・実行ガイド」を参照してください。

## 2. GUI画面の作成

マップ名を MAP001 とした場合に、生成される論理マップファイル名称

- COBOL の場合

入力論理マップファイル名称は MAP001I.cbl, 出力論理マップファイル名称は MAP001O.cbl になります。

- C 言語の場合

入力論理マップファイル名は MAP001I.h, 出力論理マップファイル名称は MAP001O.h になります。

### 2.2.2 定義パターンおよび部品の登録

定義パターンおよび部品について説明します。なお、定義パターンおよび部品の登録の詳細手順については、マニュアル「XMAP3 開発・実行ガイド」を参照してください。

#### (1) 定義パターン、部品とは

ドロワーでは、画面のレイアウト定義のほかに、パターンや部品を必要に応じて定義できます。パターンとは、マップを効率良く作るために、定型的なオブジェクトを定義してあるひな型のマップのことです。部品とは、複数のオブジェクトを一つのグループとして定義したオブジェクトのことです。使用頻度の高い各種オブジェクトをあらかじめ部品として定義しておくことで、マップを効率良く作成できます。

#### (2) 定義パターンの登録

XMAP3 では、汎用的に使用できる定義パターンを標準提供していますが、このほかに、必要に応じてユーザ独自の定義パターンを用意しておけます。ユーザ独自の定義パターンは、レイアウト定義前に作成してください。

登録した定義パターンは、画面の新規作成時に、新規作成ダイアログの定義パターンの選択一覧に表示されます。

定義パターンは、GUI画面（一次）、GUI画面（二次）ごとにそれぞれ登録できます。

#### (3) 部品の登録

よく使用する属性のオブジェクトや、幾つかのオブジェクトを組み合わせたグループを部品として登録できます。オブジェクトを組み合わせるには、固定フィールドや入出力フィールドなどの複数のオブジェクトを、グループ化することで実現できます。部品を有効に使用することで、ユーザの画面定義を効率良く作成できます。部品を使用する場合は、レイアウト定義前に作成しておく必要があります。

部品は、GUI画面（一次）、GUI画面（二次）ごとにそれぞれ登録できます。

#### (4) 定義パターンおよび部品登録時の注意

定義パターンと部品の登録は、ドロワーセットアップの「運用管理者用の設定」で「パターンの登録・更新を許可」または「部品の登録・更新を許可」を選択しているときにできます。ただし、定義パターンおよび部品は、ユーザの開発プロジェクト内で必要な



ものを登録するため、プロジェクト管理者などの特定のユーザが登録することをお勧めします。

## 2.3 GUI画面属性の設定

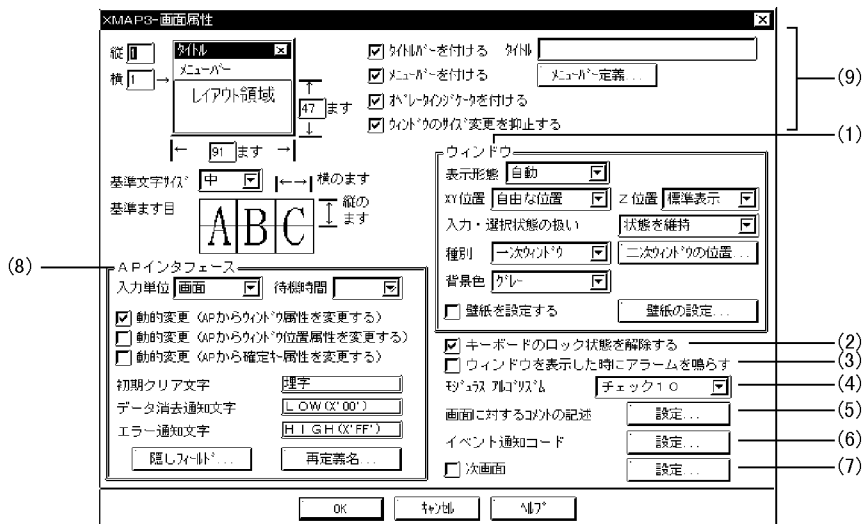
GUI画面の画面属性情報、およびイベント通知コードの詳細について説明します。

画面を制御する情報を画面属性情報といいます。画面属性情報は、ドロースेटアップで初期値が設定されています。この情報は、セットアップ時、または画面定義時に設定できます。

画面属性情報に指定できる属性を次に示します。

### 2.3.1 画面属性の詳細

画面属性は、画面属性ダイアログで指定します。なお、次の図にある番号は、該当定義項目が説明されている項番を示しています。



#### (1) ウィンドウ

##### (a) 表示形態

新しく画面を表示したときの表示中の画面の扱いを変更できます。表示中の画面を消去して次の画面を表示する場合は「全面書換」、画面を重ねて表示する場合は「一部上書」を指定します。システムに表示形態を任せる場合は、「自動」を指定します。「全面書換」「一部上書」および「自動」の詳細については、「1.5.3(2) 表示形態」を参照してください。

##### (b) XY位置

物理画面上の仮想画面の表示位置を指定します。二次ウィンドウの定義で、一次ウィンドウでの指定位置に従って表示する場合は、「一次に従う」を指定します。

**自由な位置：**

任意の位置を縦横の値で指定できます。指定した位置は、画面を最初に表示したときに有効となります。そのあと、別画面を表示したり、同一画面を再表示したりした場合は、直前の画面の位置（ウィンドウ操作によって移動した位置）のまま表示します。ただし、表示・印刷セットアップの「デザイン 1」で「画面定義を優先する」を指定している場合は、「全面書換」で表示するたびに指定が有効になるので、指定した位置に画面を表示できます。

**左上 / 左下, 右上 / 右下, 中央：**

それぞれの位置にウィンドウを表示します。この指定は、画面を最初に表示したときと、直前の画面と異なる画面を表示したときに有効となります。同一画面を再表示したときは、直前の画面の位置（ウィンドウ操作を行った位置）のまま表示します。なお、表示・印刷セットアップの「デザイン 1」の「ウィンドウ操作を優先する」または「画面定義を優先する」の選択は無効となります。

AP から動的変更で指定する場合は、「2.3.1(8)(d) 動的変更（AP からウィンドウ位置属性を変更する）」を参照してください。

**(c) Z 位置**

物理画面上に、複数の仮想画面が表示されている場合などに、それぞれの仮想画面の重なり位置を指定し、操作するウィンドウを目立たせることができます。また、ウィンドウを一時的に非表示にできます。

**標準表示, 手前に表示, 後ろに表示, 一時非表示：**

それぞれの Z 位置にウィンドウを表示します。この指定は、最初に画面を表示したときと、直前の画面と異なる画面を表示したときに有効となります。

同一画面を再表示したときは、直前の画面の位置（ウィンドウ操作を行った位置）のまま表示します。

なお、一時非表示の場合、送信キーを押したときと同じイベント通知コードが、自動的に AP に渡されます。

AP から動的変更で指定する場合は、「2.3.1(8)(d) 動的変更（AP からウィンドウ位置属性を変更する）」を参照してください。

**(d) 入力・選択状態の扱い**

画面を一部上書するとき、画面表示前の入力・選択系オブジェクトの扱いを変更できます。入力・選択系オブジェクトを未入力または未選択状態にするかどうかを指定します。転送状態とは、キー入力または AP での指定で入力済みしておくことを示します。AP で入力済みを指定した場合、AP から未入力状態に変更させたり、全面書換をしないかぎり、入力済みの状態は引き継がれます。

## 2. GUI 画面の作成

一度表示した画面に対して一部上書したときに、AP に返す値を次に示します。

操作	オブジェクトの表示属性		ウィンドウ属性（入力・選択状態の扱い）		
			未入力・未選択	状態を維持	初期状態
ヌルを入力	入力済（初期値なし）		消去通知文字	消去通知文字	消去通知文字
	入力済でない（初期値なし）		初期クリア文字	初期クリア文字	初期クリア文字
入力が選択	入力済（初期値なし）		表示データ	表示データ	消去通知文字
	選択済		選択通知	選択通知	選択通知
	選択済でない		初期クリア文字	選択通知	初期クリア文字
	初期値あり	入力済でない	初期値	初期値	初期値
	初期値なし	入力済でない	初期クリア文字	初期クリア文字	初期クリア文字
何もしない	初期値あり	入力済	初期値	初期値	初期値
		入力済でない	初期値	初期値	初期値
	初期値なし	入力済	消去通知文字	消去通知文字	消去通知文字
		選択済	選択通知	選択通知	選択通知
		入力済でない	初期クリア文字	初期クリア文字	初期クリア文字
		選択済でない	初期クリア文字	初期クリア文字	初期クリア文字

### 注

ウィンドウ属性（入力・選択状態の扱い）は、ドロースेटアップの「表示属性の標準値」の「ウィンドウ属性」タブでも指定できます。なお、ドロースेटアップでの値が有効になるのは、XMAP3 が標準提供している定義パターンを使用してドロースेटアップした場合、および XMAP3 が標準提供している定義パターンを使用して、ユーザ登録の定義パターンを作成したときです。

### 注

固定のラジオボタン、チェックボタンで動的変更しないドロースेट定義をした場合は、初期クリア文字を返します。

### (e) 種別

定義する画面が一次ウィンドウか、二次ウィンドウかを指定します。

### (f) 背景色

画面全体で共通の背景色（ウィンドウの色）を変更できます。標準値は「グレー」です。

#### (g) 壁紙を設定する

「壁紙を設定する」をチェックして[壁紙の設定]ボタンをクリックすると、壁紙として使用できるグラフィックファイルの一覧が表示されます。壁紙として使用したいファイルを選びます。なお、壁紙はレイアウト領域には表示されません。確認するときは、テスト表示を使用してください。

壁紙はビットマップファイル(.bmp)です。壁紙の表示様式については、「2.1.6(2) 壁紙の利用」を参照してください。

#### (h) 二次ウィンドウの位置

「種別」が「一次ウィンドウ」の場合に、このウィンドウに対して表示する二次ウィンドウの位置を、ウィンドウ種別のIDによって、あらかじめ指定できます。ウィンドウ種別のIDの詳細については、「2.4.3 二次ウィンドウ」を参照してください。

指定しない場合は、各二次ウィンドウに設定された位置で表示されます。

### (2) キーボードのロック状態を解除する

キーボードのロック状態を直前の画面から継続するか、解除するかが指定できます。この指定をチェックすると、画面を表示したとき、マウスやキーボードから入力できる状態になります。指定しない場合は、キーボードのロック状態が直前の画面から継続されます。画面の操作では、「取消キー」によってキーボードのロック状態を解除できます。なお、キーの詳細については、「4.2.1 キーの機能」を参照してください。

### (3) ウィンドウを表示したときにアラームを鳴らす

画面を表示したときにアラームを鳴らすかどうか指定できます。初期値は「鳴らさない」です。

### (4) モジュラスアルゴリズム

モジュラスチェックを「(モジュラス)10チェック」ですか「(モジュラス)11チェック」であるかが指定できます。モジュラスチェックの詳細については、「付録A 属性情報一覧」を参照してください。画面属性ダイアログの初期値は「(モジュラス)10チェック」です。

### (5) 画面に対するコメントの記述

画面名(日本語名)と画面に関するコメントを書きます。各オブジェクトに付ける概要や使用目的などの情報です。実際の画面表示の対象にはなりません。

コメントはドキュメンテーション支援でほかの情報とともに出力されるので、定義情報をドキュメント化したときにわかりやすい情報が得られます。また、日本語名はテスト支援の遷移図の画面名として使用されます。ドキュメンテーション支援およびテスト支援についてはマニュアル「XMAP3 開発・実行ガイド」を参照してください。

## 2. GUI 画面の作成

### (6) イベント通知コード

何によって発生するイベントか、イベント発生後の動作はどうするかを画面単位に指定します。イベント通知コードの詳細についてはイベント通知コードダイアログで指定します。

対象	通知コード	動作	コメント
ファンクシヨ (F1)	PF01	(使用しない)	
ファンクシヨ (F2)	PF02	AP通知(中断)	
ファンクシヨ (F3)	PF03	AP通知(確定)	
ファンクシヨ (F4)	PF04	AP通知(確定)	
ファンクシヨ (F5)	PF05	AP通知(確定)	
ファンクシヨ (F6)	PF06	キー押下無効	
ファンクシヨ (F7)	PF07	AP通知(確定)	
ファンクシヨ (F8)	PF08	AP通知(確定)	
ファンクシヨ (F9)	PF09	AP通知(確定)	
ファンクシヨ (F10)	PF10	AP通知(確定)	
ファンクシヨ (F11)	PF11	AP通知(確定)	
ファンクシヨ (F12)	PF12	AP通知(確定)	
ファンクシヨ (F13)	PF13	AP通知(確定)	
ファンクシヨ (F14)	PF14	AP通知(確定)	
ファンクシヨ (F15)	PF15	AP通知(確定)	

イベントが発生したとき、APに通知するかどうかはイベント通知コードダイアログの「動作」で指定できます。イベント通知コードダイアログで定義する、通知コードと動作との関係を次に示します。

対象	通知 コード	動作			
		確定キーを使用する		確定キーを使用しない	
		一次 ウインドウ	二次 ウインドウ	一次 ウインドウ	二次 ウインドウ
ファンクシ ヨ ン1~24 (F1~F24)	PF01 ~	AP通知(確定)	AP通知(確定)	AP通知(確定)	AP通知(確定)
	PF24	AP通知(中断)	AP通知(終了)	AP通知(中断)	AP通知(終了)
		ポップアップ表 示	AP通知(中断)	キー押下無効	AP通知(中断)

対象	通知 コード	動作			
		確定キーを使用する		確定キーを使用しない	
		一次 ウィンドウ	二次 ウィンドウ	一次 ウィンドウ	二次 ウィンドウ
送信	ENTR	キー押下無効	AP 通知 (取消)	使用しない	AP 通知 (取消)
割り込み	BREK	使用しない	ポップアップ表示 キー押下無効 使用しない	使用しない	キー押下無効 使用しない
スクリーン 消去	SCRN				
PA1 キー (PA1)	PA1				
PA2 キー (PA2)	PA2				
PA3 キー (PA3)	PA3				
ファンクショ ンキー 25 ~ 84 (F25 ~ F84)	A001 ~ A060				
プッシュボタ ン/メニュー バー	A061 ~ A071	AP 通知 (確定) AP 通知 (中断)	AP 通知 (確定) AP 通知 (終了) AP 通知 (中断) AP 通知 (取消)	AP 通知 (確定) AP 通知 (中断)	AP 通知 (確定) AP 通知 (終了) AP 通知 (中断) AP 通知 (取消)

## 注

既存の画面に対して確定キーとして使用する場合は、ユティリティで「セットアップ情報反映」を実行してください。

動作の詳細は次のとおりです。

動作の指定	説明
AP 通知 (確定)	画面確定をして AP にイベントを通知する。入力・選択必須のチェックが実行され、エラーがあった場合は AP に制御を返さない。
AP 通知 (終了)	「確定」と同じ動作をし、ウィンドウを閉じる。二次ウィンドウでだけ指定できる。
AP 通知 (中断)	画面確定をしないで AP にイベントを通知する。入力・選択必須のチェックは実行されない。AP にはイベント通知コードを返すが、入力データは返さない。ヘルプボタンや強制終了などに使用する。
AP 通知 (取消)	「中断」と同じ動作をし、ウィンドウを閉じる。二次ウィンドウでだけ指定できる。

## 2. GUI 画面の作成

動作の指定	説明
ポップアップ表示	使用目的が「POP・手動」のポップアップのメニュー項目を開く。AP にイベントは通知しない。なお、メニュー項目が開くのは、フォーカス・カーソルが位置づけてあるポップアップである。
キー押下無効	キー押下無効は、使用しない PF キーなどに使う。これに対応する確定キー、プッシュボタン、またはメニューバーを押すと、エラーとなって画面は入力待ち状態になる。なお、無効にする確定キーを AP から動的変更することもできる。
使用しない	使用しない PF キーなどに使う。これに対応する確定キーを押すと、画面が確定される。AP には、イベント通知コードとして初期クリア文字が返る。

### 注 1

XMAP3 で表示した画面の入出力テキスト・フィールドをコピーまたはペーストしても、AP へは何も返しません。

### 注 2

OLTP サーバ構成の場合、OpenTP1 上で「次メッセージ要求キー」として割り当てられているキーのイベントは AP に渡りません。

### 注

OLTP サーバ構成で使用している場合、リターンコードが「4」、詳細エラーコードが「(0414)<sub>16</sub>」のエラーとなります。

## (7) 次画面

AP で、画面名称を固定値として定義していなくても、次に表示させたい画面名称を実行時に取得したいとき指定します。次画面はマップ名が 6 文字のときに指定できます。なお、次画面は OLTP サーバ構成の環境で利用できます。

## (8) AP インタフェース

AP からウィンドウの属性を変更して表示する場合に指定します。

### (a) 入力単位

画面確定のタイミングを「画面」「フィールド」「イベント」および「表示直後」から選びます。入力単位の詳細については「1.5.8 入力単位」を参照してください。

### (b) 待機時間

「入力単位」で「表示直後」を選んだ場合、AP に自動的に制御が渡されるまでの時間を設定できます。「0 秒」も設定できます。

### (c) 動的変更 (AP からウィンドウ属性を変更する)

AP から表示形態などのウィンドウ属性を変更して、画面を表示します。

この場合、制御項目が論理マップに展開されます。この制御項目に修飾名という定数を



AP が設定することで、表示属性が動的に変更されます。

修飾名と表示属性との対応の標準値については「付録 A.10 修飾名と変更属性の標準値」を参照してください。なお、表示属性はドロースेटアップで変更できます。

#### (d) 動的変更 (AP からウィンドウ位置属性を変更する)

一次ウィンドウを表示する場合のウィンドウの位置 (XY 位置, Z 位置) を動的に変更できます。

この場合、制御項目が論理マップに展開されます。この制御項目に修飾名という定数を AP が設定することで、表示属性が動的に変更されます。

直前の画面と同じ画面を再表示する場合は、画面定義時にウィンドウ属性の表示形態に「全面書換」を指定するか、または AP からの動的変更で表示形態に「全面書換」を指定してください。その他の指定の場合や指定なしの場合は、位置属性は変更されません。

修飾名と表示属性との対応の標準値については「付録 A.10 修飾名と変更属性の標準値」を参照してください。なお、表示属性はドロースेटアップで変更できます。

#### (e) 動的変更 (AP から確定キー属性を変更する)

画面に対する操作に使用する各制御キーの有効/無効を動的に変更できます。

この場合、制御項目が論理マップに展開されます。この制御項目に修飾名という定数を AP が設定することで、表示属性が動的に変更されます。

修飾名と表示属性との対応の標準値については「付録 A.10 修飾名と変更属性の標準値」を参照してください。なお、表示属性はドロースेटアップで変更できます。

#### (f) 初期クリア文字

AP 実行時に、画面からの入力データを AP が受け取る前に、XMAP3 が入力論理マップの各項目をあらかじめクリアしておく文字です。なお、AP が受け取るデータの優先順位は次のとおりです。

1. キー入力データ / 選択したイベント通知コード
2. 入力済み / 選択済みのとき、表示されているデータ / イベント通知コード
3. エラー通知文字
4. 初期値
5. 初期クリア文字

#### (g) データ消去通知文字

AP 実行時に、画面からフィールドキーなどで入出力テキスト・フィールドのデータを消去したり、(00)<sub>16</sub> のデータを受信したときに AP が受け取るデータです。したがって、AP が受け取ったデータがデータ消去通知文字であれば、画面操作でデータ消去されたと判断できます。

初期クリア文字や初期値とデータ消去通知とを分けておくことで、入力操作がなかった

## 2. GUI 画面の作成

のか、データ消去されたのが区別できます。

なお、オブジェクトに「入力済み」が定義されている場合で、画面確定時に何も入力しなくてもヌル (00)<sub>16</sub> が表示されている状態のとき、データ消去通知文字が AP に返されます。

### (h) エラー通知文字

XMAP3 が入力データにエラーを検出した場合に、AP が受け取るデータです。

入力データのエラーは、半角文字だけを入力できるテキスト・フィールドに、全角文字を入力したときなどに発生します。

### (i) 隠しフィールド

実際の画面には表示されないフィールドで、入力固定項目として使用します。

入力固定項目とは、入力論理マップに必ず固定の値を返すデータ項目で、論理マップ可変部の最初のデータ項目として展開されます。

OLTP サーバ構成で使用するトランザクションコードを XMAP3 上で定義するには、隠しフィールドとして定義します。

### (j) 再定義名

定義済みの論理マップを別定義として使用するときの名称です。半角で 30 字以内で定義します。

入力論理マップ用と出力論理マップ用とで別の名称にすることもできます。

## (9) タイトルバー / メニューバー / オペレータインジケータ / ウィンドウサイズ変更抑止 / 基準文字サイズ

### (a) タイトルバーを付ける

ウィンドウのタイトルバーを付けるかどうか、およびタイトルの内容を指定します。

### (b) メニューバーを付ける

画面に対する処理をメニューから選択させたいときに指定します。指定すると、定義したウィンドウにメニューバーが付きます。メニューバーの内容の定義は、[メニューバー定義] ボタンから操作します。

### (c) オペレータインジケータを付ける

オペレータインジケータを表示するかしないかを指定できます。OLTP サーバ構成の場合には、オペレータインジケータを付ける必要があります。この場合、拡張形式は使用できません。オペレータインジケータを設定していない場合、二次ウィンドウを表示するとき TP1/NET/XMAP3 の通信エラーとなります。

なお、オペレータインジケータに表示されるメッセージについては、マニュアル

「XMAP3 開発・実行ガイド」を参照してください。

(d) ウィンドウのサイズ変更を抑止する

マウスのドラッグで、ウィンドウのサイズを自由に変更できるかできないかを指定します。標準は「変更できない」です。ウィンドウを小さくした場合は、必要に応じてスクロールバーが付けられます。スクロールバーを付けるか付けないかは、表示・印刷セットアップで設定します。なお、スクロールバーを付けるときは、スクロールすることで重要なエラーメッセージなどが隠れないよう、画面設計時に注意してください。

(e) 基準文字サイズ

画面全体の基準となる文字サイズを変更できます。

画面属性の定義では、「基準文字サイズ」の設定によって、配置できる領域が変わってきます。この基準文字サイズは、レイアウト画面上に表示されているグリッドの縦横の間隔と対応します。縦（行）の間隔は文字サイズの半分、横（列）の間隔は半角の文字サイズの幅で表示されます。GUI画面中のフィールドボックスに配置する場合の基準文字サイズについては、「2.1.1(3) ウィンドウ」の「ますの高さ」の記述を参照してください。

## 2.3.2 画面属性の動的変更

画面属性はAP実行時に変更できます。XMAP3では、ドローで定義した画面属性情報のほかに、APからの変更用としてドローセットアップで画面属性情報を定義できます。

これらの画面属性情報は、名前を付けて区別します。この名前を修飾名といいます。標準提供の画面属性情報の修飾名は「付録 A.10 修飾名と変更属性の標準値」を参照してください。標準提供の修飾名を使用すると、表示形態をAPから変更できます。APでは、論理マップの属性情報格納エリア（制御項目）に画面属性情報の修飾名を格納することで、実行時に画面属性の表示属性を変更できます。

## 2.4 GUI 画面を構成するオブジェクト

GUI 画面を構成するオブジェクトと、それぞれの機能について説明します。なお、オブジェクトは機能によって次のように分類されます。

大分類	小分類	オブジェクト名	
キャラクタコントロール	キーエントリ	入出力テキスト・フィールド	
		入出力日付テキスト・フィールド	
		入出力時刻テキスト・フィールド	
		コンボボックス (固定 / 可変)	
	ラベル	固定テキスト・フィールド	
		出力テキスト・フィールド	
		出力日付テキスト・フィールド	
		出力時刻テキスト・フィールド	
		予約テキスト・フィールド	
	選択エントリ	コンボボックス (固定 / 可変)	
		ポップアップテキスト・フィールド (固定 / 可変)	
		スピンボックス	
	候補選択コントロール	リストボックス	リスト項目
ラジオボタンボックス (固定 / 可変)		ラジオボタン	
チェックボタンボックス (固定 / 可変)		チェックボタン	
トグルフィールド			
グループコントロール	グループボックス		
	フィールドボックス		
	フレーム		
グラフィックコントロール	グラフィック	固定グラフィック	
		出力グラフィック	
	セパレータ・けい線		
	メッセージアイコン		
	塗りつぶしフィールド		
コマンドコントロール	メニューバー	ブルダウン	
		カスケード	
	プッシュボタン		

注

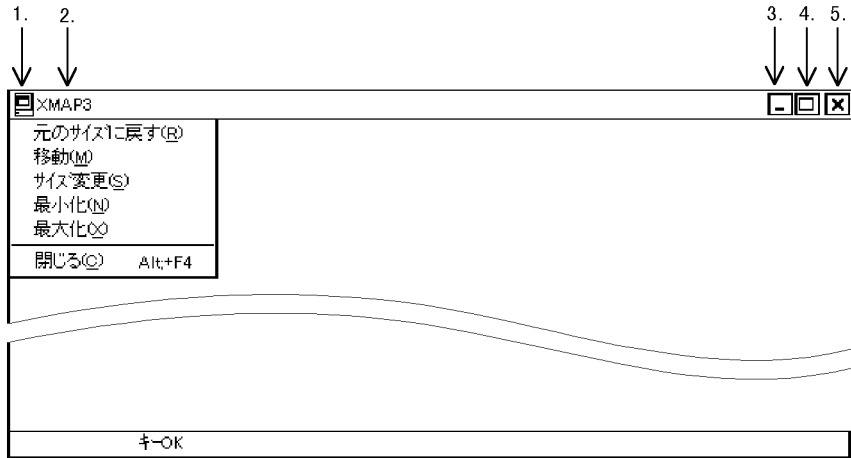
予約テキスト・フィールドは XMAP3 の開発環境を利用した、OLTP サーバ構成用のプログラムで使用できます。

## 2.4.1 タイトルバー

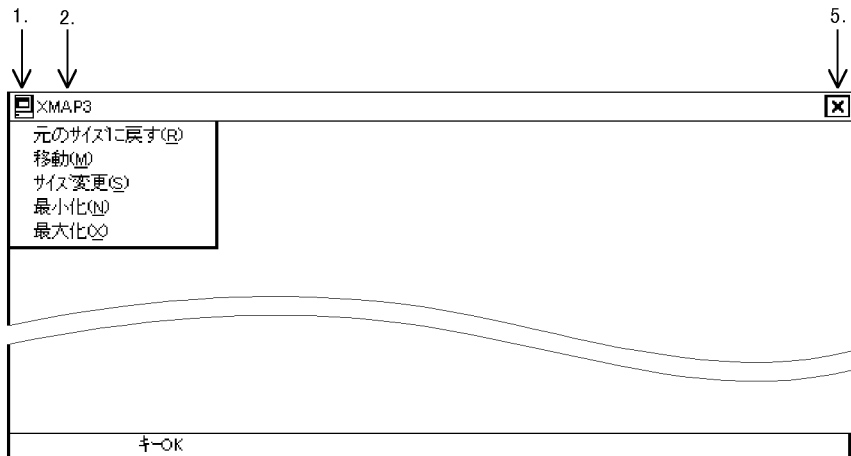
一次ウィンドウおよび二次ウィンドウのタイトルバーと、各部の表示仕様を示します。

### (1) 形式

一次ウィンドウ



二次ウィンドウ



### (2) 表示仕様

タイトルバーの表示仕様を次の表に示します。なお、タイトルバーを表示するかしない

## 2. GUI 画面の作成

かは画面属性ダイアログで指定できます。

表 2-5 タイトルバーの表示仕様

項番	名称	機能	使用条件, その他
1	タイトルバーアイコン またはコントロール ボックス	XMAP3 システムのアイコン またはコントロールメニュー を出すための, ボタンを表示 する。クリックすると最小化 メニューなどが表示される。	タイトルバーアイコンが表示され る。
2	ウィンドウタイトル	表示されているウィンドウの タイトル。通常は, 業務内容 を示す。	ドローで定義する。何も定義されて いないときは「XMAP3」が表示さ れる。表示・印刷セットアップで, 表示するかしないかを設定できる。
3	最小化ボタン	ウィンドウをアイコン化す る。	表示・印刷セットアップで, 表示す るかしないかを設定できる。
4	最大化ボタン	ウィンドウのサイズを最大に する。	表示・印刷セットアップで, 表示す るかしないかを設定できる。
5	閉じるボタン	ウィンドウを閉じ, AP を終 了させるために使用する。	表示・印刷セットアップで, 表示す るかしないかを設定できる。 二次ウィンドウでは使用できない。

### (3) AP インタフェース

- ・タイトルバーのタイトルは AP から変更できません。
- ・[ 閉じる ] ボタンはオプションです。[ 閉じる ] ボタンをクリックすると, [ 割込 ] キーと同じイベントが AP に通知されます。

## 2.4.2 メニューバー

メニューバーは, 表示された画面に対する処理を選択するときに使用します。画面遷移やポップアップメニュー (手動) 表示の契機としても使用できます。メニューバーは常に表示されているため, 一般的に使用する機能 (ファイル, 編集など) を設定することをお勧めします。

メニューバーで表示できるメニューの種類を次の表に示します。

表 2-6 メニューの種類

種類	内容
メニューバー	ウィンドウ上のタイトルバーの下に表示されるメニュー。 1 メニューバー上で最大 24 とおりの処理が選択できる。

種類	内容
プルダウンメニュー	メニューバーを選択すると表示されるメニュー。 メニューバーを選択したとき下位の処理選択があるとき（アクション 計算，印刷などアクションに対する処理が細分化するとき）に使用する。 1 プルダウンメニュー上で最大 24 とおりの処理が選択できる。
カスケードメニュー	メニューの右端にカスケードメニュー表示用の記号（矢印ボタン）が表示されているプルダウンメニューを選択すると表示されるメニュー。 プルダウンメニューの処理内容がさらに細分化される時（印刷から印刷開始まで，書式設定，印刷取消など印刷に対する処理が細分化される時）に使用する。1 カスケードメニュー上で最大 24 とおりの処理が選択できる。

選択する処理が多数あり処理単位でグループ分けができる場合は，プルダウンメニューやカスケードメニューを使用してメニューを階層化することをお勧めします。メニューを階層化することで，1 ウィンドウ中で最大 96 とおりの処理が選択できます。

### （1）選択時の AP インタフェース

メニューは，マウス，またはキーボードで選択できます。キーボードで処理を選択する場合，アクセスキーとショートカットキーを指定しておきます。

各メニューを選択したときに指定できる動作を次に示します。

#### 《メニューバー》

- プルダウンメニューを表示する。
- AP にイベント通知コードを返す。
- 手動ポップアップを表示する。
- 不活性（選択できない状態にする）
- 最後のテキストを右端に表示する。
- 動的変更（AP から表示属性を変更する）

#### 《プルダウンメニュー》

- カスケードメニューを表示する。
- AP にイベント通知コードを返す。
- 手動ポップアップを表示する。
- メニューの下にセパレータ線を引く。
- 不活性（選択できない状態にする）

#### 《カスケードメニュー》

- AP にイベント通知コードを返す。

また，メニューを選択したときに複写，貼り付け，およびポップアップ表示を指定できます。複写と貼り付けの機能は画面間での動作のため，AP に影響はありません。メニューの選択方法，および複写と貼り付け機能の操作の詳細については，「4.4 コピー & ペースト操作」を参照してください。ポップアップ表示は，ポップアップ（手動）付きの入出力テキストボックス（またはフィールドボックス内の入出力フィールド）を配置してある場合に有効となります。ポップアップの詳細については「2.4.10 ポップアップ

## 2. GUI画面の作成

「プテキスト」を参照してください。

メニュー選択時に AP に通知コードを返す場合、メニューに対応したイベント通知コードが入力論理マップに格納されます。

### 2.4.3 二次ウィンドウ

二次ウィンドウは、一次ウィンドウを消去しないで補助的なデータの表示と操作をするときに使用します。また、二次ウィンドウは、メッセージの表示や入力するデータを検索するダイアログを表示するときにも使用します。

二次ウィンドウの表示は、一次ウィンドウを表示する方法と同じです。二次ウィンドウは、必ず一次ウィンドウの前面に重ねて表示します。このとき、一次ウィンドウが消去されることはありません。また、二次ウィンドウの表示中には、一次ウィンドウの操作はできません。

また、画面属性ダイアログの「二次ウィンドウの位置」または種別「二次ウィンドウ」時の「位置」ではウィンドウ種別の ID によって二次ウィンドウの表示位置およびサイズを指定できます。ID ごとに次の表に示すように分類して使用することをお勧めします。

表 2-7 ウィンドウ種別の ID ごとの表示内容の適用例

ID	表示制御の内容
1	データ入力の補助のために表示するダイアログ。
2	メッセージを表示するダイアログ。
3	ヘルプ情報を表示するダイアログ。

#### (1) データ入力補助のダイアログとして使用する

一次ウィンドウで選択対象とする項目が多岐にわたり、ポップアップなどでは表示できない場合や、選択するデータを検索したい場合に使用します。データ入力時の煩雑さをなくすために、一次ウィンドウの一つの項目に対するデータ入力補助を目的とし、二次ウィンドウから選択するデータは基本的に一つにすることをお勧めします。

画面のサイズは一次ウィンドウの半分以下とし、一次ウィンドウと同様の機能を使用して画面を設計します。

#### (2) メッセージ表示のダイアログとして使用する

画面を操作するオペレータに対してメッセージを通知するときに使用します。

メッセージには「エラー」「インフォメーション」「クエスチョン」「ワーニング」があります。

画面のサイズは一次ウィンドウの 1/4 以下を目安とし、テキストボックスやフィールドボックスを使用した出力メッセージと、メッセージアイコンとを組み合わせで画面を設計します。



### (3) ヘルプ表示のダイアログとして使用する

一次ウィンドウに対する操作説明などのヘルプ情報を表示するときに使用します。ヘルプ情報は、確認後,[OK]のプッシュボタンなどで二次ウィンドウを閉じるように指定します。

画面のサイズは一次ウィンドウの半分以下とし、テキストボックスやフィールドボックスを使用して表示するヘルプダイアログを設計します。

## 2.4.4 入出力テキストボックス

入出力テキストボックスは、キーボードなどからデータを入力するとき、およびAPから渡されるデータを表示するときに使用します。入出力するデータの意味に応じて、種類(使用目的、詳細目的)を選びます。

初期表示値(出力データ)を表示し、書き換えが必要な場合だけ入力するような使い方ができます。

ボックスの表示属性の標準値は背景色が「白」です。テキストの表示属性の標準値は文字色が「黒」です。なお、入出力テキストボックスの表示属性の詳細については「付録A 属性情報一覧」を参照してください。

### (1) 表示形式

表示形式は、扱うデータの種別によって次のように変わります。

<p>&lt;英数, 漢字など&gt;</p> <div style="border: 1px solid black; padding: 2px; width: 150px; margin-bottom: 5px;">日立</div> <div style="border: 1px solid black; padding: 2px; width: 150px; margin-bottom: 5px;">ABC</div> <p>左詰めが表示</p>	<p>&lt;数値&gt;</p> <div style="border: 1px solid black; padding: 2px; width: 150px; margin-bottom: 5px; text-align: right;">123</div> <p>右詰めが表示</p>	<p>&lt;金額&gt;</p> <div style="border: 1px solid black; padding: 2px; width: 150px; margin-bottom: 5px; text-align: right;">1,234</div> <p>右詰めに表示 「,」が付く</p>
--	--	---

- 枠は必ず付きます。
- 文字サイズを指定できます。
- 文字を強調するために、斜体や太字を使用できます。
- 明朝体、ゴシック体を使用できます。
- 枠を付ける場合「枠高」で「2ます」を指定すると、枠付きのテキストボックスを上下に配置したとき、枠同士が接した状態でも表示できます。「枠高」で「2.5ます」または「3ます」を指定している場合、枠と枠の間が1ます分空きます。

### (2) 入力データのチェック

扱うデータの種類を決めたり、空白の使い方を指定して、入力データをチェックします。

#### (a) 使用目的による入力データのチェック

使用目的によって、入力データをチェックできます。使用目的と入力データの関係につ

## 2. GUI画面の作成

いては、「付録 A.9 表示属性の解説」の表 A-8 を参照してください。

### (b) 詳細目的による入力データのチェック

詳細目的を指定して、使用目的よりも細かいチェック条件が指定できます。詳細目的とデータ型および入力できる文字の関係については、「付録 A.9 表示属性の解説」の表 A-9 を参照してください。

### (c) 空白入力

データに対して挿入する空白を、半角だけにするか全角だけにするか混合にするかで指定します。選べる項目は、使用目的および詳細目的に応じて異なります。「禁止」を指定すると、空白の入力はできません。空白入力については、「付録 A.2 キーエントリ・ラベル系オブジェクトの表示属性一覧(テキストとフィールド)」を参照してください。

## (3) 画面確定時の動き

### (a) イベント通知コードの動作で指定する画面確定時の動き

画面属性ダイアログの「イベント通知コード」設定の「動作」に「AP 通知(確定)」を指定している場合、画面確定時に「入力必須」を指定したテキスト・フィールドの内容が入力されているかがチェックされます。ヘルプの表示など、一時的に入力を中断して AP に通知を返すときは、「AP 通知(中断)」「AP 通知(取消)」を使用して、「入力必須」を指定したテキスト・フィールドが未入力でもチェックされないようにできます。

### (b) [閉じる] ボタンでの画面確定時の注意

入出力テキストの使用目的が「金額」「数値」「日付」および「時刻」の場合、[閉じる] ボタンで画面を確定した表示画面から AP に制御が渡るとき、画面上のデータは保証されません。再表示時に前回と同じ内容を表示させたい場合は、明示的にオブジェクトの値を設定するか、画面属性ダイアログの「入力・選択状態の扱い」で「初期設定」を指定してください。

## (4) フォーカスの遷移

遷移条件が指定できます。

- 手動スキップ・右寄せ  
入力データが桁数に満たない場合、タブキーで次の入力/選択できる領域にフォーカス・カーソルを移動すると、入力データが右寄せされます。
- 手動スキップ・全桁必須  
1文字でも入力した場合、桁数いっぱいに入力しないと入力エラーになり、タブキーでのフォーカス・カーソル移動ができなくなります。
- 手動スキップ  
桁数を超えるデータを入力しようとするとエラーになります。
- 自動スキップ  
桁数いっぱいまで文字を入力すると、フォーカス・カーソルが次の入力/選択できる領域に自動的に移動します。これに対して、手動スキップの場合はタブキーを押すこ

とでフォーカス・カーソルの次の入力/選択できる領域に移動します。

## (5) 入力時の AP インタフェース

APの入力論理マップには次の内容を格納します。

### (a) 入力データ項目

#### 下位項目を利用したチェック

必要に応じて下位項目を設定できます。下位項目はデータ型が「文字 (XX)」のときに設定できます。下位項目を設定すると、データをコードデータのように区切ってチェックしたり、1文字の繰り返しデータのようにして、1文字ずつチェックしたりできます。下位項目の詳細については、「6.2.10 下位項目を利用した入出力テキスト」を参照してください。

#### 入力データチェックの通知

どのような操作が起こったかをチェックするためのコードを AP に通知します。

##### • コードエラーの通知

不正な文字を入力した場合、一般的には入力時にエラーになります。ただし、エラー通知文字 (標準は HIGH (X'FF')) を AP へ返すこともできます。なお、この機能を使用できるのは、使用目的「カナ」、詳細目的「カナ・半角」を組み合わせ指定したテキストまたはフィールドに対して、全角文字を入力した場合です。また、空白入力に「一部&全桁 (半角)」か「一部 (半角)」を指定し、表示・印刷セットアップの「日本語入力」タブの「全角スペースコードの扱い」で「全角スペースをそのまま返す」を設定すると、全角の空白を入力した場合にコードエラーになります。

##### • データ消去の通知

フィールドキーでデータを消去した場合などに、データ消去通知文字 (標準は LOW (X'00')) を AP へ返します。

##### • データ未入力の通知

###### ・初期値を返す場合

(例)

テキスト・フィールドで初期値に「ABC」の文字を設定し、データを入力しない場合に初期値の「ABC」を AP へ返します。

###### ・初期クリア文字を返す場合

初期値の文字を設定しないで、データを入力しない場合に初期クリア文字 (標準は埋字と同じ文字) を返します。

(例)

使用目的が「英数」の場合は埋字 (スペース) を、「数字」の場合は埋字 (ゼロ) を AP へ返します。

エラー通知文字、データ消去通知文字および初期クリア文字は、ドローセットアップで変更することもできます。

## 2. GUI画面の作成

### (b) 入力バイト数格納項目 (オプション)

実際にキー入力した文字のバイト数 (半角の桁数) が入ります。入力バイト数格納項目については「5.3.8 入力バイト数格納項目の論理マップ生成規則とマッピング規則」を参照してください。

### (6) 入出力テキスト・フィールドでの定義とAPの実行結果

XMAP3での入出力テキスト・フィールドの定義と、APでの実行結果の関係を次の表に示します。

表 2-8 入出力テキスト・フィールドとAPの実行結果例

適用例	ドローでの定義		実行結果		
	使用目的	詳細目的 (空白入力)	キーボード の入力	表示結果	論理マップ 格納形式
数字 コード	数字	数字	1234	<input type="text" value="1 2 3 4"/>	<input type="text" value="1 2 3 4"/>
			+123	エラー	
英数 コード	英数	英大・数	A001	<input type="text" value="A 0 0 1"/>	<input type="text" value="A 0 0 1"/>
			* 001	エラー	
名前	日本語	混在 (一部 & 全桁 (半角))	日 立	<input type="text" value="日△△立"/>	<input type="text" value="日 △ △ 立 △"/>
		漢字 (禁止)	日立	<input type="text" value="日 立"/>	<input type="text" value="日 立 △ △"/>
		漢字 (禁止)	日立	エラー	
数量	数値	数字	12	<input type="text" value="1 2"/>	<input type="text" value="0 0 0 1 2"/>
		数字・数字記号	+12	<input type="text" value="+ 1 2"/>	<input type="text" value="  + 1 2"/>
金額	金額	数字	1234	<input type="text" value="1, 2 3 4"/>	<input type="text" value="0 1 2 3 4"/>
			+1234	エラー	
フリガナ	カナ	カナ・英大・ マイナス・長音	7ハ°-ト B-1	<input type="text" value="7ハ°-ト B-1"/>	<input type="text" value="7 ハ ° - ト B - 1 △ △"/>
			7ハ°-ト b-1	エラー	
パスワード	パスワード	半角	1234	<input type="text" value="* * * *"/>	<input type="text" value="1 2 3 4 △ △"/>
ID カード	MCR	半角	> 1234	<input type="text" value="1 2 3 4"/>	<input type="text" value="1 2 3 4 △ △"/>

(凡例)

□ : 空白を示す。

注

「桁寄せ」と「埋字」は標準値で定義してあると仮定します。

## (7) 出力時の AP インタフェース

出力時の AP インタフェースについては、「2.4.7 出力テキストボックス」を参照してください。

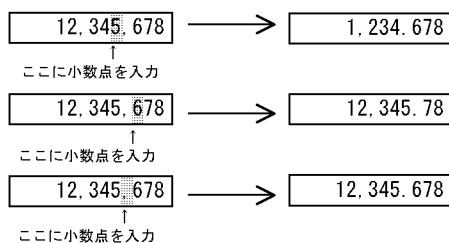
## (8) 表示属性の動的変更

入出力テキストボックスでは、表示属性を AP 実行時に変更できます。表示属性の動的変更については「1.5.6 表示属性の動的変更」を参照してください。

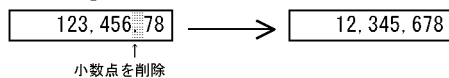
## (9) 小数点入力時の文字列の表示

使用目的が「金額」または「数値」で、詳細目的が「小数点付き」または「符号・小数点付き」の場合の小数点入力時の動きについて、使用目的が「金額」の場合の例を次に示します。

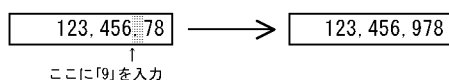
### ● 小数点「.」を入力したときの動作 (右寄せ以外)



### ● 小数点「.」を削除したときの動作



### ● 小数点「.」を数字に置き換えたときの動作



注 1

「数値」の場合は、コンマは付きません。

注 2

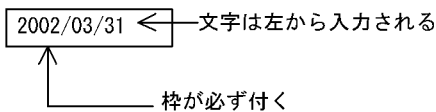
「数値」の場合は、数値の前にゼロを付けることができます。

## 2.4.5 日付テキスト・フィールドと時刻テキスト・フィールド（入出力）

日付や時刻を表すデータを入出力するオブジェクトです。キーボードなどからデータを入力します。また、AP から渡される可変のデータを表示できます。

表示されている出力データを必要に応じて編集したり、日付や時刻を表記する形式（表示書式）を選んだりできます。また、入出力対象にする単位（年／月／日、時／分／秒）なども指定できます。

### （1）表示形式（数字表示）



平成14年3月31日

### （2）表示形式（元年表示）

平成元年5月1日

### （3）表示形式の例

日付および時刻を入力するとき、出力するときおよび格納するときのデータの形式や、入出力の対象にする単位（年／月／日、時／分／秒）はドローで定義します。ドローで定義した入力形式を基に、キーボードから入力されたデータはチェックされ、出力形式（表示書式）を基に、AP から渡されるデータは表示されます。また、データは格納形式に従って AP に格納されます。日付および時刻の入力形式と格納形式の詳細については「付録 A.9 表示属性の解説」の表 A-10 および表 A-11 を参照してください。

### （4）データのチェック

キーボードから入力した日付および時刻のデータをチェックします。なお、西暦下 2 桁（YY）の場合に「00」を指定すると、西暦 2000 年（うるう年）とみなされます。

#### （a）チェック範囲

「部分」の場合

日付を「YYYY（または YY）」「MM」「DD」に分け、それぞれの数字に対し最小値から最大値の範囲でチェックします。「YYYY（または YY）」「MM」「DD」がそれぞれ「1～9999（または 0～99）」「1～12」「1～31」を超えている場合、入力エラーになります。

### 「全体」の場合

入力されたデータが、実際にありうる年月日（または年月）かをチェックします。和暦の場合、年号の区切りのチェック方法には「戸籍六法方式」「JIS規格方式」および「JIS規格と戸籍六法の混在方式」があります。どの方式を適用するかは、表示・印刷セットアップで設定します。詳細についてはマニュアル「XMAP3 開発・実行ガイド」を参照してください。方式ごとの和暦年号の区切りを次に示します。

#### 戸籍六法方式

元号	和暦年号の区切り	区切りの年に該当する西暦
明治	1年9月8日～45年7月29日	1912（明治45年）
大正	1年7月30日～15年12月24日	1926（大正15年）
昭和	1年12月25日～64年1月7日	1989（昭和64年）

#### JIS規格方式

元号	和暦年号の区切り	区切りの年に該当する西暦
明治	1年9月8日～45年7月30日	1912（明治45年）
大正	1年7月31日～15年12月25日	1926（大正15年）
昭和	1年12月26日～64年1月7日	1989（昭和64年）

#### JIS規格と戸籍六法の混在方式

元号	和暦年号の区切り	区切りの年に該当する西暦
明治	1年9月8日～45年7月30日	1912（明治45年）
大正	1年7月30日～15年12月25日	1926（大正15年）
昭和	1年12月25日～64年1月7日	1989（昭和64年）

#### (b) 時刻のチェック

時刻を「HH」「MM」「SS」に分け、それぞれの数字に対し最小値から最大値の範囲でチェックします。「HH」「MM」「SS」がそれぞれ「0～23」「0～59」「0～59」を超えている場合、入力エラーになります。

#### (5) データ入力後の動作

次に示す項目で、データを入力したあとのフォーカス・カーソルの遷移や、APへ制御を返すタイミングなどを定義します。

- 遷移属性

## 2. GUI画面の作成

- 入力済み（入力しなくても表示データを AP に返す）
- 自動送信（入力直後に AP へ制御を渡す）

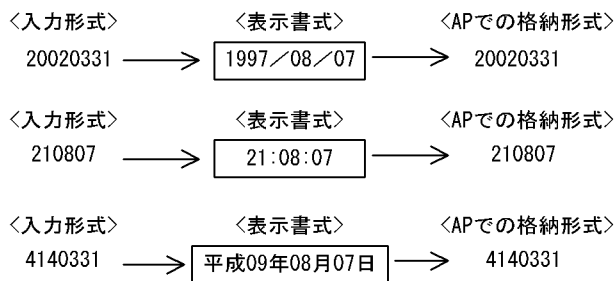
### （6）画面確定時のチェック

画面属性ダイアログの「イベント通知コード」設定の「動作」に「AP 通知（確定）」を指定している場合、画面確定時に「入力必須」を指定したテキスト・フィールドの内容が入力されているかがチェックされます。ヘルプの表示など、一時的に入力を中断して AP に通知を返すときは、「AP 通知（中断）」「AP 通知（取消）」を使用して、「入力必須」を指定したテキスト・フィールドが未入力でもチェックされないようにできます。

### （7）入力と AP インタフェース

デリミタ（ / や - など）を除いた形式のデータを AP に渡します。和暦の場合は、先頭に元号を指定します。時刻の場合は、24 時間表示で入力します。

（例）



注 和暦の元号は1桁の数字(1~4)で表します。数字と元号との対応を次に示します。

- 1：明治
- 2：大正
- 3：昭和
- 4：平成

注

「年月日」および「時刻」のデータは一まとまりに扱われます。「年」「月」「日」および「時」「分」「秒」それぞれのデータを分けて扱いたいときには入出力日付テキストまたは入出力時刻テキストの属性を定義するダイアログで下位項目を設定してください。入出力日付テキスト・フィールドまたは入出力日付テキスト・フィールドでの下位項目の利用については「6.2.11 下位項目を利用した日付テキスト・フィールドと時刻テキスト・フィールド（入出力）」を参照してください。

### （8）出力時の AP インタフェース

下位項目を指定してデータを細分化できます。下位項目はデータ型が「文字 (XX)」のときに設定できます。



## (9) 表示属性の動的変更

入出力の日付・時刻は、表示属性を AP 実行時に変更できます。表示属性の動的変更については「1.5.6 表示属性の動的変更」を参照してください。

## 2.4.6 固定テキストボックス

固定テキストボックスについて説明します。固定テキストボックスは、ドロワーで定義した固定の文字列を表示するときに使用します。全角文字を使用する「日本語」と半角文字だけを使用する「英数」があります。

なお、固定テキストボックスの表示属性の詳細については「付録 A 属性情報一覧」を参照してください。

### (1) 表示形式

画面属性ダイアログの基準文字サイズで「大」「中」「小」のどれかを選ぶと、選んだサイズの文字がドロワーの画面に表示されます。表示されている文字に対し、さらに文字サイズを指定できます。また、文字を強調するために太字や斜体も使用できます。

<拡大／強調>



<枠付／枠なし>



- 文字を強調するために、斜体や太字を使用できます。ただしオブジェクトを斜体で配置する場合は、右端に1ますの余白が必要です。
- 文字サイズを指定できます。
- 明朝体、ゴシック体を使用できます。
- 枠を付ける場合「枠高」で「2ます」を指定すると、枠付きのテキストボックスを上下に配置したとき、枠同士が接した状態でも表示できます。「枠高」で「2.5ます」または「3ます」を指定していると、枠と枠の間が一ます分空きます。

## (2) 表示属性の動的変更

固定テキストの表示属性は実行時に変更できます。表示属性の動的変更については「1.5.6 表示属性の動的変更」を参照してください。なお、動的変更を指定した場合、画面初回表示および画面切り替え時のマッピングオプションに「論理マップだけ」を指定すると、ドローで定義した固定の文字列が表示されません。

## 2.4.7 出力テキストボックス

出力テキストボックスは、AP から渡される可変データを表示するときに使用します。表示するデータに応じて、データの種別（使用目的）を選べます。出力テキストボックスの表示属性の詳細については「付録 A 属性情報一覧」を参照してください。

### (1) 表示形式

画面属性ダイアログの基準文字サイズで、表示するデータの文字サイズを「大」「中」「小」から選べます。また、表示されている文字に対し、さらに文字サイズを指定できます。また、文字を強調するために太字や斜体も使用できます。

<拡大／強調>

日立 日立 日立 日立 日立

標準文字 平体 長体 倍角

日立 日立 日立

太字 斜体 太字&斜体

<枠付／枠なし>

¥1920 ¥1920

枠付 枠なし

- 文字を強調するために、斜体や太字を使用できます。ただしオブジェクトを斜体で配置する場合は、右端に1ますの余白が必要です。
- 文字サイズを指定できます。
- 明朝体、ゴシック体を使用できます。
- 枠を付ける場合「枠高」で「2ます」を指定すると、枠付きのテキストボックスを上下に配置したとき、枠同士が接した状態でも表示できます。「枠高」で「2.5ます」ま

たは「3ます」を指定している、枠と枠の間が一ます分空きます。

## (2) 出力時の AP インタフェース

COBOL の場合、ピクチャ文字による編集ができます。なお、C 言語の場合、すべて「char」として扱われます。

### (a) 出力データ

#### 数字

数字データ、数字編集データを表示できます。数字編集データは、指定した PICTURE 句に従って表示されます (COBOL だけ有効)。

使用できるピクチャ文字:「\*」「+」「-」「¥」「9」「,」「.」「Z」「/」の組み合わせで指定します。

#### 日本語

全角または全角と半角の混在文字データを表示できます。全角と半角の混在文字データを表示する場合は、下位項目を指定してデータを細分化できます。

#### 英数

半角の文字データを表示できます。また、下位項目を指定してデータを細分化できます。

### (b) データ格納形式

出力フィールドでは、埋字や桁寄せなどのデータの編集もできます。AP がデータ有無コードを格納し、出力論理マップに代入した有効データの長さが出力フィールドデータ名の領域の長さより短い場合、出力フィールドデータ名の領域の余った部分を埋め字に置き換えます。また、出力データを出力フィールドデータ名の領域の右側に格納するか、左側に格納するかは、桁寄せ向きに従います。

## (3) 表示属性の動的変更

出力テキストの表示属性は実行時に変更できます。表示属性の動的変更については「1.5.6 表示属性の動的変更」を参照してください。

## 2.4.8 日付テキスト・フィールドと時刻テキスト・フィールド (出力)

日付や時刻を表すデータを表示するオブジェクトです。AP から渡される可変のデータを出力します。日付や時刻を表示する形式 (表示書式) は日付または時刻の定義ダイアログで選んで定義します。出力対象にする単位 (年/月/日、時/分/秒) なども指定できます。

### (1) 表示形式

日付・時刻の出力時の表示形式の例を次に示します。なお、日付および時刻の表示書式と格納形式の詳細については「付録 A.3 キーエントリ・ラベル系オブジェクトの表示属

## 2. GUI 画面の作成

性一覧（日付テキスト・フィールドと時刻テキスト・フィールド）」を参照してください。

定義する表示書式	実際の表示形式	説明
1997 / 08 / 07	2002 / 03 / 31	デリミタ「 / 」 ゼロサプレス <sup>1</sup> なし
1997 . 08 . 07	2002 . 03 . 31	デリミタ「 . 」 ゼロサプレスなし
1997 / 8 / 7	2002 / 3 / 31	デリミタ「 / 」 ゼロサプレスあり
平成 9 年 8 月 7 日 <sup>2</sup>	平成 14 年 3 月 31 日	元号「平成」 ゼロサプレスあり
21 : 08 : 07	21 : 08 : 07	デリミタ「 : 」 ゼロサプレスなし

### 注 1

ゼロサプレスでは、日付・時刻の前にある 0 を削除し、空白で表示します。

### 注 2

和暦表示の場合、元号が変わった最初の年を「1 年」と表示するか「元年」と表示するかを表示・印刷セットアップで設定できます。詳細についてはマニュアル「XMAP3 開発・実行ガイド」を参照してください。

## (2) AP でのデータと出力結果

AP では、データはデリミタなしの形式で扱われます。これらのデータは、出力時に、表示書式に合わせて編集されます。

### (例)

AP でのデータ格納形式	ドローで定義する表示書式 <sup>1</sup>	出力結果
20020331	1997 - 08 - 07	2002 - 03 - 31
4140331 <sup>2</sup>	平成 9 年 8 月 7 日	平成 14 年 3 月 31 日
101000	21 : 08 : 07	10 : 10 : 00
101000	PM 09 : 08 : 07 <sup>3</sup>	AM 10 : 10 : 00
221000		PM 10 : 10 : 00

### 注 1

表示書式の日付および時刻（1997 - 08 - 07 や 21 : 08 : 07）はパラメタです。実際の表示内容ではありません。

### 注 2

和暦の場合、先頭の数字が元号に変換されます。先頭の数字と元号の関係を次に示します。それぞれに 3 種類の形式があります。

- 1.....明 / 明治
- 2.....大 / 大正
- 3.....昭 / 昭和
- 4.....平 / 平成

注 3

時刻の出力では、24 時間表示のデータを、定義する表示書式に従って AM / PM 付きに変換して表示します。

### (3) 出力時の AP インタフェース

下位項目を指定してデータを細分化できます。下位項目はデータ型が「文字 (XX)」のときに設定できます。出力日付テキスト・フィールドまたは出力日付テキスト・フィールドでの下位項目の利用については「6.2.4 下位項目を利用した日付テキスト・フィールドと時刻テキスト・フィールド (出力)」を参照してください。

### (4) 表示属性の動的変更

日付・時刻 (出力) テキストの表示属性は実行時に変更できます。表示属性の動的変更については「1.5.6 表示属性の動的変更」を参照してください。

## 2.4.9 コンボボックス

コンボボックスには、データをメニュー項目選択およびキー入力を入力するもの (キー入力コンボボックス) と、メニュー項目選択だけで入力するもの (メニュー選択コンボボックス) があります。それぞれに固定と可変があります。ドロワーでこれらのオブジェクトを定義するときは、コンボボックスアイコンをクリックして、種類を切り替えてから使用します。

なお、このマニュアルでは、特に書き分けていない場合、コンボボックスはメニュー選択コンボボックスとキー入力コンボボックスを指します。

### (1) メニュー選択コンボボックス

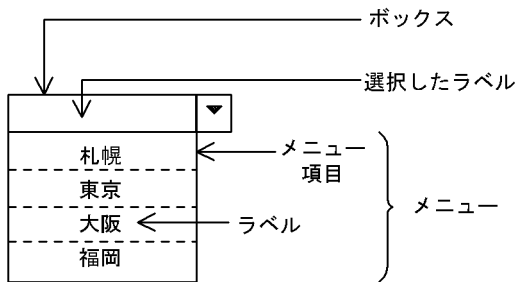
メニュー選択コンボボックスは、入力するデータを、候補 (メニュー項目) から選択することで指定するオブジェクトです。このとき、キー入力でメニューの内容を編集することはできません。

メニュー選択コンボボックスのメニューは、オブジェクトとして常に表示されてはいません。したがって、必要な場合だけ入力するオプションを選択したり、画面上に入力候補となるデータを表示する領域を十分に確保できないときに使用することをお勧めします。

なお、メニュー選択コンボボックスの表示属性の詳細については、「付録 A 属性情報一覧」を参照してください。

## 2. GUI 画面の作成

### (a) 表示形式



### 注

メニューのタイトルはありません。

### (b) 使用目的

使用目的によって、入力データをチェックできます。使用目的と入力データの関係については、「付録 A.9 表示属性の解説」の表 A-8 を参照してください。

### (c) 表示サイズとメニュー項目数

メニュー選択コンボボックスの固定コンボボックスおよび可変コンボボックスを表示する領域（表示サイズ）は、必ずウィンドウに収まるようにドロー上で定義します。コンボボックスのメニュー項目数が、表示領域に収まらない場合は、自動的にスクロールバーが付けられます。ドローで定義したコンボボックスのメニューの行数よりも少ないメニュー項目を AP から設定した場合は、表示するメニューサイズを項目数に合わせて自動的に縮小します。なお、メニュー項目は最大 128 個、指定できます。

### (d) キーを押したときの動作の選択

メニュー選択コンボボックスはキーで操作できます。キーで操作するときのメニュー選択コンボボックスの動きは、表示・印刷セットアップで設定できます。次に、表示・印刷セットアップで設定できる、キーを押したときのメニュー選択コンボボックスの動きを示します。なお、表示・印刷セットアップでの設定の詳細についてはマニュアル「XMAP3 開発・実行ガイド」を参照してください。

#### カーソルキーでメニューを開くモード

メニュー選択コンボボックスにフォーカスが位置しているときに、カーソル移動キーを押すと、メニューが表示されます。表示されたメニューから、メニュー項目を選択します。

#### カーソルキーでメニューを選択するモード

メニュー選択コンボボックスにフォーカスが位置しているときに、カーソル移動キーを押すと、メニューを開かない状態で、メニュー項目を選択します。カーソル移動キーを押すごとに、次のメニュー項目が選ばれ、逐次、選択したラベルとして表示されます。

## (e) 入出力時の AP インタフェース

メニュー選択コンボボックスには、固定コンボボックスと可変コンボボックスとがあります。固定コンボボックスは、画面設計時に入力データの候補となるメニューが決まっている場合に使用します。可変コンボボックスは、メニューのラベルおよびメニューの通知コードを AP から設定したい場合に使用します。

メニュー選択コンボボックスで選択されたメニューはボックス内に表示され、画面確定後には、メニューに対応した通知コードが入力論理マップに格納されます。

## (f) 表示属性の動的変更

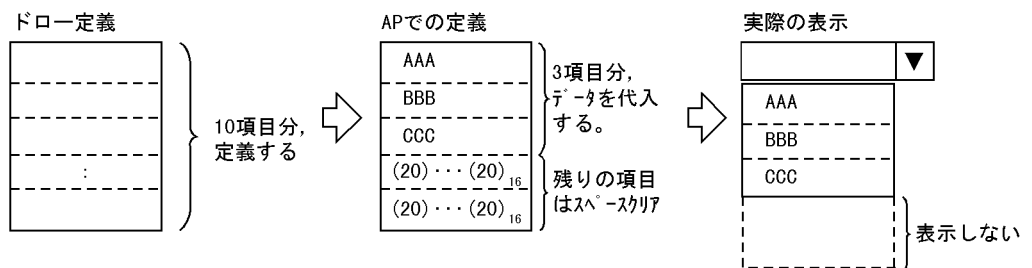
メニュー選択コンボボックスの表示属性を AP 実行時に変更できます。表示属性の動的変更については「1.5.6 表示属性の動的変更」を参照してください。

## (g) AP から初期値を変更する

「AP から初期値を選択する」を指定すると、初期値用のデータエリア「マップ - FIELD」が出力論理マップ中に生成されます。ここに値を代入すると、コンボボックスの初期表示時に、初期値が選択された状態で表示されます。

## (h) メニュー項目数の動的変更（可変コンボボックスの場合）

コンボメニューのダイアログでは、使用するメニュー項目数の最大値を指定しておきます。しかし実際の表示では、メニュー項目数が可変になる場合があります。上から順に項目のデータを代入し、残った項目のラベルに半角の空白（スペースコード）を代入すると、空白を代入したメニュー項目は表示されません。なお、途中のメニュー項目を詰めて表示することはできません。空白のラベルをメニューの最後に表示させたい場合は、ラベルのデータに全角のスペースを代入してください。途中のメニュー項目の空白はデータとみなされます。メニュー項目数の動的変更の詳細については「6.2.32 コンボボックスの項目数の動的変更」を参照してください。



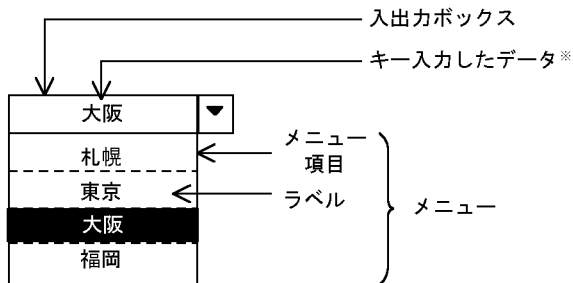
## (2) キー入力コンボボックス

キー入力コンボボックスは、入力するデータを、候補（メニュー項目）から選択したり、直接、キー入力したりして指定するオブジェクトです。キー入力コンボボックスの規則はメニュー選択コンボボックスの規則とほとんど同じです。ここでは、メニュー選択コンボボックスと異なる規則について説明します。なお、キー入力コンボボックスの表示

## 2. GUI画面の作成

属性の詳細については、「付録 A 属性情報一覧」を参照してください。

### (a) 表示形態



注※ メニューで選択したラベルも表示できます。

#### 注 1

メニューのタイトルはありません。

#### 注 2

メニューが開いている状態で入出力ボックスにデータを入力した場合、入力したデータと一致するコードがある場合は、対応するメニュー項目が選択された状態で表示されます。キー入力したデータと一致するコードがない場合、メニューを表示しても、何も選ばれていない状態で先頭のメニュー項目が表示されます。

### (b) 使用目的と詳細目的

APが受け取る通知コードの「データ型」「桁寄せ」および「埋字」に設定できる値は、「使用目的」および「詳細目的」によって決まります。キー入力コンボボックスの「使用目的」については「付録 A.9 表示属性の解説」の表 A-8 を参照してください。「使用目的」と「詳細目的」の組み合わせについては「付録 A.9 表示属性の解説」の表 A-9 を参照してください。

### (c) ボックス内のデータ表示

キー入力コンボボックスにフォーカスがあるときで、メニュー項目が表示されていない場合、ボックス内のデータにはカーソルが位置づきます。また、ボックス内のデータをワンタッチクリアすることもできます。ワンタッチクリアについては「4.2.2 キーでのデータ入力と編集」を参照してください。

### (d) 入出力時の AP インタフェース

出力時の AP インタフェース

ラベル初期表示したい場合、ラベルに対応するコードを出力論理マップに設定します。AP中の論理テーブルに対応するコードがない場合は、設定したコードが初期表示として表示されます。何も設定されていないときは、何も表示されません。



### 入力時の AP インタフェース

キー入力コンボボックスのラベルには、直接、データを入力できます。入力されたデータは、入力論理マップに格納されます。

なお、メニューを選択してデータを入力した場合は、コンボボックスと同じように、ラベルに対応するコードが AP に渡ります。

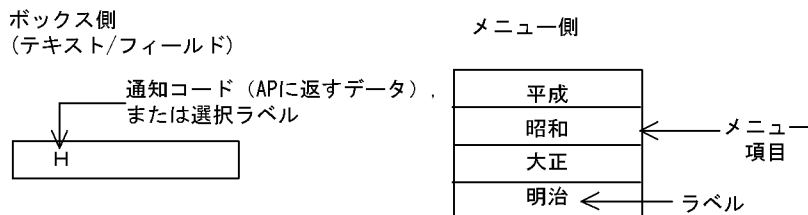
### (3) コンボボックス使用時の注意事項

大量の項目を持つコンボボックスおよびキー入力コンボボックスを、画面に大量に定義すると、生成されるウィンドウ数が Windows の上限値を超えてしまい、画面を表示する際に AP にエラーが通知されることがあります。この場合は、定義するコンボボックスの個数またはメニュー項目数を減らすようにしてください。

## 2.4.10 ポップアップテキスト

ポップアップテキストは、決められた候補（メニュー項目）から一つを選択するとき 사용합니다。固定と可変があります。メニュー項目に入力データの候補を表示することで、データの入力を補助します。入力するデータの内容が決まっている場合に使用することをお勧めします。なお、ポップアップテキストの表示属性の詳細については「付録 A 属性情報一覧」を参照してください。

### (1) 表示形式（可変・固定）



### (2) 使用目的（可変・固定）

使用目的によって、入力データをチェックできます。ただし、ラベルのデータ型は、使用目的の指定と連動しません。ラベルのデータ型の「文字 (XX)」は「日本語」の属性になります。また、使用目的に「自動 - 日本語」または「手動 - 日本語」を選択すると、詳細目的として「混在 (全角・半角)」または「漢字」を選択できます。

使用目的と入力データの関係、および詳細目的については、「付録 A.9 表示属性の解説」の表 A-8 および表 A-9 を参照してください。

### (3) 表示サイズとメニュー項目数（可変・固定）

固定ポップアップメニューおよび可変ポップアップメニューを表示する領域（表示サイズ）は、必ずウィンドウに収まるようにドロウ上で定義します。ポップアップメニューのメニュー項目数が、表示領域に収まらない場合は、自動的にスクロールバーが付けら

## 2. GUI 画面の作成

れます。ドローで定義したポップアップのメニューの行数よりも少ないメニュー項目を AP から設定した場合は、表示するメニューサイズを項目数に合わせて自動的に縮小します。詳細については、「2.4.9(1)(h) メニュー項目数の動的変更（可変コンボボックスの場合）」を参照してください。なお、表示できるメニュー項目数は次のようになります。

- ドローまたは AP で設定したメニュー項目を表示する場合：128 個 / 1 階層（階層は 1 階層だけ）
- ポップアップメニューファイルのメニュー項目を表示する場合  
：256 項目 / 1 階層（ファイルが階層構造の場合）（階層は 8 階層まで）  
：32,000 項目 / ファイル

### （4）入出力時の AP インタフェース（可変・固定）

ポップアップテキストには、固定ポップアップと可変ポップアップとがあります。固定ポップアップは、画面設計時に入力データの候補となるメニューが決まっている場合に使用します。可変ポップアップは、メニューのラベルおよびメニューの通知コードを、AP またはポップアップメニューファイルから設定したい場合に使用します。

ポップアップテキストで選択されたメニューはボックス内に表示され、画面確定後には、メニューに対応した通知コードが入力論理マップに格納されます。

ポップアップメニューの表示方法を次の表に示します。

表 2-9 ポップアップメニューの表示方法

表示方法	内容
自動で表示する POP（自動）	メニュー項目の選択でだけデータを入力する標準的な使い方。 オブジェクトにフォーカスが位置づくとき、メニューが自動的に表示される。
手動で表示する POP（手動）	キーボードからの入力時、メニュー項目の選択との 2 とおりでデータを入力できる使い方。 メニューは次に示す方法で表示される。 <ul style="list-style-type: none"><li>• 手動で表示する メニューは、オブジェクトにフォーカスを位置づけ、あらかじめポップアップ表示用に対応づけたファンクションキーかプッシュボタンを押すと、表示される。なお、同じキーまたはボタンを押して、表示されているメニューを閉じることもできる。キー操作に慣れているユーザはキー入力での操作でき、初心者はボタンなどで表示されているメニューから選択して操作できる。</li><li>• 「最初にメニューを開く」を指定して表示する ポップアップメニューダイアログの「最初にメニューを開く」を指定しておくと、オブジェクトにフォーカスが位置づいたとき、自動的にメニューが表示される。</li></ul> <p>注 アクセスキーでメニュー項目を選択する場合は、キー操作または「最初にメニューを開く」指定をしてメニューを表示しておく必要がある。</p>

### （5）ポップアップ表示属性の遷移条件（可変・固定）

ポップアップでメニューを選択した場合、入出力テキストに表示される選択ラベルは、そのポップアップの入出力テキスト属性で指定する「遷移条件」の動作はしません。選

択したメニューに対応する選択ラベルはそのままの状態の入出力テキストに表示され、右寄せなどの表示はされません。また、次のボックスに自動的にフォーカス遷移する「自動スキップ」の属性は、メニューの表示属性の「選択したらメニューを閉じる」の指定に従います。

ポップアップ表示属性ダイアログの「遷移条件」の定義内容は、使用目的が「POP（手動）」のとき、入出力テキストにキー入力されたテキストにだけ適用されます。使用目的ごとの遷移条件の適用を次の表に示します。

使用目的		遷移条件 (手動スキップ・自動スキップ)	桁寄せ (右寄せ・全桁必須)
POP（自動）		x	x
POP（手動）	メニュー選択		x
	キー入力		

(凡例)

：適用される。

x：適用されない。

注

「自動スキップ」を指定した場合は、「選択したらメニューを閉じる」を指定してください。「手動スキップ」を指定した場合は、「選択したらメニューを閉じる」を指定しないでください。

## (6) アクセスキーと選択ラベル（可変・固定）

ポップアップでは、アクセスキーを利用してポップアップメニューを選んだり、選んだメニュー項目が反映される選択ラベルを使用したりできます。アクセスキーおよび選択ラベルについては「6.2.17 可変ポップアップテキストを使用したデータの入力」または「6.2.19 固定ポップアップテキストを使用したデータの入力」を参照してください。

### (a) アクセスキーの利用

アクセスキーは、可変ポップアップおよび固定ポップアップの使用目的が「POP（自動）」または「POP（手動）」のときに利用できます。また、ポップアップメニューファイルにアクセスキーを記述して、メニューをファイルで指定する可変ポップアップでも利用できます。

### (b) アクセスキーの入力時の注意

アクセスキーは4桁まで指定できます。ドローでアクセスキーの桁数を指定するには、定義画面のアクセスキーの「長さ」で指定します。ポップアップメニューファイルでアクセスキーの桁数を指定する場合については、マニュアル「XMAP3 開発・実行ガイド」を参照してください。また、ポップアップメニューファイルで階層化されているメニュー項目にアクセスキーを入力するときの注意事項については、「2.4.10(10)(a) ポップアップメニューファイル使用時の注意事項」を参照してください。

次にアクセスキーを入力するときの注意事項を示します。

## 2. GUI 画面の作成

- アクセスキーは、メニューが表示されているときだけ入力できます。メニューが表示されていないときに入力すると、通知コードとして AP に渡されますので注意してください。
- アクセスキーは全桁入力したときに、有効になります。指定した長さより短いアクセスキーを入力して確定させようとしても、エラーになります。

### (7) 表示属性の動的変更 (可変・固定)

ポップアップテキストの表示属性を AP 実行時に変更できます。表示属性の動的変更については「1.5.6 表示属性の動的変更」を参照してください。

### (8) AP から初期値を変更する (可変・固定)

「AP から初期値を選択する」を指定すると、初期値用のデータエリアが出力論理マップ中に生成されます。ここに値を代入すると、ポップアップテキストの初期表示時に、初期値に対応するメニュー項目が選択された状態で表示されます。

### (9) メニュー項目数の動的変更 (可変)

ポップアップのダイアログでは、使用するメニュー項目数の最大値を指定しておきます。しかし実際の表示では、メニュー項目数が可変になる場合があります。上から順に項目のデータを代入し、残った項目のラベルに半角の空白 (スペースコード) を代入すると、空白を代入したメニュー項目は表示されません。なお、途中のメニュー項目を詰めて表示することはできません。途中のメニュー項目の空白はデータとみなされます。空白のラベルをメニューの最後に表示させたい場合は、ラベルのデータに全角のスペースを代入してください。メニュー項目数の動的変更の詳細については「6.2.32 コンボボックスの項目数の動的変更」を参照してください。なお、コンボボックスはポップアップに置き換えてお読みください。

### (10) ポップアップメニューをファイルで指定する (可変)

XMAP3 では、ポップアップテキストに表示するメニューデータを AP 実行時にファイルから渡せます。このファイルをポップアップメニューファイルといいます。ポップアップメニューファイルは、あらかじめ用意しておきます。

ポップアップメニューファイルは、メニュー項目が大量にある場合や、メニューを大分類、小分類にして表示したい場合に使用します。また、ホスト連携機能を利用する場合、PC 側にポップアップメニューファイルを持つことで、ホストと PC 間で転送するデータ量を減らせます。ポップアップメニューファイルの使い方については、「6.2.18 ポップアップメニューファイルを使用したデータの入力」を参照してください。

#### (a) ポップアップメニューファイル使用時の注意事項

次にポップアップメニューファイルを使用するときの注意事項を示します。

ポップアップメニューファイルの準備

ポップアップメニューファイルを使用する場合は、あらかじめポップアップメニュー

ファイルを作成し、各クライアント PC 側に用意しておきます。ポップアップメニューファイルは、ユティリティの「ポップアップメニューエディタ」で作成できます。また、ポップアップメニューファイルを格納しておくフォルダは、実際に表示を実行する PC の「ポップアップメニューフォルダ (ディレクトリ)」です。ポップアップメニューフォルダのパスは、表示・印刷セットアップで設定します。ポップアップメニューファイルの準備および規則の詳細については、マニュアル「XMAP3 開発・実行ガイド」を参照してください。

#### ファイル名の初期値の指定

ポップアップメニューファイルで指定したメニューをテスト表示時に確認したい場合は、準備したポップアップメニューファイル名を、定義画面のファイル名の「初期値」に指定しておきます。このとき、必ず拡張子「.csv」を付けて指定してください。拡張子が「.csv」でないファイル名を指定すると、エラーになります。なお、拡張子は、大文字でも小文字でもかまいません。

#### 注

CSV 形式 (Comma Separated Values 形式)

ポップアップメニューファイルで指定できるメニューの階層

ポップアップメニューファイルで指定できるメニューの階層は、1 階層から 8 階層までです。

ポップアップメニューファイルがない場合

AP で指定したポップアップメニューファイルがなかったり、「メニューデータをファイルで指定する」を指定して、ファイル名の指定をしていなかったりした場合、ポップアップテキストには 1 階層の空のメニューが作成されます。このメニューにフォーカスが位置づくとき、エラーになります。

メニュー項目の制限値を超えている場合

一つのファイルに指定できるメニュー項目数は、32,000 項目です。32,000 項目を超えるメニュー項目が指定されているファイルを読み込んだ場合、超過したメニュー項目は表示されません。

ポップアップメニューファイル読み込みの契機 (タイミング)

ポップアップメニューファイルを読み込むタイミングは、表示・印刷セットアップで設定できます。メニュー項目が大量の場合など、必要に応じて読み込むタイミングの設定を変更してください。ポップアップメニューファイル読み込みのタイミングの設定については、マニュアル「XMAP3 開発・実行ガイド」を参照してください。

選択ラベルおよびアクセスキーの利用

ポップアップメニューファイルでの定義を有効にするために、次の点に注意してください。

- 選択ラベルを定義している場合

ポップアップメニューファイルで選択ラベルを定義している場合は、ドロワーのポップアップメニューダイアログで「選択ラベルと通知コードを同じ値にする」の指定を外してください。

## 2. GUI 画面の作成

- アクセスキーを定義している場合  
ポップアップメニューファイルでアクセスキーを定義している場合は、ドロウのポップアップメニューダイアログで「アクセスキーを指定する」を指定してください。

### 選択ラベルおよびラベルの大きさの指定

選択ラベルおよびラベルの大きさは、ポップアップメニューファイルで定義した選択ラベルおよびラベルのデータ長にかかわらず、ドロウで定義した大きさになります。ポップアップメニューダイアログの、選択ラベルの「桁」およびラベルの「桁」で大きさを指定してください。指定できる「桁」の大きさについては、「付録 D 画面定義の制限値」を参照してください。

### 使用目的の指定

ポップアップメニューファイルで定義する通知コードを入力データとして正しく扱うには、ポップアップテキストの使用目的の指定を考慮する必要があります。使用目的と通知コードの関係については、「付録 A.9 表示属性の解説」の表 A-8 を参照してください。

AP 実行時の通知コードの扱いについては、「5.4.1(2) ポップアップフィールドの論理マップ生成規則とマッピング規則」を参照してください。

### 階層化されたメニューでのアクセスキー入力

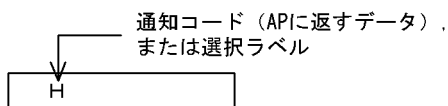
階層化されたポップアップメニューでは、アクセスキー入力時に表示されている、最下位のメニュー項目が選択対象になります。階層を超えて、別の階層のアクセスキーを入力できません。上位の層のメニュー項目のアクセスキーを入力する場合は、カーソル移動キーで対象の階層の表示に戻してから入力してください。また、最下位のメニュー項目を選択する場合、マウス操作やカーソル移動キーの操作で最終的に選択するメニュー項目を表示させて選択するか、手動ポップアップであれば、最終的に AP に渡される通知コードを、直接、テキストボックスへ入力して選択してください。

### (b) メニューの表示 (標準メニューの場合)

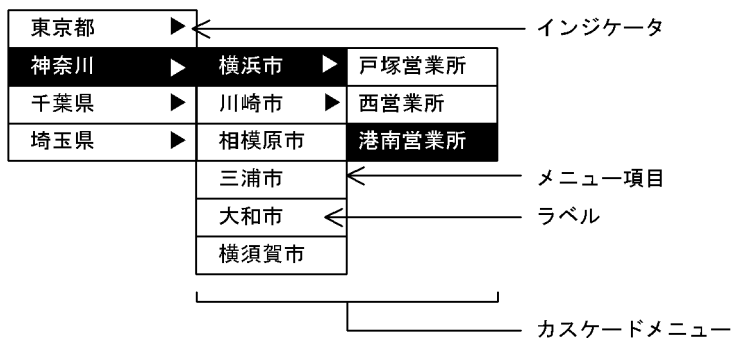
ポップアップメニューファイルを使用する場合、メニュー表示の形式を選択できます。メニュー形式で「標準 (メニュー)」を指定すると、ポップアップメニューファイルに格納されているメニュー項目が階層構造の場合は、下位のメニュー項目がカスケードメニューで表示されます。

## 表示形式

## ボックス側(テキスト)



## メニュー側



## メニュー項目の選択

標準表示のメニュー項目は、マウス操作でもキー操作でも選択できます。

## • マウスでの操作

マウス操作でメニュー項目を選択する場合、ドラッグするだけでは、下位階層のカスケードメニューは表示されません。必ず、インジケータが付いているメニュー項目をクリックしてください。

## • キーでの操作

フォーカスの移動：上下のカーソルキー。

下位階層のカスケードメニューの表示：インジケータが付いているメニュー項目にフォーカスを位置づけ、[スペース]キーまたは[Enter]キーを押す。

キー操作の詳細については「4. マウスとキー操作」を参照してください。

## 選択後のメニュー表示

標準表示では、メニュー項目を選択後、メニューを閉じるかどうかを選択できます。

## アクセスキーの使用

アクセスキーを入力する場合は、必ずメニューを表示してください。入力したアクセスキーに該当するメニュー項目にインジケータが付いている場合は、下位階層のカスケードメニューが表示され、アクセスキーに対応するメニュー項目にフォーカスが位置づきます。

## 通知コードの使用

通知コードをAPから指定したり、メニューが表示されていない状態で、入出力テキストボックスに通知コードを入力すると、指定された通知コードに対応するラベルが選択状態になります。このとき、APに制御が渡されると、指定された通知コードを

## 2. GUI画面の作成

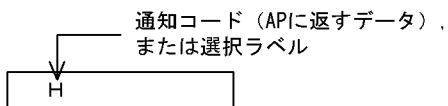
APが受け取ります。

### (c) メニューの表示 (リストの場合)

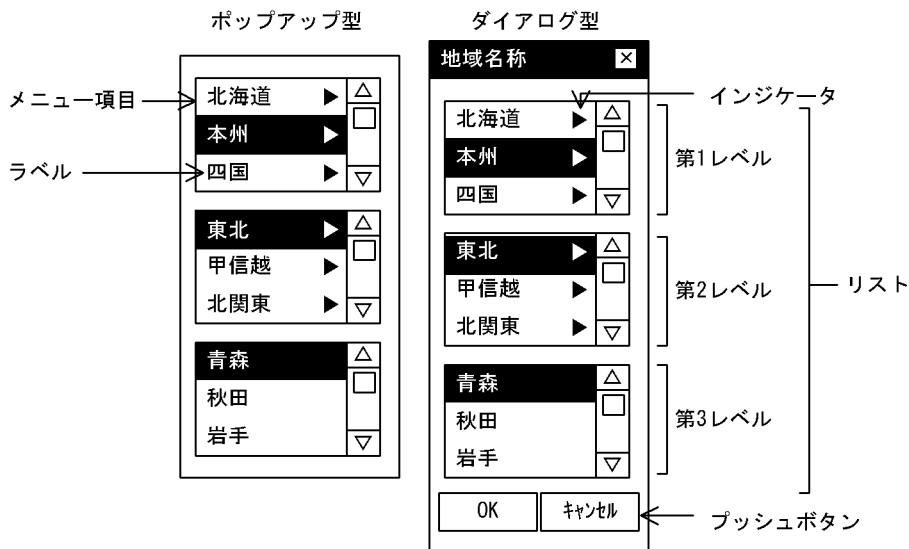
ポップアップメニューファイルを使用する場合、メニュー表示の形式を選択できます。メニュー形式で「リスト - ポップアップ型」または「リスト - ダイアログ型」を指定すると、ポップアップメニューファイルに格納されているメニュー項目が階層構造の場合、下位のメニュー項目は下位レベルのリストとして表示されます。メニュー形式で「リスト - ポップアップ型」を指定すると、メニューをポップアップ型で表示できます。「リスト - ダイアログ型」で指定すると、メニューをダイアログ型で表示できます。

表示形式

ボックス側(テキスト)



メニュー側



ポップアップ型とダイアログ型の違い

ポップアップ型とダイアログ型の違いを示します。

ポップアップ型

- ポップアップ型は、階層構造のメニュー項目を第1～第8レベルのリストで表示します。
- ポップアップ型では、下位項目を持たないメニュー項目にフォーカスが位置づいたとき、メニュー項目が選択されたとみなされます。
- ポップアップ型では、メニュー項目を選択後、メニューを閉じるかどうかを選択で



きます。

- ポップアップ型が表示されても、実行画面は不活性になりません。
- ポップアップ型で表示しているメニューは、単体では移動できません。
- 使用目的で「POP（手動）」が指定されている場合、メニュー項目が選択されて、実行画面のテキストボックスにデータが入力されるときに、メニューは閉じられます。「POP（自動）」が指定されている場合は、メニューは閉じられません。
- ポップアップ型で表示しているメニューには、タイトルを付けません。

#### ダイアログ型

- ダイアログ型は、階層構造のメニュー項目を第1～第8レベルのリストで表示し、[OK] ボタンと[キャンセル] ボタンを付けます。
- ダイアログ型では、[OK] ボタンをクリックしたときにフォーカスが位置づいていたメニュー項目が、選択されたメニュー項目とみなされます。このとき、選択されたメニュー項目は下位項目を持たないメニュー項目でなければなりません。下位項目を持っているメニュー項目を選んで[OK] ボタンをクリックすると、エラーになります。
- ダイアログ型では[OK] ボタンをクリックすると、選んだメニュー項目を確定し、メニューを閉じます。
- ダイアログ型が表示されると、実行画面は不活性になります。
- ダイアログ型で表示している場合、メニューだけの移動ができます。
- ダイアログ型で表示しているメニューには、タイトルが必ず付けられます。

#### メニュー項目の選択

リスト表示のメニュー項目は、マウス操作でもキー操作でも選択できます。

- キーでの操作
  - 同レベル内のフォーカスの移動：上下のカーソルキーを押す。
  - メニュー項目の選択：[スペース] キーまたは[Enter] キーを押す。
  - 第2～第8レベルのリストの表示：インジケータが付いているメニュー項目にフォーカスを位置づけ[スペース] キーまたは[Enter] キーを押す。
  - メニュー内でのフォーカス移動：[Tab] キーを押す。
  - 第1レベルにフォーカスを戻す：[Home] キーを押す。

なお、[キャンセル] ボタンにフォーカスが位置づいているときに[Tab] キーを押すと、メニューは閉じられます。

キー操作の詳細については「4. マウスとキー操作」を参照してください。

#### 注

ダイアログ型で表示している場合、フォーカスはプッシュボタンにも位置づきません。

#### アクセスキーの使用

アクセスキーを入力すると、対応するメニュー項目が選択状態になります。対応するメニュー項目にインジケータが付いている場合は、第2～第8レベルのリストが表示され、アクセスキーに対応するメニュー項目にフォーカスが位置づきます。

## 2. GUI画面の作成

### 通知コードの使用

通知コードをAPから指定したり、メニューが表示されていない状態で、入出力テキストボックスに通知コードを入力すると、指定された通知コードに対応するラベルが選択状態になります。このとき、APに制御が渡されると、指定された通知コードをAPが受け取ります。

### メニュー項目数の動的変更

リスト表示をしているポップアップメニューでは、メニュー項目数の動的変更はできません。

## 2.4.11 スピンボックス

スピンボックスでは、ボックスに表示されている数値データを、ボックスの右端にあるアップダウンボタンで増減させて指定します。データを増減するときの値（増分値）はドロウ定義時に設定します。APからは変更できません。

なお、スピンボックスの最小値と最大値の差分は、増分値の倍数になるようドロウで定義してください。最大値に、最小値と最大値の差分が増分値の倍数にならない値を指定した場合には、増加の最終値は最大値になります。また減少操作での最終値は最小値となります。また、最大値後に減少操作をした場合の値は、増加操作時の変位と異なる変位を取るようになります。

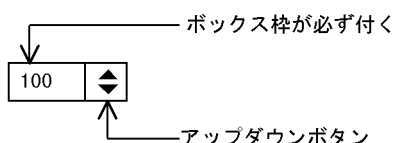
(例)

最大値「13」、最小値「3」、増分値「4」の場合

この場合、最小値と最大値の差分は「10」となるので、増分値「4」の倍数にはなりません。このときのアップダウンボタンでの値の動きを次に示します。

1. 初期値が「3」の状態では値を増加させると、増分値が「4」のため、「7」「11」「15」の順に値が増加します。ただし、この場合、最大値に「13」を指定しているため、値は「7」「11」「13」の順に増加し、最終値は「13」になります。
2. 最終値「13」の状態からアップダウンボタンで値を減少させると、「9」「5」「3」の順に値が減少していきます。  
このように、最小値と最大値の差分が増分値の倍数にならない場合、減少操作をすると、増加したときと異なる変位を取ります。

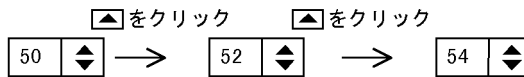
(1) 表示形式



## (2) スピンドボックスの例

### (a) 増分値について

最小値を「1」、最大値を「100」とし、増分値を「2」、APからの表示データを「50」で設定した場合。



### 注

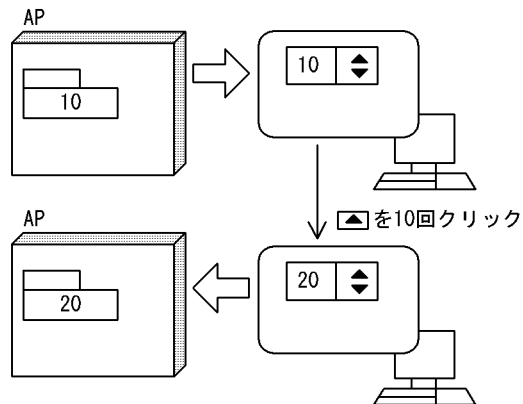
増分値には数字だけ指定できます。

### (b) スピンドボックスとAPの関係について

スピンドボックスとAPの関係を示します。

#### スピンドボックスの定義

最大値：100  
最小値：0  
増分値：1



### (3) 出力時のAPインタフェース

APから、論理マップに初期値を設定します。

### (4) 入力時のAPインタフェース

最終的に表示された値が、APに渡ります。

### (5) 表示属性の動的変更

スピンドボックスの表示属性をAP実行時に変更できます。表示属性の動的変更について

## 2. GUI 画面の作成

は「1.5.6 表示属性の動的変更」を参照してください。

### 2.4.12 リストボックス

画面表示しているリストの中から選択させるときに使用します。表示するデータ量や選択肢の数が多きときに使用することをお勧めします。例えば、マスタファイル検索結果としての複数のデータの中からデータを選択させる場合などに適しています。表示する範囲が限られていても、スクロールバーを付けて定義した表示範囲よりも大きなリスト項目を表示できます。

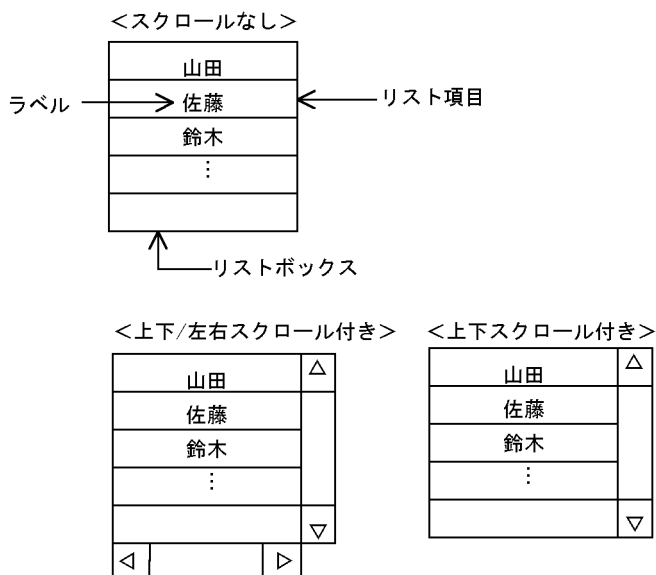
リストボックスには、表示データから一つのリスト項目だけ選択できる「単一選択リストボックス」と、複数のリスト項目を選択できる「複数選択リストボックス」があります。単一選択リストボックスでリスト項目を選択する場合、フォーカスが位置づいたリスト項目を自動的に選択する方法と、フォーカスに関係なくキー操作やマウス操作などでリスト項目を選択する方法があります。どちらの方法でリスト項目を選択するかは、表示・印刷セットアップで設定します。詳細は、マニュアル「XMAP3 開発・実行ガイド」を参照してください。

リストボックスの表示属性の標準値は文字色が「黒」、「ボックス枠を付ける」、「垂直スクロールバーを付ける」です。リストボックスの表示属性の詳細については、「付録 A 属性情報一覧」を参照してください。

#### 注

フォーカスが位置づいていない状態でリストボックスのスクロールバーをマウスでスクロールしても、リストボックスにフォーカスは移動しません。リストボックスにフォーカスを位置づけたい場合は、[ Tab ] キーを押すか、リスト項目をマウスでクリックしてください。

### (1) 表示形式



### (2) 使用目的

使用目的によって、入力データをチェックできます。ただし、ラベルのデータ型は、使用目的の指定と連動しません。ラベルのデータ型の「文字 (XX)」は「日本語」の属性になります。使用目的と入力データの関係については、「付録 A.9 表示属性の解説」の表 A-8 を参照してください。

### (3) 入出力時の AP インタフェース

リストボックスに表示するリスト項目のラベルと通知コードは AP で指定します。リストボックスからリスト項目を選択したあとで画面が確定すると、選んだリスト項目のラベルに対応した通知コードを入力論理マップに格納します。

### (4) 表示属性の動的変更

リストボックスの表示属性を AP 実行時に変更できます。表示属性の動的変更については「1.5.6 表示属性の動的変更」を参照してください。

### (5) リスト項目数の動的変更

リストボックスのダイアログでは、使用するリスト項目数の最大値を指定しておきます。しかし実際の表示では、リスト項目数が可変になる場合があります。上から順に項目のデータを代入し、残った項目のラベルに半角の空白 (スペースコード) を代入すると、空白を代入したメニュー項目は表示されません。なお、途中の項目を詰めて表示することはできません。途中のリスト項目の空白はデータとみなされます。リスト項目数の動的変更の詳細については「6.2.32 コンボボックスの項目数の動的変更」を参照してくだ

さい。なお、コンボボックスはリストボックスに置き換えてお読みください。

## 2.4.13 ラジオボタンボックス

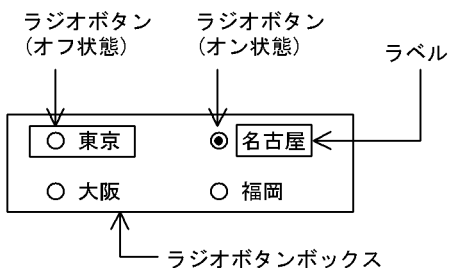
入力データを、画面表示している候補（ラジオボタンボックス）をオン状態にして指定します。ラジオボタンでは、ラジオボタンボックスの中からボタンを一つだけ選択できます。

ラジオボタンボックスでラジオボタンを選択する場合、フォーカスが位置づいたラジオボタンを自動的に選択する方法と、フォーカスに関係なくキー操作やマウス操作などでラジオボタンを選択する方法があります。どちらの方法でラジオボタンを選択するかは、表示・印刷セットアップで設定します。詳細は、マニュアル「XMAP3 開発・実行ガイド」を参照してください。

ラジオボタンボックスには、ボタンラベルをドロウで定義する「固定ラジオボタン」と、AP から指定する「可変ラジオボタン」があります。ラジオボタンは、表示データ量や選択肢の数が少ないときに使用します。また、「YES」か「NO」の選択などの単純な選択をするのに適しています。

なお、ラジオボタンボックスの表示属性の詳細については、「付録 A 属性情報一覧」を参照してください。

### (1) 表示形式



#### 注 1

ラベルのないボタンも作成できます。

#### 注 2

ボタンの並びは、ラジオボタンボックスで「ボタンを横に並べる」を指定したときは左から右へ、指定していないときは上から下へとなります。

### (2) 使用目的

使用目的によって、入力データをチェックできます。ただし、ラベル（可変）のデータ型は、使用目的の指定と連動しません。ラベルのデータ型の「文字 (XX)」は「日本語」の属性になります。使用目的と入力データの関係については、「付録 A.9 表示属性の解説」の表 A-8 を参照してください。

### (3) 入出力時の AP インタフェース

ラジオボタンには、固定ラジオボタンと可変ラジオボタンがあります。固定ラジオボタンはラベル、およびラベルに対応する通知コードを画面定義時に指定します。可変ラジオボタンはラベル、およびラベルに対応する通知コードを AP で指定します。

ラジオボタンボックスからボタンを選択したあとで画面を送信すると、ボタンに対応する通知コードが AP の入力論理マップに格納されます。

### (4) 表示属性の動的変更

ラジオボタンの活性 / 不活性や表示属性を AP 実行時に変更できます。表示属性の動的変更については「1.5.6 表示属性の動的変更」を参照してください。

## 2.4.14 チェックボタンボックス

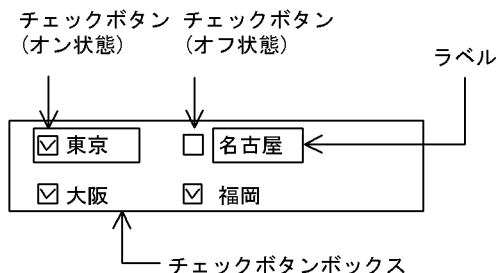
入力データを、画面表示している候補（チェックボタンボックス）をオン状態にして指定します。チェックボタンでは、チェックボタンボックスの中からボタンを複数選択できます。

チェックボタンには、ボタンラベルをドロウで定義する「固定チェックボタン」と、AP から指定する「可変チェックボタン」があります。

チェックボタンは、表示データ量や選択肢の数が少ないときに使用します。また、ボタンを選択しなくてもエラーにならないため、オプション的な選択をするのに適しています。チェックボタンでは、表示データ量は漢字で最大 32 文字、選択肢の数は 64 個までですが、表示データ量が漢字で 10 文字程度、選択肢の数が 10 個程度のときに使用することをお勧めします。

チェックボタンの表示属性の標準値は文字色が「黒」です。なお、チェックボタンボックスの表示属性の詳細については、「付録 A 属性情報一覧」を参照してください。

### (1) 表示形式



#### 注 1

ラベルのないボタンも作成できます。

## 2. GUI 画面の作成

### 注 2

ボタンの並びは、チェックボックスで「ボタンを横に並べる」を指定したときは左から右へ、指定していないときは上から下へとなります。

### (2) 使用目的

使用目的によって、入力データをチェックできます。ただし、ラベルのデータ型は、使用目的の指定と連動しません。ラベルのデータ型の「文字 (XX)」は「日本語」の属性になります。使用目的と入力データの関係については、「付録 A.9 表示属性の解説」の表 A-8 を参照してください。

### (3) 入出力時の AP インタフェース

チェックボタンには、固定チェックボタンと可変チェックボタンがあります。固定チェックボタンはラベル、およびラベルに対応する通知コードを画面定義時に指定します。可変チェックボタンはラベル、およびラベルに対応する通知コードを AP で指定します。

チェックボックスからボタンを選択したあとで画面を送信すると、ボタンに対応する通知コードが入力論理マップに格納されます。

### (4) 表示属性の動的変更

チェックボタンの活性 / 不活性や表示属性を AP 実行時に変更できます。表示属性の動的変更については「1.5.6 表示属性の動的変更」を参照してください。

## 2.4.15 トグルフィールド

表の各行の末尾にフィールドを付け、「確認」や「了解」「承認」などの印を付けます。また、「オン / オフ」での選択ができます。「オン / オフ」は「」、「YES / NO」および「 / x」で表されます。オン状態の印が付けられたデータは、入力データとして AP に渡されます。

なお、インストール直後にトグルフィールドが正しく表示されない場合、表示に必要なフォントが OS に認識されていない可能性があります。OS を再起動してください。

### (1) 表示形式

トグル種別	チェック	/ x	Y / N
オン	<input checked="" type="checkbox"/>		Y
オフ	<input type="checkbox"/>	x	N

### 注

サイズは文字と同じです。全角 1 文字に相当します。



## (2) 使用目的

使用目的によって、入力データをチェックできます。使用目的と入力データの関係については、「付録 A.9 表示属性の解説」の表 A-8 を参照してください。

## (3) 入力時の AP インタフェース

オン状態のとき、通知コードが AP に返ります。

## (4) 表示属性の動的変更

トグルフィールドの表示属性を AP 実行時に変更できます。表示属性の動的変更については「1.5.6 表示属性の動的変更」を参照してください。

## 2.4.16 グループボックス

画面上のオブジェクト（セパレータを除く）を囲んでグループ化するオブジェクトです。必要に応じて、グループボックス単位にフォーカス遷移の順序を設定できます。

### (1) 表示形式

グループボックスには、ボックス枠やタイトルの有無、および背景色を指定できます。また、タイトルを付けた場合、タイトルの文字色も指定できます。

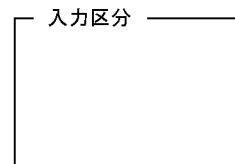
<枠なし>



<枠あり>



<タイトル付き>



### (2) フォーカス順序

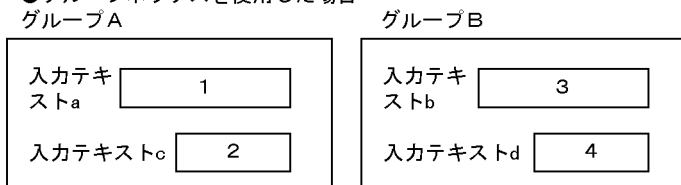
#### (a) グループボックスを使用した場合と使用しない場合

グループボックスを使用した場合と使用していない場合のフォーカスの移動順序を次の図に示します。

## 2. GUI 画面の作成

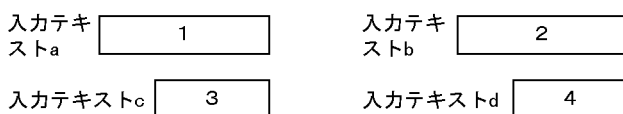
図 2-3 グループボックスを使用した場合と使用していない場合のフォーカスの移動順序

### ●グループボックスを使用した場合



- ・グループボックス単位で A→B の順にフォーカスが移動する
- ・グループボックス内で A (a→c), B (b→d) と移動する

### ●グループボックスを使用しない場合



a→b→c→d の順 (左上から右下) にフォーカスが移動する

### (b) フォーカス順序を設定した場合

複数の入力オブジェクトをグループ化すると、グループボックス間のフォーカスの移動順序を設定できます。フォーカス順序の設定の詳細については、「2.4.26 フォーカスとカーソル」を参照してください。

### (3) タイトルの設定

グループボックスにタイトルを付けられます。どのような種類のオブジェクトがグループ化されているかがわかるようなタイトルを付けたり、目立つような色で表示したりすると、画面上での検索に役立ちます。タイトルはグループボックスダイアログで設定します。

## 2.4.17 フィールドボックス

多量のデータをまとめて入出力するためのボックスオブジェクトです。

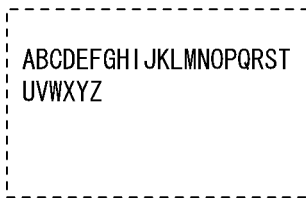
次のような場合に適しています。

- ・ けい線を使って、表形式でオブジェクトを並べる
- ・ CUI 画面を基本にして一部だけを GUI 化 (ファンクションキーによる選択部分だけのボタン化など)
- ・ 文章データなど、複数行にわたる文字データの表示

なお、フィールドボックスの表示属性の詳細については、「付録 A 属性情報一覧」を参照してください。

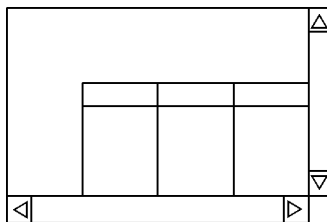
## (1) 表示形式

&lt;枠なし、2行にわたる文章&gt;



&lt;枠あり、表&gt;


&lt;スクロールバーあり&gt;



注

フィールドボックスは1画面に4個まで配置できます。

## (2) フィールドボックスの属性

フィールドボックスでは次に示す表示属性などが指定できます。詳細は「付録A 属性情報一覧」を参照してください。

- ボックス枠の付加
- スクロールバーの付加

表示データ(プレーン)がボックスサイズより大きい場合、スクロールすることで全体を表示できます。

注

フォーカスが位置づいていない状態でフィールドボックスのスクロールバーをマウスでスクロールしても、フィールドボックスにフォーカスは移動しません。フィールドボックスにフォーカスを位置づけたい場合は、[ Tab ] キーを押すか、表示データ(プレーン)の領域をマウスでクリックしてください。

入力可能なオブジェクトがフィールドボックス内に存在しない場合でも、キー操作でフィールドボックスをスクロールするため、フィールドボックスにフォーカスを移動できます。

- スクロール禁止領域

常にデータを表示しておくための、スクロールしない領域を指定できます。

フィールドボックスは、表示属性の標準値として背景色が透明、けい線、カーソル、およびアンダードットの色が黒で設定されています。

フィールドボックスでは、表示属性としてけい線やスクロール禁止領域の表示色など

## 2. GUI 画面の作成

が指定できます。詳細については、「付録 A 属性情報一覧」を参照してください。

- 初期フォーカスを本ボックスに設定する

### (3) カーソル表示

入力可能なオブジェクト がフィールドボックス内に定義されている場合、そのオブジェクトすべてに標準属性または動的変更によって入力不可が指定され、フィールドボックス内に入力できるオブジェクトが一つもないときでも、フォーカスが位置づくときカーソルは表示されます。入力可能なオブジェクト がフィールドボックス内に定義されていない場合、カーソルは表示されません。

注

入力可能なオブジェクトとは、入出力フィールド、入出力日付フィールド/入出力時刻フィールド、ポップアップフィールド、トグルフィールドを指します。

### (4) フィールドボックス内の文字とけい線

文字のサイズ、書体、強調およびけい線の色はフィールドボックス単位で指定できます。文字の色やけい線の種別は、オブジェクトごとに指定できます。文字拡大（サイズ）および強調の例を次に示します。

なお、フィールドを斜体で配置する場合は、右端に 1 ますの余白が必要です。また、フィールドボックス内で文字を斜体にすると最後の文字が正しく表示されないことがあります。実際の表示には影響しません。

<拡大/強調>

日立 日立 日立 日立

標準文字 平体 長体 倍角

日立 日立 日立

太字 斜体 太字&斜体

### (5) AP インタフェース

フィールドボックス自体は AP とのインタフェースは持ちません。背景色などの表示属性はドローで定義します。

フィールドボックス内のオブジェクトのインタフェースについては、各オブジェクトの説明を参照してください。

## (6) フィールドボックス内のオブジェクト

フィールドボックスの中には次に示すオブジェクトを配置できます。

- 固定フィールド
- 出力フィールド
- 入出力フィールド
- 入出力日付フィールド / 入出力時刻フィールド
- 出力日付フィールド / 出力時刻フィールド
- 予約フィールド
- ポップアップフィールド
- トグルフィールド
- 塗りつぶしフィールド
- けい線
- フレーム

### (a) 固定フィールド

固定フィールドは、画面上の表のタイトルや注釈などの変化しないテキストを配置するときに使用します。固定のテキストとして、漢字を含めた日本語、または半角の英数字を設定できます。

#### 表示形式

表示されている文字に対し、文字サイズを指定できます。

#### 主な機能

- 使用目的の指定  
表示する文字に応じて、ドロワー上で使用目的（日本語、英数）を選べます。
- 複数行にわたるサイズの配置  
行を折り返して複数行にわたるサイズで配置できます。ただし、文字の拡大や反復定義との併用はできません。

### (b) 出力フィールド

出力フィールドは、APで指定したデータを表示するときに使用します。キーボードやマウスを使用しての文字入力はできません。テキストの表示属性の標準値は文字色が「黒」、文字の拡大が「標準」です。指定できる表示属性の詳細については、「付録A 属性情報一覧」を参照してください。

#### 表示形式

フィールドボックスのダイアログの基準文字サイズで、表示するデータの文字サイズを「大」「中」「小」から選べます。

#### 主な機能

- 反復表示  
一つのオブジェクトを縦または横方向に繰り返して定義できます。表形式で、同じ属性を並べるときなどに使用できます。反復を使用すると、同じ表示属性のオブジェクトを一つずつ配置するよりも効率良く定義できます。反復の操作は [ Shift ]

## 2. GUI 画面の作成

キーを押しながらマウスをドラッグすることで、オブジェクトを縦方向に繰り返して定義できます。反復したオブジェクトのデータ項目は配列（COBOL の場合、OCCURS 展開）となります。

- 使用目的の指定  
表示する文字に応じて、使用目的（数字、英数、日本語）を選べます。
- 複数行にわたるサイズの配置  
行を折り返して複数行にわたるサイズで配置できます。ただし、文字の拡大や反復定義との併用はできません。
- 出力データの編集  
出力フィールドでは、埋字や桁寄せなどのデータの編集もできます。AP がデータ有無コードを格納し、出力論理マップにセットした有効データの長さが出力フィールドデータ名の領域の長さより短い場合、出力フィールドデータ名の領域の余った部分を埋字に置き換えます。また、出力データを出力フィールドデータ名の領域の右側に格納するか、左側に格納するかは、桁寄せ向きに従います。

### 出力データ

出力フィールドに設定できるデータを次に示します。

- 数字  
数字データ、数字編集データを表示できます。数字編集データは、指定した PICTURE 句に従って表示されます（COBOL だけ有効）。  
使用できるピクチャ文字：「\*」「+」「-」「¥」「9」「,」「.」「Z」「/」の組み合わせで指定します。
- 日本語  
全角、または全角と半角の混在文字データを表示できます。
- 英数  
半角の文字データを表示できます。
- 実行時の属性変更の AP インタフェース  
出力フィールドの表示属性を AP 実行時に変更できます。表示属性の動的変更については「1.5.6 表示属性の動的変更」を参照してください。

### (c) 入出力フィールド

入出力フィールドは、キーボードなどからデータを入力するとき、および AP から渡されるデータを表示するときに使用します。入出力するデータの意味に応じて、種類（使用目的、詳細目的）を選びます。

初期表示値（出力データ）を表示し、書き換えが必要な場合だけ入力するような使い方ができます。

テキストの表示属性の標準値は文字色が「黒」です。指定できる表示属性の詳細については、「付録 A 属性情報一覧」を参照してください。

なお、入出力フィールドの詳細については「2.4.4 入出力テキストボックス」を参照してください。「入出力テキストボックス」は「入出力フィールド」と置き換えてお読みください。ここでは、入出力テキストボックスと異なる点を示します。

## 表示形式

入出力テキストボックスと同じです。ただし、入出力フィールドの場合、アンダードットが付きます。枠は付きません。また、入力データの長さが、入力開始位置から行の最後までまでの長さを超える場合、入力しきれないデータを継続行として次行に入力できます。このとき、データの入力開始位置はフィールドのどこであってもかまいません。

入出力フィールドが複数行にわたる場合、復改キーでカーソルの改行動作が行われます。改行入力属性を指定している場合、改行位置に改行コードが付きます。改行入力属性を指定していない場合、改行コードは付きません。

改行入力属性を指定している場合、データの途中の改行コードは、画面表示、および画面入力時に改行コードとして入出力されます。

改行入力属性を指定していない場合、データの途中の改行コードは、画面表示、および画面入力時に不当な文字データとして入出力されます。

なお、改行入力属性を指定している場合に、改行位置からフィールドの末尾までにデータが入力されている、または埋字に「LOW (X'00')」以外が指定されていると、改行コードが挿入されないことがあります。そのため、埋字に「LOW (X'00')」と指定するか、表示後に文字データ以降をフィールドキーでクリアしてください。

(例1)

アンダードットの例

12345678

(例2)

複数行にわたるデータの例



## 反復表示

一つのオブジェクトを縦または横方向に繰り返して定義できます。表形式で、同じ属性を並べるときなどに使用できます。反復を使用すると、同じ表示属性のオブジェクトを一つずつ配置するよりも効率良く定義できます。反復の操作は [ Shift ] キーを押しながらマウスをドラッグすることで、オブジェクトを縦方向に繰り返して定義できます。反復したオブジェクトのデータ項目は配列 (COBOL の場合、OCCURS 展開) となります。

(d) 日付フィールドと時刻フィールド (入出力 / 出力)

日付フィールドと時刻フィールド (入出力) の詳細は「2.4.5 日付テキスト・フィールドと時刻テキスト・フィールド (入出力)」を参照してください。日付フィールドと時刻フィールド (出力) の詳細は「2.4.8 日付テキスト・フィールドと時刻テキスト・

## 2. GUI 画面の作成

フィールド（出力）」を参照してください。

### (e) ポップアップフィールド

ポップアップフィールドは、決められた候補（メニュー項目）から一つを選択するとき  
に使用します。メニュー項目に入力データの候補を表示することで、データの入力を補  
助します。入力するデータの内容が決まっている場合に使用することをお勧めします。  
なお、ポップアップフィールドの表示属性の詳細については「付録 A 属性情報一覧」  
を参照してください。ポップアップフィールドの詳細については「2.4.10 ポップアップ  
テキスト」を参照してください。その場合、「ポップアップテキスト」は「ポップアップ  
フィールド」と置き換えてお読みください。ここでは、ポップアップテキストと異なる  
点を示します。

#### 注

ポップアップフィールドでは、ポップアップメニューファイルは利用できません。

#### 表示形式

ポップアップテキストと同じです。ただし、枠は付きません。また、キー入力できる  
手動ポップアップのときは、アンダードットが付きます。

(例)

.....

#### 反復表示

一つのオブジェクトを縦または横方向に繰り返して定義できます。表形式で、同じ属  
性を並べるときなどに使用できます。反復を使用すると、同じ表示属性のオブジェク  
トを一つずつ配置するよりも効率良く定義できます。反復の操作は [ Shift ] キーを押  
しながらマウスをドラッグすることで、オブジェクトを縦方向に繰り返して定義でき  
ます。反復したオブジェクトのデータ項目は配列（COBOL の場合、OCCURS 展開）  
となります。

### (f) トグルフィールド

「YES / NO」などの印を付けるためのオブジェクトです。表の各行末に配置して、「オ  
ン / オフ」や「済 / 未」を表す場合などに使います。トグルフィールドの詳細は「2.4.15  
トグルフィールド」を参照してください。

### (g) 塗りつぶしフィールド

フィールドボックス内の領域に色を付けるオブジェクトです。ほかのオブジェクトを重  
ねて配置できます。1 行おきに色のパターンを付けて行を見やすくする場合などに使いま  
す。塗りつぶしフィールドの詳細は「2.4.23 塗りつぶしフィールド」を参照してくださ  
い。

### (h) けい線

配置した複数のオブジェクトを区切ったり、強調させたりするオブジェクトです。けい



線の詳細については「2.4.21 けい線」を参照してください。

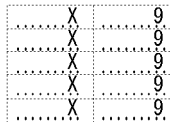
(i) フレーム（横／縦繰り返し）

フレームで指定した領域内の先頭行または先頭列にオブジェクトを配置すると、配置したオブジェクトがフレームで指定した行数分または列数分、反復表示されます。入出力フィールドと出力フィールドを1行として反復表示することもできます。APでは、この1列および1行を集団項目として扱います。また、論理マップでは反復表示されたオブジェクトを配列として展開します。

表示形式

フレーム自体は表示されません。ドロー上では、オブジェクトをまとめる点線の枠で表示されます。

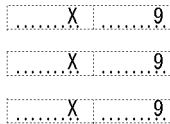
<縦に繰り返す>



<横に繰り返す>



<1行置きに繰り返す>



フレームに配置できるオブジェクト

GUIのフィールドボックスで設定するフレームに配置できるオブジェクトは次のとおりです。なお、フレームの1枠に範囲指定で選択した複数オブジェクトを移動・複写することはできません。コピーしてから貼り付けてください。

- 入出力フィールド
- 固定フィールド
- 出力フィールド
- 入出力日付・時刻フィールド，出力日付・時刻フィールド（デリミタ含む）
- 縦／横けい線
- ポップアップフィールド
- トグルフィールド

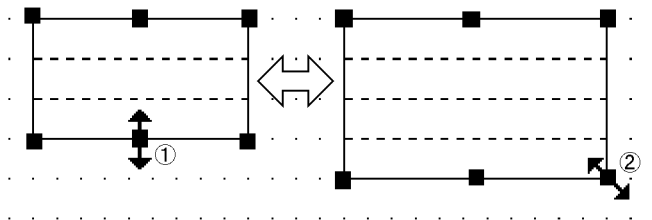
フレームの操作

フレームのハンドル（ ）をドラッグしたり，行または列を選択して行の高さや幅を変更できます。

- フレームの反復回数とサイズの変更

フレーム全体を選択し，表示されるハンドル（ ）をドラッグします。

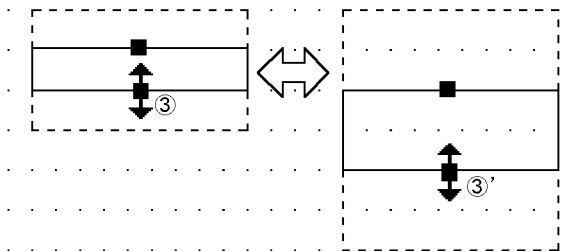
## 2. GUI 画面の作成



- (例) ①を下にドラッグすると反復回数が増加します。  
①を上ドラッグすると反復回数が減少します。  
②を左上にドラッグすると、サイズが小さくなります。  
②を右下にドラッグすると、サイズが大きくなります。

### • 行の高さまたは列の幅の変更

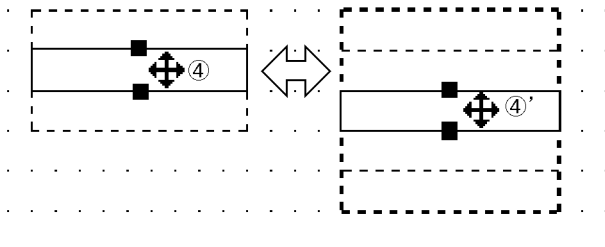
フレームの先頭または最終以外の行（または列）をクリックし、行（または列）に表示されるハンドル（ ）をドラッグします。フレーム内のすべての行の高さ（または列の幅）が変更されます。なお、拡大する場合は、あらかじめフレームの変更する方向の領域を空けておく必要があります。



- (例) ③を下にドラッグすると、行の高さが大きくなります。  
③'を上ドラッグすると、行の高さが小さくなります。

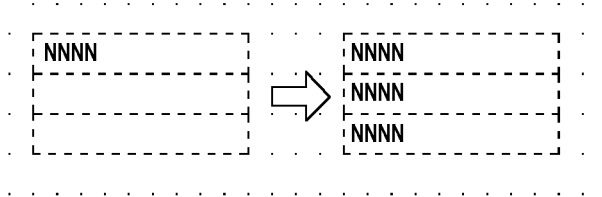
### • フレームの繰り返し間隔の変更

フレームの先頭または最終以外の行（または列）をクリックし、選択した行（または列）をドラッグします。フレームの行（または列）が1行（または1列）おきに繰り返されます。

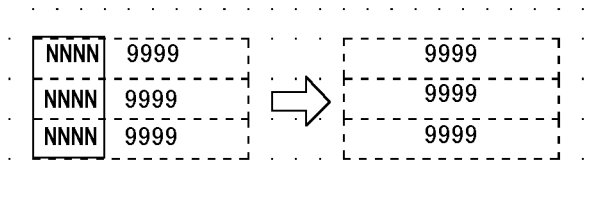


- (例) ④を下にドラッグすると、フレームの間隔が空きます。  
④'を上ドラッグすると、繰り返しがなくなります。

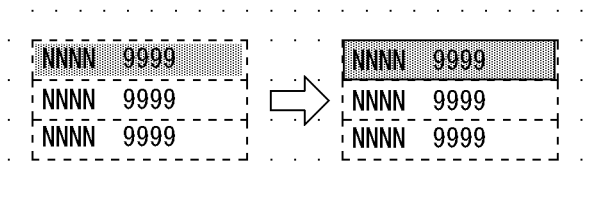
- フレームへのオブジェクトの追加  
1 行目（または 1 列目）にオブジェクトを配置します。配置すると行（または列）方向にオブジェクトが繰り返されます。



- フレーム内のオブジェクトの削除  
1 行目（または 1 列目）のオブジェクトを選択します。行（または列）のオブジェクトが選択状態になったら [ 削除 ] を選択します。



- 塗りつぶしのフィールドを選択する  
塗りつぶしとオブジェクトのフィールドが重なっている場合に、塗りつぶしを選択したいとき、[ Shift ] キーを押しながらクリックすると、塗りつぶしが選択できます。



#### AP インタフェース

フレームは、AP インタフェースを持ちません。フレーム内に配置された各オブジェクトの AP インタフェースは有効です。

### 2.4.18 固定グラフィック

GUI 画面上に、固定グラフィックを貼り付けられます。固定グラフィックを使用するには、あらかじめ Windows 形式の固定グラフィックファイルを用意しておきます。グラフィックファイルのサイズやデータ形式については「付録 D 画面定義の制限値」を参

## 2. GUI 画面の作成

照してください。また、XMAP3 が標準提供しているクリップアートを利用することもできます。

固定グラフィックには、レイアウト領域に配置したあと、ボックスサイズを基準にしてサイズを変更できる表示方法と、レイアウト領域に配置したままのサイズで固定する表示方法があります。XMAP3 の表示領域は行列単位でサイズを指定するのに対し、固定グラフィックはドット単位でサイズを変更する上、ディスプレイの画面解像度によって行列当たりのドット数が異なるため、表示の際に何を優先するかで使い分けてください。

### (1) 固定グラフィックをボックスサイズに合わせて表示する

異なる画面解像度のディスプレイ上で、画面上の固定グラフィックサイズを同じにしたい場合に使用します。固定グラフィックダイアログで「ボックスサイズに合わせて表示する」を指定しておくで、固定グラフィックをレイアウト領域に配置したあとで、指定した領域サイズに合わせて固定グラフィックを拡大・縮小できます。ただし、この場合、読み込んだファイル中の固定グラフィックに比べて表示が不鮮明になることがあります。どのディスプレイ上でも、見た目にあまり差がないような固定グラフィックを表示するためには、領域サイズを次のように決めることをお勧めします。

- 画面の定義前に、異なる画面解像度のディスプレイ上で、どちらの解像度でも固定グラフィックが鮮明に読み込める領域サイズを決める。
- いちばん使用頻度の多い画面解像度のディスプレイに合わせて領域サイズを決める。
- サイズが小さい画像（ビットマップおよび JPEG）を大きい領域に表示すると、正常に表示されないことがあります。この場合、画像自体のサイズを大きくするか、表示領域のサイズを小さくしてください。

### (2) 読み込んだままの固定グラフィックを表示する

固定グラフィックを、読み込んだファイルと同じくらい鮮明に表示したい場合は、固定グラフィックダイアログで「ボックスサイズに合わせて表示する」を指定しません。この場合、固定グラフィックのサイズをレイアウト領域上で変更できないので、表示領域が読み込んだ固定グラフィックファイルよりも小さい場合、はみ出した部分はカットされます。

大きいサイズの固定グラフィックを鮮明に表示するためには、640 × 480 ドットの画面解像度のディスプレイをベースに領域サイズを決めることをお勧めします。

### (3) 固定グラフィックの大きさとカラーの扱い

画面上に配置する固定グラフィックのサイズと、カラーの扱いについては、「付録 D 画面定義の制限値」を参照してください。

## 2.4.19 出力グラフィック

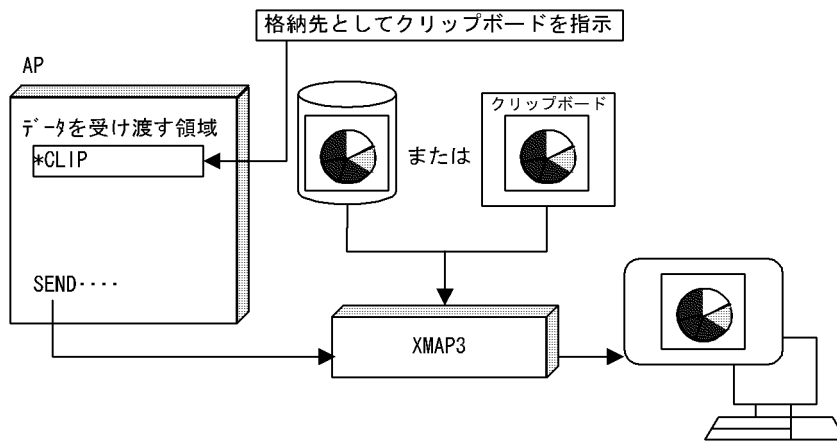
描画ソフトであらかじめ作成しておいたグラフィックファイルを、ファイルやクリップボードを経由して AP が実行時に表示する機能です。グラフィックデータを画面に表示す

る命令は、AP側で指定します。XMAP3ではグラフィックデータを受け渡す領域を定義します。APから表示の命令が来ると、XMAP3は「グラフィックデータの渡し方」に定義したファイル名、または取り込み元がクリップボードである指示を基にしてグラフィックデータを取り込み、画面上に表示します。なおグラフィックファイルのサイズやデータ形式については「付録D 画面定義の制限値」を参照してください。

### (1) 表示処理

グラフィックデータ出力の仕組みを次の図に示します。

図 2-4 グラフィックデータの出力



### (2) 出力時の AP インタフェース

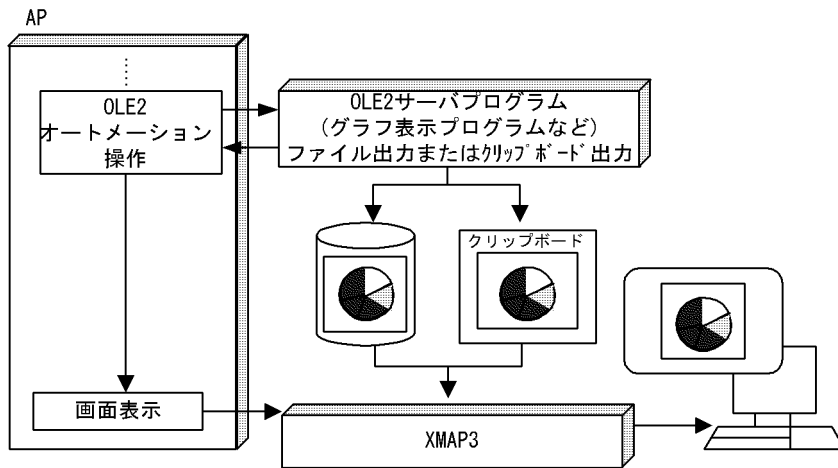
#### イメージデータを表示する場合

イメージデータをスキャナやイメージ OCR などから入力し、ファイルに格納しておきます。APからこのファイル名を論理マップに代入し、SEND要求をすると、XMAP3 実行支援が指定されたファイルからデータを入力して表示します。APから直接イメージデータを論理マップに代入することはできません。グラフィックデータを格納しておくフォルダは実行支援の動作をしている PC の「グラフィックフォルダ (ディレクトリ)」です。グラフィックパスは表示・印刷セットアップで設定します。

#### グラフを表示する場合

描画ソフトを使用してグラフを作成し、グラフィックデータとしてファイルまたはクリップボードに格納します。日立 COBOL で提供する OLE2 オートメーションインタフェース機能を使用することで Excel などのグラフ作成ツールを呼び出せます。APからこのファイル名を論理マップに代入し、SEND 要求をすると、XMAP3 実行支援が指定されたファイルからデータを入力して表示します。APから直接イメージデータを論理マップに代入することはできません。グラフィックデータを格納しておくフォルダは実行支援の動作をしている PC の「グラフィックフォルダ (ディレクトリ)」です。グラフィックパスは表示・印刷セットアップで設定します。

## 2. GUI 画面の作成



### (3) 実行時の注意事項

#### グラフの変更による再表示

グラフを変更して再表示するには、再度グラフ表示アプリケーションを操作して、グラフデータを作成します。OLE2 処理を終了している場合、AP の開始から始める必要があります。ただし、OLE2 処理はほかの AP を起動する処理などを行うため、処理が遅くなることが考えられます。表示したグラフを変更することが考えられる場合、グラフ作成ごとにグラフ表示アプリケーションの開始や終了をするのではなく、表示アプリケーションを起動しておくなどの配慮が必要です。

#### グラフィックファイルのフォーマット

XMAP3 が扱えるグラフィックファイルフォーマットは、ファイル出力の場合はファイル拡張子、クリップボード出力の場合はクリップボード格納形式で判断します。

#### グラフィックファイルのサイズ

グラフィックデータのサイズの制限値は表示処理に必要な Windows リソース（使用できるメモリ量など）に制限される以外はありません。ただし、サイズが大きくなると、データ読み込みや展開などの表示処理に時間が掛かるため、100KB 以内しておくことをお勧めします。

なお、サイズが小さい画像（ビットマップおよび JPEG）を大きい領域に表示すると、正常に表示されないことがあります。この場合、画像自体のサイズを大きくするか、表示領域のサイズを小さくしてください。

#### ウィンドウのサイズ/大きさの制限

- 1 ウィンドウ中の領域の数、または大きさの制限値はウィンドウサイズによる制限以外はありません。ただし、数やサイズが大きくなってくると、表示処理に要する時間も大きくなるため、できる限り 1 個にすることを勧めます。
- クリップボードは OS が用意するものを使用するため、グラフィックデータは 1 度に 1 種類だけの保存・表示になります。そのため、1 ウィンドウの中に、複数グラフィックデータを表示する場合は、ファイルからの読み込みを併用することになり

ます。

- グラフィックファイルの格納場所は、表示・印刷セットアップで指定した特定のディレクトリ下1か所だけになります。ファイル数の制限値はOSの制限値になります。
- グラフィックデータの表示はファイル、またはクリップボード内のデータをそのまま表示します。つまり、回転・縮小・拡大・減色などの操作は、ファイルなどに書き込むAPでオブジェクトのプロパティを操作するなどして、あらかじめ処理しておいてください。ただし、表示領域サイズがグラフィックデータサイズと異なる場合、領域サイズに合うようにグラフィックデータを拡大縮小する処理は、画面定義時のオプションとして指定できます。
- グラフィックデータサイズが表示領域サイズより大きい場合、画面定義時のオプションで、領域サイズに合うようにグラフィックデータを拡大縮小しないで、スクロール表示できます。その際、リストボックスと同様、スクロールバーはグラフィックデータサイズと表示領域サイズの大小に関係なく常に表示されます。

#### グラフィックデータの表示

- グラフィックデータ表示領域はキーボードからデータ入力されない領域のため、タブキーなどでフォーカスが位置つきません。そのため、スクロールバーの操作はマウスだけになります。
- 表示色数が少ないディスプレイ（256色）に対して、表示色が多いグラフィックデータを表示する場合は、OSが色の再割り当てをするため、意図した色が表示されない場合があります。グラフィックデータに使用する色数はできる限り16色以内に抑えた方が安全です。
- グラフィックデータは一般的にデータ量が大きく、そのサイズや個数が多くなるにつれ、処理に要する時間も比例して大きくなります。表示に要する時間を短縮するためには、1画面に表示するグラフィックデータは1個にしたり、大きさを小さくしたりするなどして処理時間を軽減してください。

#### クリップボードの利用

- クリップボードへの入出力は、ファイルへの入出力よりも短い時間で処理できます。表示するグラフィックデータは1個にして、クリップボード経由でXMAP3に引き渡すなどして処理時間を軽減してください。また、ファイルを使用する場合は、RAMディスクに割り当てると表示性能を確保できる場合があります。
- クリップボードはシステムに一つだけです。そのため、クリップボードを使用する複数のプログラムを同時に動作させると、意図したもの以外のグラフィックが表示・印刷されることがあります。画面表示プログラムと帳票印刷プログラムを同時に実行したときも同じことが考えられますので、そのような運用をする場合には、ファイルからの読み込みで対処してください。

## 2.4.20 セパレータ

セパレータは縦方向、または横方向に1本の線を引きます。配置したオブジェクトの間を区切るときに使用します。なお、セパレータは、APとのインタフェースを持ちません。そのため、セパレータを追加、修正、または削除してもAPに影響はありません。

## 2. GUI画面の作成

また、セパレータはグループボックス中に引いたり、ボックスを通過して引いたりすることはできません。

### (1) セパレータの種類

セパレータの種類を次に示します。

#### (a) 凹型

立体的な影が付いて、周りよりも一段低くなっているような線が引けます。3D画面で表示できます。2D画面では立体的に表示しないで、平面的な実線として表示します。

#### (b) 凸型

立体的な影が付いて、周りよりも一段高くなっているような線が引けます。3D画面で表示できます。2D画面では立体的に表示しないで、平面的な実線として表示します。

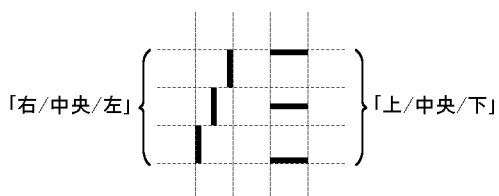
#### (c) 実線 / 点線 / 破線

3D画面および2D画面で表示できます。

### (2) セパレータの位置

セパレータの位置は縦線の場合は「右 / 中央 / 左」、横線の場合は「上 / 中央 / 下」から選べます。これらは縦（または横）セパレータダイアログで設定します。また、横方向のセパレータの開始位置、および終了位置と同じ領域には、テキストやボタンを配置できます。

#### (a) 線の位置



## 2.4.21 けい線

けい線は縦方向、または横方向に1本の線を引きます。配置した複数のオブジェクトの間を区切るときに使用します。

なお、けい線は、APとのインタフェースを持ちません。そのため、けい線を追加、修正、または削除してもAPに影響はありません。また、けい線はフィールドを通過して引いたりすることはできません。

### (1) けい線の線種

けい線の線種を次に示します。



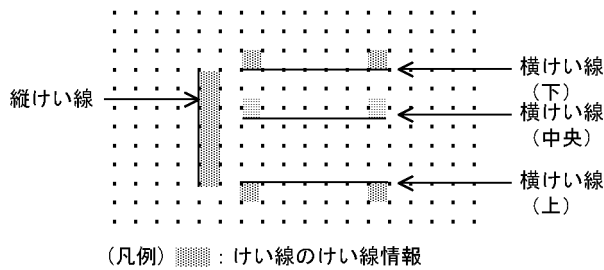
### 実線 / 点線 / 破線 / 二重線

それぞれに線の太さが選べます。線の太さは「中 / 太 / 極太」から選べます。線の種類、太さおよび線の位置はけい線ダイアログで設定します。なお、二重線の場合、線の太さは選べません。

### (2) 線の位置

GUI画面のフィールドボックス内の縦けい線は、列の「右 / 中央 / 左」、横けい線は行の「上 / 中央 / 下」に引けます。

図 2-5 けい線の位置



### (3) 矩形および表けい線

けい線はオブジェクトを区切る以外にも次のように使用できます。なお、線の属性の設定（線種や位置など）はけい線の場合と同じです。

#### (a) 矩形

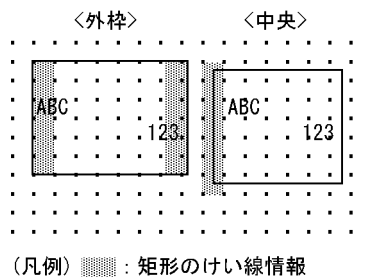
指定した範囲をけい線で囲むときに使用します。

#### 矩形の位置

矩形の位置は「外枠 / 中央」から選べます。これらは矩形ダイアログで設定します。線種および線の太さはけい線と同じです。

外枠の場合は、矩形の内側に隣接したフィールドが配置できます。

図 2-6 矩形の位置



## 2. GUI画面の作成

### (b) 表けい線の枠

表を作成するときに使用します。表に見出しを付けるか付けないかも指定できます。

### (c) 表けい線の繰り返し

縦方向、または横方向に等間隔のけい線を引くときに使用します。横方向の繰り返しの間隔は「2行 / 1行 / 自動」から選べます。

また、けい線は縦けい線の位置が「中央」の場合に半角文字で1文字分のけい線情報用の領域を占有します。このため、けい線を画面の上に配置する場合には、ほかの項目と重ならないようにします。

なお、けい線は複数行にわたるフィールドの途中の行には引けません。また、フィールドを通過して引くこともできません。





## 2.4.22 メッセージアイコン

ユーザに対して何らかの注意をする際に使用します。メッセージアイコンは、1画面に一つだけ配置できます。

基本的にメッセージアイコンは、メッセージと一緒に出力することをお勧めします。メッセージアイコンの種類を次の表に示します。アイコンの種類はドロウ定義時に設定します。

なお、メッセージアイコンはAPとのインタフェースは持ちません。

表 2-10 メッセージアイコンの種類

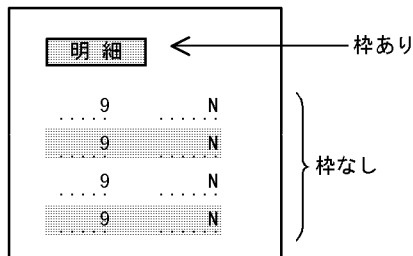
種類	内容
	クエスチョン ユーザに質問するメッセージと一緒に使用します。
	ワーニング ユーザに何らかの注意を喚起したいときに使用します。
	インフォメーション 処理に関する情報を表示するときに使用します。
	エラー エラー時のメッセージと一緒に使用します。

### 2.4.23 塗りつぶしフィールド

フィールドボックス内に配置してあるオブジェクトに対して塗りつぶしの領域をかぶせて指定すると、オブジェクトの背景が塗りつぶされます。画面上に、部分的に背景色が異なる状態を作れます。フレームなどで反復表示されているオブジェクトを行ごとに色分けして見やすくする場合などに利用できます。

すべての背景を塗りつぶす場合は、反復されている領域すべてに対して塗りつぶしの領域をかぶせます。塗りつぶす色は、ドロー定義時に設定します。

#### (1) 表示形式



### 2.4.24 プッシュボタンボックス (コマンドボタン)

プッシュボタンは、イベント (実行する処理) を候補 (ボタン) の選択によって指定するオブジェクトです。ボタンを選択したときの動作として、「イベントを AP に通知」「二次ウィンドウを閉じる」「ポップアップメニューを表示」などを設定できます。PF キーの代わりとしても使用できます。また、プッシュボタンにはアクセスキーを指定できます。

ボタンのサイズは、画面の解像度に応じて指定できます。また、グラフィックデータ (.bmp) をボタンのラベルとして貼り付けられます。ボタンのラベルはドローで定義し、AP 実行時に変更できます。

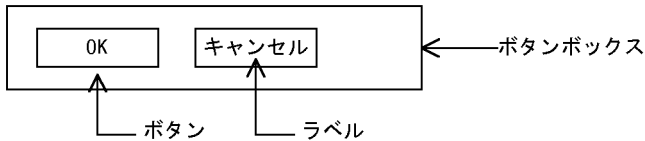
なお、プッシュボタンボックスの表示属性の詳細については「付録 A 属性情報一覧」を参照してください。

#### 注

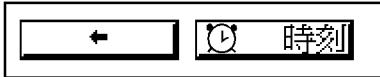
ボタンのラベルとして貼り付けるグラフィックデータを複数、登録している場合、画面表示時にどのグラフィックデータが使用されるかはプッシュボタンのボタン高で決まります。プッシュボタンのボタン高に収まるサイズのグラフィックデータの中で、最大のものが使用されます。解像度ごとのボタン高の違いについては、「(3) グラフィックをラベルとしてプッシュボタンに設定する」を参照してください。

## 2. GUI画面の作成

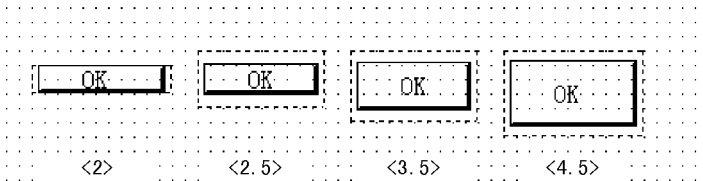
### (1) 表示形式



グラフィック付き、グラフィック+テキスト付きプッシュボタン



ボタン高



### (2) AP インタフェース

#### (a) 入力時の AP インタフェース

プッシュボタンが選択されるとイベントが発生します。AP に通知するイベントの場合、AP では定義時に指定したイベント通知コードが入力論理マップに格納されます。

イベント通知コードとしては「送信」「スクリーン」「割込」「ファンクションキー (PF1 ~ PF84)」「PA1 ~ PA3」および「ボタンイベント (A061 ~ A071)」が使用できます。

#### (b) 出力時の AP インタフェース

動的変更によって、ボタンの「活性/不活性」やラベルテキストを変更できます。

ユーザ AP でラベルテキスト項目名に変更内容を指定した場合は、ドロワーのプッシュボタンダイアログで指定した「テキスト揃え」は適用されません。実行時には、ユーザ AP で指定したボタンのラベルテキストは左寄せ固定となります。また、ユーザ AP でラベルテキスト出力項目名に、ビットマップおよびビットマップファイル名を指定できません。

### (3) グラフィックをラベルとしてプッシュボタンに設定する

プッシュボタンダイアログの「ラベルの種類」から「ビットマップ」または「ビットマップとテキスト」の組み合わせを選択し、ビットマップの [設定] ボタンをクリックすると、プッシュボタンのラベルとして固定グラフィックや固定グラフィックとテキストの組み合わせを表示できます。使用する固定グラフィックは、あらかじめユーザが用意しておきます。なお、グラフィックのサイズが大きい場合や、テキストが長い場合に

は、グラフィックとテキストが重なり合うことがありますので、必要に応じてボタンのサイズを大きくしてください。

なお、プッシュボタンに表示するグラフィックは、画面の解像度に応じて交代で表示できます。最もよく使用する解像度のグラフィックを「標準用」として指定し、必要に応じて「交代用」を指定します。

次に示す表は、XMAP3 が提供する標準の文字サイズに対する値です。

表 2-11 640 × 480 または 800 × 600 の解像度 (2D)

文字	ボタン高			
	4.5	3.5	2.5	2
大	48 × 48	36 × 36	24 × 24	22 × 22
標準	32 × 32	24 × 24	16 × 16	14 × 14
小	24 × 24	18 × 18	12 × 12	10 × 10

表 2-12 640 × 480 または 800 × 600 の解像度 (3D)

文字	ボタン高			
	4.5	3.5	2.5	2
大	44 × 44	32 × 32	22 × 22	20 × 20
標準	28 × 28	20 × 20	14 × 14	12 × 12
小	24 × 24	18 × 18	12 × 12	10 × 10

表 2-13 1,024 × 768 または 1,280 × 1,024 の解像度 (2D)

文字	ボタン高			
	4.5	3.5	2.5	2
大	64 × 64	48 × 48	32 × 32	30 × 30
標準	48 × 48	36 × 36	24 × 24	22 × 22
小	32 × 32	24 × 24	16 × 16	14 × 14

表 2-14 1,024 × 768 または 1,280 × 1,024 の解像度 (3D)

文字	ボタン高			
	4.5	3.5	2.5	2
大	44 × 44	32 × 32	22 × 22	20 × 20
標準	28 × 28	20 × 20	14 × 14	12 × 12
小	24 × 24	18 × 18	12 × 12	10 × 10

## 2. GUI画面の作成

### (4) フォーカスが遷移しないボタン (マウスでの選択だけ有効)

フォーカス遷移で選択できないようにボタンを設定します。誤入力などを防ぐのに有効です。プッシュボタンボックスダイアログの「フォーカス遷移の対象としない」を指定すると、そのボタンボックス中のプッシュボタンへは、フォーカスが遷移しなくなります。

### (5) キーで選択するボタン

マウスだけでなく、文字キーでもプッシュボタンを選択できるように設定します。プッシュボタンダイアログの「アクセスキーを使用する」を指定すると、アクセスキーが対応するキーを選択することで、プッシュボタンを押せます。このアクセスキーに「0～9」の数字キーを指定すると、テンキーでも操作できるようになります。

なお、アクセスキーが有効なのは、フォーカスがプッシュボタンボックスに位置づいているときです。

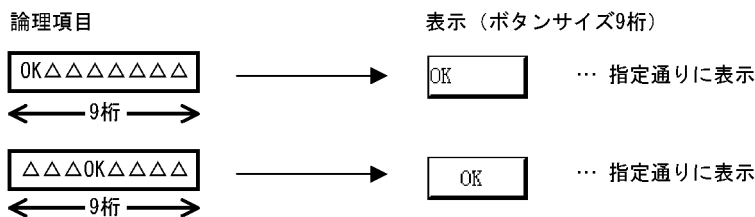
### (6) 表示属性の動的変更

プッシュボタンの活性 / 不活性や表示属性を実行時に変更できます。表示属性の動的変更については「1.5.6 表示属性の動的変更」を参照してください。

#### (a) ラベルテキストを動的変更するときの注意

##### テキスト揃え

ラベルテキストを動的変更する場合、ラベルテキストの中央寄せなどの桁揃えは、論理項目の設定値で調整します。



##### プッシュボタンのサイズ変更

プッシュボタンのサイズを変更した場合、変更した値がすべてのプッシュボタンのボタンサイズ、およびラベルテキストの論理項目の桁に設定されます。このため、「AP からテキストを変更する」を指定しているボタンが含まれるプッシュボタンボックスで、ボタンのサイズを変更したときは、AP を再コンパイルしてください。

##### テキストのデータ長の変更

ターゲットによって、データ長をユーザ任意の値に設定できるようになります。「テキストのデータ長を変更する」を指定した場合、ラベルテキストの論理項目の桁が固定になります。このため、同じボックス内のすべてのボタンで、ボタンサイズの変更、および

ラベルテキストの論理項目の桁の変更ができなくなります。ターゲットとデータ長の関係については、マニュアル「XMAP3 開発・実行ガイド」を参照してください。

#### プッシュボタンの貼り付け

「AP からテキストを変更する」を指定しているプッシュボタンを貼り付ける場合、貼付け先のプッシュボタンボックスのボタン長に合わせて桁が変更されます。テキストのデータ長の変更は解除されます。

### 2.4.25 MCR

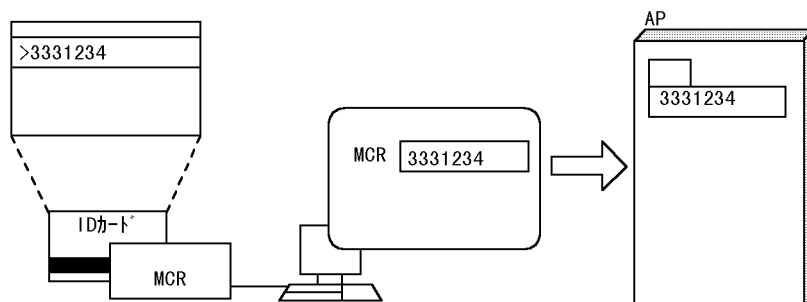
MCR のオブジェクトは入出力テキスト・フィールドダイアログで、使用目的の「MCR」を選んで指定します。通常、MCR カードデータを読み込ませた直後に画面が送信されるため、1 画面に対し MCR オブジェクトを一つだけ指定するようにしてください。

入出力テキスト・フィールドダイアログで、使用目的に「MCR」以外を指定した項目にも MCR から入力できます。キーボードで入力した場合と同様の動作となります。

MCR は、ターミナルサービス環境では使用できません。また、Windows XP のリモートデスクトップ環境でも使用できません。

#### (1) MCR からの入力

MCR からの入力に使用するカードには、最大 64 バイトの文字を書き込めます。この文字列の先頭に、ID コードを付けておきます。XMAP3 は「<」または「>」を ID コードとみなします。MCR で入力されたデータは、キーボードで入力したデータと同様に AP に格納されます。ただし、このとき「<」および「>」は削除されます。なお、MCR を使用するときは、表示・印刷セットアップの「接続機器」タブで「使用しない」チェックを外してください。標準ではチェックされています。



フィールドや入出力テキストボックスにカード入力する場合、ID コードの有無や違いによって、XMAP3 の動作が異なります。ID コードの違いによる XMAP3 の動作の違いを説明します。

### (2) ID コード無しの場合

MCR カードデータの先頭が「<」または「>」でない場合、つまり ID コードがない場合には、入出力テキスト・フィールドダイアログの使用目的に「MCR」以外を指定するようにしてください。このとき、XMAP3 は通常のキーボード入力と同じように動作します。MCR カードデータが「TOKYO」だとすると、カーソル位置から「TOKYO」と表示します。プッシュボタン、送信（実行）キーまたはファンクションキーを押すと、画面が確定します。

入出力テキスト・フィールドダイアログの使用目的に「MCR」を指定した場合、画面上は入力データが表示されますが、入力論理マップに返されるフィールドデータは入力操作なしのマッピング規則に従います。



(凡例) █: カーソル

### (3) ID コードが「<」の場合

MCR カードデータの先頭が「<」の場合、入出力テキスト・フィールドダイアログの使用目的に「MCR」以外を指定するようにしてください。このとき、XMAP3 は「<」を ID コードとみなします。MCR カードデータが「<TOKYO」だとすると、XMAP3 は「<」に続くカードデータ「TOKYO」をカーソル位置から表示したあと、すぐに送信要求を発生させます。カードデータが「<」だけの場合には、送信要求だけを発生させません。

入出力テキスト・フィールドダイアログの使用目的に「MCR」を指定した場合、画面上は入力データが表示されますが、入力論理マップに返されるフィールドデータは入力操作なしのマッピング規則に従います。



(凡例) █: カーソル

### (4) ID コードが「>」の場合

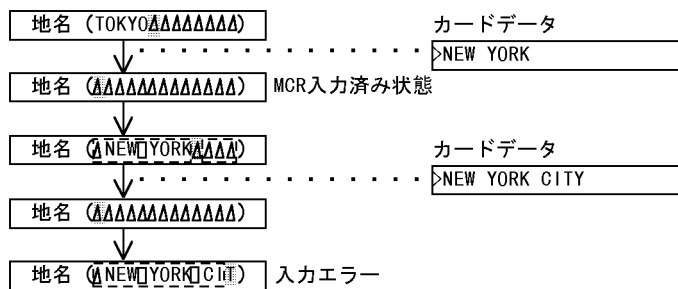
MCR カードデータの先頭が「>」の場合、入出力テキスト・フィールドダイアログの使用目的に「MCR」を指定するようにしてください。このとき、XMAP3 は「>」を ID コードとみなします。ID コードが「>」のカードデータが入力されると、XMAP3 はまず MCR 入力フィールド全体を NULL クリアし、カーソル位置を先頭に戻します。この



とき、そのMCR入力フィールドはMCR入力済み状態になります。

次に、XMAP3はMCR入力フィールドの2桁目からカードデータを入力し、入力後すぐに送信要求を発生させます。カードデータが「>」だけの場合には、送信要求だけを発生させます。なお、IDコードが「>」のカードデータを入力した場合、カーソル位置は移動しますが、カードデータの内容は画面に表示されません。MCRカードデータが「>NEWYORK」だとすると、画面には何も表示されずにカーソル位置だけが10桁目に移動します。IDコードに続くカードデータがフィールド長を超えると、入力エラーになりますので、IDコードに続くカードデータよりも1バイト長くフィールド長を定義しておく必要があります。

入出力テキスト・フィールドダイアログの使用目的に「MCR」以外を指定した場合、入力できません。



(凡例) ■: カーソル □: スペースデータ ▲: NULL □: 非表示

- データ入力時、使用目的がMCRの入出力テキスト・フィールドは、半角文字を入力できるフィールドで、桁あふれチェック、非表示の表示属性となります。MCR入力済み状態になったMCR入出力テキスト・フィールドには、その後IDコードが「>」のカードだけが入力できます。IDコードのないカードやIDコードが「<」のカードで入力したり、キーボードから入力したりすると入力エラーになります。MCR入力済み状態は、次の場合に解除されます。
- MCR入力済み状態のフィールド、または入出力テキストボックスにカーソルが位置する場合で、フィールドキーを押したとき。フィールド、または入出力テキストボックス全体がNULLクリアされ、カーソルはフィールド、または入出力テキストボックス先頭に位置づけられます。
- データキーを押した場合。
- 画面表示時にMCR入力済み状態のフィールド、または入出力テキストボックスにデータが出力された場合。
- 画面表示時にMCR入力済み状態のフィールド、または入出力テキストボックスがNULLクリアされた場合。

IDコードの違いによるXMAP3の動作の違いを次の表に示します。

## 2. GUI 画面の作成

論理マップに返される項目データ，およびイベント通知コードの詳細については，「5.3.15 MCR 入力のマッピング規則」，および「5.3.4 イベント通知コード」を参照してください。

表 2-15 ID コードの違いによる XMAP3 の動作の違い（使用目的が MCR の場合）

ID コード	<	>	なし
文字 JIS	(3C) <sub>16</sub>	(3E) <sub>16</sub>	-
フィールドおよびテキストボックスへの入力 <sup>1</sup>	ID コードに続くカードデータ（ID コードを除く）を文字キーと同様にカーソル位置から入力する。	ID コードに続くカードデータ（ID コードを除く）をカーソルが位置するフィールド先頭の次の桁から入力する <sup>2</sup> 。	カードデータの先頭から文字キーと同様にカーソル位置から入力する。
画面への表示	表示する。	表示しない。	表示する。
送信要求の発生のタイミング	カードを読み込ませた直後。	カードを読み込ませた直後。	画面を確定させるキー（実行キー，ファンクションキーなど）を押したあと。
論理マップに返る項目データ	入力操作なしとなり，カードデータは返らない。	カードデータが返る。	入力操作なしとなり，カードデータは返らない。

（凡例）

- : 該当しない。

注 1

入力禁止のフィールド，およびテキストボックスには入力できません。

注 2

ID コードに続くカードデータがフィールド長を超えると入力エラー（桁あふれ）になります。この場合，自動送信しないため，論理マップにカードデータは返りません。

表 2-16 ID コードの違いによる XMAP3 の動作の違い（使用目的が MCR 以外の場合）

ID コード	<	>	なし
文字 JIS	(3C) <sub>16</sub>	(3E) <sub>16</sub>	-
フィールドおよびテキストボックスへの入力	ID コードに続くカードデータ（ID コードを除く）を文字キーと同様にカーソル位置から入力する。	入力できない（ブザーが鳴る）。	カードデータの先頭から文字キーと同様にカーソル位置から入力する。
画面への表示	表示する。	表示しない。	表示する。

IDコード	<	>	なし
文字JIS	(3C) <sub>16</sub>	(3E) <sub>16</sub>	-
送信要求の発生のタイミング	カードを読み込ませた直後。	画面を確定させるキー（実行キー、ファンクションキーなど）を押したあと。	画面を確定させるキー（実行キー、ファンクションキーなど）を押したあと。
論理マップに戻る項目データ	カードデータが返る。	入力操作なしとなり、カードデータは返らない。	カードデータが返る。

（凡例）

- : 該当しない。

注

入力禁止のフィールド、およびテキストボックスには入力できません。

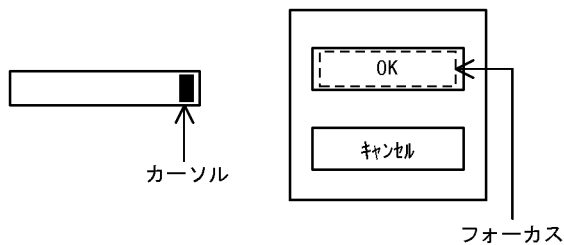
## 2.4.26 フォーカスとカーソル

フォーカスまたはカーソルが位置づいているオブジェクトは、ユーザが操作できる状態になっています。オブジェクトに対する操作は、必ずマウス、またはキーボードでフォーカス・カーソルを位置づけたあとで行います。

フォーカスとは、オブジェクトに対して入力権があることを示す印です。キー制御は、この入力権のあるオブジェクトに対して行われます。カーソルは、フォーカスのあるオブジェクトがキーボードからデータ入力できる状態であるとき、その一文字に対し、入力権があることを示す印です。データの入力や削除は、カーソル位置が示す個所が対象となります。

### （1）表示形式と制御方法

#### （a）表示形式



#### （b）制御方法

XMAP3では、フォーカス・カーソルを次のように制御します。これらの制御方法はドロースेटアップで設定します。詳細についてはマニュアル「XMAP3 開発・実行ガイド」を参照してください。

## 2. GUI 画面の作成

- フォーカスとカーソルを同時に制御する  
フォーカスとカーソルを共通の項目を使用して制御します。フィールドボックスを定義した画面で、プッシュボタンボックスのフォーカス位置を制御する場合には、フォーカスとカーソルを別々に制御してください。
- フォーカスとカーソルを別々に制御する  
フォーカス位置とカーソル位置の情報を格納する項目が別々に生成されます。なお、フォーカスとカーソルを別々に制御する場合は、カーソルの種類を次の二つから選びます。

### 論理カーソル

カーソル位置をカーソル定数で制御する通常の制御方法です。カーソル定数は XMAP3 が生成するため、画面のオブジェクトの位置を変更しても AP を変更する必要はありません。

### 行列 (2 進) カーソル

カーソル位置を 2 進数の行列位置で制御します。主にホスト系からの移行上で、行列での制御を特に必要とする場合に使用します。画面のフィールドの位置を変更した場合、AP の設定値も合わせて変更する必要があります。

## (2) 初期フォーカス・カーソル

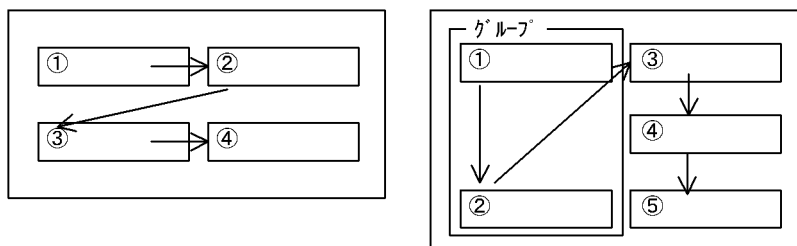
各ダイアログのフォーカス・カーソル設定によって、画面を表示したとき、最初にフォーカス・カーソルを位置づけるオブジェクトを設定できます。

## (3) 初期フォーカス・カーソルの AP からの設定

フォーカス定数またはカーソル定数という値を論理マップに代入すると、初期フォーカスを位置づけるオブジェクトを AP から設定できます。これらの値が設定されていないと、ドロウで定義した初期フォーカス・カーソルの指定が採用されます。フォーカス定数およびカーソル定数については「5.5 定数部の論理マップ生成規則とマッピング規則」を参照してください。

## (4) フォーカス・カーソルの移動順序

タブキーを押したり、メニューを選択したりしたあと、フォーカス・カーソルが次の項目やフィールドに移動するときの順序です。標準では、画面の左上から右下に移動します。グループボックスで囲まれている複数のオブジェクトは、一つのオブジェクトとしてみなされます。



#### フォーカス対象外

プッシュボタンにフォーカスが遷移しないように設定できます。フォーカス対象外については「2.4.24(4) フォーカスが遷移しないボタン（マウスでの選択だけ有効）」を参照してください。

### (5) グループ間のフォーカス順設定

グループボックスで囲まれたオブジェクトのフォーカス順序を設定できます。フォーカス順はAPから動的変更できません。入力単位が「フィールド」のときは、APからフォーカス・カーソルを設定するのでフォーカス順の設定は無効になります。

#### (a) オブジェクトの遷移条件

オブジェクトの遷移条件を示します。オブジェクトの種類によって、指定できるものが異なります。

##### キーエントリ系オブジェクトの場合

- 手動スキップ・右寄せ  
入力データが桁数に満たない場合、[ Tab ] キーで次の入力 / 選択できる領域にフォーカス・カーソルを移動すると、入力データが右寄せで表示されます。
- 手動スキップ・全桁必須  
一文字でも入力した場合、桁数いっぱいに入力しなければ入力エラーになり、[ Tab ] キーでのフォーカス・カーソル移動ができなくなります。
- 手動スキップ  
桁数いっぱいまで入力し、[ Tab ] キーを押すとフォーカス・カーソルが次の入力 / 選択できる領域に移動します。桁数を超えるデータを入力するとエラーになります。
- 自動スキップ  
桁数いっぱいまで入力すると、フォーカス・カーソルが次の入力 / 選択できる領域に自動的に移動します。

##### 選択エントリ系オブジェクトの場合

- 自動スキップ（選択したら次のボックス・ボタン・項目にフォーカスを移す）  
ボックス内での選択操作が終わると、フォーカス対象の次のボックスへフォーカスが自動的に遷移します。複数選択できるオブジェクト（複数選択リストボックス、チェックボタンボックス）の場合は、末尾の候補が選択されるとフォーカスが遷移します。

## 2. GUI画面の作成

### (b) フォーカス移動順序の設定

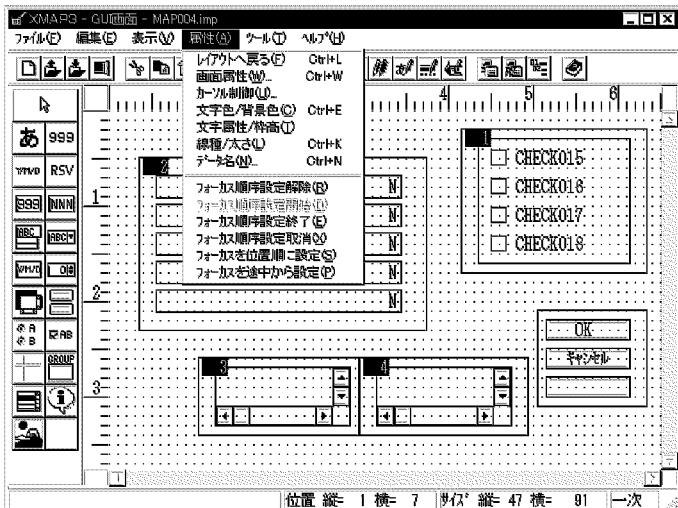
フォーカス順序の設定は、ドロワーのメニューバー「属性」の「フォーカス順序設定開始」で開始します。

フォーカス順序を設定するメニューを次に示します。

- フォーカス順序設定解除
- フォーカス順序設定開始
- フォーカス順序設定終了
- フォーカス順序設定取消
- フォーカスを位置順に設定
- フォーカスを途中から設定

上記のメニューを使用してフォーカス順序を設定します。次に、「フォーカス順序設定開始」を選択したときのレイアウト画面を次の図に示します。

図 2-7 フォーカス順序設定開始の画面



#### フォーカス順序の対象となる条件

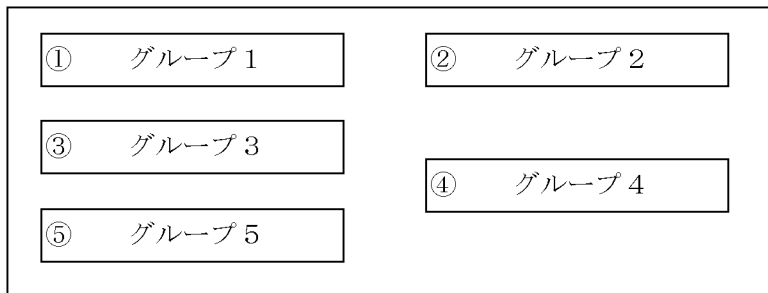
入力項目（入力操作が伴うオブジェクト）に対してフォーカスを位置づけることができます。オブジェクトにフォーカス順序を位置づけるためには、グループボックスで囲む必要があります。ただし、プッシュボタンボックスでフォーカス対象外に指定した場合は、フォーカス順序の対象外になります。

フォーカス順序を設定するときには、設定するすべてのオブジェクトがレイアウト領域上に表示されるようにする必要があります。表示されていない場合、レイアウト領域を広げるかズームで全体が見えるようにしてから設定してください。

また、同一グループ内に、プッシュボタン（フォーカス対象外）以外のフォーカス対象オブジェクトがある場合は、グループにはフォーカスが位置づきます。ただし、プッシュボタンボックスにはフォーカスは位置づきません。

## 標準のフォーカス順序

[フォーカス順序設定開始]メニューで決められるフォーカス順序は、次に示すように左から右、上から下の順になります。

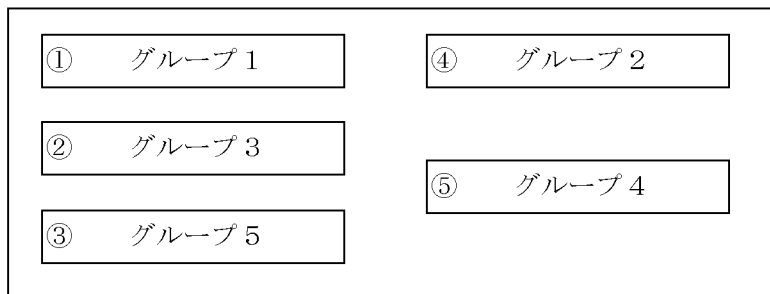


注 ①～⑤がフォーカス順序になります。

[フォーカスを位置順に設定]を選んだ場合も、上記と同じフォーカス順序になります。

## フォーカス位置の設定

各ボックスのフォーカス位置を任意の位置から任意の順番で設定する場合、マウスで設定順にクリックしていきます。

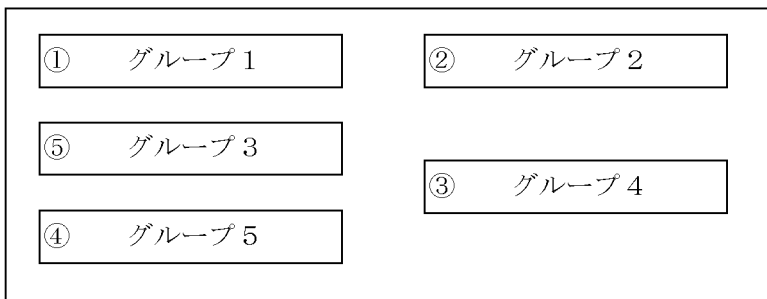


注 ①～⑤がフォーカス順序になります。  
上記の例では、グループ1, 3, 5, 2, 4の順に選んだ場合です。

## 先頭位置以外からのフォーカス順序の設定

画面に定義したグループの任意の位置からのフォーカス順序を変更したい場合は、[フォーカスを途中から設定]メニューを使用します。[フォーカスを途中から設定]メニューを選ぶと何番目のオブジェクトから変更するかを問い合わせるフォーカススキップダイアログが表示されます。

## 2. GUI 画面の作成



注 ①～⑤がフォーカス順序になります。

上記例は、「フォーカススキップ」ダイアログで「3」を指定し、その後、グループ 4 をマウスで選んだ場合のフォーカス順序の例です。前回設定してある順序の 1, 2 以外のグループをマウスで指定すると、マウス操作の順に「3」から設定されます。

グループ 1, 2 は変わらず、グループ 3～5 の位置順が変更されます。

### フォーカス順序設定の解除

[フォーカス順序設定開始]メニューを使用して設定したフォーカス順序を、画面での配置順に戻す場合は、[属性]メニューの[フォーカス順序設定解除]を選びます。

## (6) フォーカス設定の動的変更

画面が表示されたときに、どのオブジェクトにフォーカスを位置づけるかを AP で指定できます。フォーカス設定の詳細については「6.2.39 初期フォーカスの動的変更」を参照してください。また動的変更については、「1.5.6 表示属性の動的変更」を参照してください。

## (7) カーソル制御項目とフィールドボックスの関連づけ

ドロースेटアップで「フォーカスとカーソルを別々に制御する」と設定して作成した画面に、フィールドボックスを 2 個以上配置した場合、カーソル制御項目とフィールドボックスの関連づけを変更するには、ドローのメニューバー「属性」の「カーソル制御」を選択して、カーソル制御ダイアログを開きます。カーソル制御ダイアログを次の図に示します。

図 2-8 カーソル制御ダイアログ



リストボックス中の、変更したいフィールドボックスの行と列が表示されている行をク



リックすると、下段の編集領域に、フィールドボックスの行、列、およびカーソル制御項目名が表示されます。入力カーソル制御項目名が表示されているボックスの右端のボタンをクリックすると、カーソル制御項目の一覧が表示されます。その中から、変更したいカーソル制御項目名を選択してください。

ほかのフィールドボックスが、選択したカーソル制御項目に関連づいていた場合には、そのフィールドボックスには、どのフィールドボックスとも関連づけられていないカーソル制御項目名が、代わりに割り当てられます。

## 2.4.27 予約テキスト・フィールド

OpenTP1 がサポートする予約項目名称を定義するオブジェクトです。論理マップ上にデータを設定していなくても、実行時に予約項目名に応じた内容で画面に出力されます。予約テキスト・フィールドは OLTP サーバ構成の環境で利用できます。

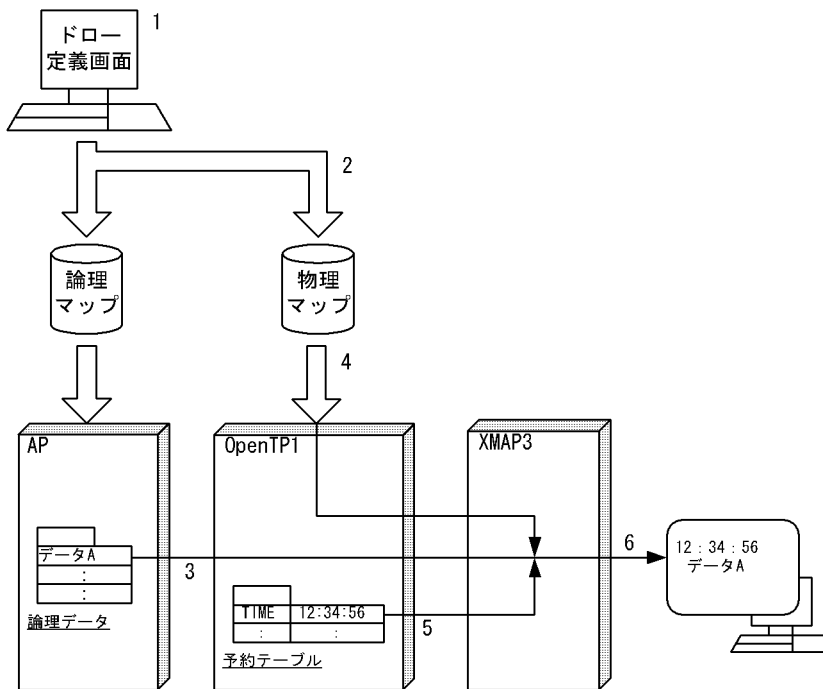
なお、予約テキスト・フィールドの表示属性の詳細については「付録 A 属性情報一覧」を参照してください。

### (1) 表示処理

予約テキスト・フィールドの出力の仕組みを次に示します。

1. ドロー定義画面で予約テキスト（または予約フィールド）「TIME」を定義します。
2. 物理マップ中に予約項目情報「TIME」が設定されます。
3. AP からの SEND 文によって論理データが指定されます。
4. OpenTP1 に物理マップが読み込みます。
5. OpenTP1 によって予約テーブルが生成され、論理データ、物理マップ、および予約テーブルなどが XMAP3 に渡されます。
6. XMAP3 によって画面にデータが表示されます。

## 2. GUI 画面の作成



### (2) 予約項目名

予約テキスト・フィールドに定義する項目名称は OpenTP1 で提供されている予約項目名を指定します。OpenTP1 が提供する予約項目名はドローで定義しておきます。論理マップにデータを設定しなくても、OpenTP1 から生成された日付、時刻などのデータを、項目名に対応したフォーマットで画面上に表示できます。

予約項目名については、マニュアル「OpenTP1 プロトコル TP1/NET/XMAP3 編」を参照してください。

## 2.4.28 次画面

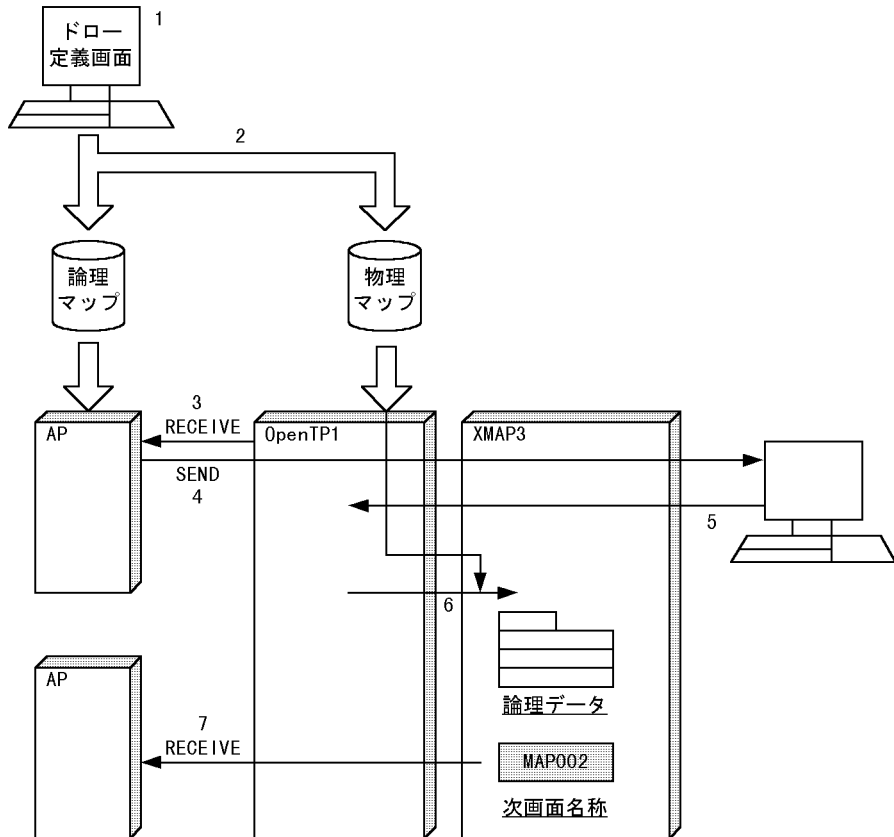
OpenTP1 との連携時、次に表示したい画面を定義するオブジェクトです。通常、ある画面から次に表示したい画面は AP 上で画面名称を定義しておく必要があります。このようにときに次に表示したい画面をドロー上で定義しておく、AP で画面名称を定義していなくても実行時に画面名称を取得して表示できます。次画面は OLTP サーバ構成の環境で利用できます。

### (1) 表示処理

次画面の表示の仕組みを次に示します。

1. ドロー定義画面で次画面名称「MAP002」を定義します。
2. 物理マップ中に次画面名称「MAP002」が設定されます。

3. OpenTP1 によって AP が起動され、AP から RECEIVE 要求が発行されます。
4. AP から SEND 要求が発行され画面が表示されます。
5. 表示された画面上で入力操作を実行するとイベントが発生します。
6. 発生したイベントに対する物理マップで XMAP3 に RECEIVE 要求され、OpenTP1 に論理データと物理マップ中の次画面名称「MAP002」を返します。
7. OpenTP1 によって、次のトランザクションに対応する AP が起動されます。このとき、マッピングによって生成された論理データと次画面名称「MAP002」は、RECEIVE 要求で AP に渡されます。



## 2.5 部品の利用

---

あらかじめ登録しておいた部品をレイアウト画面上に配置するときの手順を次に示します。

1. ドローのメニューバーの [ 編集 ] から [ 部品貼り付け ] を選ぶ。  
部品貼り付けダイアログが表示されます。
2. 使用する部品名称を選ぶ。  
部品がレイアウト画面の左上に貼り付けられますので、配置したい位置に移動してください。

なお、ユーザが貼り付けた部品と、すでに定義されているオブジェクトとの関係は次のようになります。

### データ名

貼り付けた部品と既存のオブジェクトとでデータ名が重複している場合、部品で使われているデータ名が XMAP3 の命名規則に従って変更されます。データ名が重複していない場合は、部品のデータ名がそのまま使われます。

- 言語種別が異なる場合  
部品の中に、既存のオブジェクトの言語種別では使用できない文字（ハイフンとアンダーラインは除く）が指定されている場合、XMAP3 の命名規則に従って変更されます。
- マップ名の長さが異なる場合  
マップ名の長さが 6 文字で設定されている場合、データ名として指定できる文字数は 22 文字までですが、マップ名の長さが 7 文字で設定されている場合は、21 文字までです。このため、22 文字のデータ名を、マップ名の長さが 7 桁で設定されているレイアウトに貼り付けると、XMAP3 の命名規則に従って変更されます。

### 通知コード

通知コードの重複はチェックされません。そのため、部品を貼り付けたあと、必要に応じてダイアログを開き、変更する必要があります。

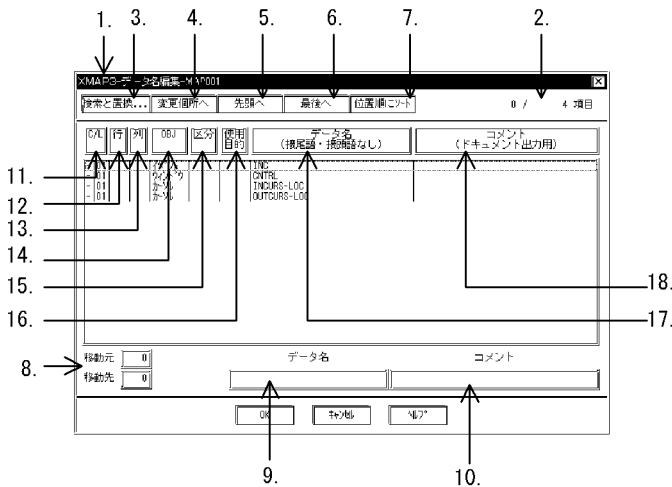
## 2.6 データ名の編集

ドローでオブジェクトをレイアウトし、そのままマップを生成すると、XMAP3 が提供している標準のデータ名が使われます。XMAP3 が提供しているデータ名は、「接頭語 - オブジェクトごとの名称 nnnn - 接尾語」で構成されています。そのため、オブジェクトが同じである場合には、オブジェクトの定義順に付けられる 4 桁の数字 (nnnn) の違いで判別しなくてはなりません。

XMAP3 では、ドローのデータ名編集機能を使用して、レイアウト画面上に定義しているすべてのオブジェクトのデータ名を表示し、必要に応じて変更できます。

### 2.6.1 データ名編集ダイアログの機能

データ名編集ダイアログには、論理マップに展開されるデータ名および生成元のオブジェクトでの定義内容の一覧が表示されます。このダイアログ上でデータ名を変更したり、論理マップ生成順を指定したりできます。



#### (1) データ名編集ダイアログの説明

データ名編集ダイアログの表示内容およびボタンについて次の表に説明します。

表 2-17 データ名編集ダイアログの説明

各部名称	機能
1. タイトルバー	ダイアログのタイトルとマップ名を表示する。
2. 99999 / 99999 項目	「選択されている項目の番号 / 項目の総数」を表示する。
3. 検索と置換 ...	検索と置換ダイアログを表示する。

## 2. GUI 画面の作成

各部名称	機 能
4. 変更箇所へ	カーソルをデータ名が変更された項目に移動し選択する。
5. 先頭へ	カーソルを先頭の項目に移動し選択する。
6. 最後へ	カーソルを最後の項目に移動し選択する。
7. 位置順にソート	論理マップ生成時のデータ名の展開順を指定する。 ボタンをクリックした場合： データ名がレイアウト上の位置順に表示される。 初期表示は、ドローセットアップで設定した値。データ名編集ダイアログで変更すると、データ名はダイアログでの指定に従って並べ替えられる。
8. 移動元・移動先	項目を移動するとき、移動元・移動先の項目番号を表示する。
9. データ名	データ名一覧で選択したデータ名を表示する。データ名はここで変更する。
10. コメント	データ名一覧で選択したデータ名にコメントがある場合、そのコメントを表示する。コメントはここで変更する。

表 2-18 データ名一覧に表示される情報

タイトル	機能
11. C	チェックマークを表示する。 ：書き込み禁止を示す *：データ名は変更されていることを示す
11. L	論理マップ展開レベル 論理マップに展開されたときのデータ項目のレベル。項目を移動する場合、移動元と移動先の論理マップ展開レベルは同じでなければならない。
12. 行	データ名に対応するオブジェクトの位置（行）を示す。
13. 列	データ名に対応するオブジェクトの位置（列）を示す。
14. OBJ	データ名に対応するオブジェクトの種類を示す。
15. 区分	データ名に対応するオブジェクトの細分類を示す。
16. 使用目的	データ名に対応するオブジェクトの使用目的を示す。
17. データ名	接頭語および接尾語を除いたデータ名を示す。
18. コメント	ドキュメント出力用のコメントを示す。

### 注

データ名編集ダイアログを表示している間に変更したデータ名にだけ付きます。ダイアログを閉じると、すべてリセットされます。

## 2.6.2 データ名編集ダイアログでの操作と注意事項

データ名編集ダイアログでは、データ名の変更、ソート、検索および置換ができます。

### (1) データ名の順序をソートする

レイアウト画面でのオブジェクトの位置順をそのまま反映する場合は、「位置順にソー

ト」を指定しておきます。ユーザが独自にソートする場合は、移動させるオブジェクトの上で左ボタンをクリックし、そのまま移動させたい位置にドラッグします。

#### ソート時の注意

- [位置順にソート] ボタンをクリックすると、データ名が位置順に並びます。位置順にソートしたあとで、任意の順番に並べ替えることもできます。
- 移動元と移動先の論理マップ展開レベルが同じならば、移動できます。
- 1回の操作で移動できるデータ名は一つです。ただし、データ名に対応するオブジェクトが下位項目を持つ場合（フレームでの反復や、下位項目を指定した場合）、下位項目も一緒に移動します。
- 移動先が移動元の上位項目である場合、または下位項目である場合は、移動できません。
- 下位項目同士のソートはできません。

### (2) データ名とコメントの変更

データ名一覧から、変更したいデータ名を選ぶと、データ名とコメントの欄に選んだデータ名とコメントが表示されます。変更したいデータ名またはコメントを入力すると、データ名一覧に変更したデータ名およびコメントが反映されます。

#### データ名とコメント変更時の注意

- 文字数  
接頭語（マップ名）を含めて 29 文字以内です。このため、ドローセットアップの「マップ名の長さ」の設定が「7」の場合は 21 文字以内、「6」の場合は 22 文字以内で指定します。全角文字は 2 文字分として数えます。
- 文字種  
COBOL の場合は、「A ~ Z」「0 ~ 9」「-（ハイフン）」「大文字の半角カナ」「全角（混在）」  
C の場合は、「A ~ Z」「a ~ z」「0 ~ 9」「\_（アンダーバー）」
- 再定義名は半角で 30 文字まで入力できます。
- コメントは、半角で 40 文字、全角で 20 文字まで入力できます。
- コメントを記述できないオブジェクト（各オブジェクトの定義ダイアログでコメントの入力ができないもの）の場合は、コメントの欄は不活性になります。

### (3) データ名の検索と置換

データ名を検索または置換する場合は、[検索と置換] ボタンを選んで、検索と置換ダイアログを表示します。

#### データ名の検索時、および置換時の注意

- 検索するデータ名は最低 1 文字、指定してください。ただし、メタキャラクターとしての「\*」は使用できません。
- 置換後のデータ名の長さが上限を超えた場合、上限値以降の文字列は切り捨てられます。

#### (4) ほかの画面からのデータ名のコピー

データ名編集ダイアログに表示されているデータ名を、あらかじめ作成しておいたデータ名辞書、Excel の表およびテキストファイルから読み込んだデータ名で変換できます。

1. Excel またはテキストファイル上で、変換に使用するデータ名にフォーカスを位置づけ、[Ctrl] + [C] でクリップボードにコピーします。
2. データ名編集ダイアログを開いて、変換したいデータ名のフィールドを選びます。フィールドが選ばれ、データ名の項目欄にフォーカスが移ります。
3. フォーカスが位置づけられたデータ名の欄に [Ctrl] + [V] で、クリップボードにコピーしておいたデータ名を貼り付けます。



## 2.7 テスト表示

---

GUI画面の設計が終了したら、実行時の表示状態を確認するためにテスト表示をします。テスト表示をするときは、[ツール]メニューの[テスト表示]を選ぶか、ツールバーから[テスト表示]ボタンを選びます。また、Webブラウザテスト表示を選択すると、ブラウザを使ったテスト表示ができます。

テスト表示では、グラフィックデータの表示<sup>1</sup>、ポップアップメニューファイルから読み込んだポップアップメニューの表示<sup>2</sup>、データの入力、ボタンのオン/オフ選択およびフォーカスの移動などが確認できます。

テスト表示では、画面確定を容易にするために、正しい入力や操作をしないと画面が確定できないような属性を無効にしています。このため、テスト表示では、実際のAPでの実行結果と動作が異なることがあります。無効となる属性については、「付録A 属性情報一覧」を参照してください。

なお、予約テキスト・フィールドは「X」(項目長分だけXで埋められた形式)で表示されます。また、次画面の機能をテスト表示することはできません。

注 1

出力グラフィックの場合、XMAP3が標準で提供しているデータしか、テスト表示できません。

注 2

ポップアップメニューファイルから読み込んだポップアップメニューを表示したい場合は、ドロウの定義画面で、ファイルの初期値にファイル名を指定し、かつポップアップメニューファイルを用意しておく必要があります。



# 3

## CUI 画面の作成

CUI 画面のデザインは，XMAP3 のドローを使用して設計します。この章では，CUI 画面の設計で使用できる機能について説明します。

---

3.1 CUI 画面の設計

---

3.2 CUI 画面のレイアウト手順

---

3.3 CUI 画面属性の設定

---

3.4 CUI 画面を構成するオブジェクト

---

3.5 部品の利用

---

3.6 データ名に関するコメントの編集

---

3.7 テスト表示

---

## 3.1 CUI 画面の設計

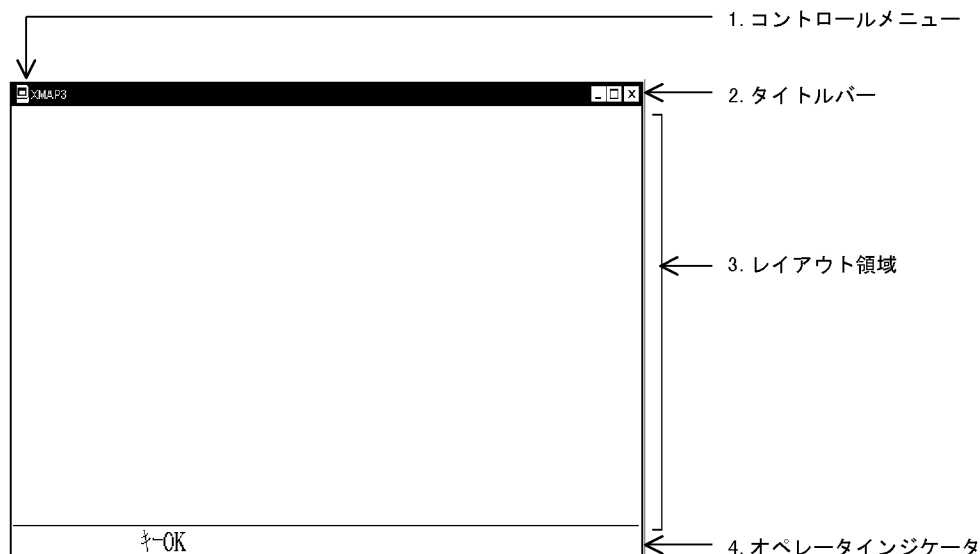
---

CUI 画面の概要を設計します。

### 3.1.1 CUI 画面の構成

CUI 画面は次のようなウィンドウで構成されています。

#### (1) ウィンドウ各部の名称



#### 1. コントロールメニュー

コントロールメニューとは、画面の左上にある枠をマウスでクリックすると表示されるシステムメニューのことです。タイトルバー左端にタイトルを位置づけ、マウスの左ボタンを押すと表示されます。コントロールメニューをクリックすると、Windows のシステムメニューが表示されます。システムメニューには、[ 移動 ][ サイズ変更 ] [ 最小化 ][ 最大化 ] があります。

#### 2. タイトルバー

タイトルは左側に表示されます。タイトルは「XMAP3」固定です。

#### 3. レイアウト領域

実際の業務処理に使用するオブジェクトを配置します。

これ以降、表示レイアウトなどの説明での「作業領域」は、断りがないかぎりレイアウト領域を示します。

#### 4. オペレータインジケータ

キーボード状態、送信状態などのメッセージを XMAP3 が表示します。ユーザプログラムからの操作はできません。表示位置は、XMAP3 の表示・印刷セットアップで変

更します。表示は、「上に表示」「下に表示」「表示しない」「拡張形式」が選択できます。なお、OLTP サーバ構成の環境で使用するときは、「上に表示」または「下に表示」のどちらかを選択します。

《レイアウト領域のグリッド》

グリッドのます目の最小単位は、半角文字 1 文字分です。

### 3.1.2 画面サイズ

CUI 画面の画面サイズは、「24 行 × 80 列」固定です。

グリッド

半角文字を基準としたます目で、24 行 × 80 列となります。

- 半角文字は 1 ます

A

- 全角文字は 2 ます

漢

### 3.1.3 背景色および画面の表示フォント

背景色

背景色は、黒で固定です。

画面の表示フォント

XMAP3 で表示に使用するフォントは、フォント構成ファイル (X3PFONT) の定義情報に従います。

## 3.2 CUI 画面のレイアウト手順

---

画面概要の設計後、実際にドローを起動して画面上にオブジェクトを配置し、レイアウトを定義します。

CUI 画面を新規に作成するときのレイアウト手順について説明します。

1. スタートメニューから [ XMAP3 ] - [ ドロー ] を選び、ドローを起動する。  
新規・更新選択ダイアログが表示されます。
2. 新規・更新選択ダイアログで [ 対象 ] から「マップ」 を選択し、「画面・帳票・書式を新規作成する」を選択する。  
選択後、[ OK ] ボタンをクリックすると新規作成ダイアログが表示されます。
3. 新規作成ダイアログでマップ名を指定する。  
マップ名の決定の詳細については、「3.2.1 マップ名の決定」を参照してください。
4. 新規作成ダイアログで「定義種別」を選択する。  
「定義対象の選択」から「CUI 画面」を選択します。
5. 新規作成ダイアログで定義パターンを選択する。  
定義パターンを選択します。選択できる定義パターンには、XMAP3 が用意しているパターンまたはユーザが登録したパターンがあります。XMAP3 で提供している定義パターンについては、「付録 B XMAP3 で提供している定義パターン」を参照してください。  
また、提供しているパターンを利用しないときは、「フリー」を選んで画面属性ダイアログで設定します。
6. 新規作成ダイアログで言語種別を選択する。  
COBOL または C のどちらかを選択します。  
なお、XMAP3/Web for Cosminexus を使用して Cosminexus 上で実行する CUI 画面を開発する場合で、AP 開発言語が Java のときは、C を選択してください。
7. [ OK ] ボタンをクリックする。  
画面定義ウィンドウが表示されます。
8. 画面定義ウィンドウで、画面属性を設定する。  
ウィンドウの表示形態やイベント通知コードなど、画面を制御する情報（画面属性情報）を定義します。[ 属性 ] - [ 画面属性 ] を選択して表示される画面属性ダイアログで値を設定します。
9. 画面レイアウトを定義する。  
画面定義ウィンドウのツールボックスからオブジェクトのボタンを選んで、レイアウト領域にオブジェクトを配置し、画面をレイアウトします。  
オブジェクトについては、「3.4 CUI 画面を構成するオブジェクト」を参照してください。

10. 定義内容を保存し、終了する。

画面定義ウィンドウのツールバーから [ ドローの終了 ] を選ぶと定義内容が保存され、画面定義が終了します。このとき、定義ファイルに加え、論理マップと物理マップが生成されます。

注

[ 対象 ] から [ パターン ] を選択した場合には定義パターンを、[ 部品 ] を選択した場合には部品を作成できます。定義パターンや部品の作成については、「3.2.2 定義パターンおよび部品の登録」を参照してください。

### 3.2.1 マップ名の決定

画面の定義情報と生成される物理マップや論理マップなどの名称の整合性を図るために、マップ名を設定します。このマップ名は、各ファイル名称の先頭の文字列（拡張子を除く文字列）などになります。

マップ名に MAP001 という名称を付けた場合のマップ定義ファイル名称  
MAP001.imp になります。

マップ名を MAP001 とした場合に生成される物理マップファイル名称  
マップ名が 6 文字 の場合、MAP001NC.pmp になります。マップ名が 7 文字 の場合は、MAP001S.pmp になります。

注

マップ名の 6 文字または 7 文字については、ドローセットアップで設定します。  
ドローセットアップについては、マニュアル「XMAP3 開発・実行ガイド」を参照してください。

マップ名を MAP001 とした場合に、生成される論理マップファイル名称

- COBOL の場合

入力論理マップファイル名称は MAP001I.cbl、出力論理マップファイル名称は  
MAP001O.cbl になります。

- C 言語の場合

入力論理マップファイル名は MAP001I.h、出力論理マップファイル名称は  
MAP001O.h になります。

### 3.2.2 定義パターンおよび部品の登録

定義パターンおよび部品について説明します。なお、定義パターンおよび部品の登録の詳細手順については、マニュアル「XMAP3 開発・実行ガイド」を参照してください。

#### (1) 定義パターン、部品とは

ドローでは、画面のレイアウト定義のほかに、パターンや部品を必要に応じて定義できます。パターンとは、マップを効率良く作るために、定型的なオブジェクトを定義してあるひな型のマップのことです。部品とは、複数のオブジェクトを一つのグループとし

### 3. CUI 画面の作成

て定義したオブジェクトのことです。使用頻度の高い各種オブジェクトをあらかじめ部品として定義しておくことで、マップを効率良く作成できます。

#### (2) 定義パターンの登録

XMAP3 では、汎用的に使用できる定義パターンを標準提供していますが、このほかに、必要に応じてユーザ独自の定義パターンを用意しておけます。ユーザ独自の定義パターンは、レイアウト定義前に作成してください。

登録した定義パターンは、画面の新規作成時に、新規作成ダイアログのパターンの選択一覧に表示されます。

#### (3) 部品の登録

よく使用する属性のオブジェクトや、幾つかのオブジェクトを組み合わせたグループを部品として登録できます。オブジェクトを組み合わせるには、固定フィールドや入出力フィールドなどの複数のオブジェクトを、グループ化することで実現できます。部品を有効に使用することで、ユーザの画面定義を効率良く作成できます。部品を使用する場合は、レイアウト定義前に作成しておく必要があります。

#### (4) 定義パターンおよび部品登録時の注意

定義パターンと部品の登録は、ドロースタックアップの「運用管理者用の設定」で「パターンの登録・更新を許可」または「部品の登録・更新を許可」を選択しているときにできます。ただし、定義パターンおよび部品は、ユーザの開発プロジェクト内で必要なものを登録するため、プロジェクト管理者などの特定のユーザが登録することをお勧めします。



## 3.3 CUI画面属性の設定

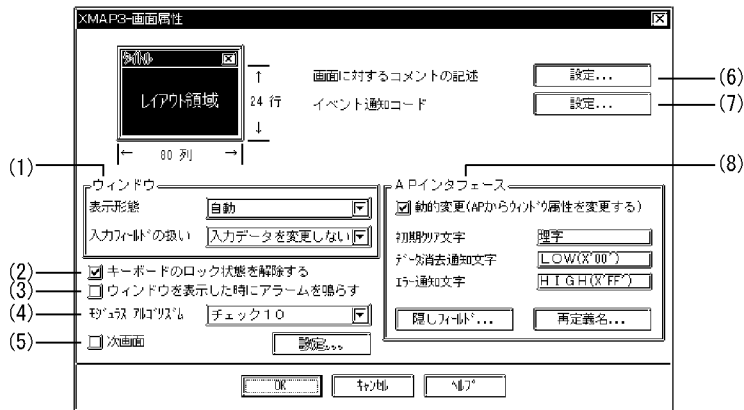
CUI画面の画面属性情報、およびイベント通知コードの詳細について説明します。

画面を制御する情報を画面属性情報といいます。画面属性情報は、ドローセットアップで初期値が設定されています。この制御情報は、セットアップ時、または画面定義時に設定できます。

画面属性情報に指定できる属性を次に示します。

### 3.3.1 画面属性の詳細

画面属性は、画面属性ダイアログで指定します。なお、次の図にある番号は、該当定義項目が説明されている項番を示しています。



#### (1) ウィンドウ

##### (a) 表示形態

新しく画面を表示したときの表示中の画面の扱いを変更できます。表示中の画面を消去して次の画面を表示する場合は「全面書換」、画面を重ねて表示する場合は「一部上書」を指定します。システムに表示形態を任せる場合は、「自動」を指定します。「全面書換」「一部上書」および「自動」の詳細については、「1.5.3(2) 表示形態」を参照してください。

### 3. CUI 画面の作成

#### (b) 入力フィールドの扱い

一度表示した画面に対して一部上書したときに、AP に返す値を次に示します。

操作	オブジェクトの表示属性		ウィンドウ属性 (入力フィールドの扱い)		
			入力データを AP に返す	入力データを変更しない	入力データを消去する
ヌルを入力	初期値なし	入力済	消去通知文字	消去通知文字	消去通知文字
	初期値なし	入力済でない	初期クリア文字	初期クリア文字	初期クリア文字
入力	初期値なし	入力済	表示データ	表示データ	消去通知文字
	初期値あり	入力済でない	初期値	初期値	初期値
	初期値なし	入力済でない	初期クリア文字	初期クリア文字	初期クリア文字
何もしない	初期値あり	入力済	初期値	初期値	初期値
		入力済でない	初期値	初期値	初期値
	初期値なし	入力済	消去通知文字	消去通知文字	消去通知文字
		入力済でない	初期クリア文字	初期クリア文字	初期クリア文字

#### 注

ウィンドウ属性 (入力フィールドの扱い) は、ドローセットアップの「表示属性の標準値」の「ウィンドウ属性」タブでも指定できます。なお、ドローセットアップでの値が有効になるのは、XMAP3 が標準提供している定義パターンを使用してドローを新規作成した場合、および XMAP3 が標準提供している定義パターンを使用して、ユーザ登録の定義パターンを作成したときです。

#### (2) キーボードのロック状態を解除する

キーボードのロック状態を直前の画面から継続するか、解除するかが指定できます。この指定をチェックすると、画面を表示したとき、マウスやキーボードから入力できる状態になります。指定しない場合は、キーボードのロック状態が直前の画面から継続されます。画面の操作では、「取消キー」によってキーボードのロック状態を解除できます。なお、キーの詳細については、「4.2.1 キーの機能」を参照してください。

#### (3) ウィンドウを表示したときにアラームを鳴らす

画面を表示したときにアラームを鳴らすかどうか指定できます。初期値は「鳴らさない」です。

#### (4) モジュラスアルゴリズム

モジュラスチェックを「(モジュラス)10チェック」ですか「(モジュラス)11チェック」であるかが指定できます。モジュラスチェックの詳細については、「付録 A 属性情報一覧」を参照してください。セットアップの初期値は「(モジュラス)10チェック」です。

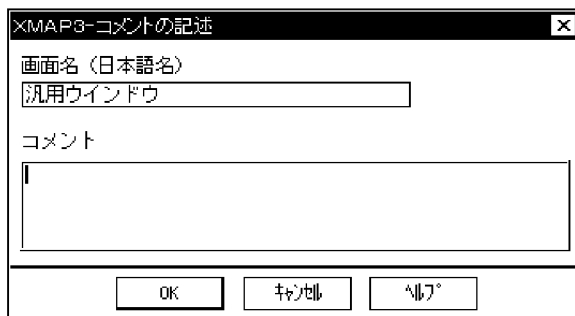
#### (5) 次画面

APで、画面名称を固定値として定義していなくても、次に表示させたい画面名称を実行時に取得したいとき指定します。次画面はマップ名が6文字のときに指定できます。なお、次画面はOLTPサーバ構成の環境で利用できます。

#### (6) 画面に対するコメントの記述

画面名(日本語名)と画面に関するコメントを書きます。各オブジェクトに付ける概要や使用目的などの情報です。実際の画面表示の対象にはなりません。

コメントはドキュメンテーション支援でほかの情報とともに出力されるので、定義情報をドキュメント化したときにわかりやすい情報が得られます。また、日本語名はテスト支援の遷移図の画面名として使用されます。ドキュメンテーション支援およびテスト支援についてはマニュアル「XMAP3 開発・実行ガイド」を参照してください。



XMAP3-コメントの記述

画面名 (日本語名)  
汎用ウインドウ

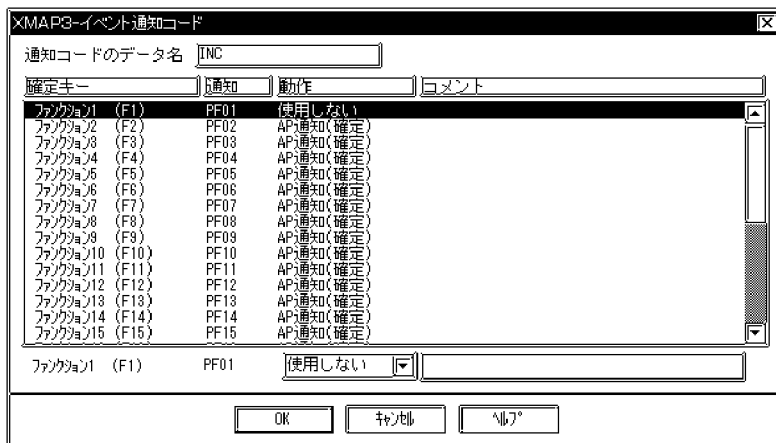
コメント

OK    キャンセル    ヘルプ

### 3. CUI 画面の作成

#### (7) イベント通知コードのコメント

確定キーに対応するイベント通知コードに、イベント発生後の動作やコメントを指定します。[設定] ボタンから表示されるダイアログで定義します。



イベントが発生したとき、AP に通知するかどうかはイベント通知コードダイアログの「動作」で指定できます。動作の詳細は次のとおりです。

動作の指定	説明
AP 通知 (確定)	画面確定をして AP にイベントを通知する。入力・選択必須のチェックが実行され、エラーがあった場合は AP に制御を返さない。
使用しない	使用しない PF キーなどに使う。これに対する確定キーを押すと、画面が確定される。AP にはイベント通知コードとして、初期クリア文字が返る。

注

OLTP サーバ構成の環境で、OpenTP1 上で「次メッセージ要求キー」として割り当てられているキーのイベントは AP に渡りません。

注

OLTP サーバ構成で使用している場合、リターンコードが「4」、詳細エラーコードが「(0414)<sub>16</sub>」のエラーとなります。

#### (8) AP インタフェース

AP からウィンドウの属性を変更して表示する場合に指定します。

##### (a) 動的変更 (AP からウィンドウ属性を変更する)

AP から表示形態などのウィンドウ属性を変更して、画面を表示します。

この場合、制御項目が論理マップに展開されます。この制御項目に修飾名という定数を

AP が設定することで、表示属性が動的に変更されます。

修飾名と表示属性との対応の標準値については「付録 A.10 修飾名と変更属性の標準値」を参照してください。なお、表示属性はドロースेटアップで変更できます。

#### (b) 初期クリア文字

AP 実行時に、画面からの入力データを AP が受け取る前に、XMAP3 が入力論理マップの各項目をあらかじめクリアしておく文字です。なお、AP が受け取るデータの優先順位は次のとおりです。

1. キー入力データ / 選択したイベント通知コード
2. 入力済みのとき、表示されているデータ / イベント通知コード
3. エラー通知
4. 初期値
5. 初期クリア文字

#### (c) データ消去通知文字

AP 実行時に、画面からフィールドキーで入出力フィールドのデータを消去したり、(00)<sub>16</sub> のデータを受信したときに AP が受け取るデータです。したがって、AP が受け取ったデータがデータ消去通知文字であれば、画面操作でデータ消去されたと判断できます。

初期クリア文字や初期値とデータ消去通知とを分けておくことで、入力操作がなかったのか、データ消去されたのかが区別できます。

なお、オブジェクトに「入力済み」が定義されている場合で、画面確定時に入力されたデータがないときは、データ消去通知文字が AP に返されます。

#### (d) エラー通知文字

XMAP3 が入力データにエラーを検出した場合に、AP が受け取るデータです。

入力データのエラーは、半角文字だけを入力できるテキスト・フィールドに、全角文字を入力したときなどに発生します。

#### (e) 隠しフィールド

実際の画面には表示されないフィールドで、入力固定項目として使用します。

入力固定項目とは、入力論理マップに必ず固定の値を返すデータ項目で、論理マップ可変部の最初のデータ項目として展開されます。

OLTP サーバ構成の環境で、使用するトランザクションコードを XMAP3 上で定義するには、隠しフィールドとして定義します。

#### (f) 再定義名

定義済みの論理マップを別定義として使用する際の名称です。半角で 30 字以内で定義します。

### 3. CUI 画面の作成

入力論理マップ用と出力論理マップ用とで別の名称にすることもできます。

#### 3.3.2 画面属性の動的変更

画面属性は AP 実行時に変更できます。XMAP3 では、ドローで定義した画面属性情報のほかに、AP からの変更用としてドローセットアップで画面属性情報を定義できます。

これらの画面属性情報は、名前を付けて区別します。この名前を修飾名といいます。標準提供の画面属性情報の修飾名は「付録 A.10 修飾名と変更属性の標準値」を参照してください。標準提供の修飾名を使用すると、表示形態を AP から変更できます。AP では、論理マップの制御項目に画面属性情報の修飾名を格納することで、実行時に画面属性の表示属性を変更できます。

## 3.4 CUI画面を構成するオブジェクト

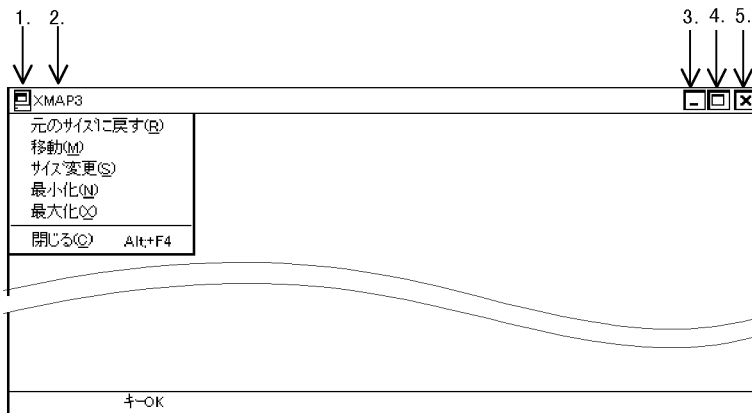
CUI画面を構成するオブジェクトと、それぞれの機能について説明します。なお、オブジェクトは機能によって分類されます。オブジェクト分類については「2.4 GUI画面を構成するオブジェクト」の一覧表を参照してください。

### 3.4.1 タイトルバー

CUI画面および各部の表示仕様を次に示します。

#### (1) 形式

図 3-1 CUI画面



#### (2) 表示仕様

タイトルバーの表示仕様を次の表に示します。

表 3-1 タイトルバーの表示仕様およびその他の領域

項番	名称	機能	使用条件, その他
1	タイトルバーアイコンまたはコントロールボックス	XMAP3 システムのアイコンまたはコントロールメニューを出すための、ボタンを表示します。クリックすると最小化メニューなどが表示されます。	タイトルバーアイコンが表示されません。 表示・印刷セットアップで、表示するかしないかを設定できます。
2	ウィンドウタイトル	表示されているウィンドウのタイトルです。	常に「XMAP3」が表示されます。 表示・印刷セットアップで、タイトルバーを表示するかしないかを設定できます。
3	最小化ボタン	ウィンドウをアイコン化します。	表示・印刷セットアップで、表示するかしないかを設定できます。

### 3. CUI 画面の作成

項番	名称	機能	使用条件, その他
4	最大化ボタン	ウィンドウのサイズを最大にします。	表示・印刷セットアップで、表示するかしないかを設定できます。
5	閉じるボタン	ウィンドウを閉じ、APを終了させるために使用します。	表示・印刷セットアップで、表示するかしないかを設定できます。

#### (3) AP インタフェース

- ・タイトルバーのタイトルは AP から変更できません。
- ・[ 閉じる ] ボタンはオプションです。[ 閉じる ] ボタンをクリックすると、[ Break ] キーと同じイベントが AP に通知されます。

### 3.4.2 AP への通知

キーボード上で確定キーが押されたというイベントが発生した場合、イベント通知コードと呼ばれるコードが発生します。

ドロセットアップで確定キーに対する任意のコードが設定できます。通常は、ドロセットアップの標準値を使用することをお勧めします。

#### (1) AP への通知

イベント通知コードとして使用できるものと、それらの送信方法、および送信のタイミングについては「1.5.9 画面確定と AP への通知」を参照してください。

### 3.4.3 入出力フィールド

入出力フィールドは、キーボードなどからデータを入力するとき、および AP から渡されるデータを表示するときに使用します。初期表示値（出力データ）を表示し、書き換えが必要な場合だけ入力するような使い方ができます。

フィールドの表示属性の標準値は文字色が「緑」です。指定できる表示属性の詳細については、「付録 A 属性情報一覧」を参照してください。

#### (1) 表示形式

入出力テキストボックスと同じです。ただし、入出力フィールドの場合、アンダードットが付きます。枠は付きません。また、入力データの長さが、入力開始位置から行の最後までを越える場合、入力しきれないデータを継続行として次行に入力できます。このとき、データの入力開始位置はフィールドのどこであってもかまいません。

(例)

.....12345678



### 反復表示

フレームを利用して反復表示ができます。フレームの詳細は「3.4.6 フレーム（横／縦繰り返し）」を参照してください。

## (2) 入力データのチェック

扱うデータの種類を決めたり、空白の使い方を指定して、入力データをチェックします。

### (a) 使用目的による入力データのチェック

使用目的によって、入力データをチェックできます。使用目的と入力データの関係については、「付録 A.9 表示属性の解説」の表 A-8 を参照してください。

## (3) カーソルの遷移

遷移条件が指定できます。

- 手動スキップ・右寄せ  
入力データが桁数に満たない場合、タブキーで次の入力できる領域にカーソルを移動すると、入力データが右寄せされます。
- 手動スキップ・全桁必須  
1文字でも入力した場合、桁数いっぱいに入力しないと入力エラーになり、タブキーでのフォーカス・カーソル移動ができなくなります。
- 手動スキップ  
桁数を超えるデータを入力しようとするとエラーになります。
- 自動スキップ  
桁数いっぱいまで文字を入力すると、カーソルが次の入力できる領域に自動的に移動します。これに対して、手動スキップの場合はタブキーを押すことでカーソルが次の入力できる領域に移動します。

## (4) 入力時の AP インタフェース

AP の入力論理マップには次の内容を格納します。

### (a) 入力データ項目

#### 下位項目を利用したチェック

必要に応じて下位項目を設定できます。下位項目を設定すると、データをコードデータのように区切ってチェックしたり、1文字の繰り返しデータのようにして、1文字ずつチェックしたりできます。下位項目の詳細については、「6.2.10 下位項目を利用した入出力テキスト」を参照してください。

#### 入力データのチェック

どのような操作が起こったかをチェックするためのコードを通知します。

- コードエラーの通知  
オブジェクトで受け付けるデータやチェック内容を指定し、不正な文字を入力した場合にエラー通知文字（標準は HIGH (X'FF') ) を AP へ返します。

### 3. CUI 画面の作成

- データ消去の通知  
フィールドキーでデータを消去した場合などに、データ消去通知文字（標準は LOW (X'00')）を AP へ返します。
- データ未入力の通知
  - ・初期値を返す場合  
(例)  
テキスト・フィールドで初期値に「ABC」の文字を設定し、データを入力しない場合に初期値の「ABC」を AP へ返します。
  - ・初期クリア文字を返す場合  
初期値の文字を設定しないで、データを入力しない場合に初期クリア文字（標準は埋字と同じ文字）を返します。  
(例)  
使用目的が「英数」の場合は埋字（スペース）を、「数字」の場合は埋字（ゼロ）を AP へ返します。  
エラー通知文字、データ消去通知文字および初期クリア文字は、ドロースettingsで変更することもできます。

#### (b) 入力バイト数格納項目（オプション）

実際にキー入力した文字のバイト数（半角の桁数）が入ります。入力バイト数格納項目については「5.3.8 入力バイト数格納項目の論理マップ生成規則とマッピング規則」を参照してください。

#### (5) 入出力フィールドでの定義と AP の実行結果

XMAP3 での入出力フィールドの定義と、AP での実行結果の関係は「付録 A.9 表示属性の解説」の表 A-8 を参照してください。なお、CUI 画面の入出力フィールドでは「詳細目的」は指定できません。

#### (6) 出力時の AP インタフェース

出力時の AP インタフェースについては、「3.4.5 出力フィールド」を参照してください。

#### (7) 表示属性の動的変更

入出力フィールドでは、表示属性を AP 実行時に変更できます。表示属性の動的変更については「1.5.6 表示属性の動的変更」を参照してください。

### 3.4.4 固定フィールド

固定フィールドは、ドロースで定義した固定の文字列を表示するときに使用します。全角文字を使用する「日本語」と半角文字だけを使用する「英数」があります。

固定フィールドの表示属性の標準値は、文字色が「黄色」、文字の拡大が「標準」、テキスト揃えが「左」です。指定できる表示属性の詳細については、「付録 A 属性情報一

覧」を参照してください。

### (1) 表示形式

表示されている文字に対し、文字サイズを指定できます。

<拡大>

あ	あ	あ	あ
標準文字	平体	長体	倍角

### (2) 主な機能

- 反復表示  
フレームを使用して、繰り返し表示ができます。フレームの詳細は「3.4.6 フレーム（横／縦繰り返し）」を参照してください。
- 使用目的の指定  
表示する文字に応じて、ドロー上で使用目的（日本語、英数）を選べます。
- 複数行にわたるサイズの配置  
行を折り返して複数行にわたるサイズで配置できます。ただし、文字の拡大や反復定義との併用はできません。

## 3.4.5 出力フィールド

出力フィールドは、APから渡される可変データを表示するときに使用します。表示するデータに応じて、データの種別（使用目的）を選べます。出力フィールドの表示属性の標準値は文字色が「緑」、文字の拡大が「標準」です。出力フィールドの表示属性の詳細については、「付録A 属性情報一覧」を参照してください。

### (1) 表示形式

表示されている文字に対し、文字サイズを指定できます。

<拡大>

日立	日立	日立	日立
標準文字	平体	長体	倍角

### 3. CUI 画面の作成

#### (2) 主な機能

- 反復表示  
フレームを使用して、繰り返し表示ができます。フレームの詳細は「3.4.6 フレーム (横/縦繰り返し)」を参照してください。
- 使用目的の指定  
表示する文字に応じて、使用目的 (数字、英数、日本語) を選べます。
- 複数行にわたるサイズの配置  
行を折り返して複数行にわたるサイズで配置できます。ただし、文字の拡大や反復定義との併用はできません。
- 出力データの編集  
出力フィールドでは、埋字や桁寄せなどのデータの編集もできます。AP がデータ有無コードを格納し、出力論理マップにセットした有効データの長さが出力フィールドデータ名の領域の長さより短い場合、出力フィールドデータ名の領域の余った部分を埋字に置き換えます。また、出力データを出力フィールドデータ名の領域の右側に格納するか、左側に格納するかは、桁寄せ向きに従います。
- 下位項目の設定  
データ型が「文字 (XX)」の場合は、下位項目を指定してデータを細分化できます。

#### (3) 出力時の AP インタフェース

COBOL の場合、ピクチャ文字による編集ができます。なお、C 言語の場合、すべて「char」として扱われます。

##### (a) 出力データ

###### 数字

数字データ、数字編集データを表示できます。数字編集データは、指定した PICTURE 句に従って表示されます (COBOL だけ有効)。

使用できるピクチャ文字:「\*」「+」「-」「¥」「9」「,」「.」「Z」「/」の組み合わせで指定します。

###### 日本語

全角または全角と半角の混在文字データを表示できます。全角と半角の混在文字データを表示する場合は、下位項目を指定してデータを細分化できます。

###### 英数

半角の文字データを表示できます。また、下位項目を指定してデータを細分化できます。

##### (b) 出力データの編集

出力フィールドでは、埋字や桁寄せなどのデータの編集もできます。AP がデータ有無コードを格納し、出力論理マップに代入した有効データの長さが出力フィールドデータ名の領域の長さより短い場合、出力フィールドデータ名の領域の余った部分を埋め字に置き換えます。また、出力データを出力フィールドデータ名の領域の右側に格納するか、左側に格納するかは、桁寄せ向きに従います。

#### (4) 表示属性の動的変更

出力フィールドの表示属性は実行時に変更できます。表示属性の動的変更については「1.5.6 表示属性の動的変更」を参照してください。

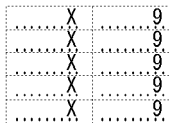
### 3.4.6 フレーム（横／縦繰り返し）

フレームで指定した領域内の先頭行または先頭列にオブジェクトを配置すると、配置したオブジェクトがフレームで指定した行数分または列数分、反復表示されます。入出力フィールドと出力フィールドを1行として反復表示することもできます。APでは、この1列および1行を集団項目として扱います。また、論理マップでは反復表示されたオブジェクトを配列として展開します。

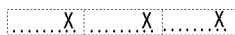
#### (1) 表示形式

フレーム自体は表示されません。ドロー上では、オブジェクトをまとめる点線の枠で表示されます。

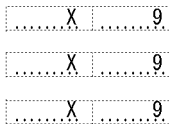
〈縦に繰り返す〉



〈横に繰り返す〉



〈1行置きに繰り返す〉



#### (2) フレームに配置できるオブジェクト

CUI画面で設定するフレームに配置できるオブジェクトは次のとおりです。

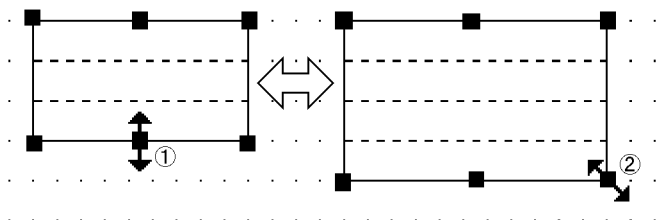
- 入出力フィールド
- 固定フィールド
- 出力フィールド
- 縦／横けい線

#### (3) フレームの操作

フレームのハンドル( )をドラッグしたり、行または列を選択して行の高さや幅を変更できます。

- フレームの反復回数とサイズの変更  
フレーム全体を選択し、表示されるハンドル( )をドラッグします。

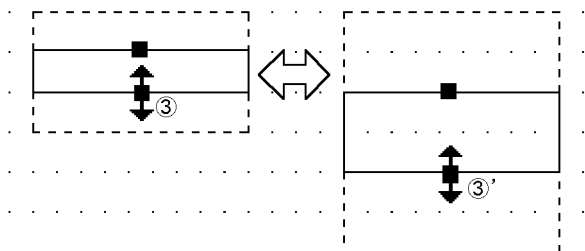
### 3. CUI 画面の作成



- (例) ①を下にドラッグすると反復回数が増加します。  
①を上ドラッグすると反復回数が減少します。  
②を左上にドラッグすると、サイズが小さくなります。  
②を右下にドラッグすると、サイズが大きくなります。

#### • 行の高さまたは列の幅の変更

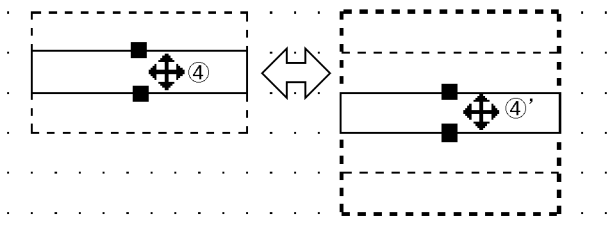
フレームの先頭または最終以外の行（または列）をクリックし、行（または列）に表示されるハンドル（ ）をドラッグします。フレーム内のすべての行の高さ（または列の幅）が変更されます。なお、拡大する場合は、あらかじめフレームの変更する方向の領域を空けておく必要があります。



- (例) ③を下にドラッグすると、行の高さが大きくなります。  
③'を上ドラッグすると、行の高さが小さくなります。

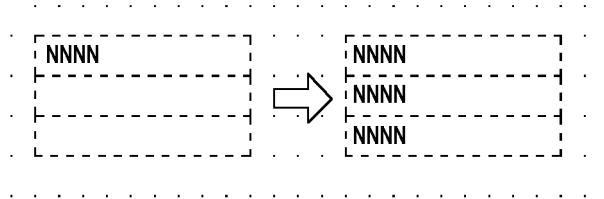
#### • フレームの繰り返し間隔の変更

フレームの先頭または最終以外の行（または列）をクリックし、選択した行（または列）をドラッグします。フレームの行（または列）が1行（または1列）おきに繰り返されます。

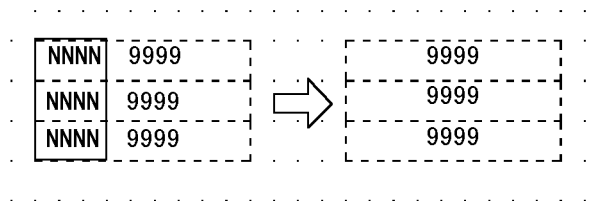


- (例) ④を下にドラッグすると、フレームの間隔が空きます。  
④'を上ドラッグすると、繰り返しがなくなります。

- フレームへのオブジェクトの追加  
1 行目（または 1 列目）にオブジェクトを配置します。配置すると行（または列）方向にオブジェクトが繰り返されます。



- フレーム内のオブジェクトの削除  
1 行目（または 1 列目）のオブジェクトを選択します。行（または列）のオブジェクトが選択状態になったら [ 削除 ] を選択します。



#### （4）AP インタフェース

フレームは AP インタフェースを持ちません。フレーム内に配置されたオブジェクトごとの AP インタフェースは有効です。

### 3.4.7 けい線

けい線は縦方向、または横方向に 1 本の線を引きます。配置した複数のオブジェクトの間を区切るときに使用します。

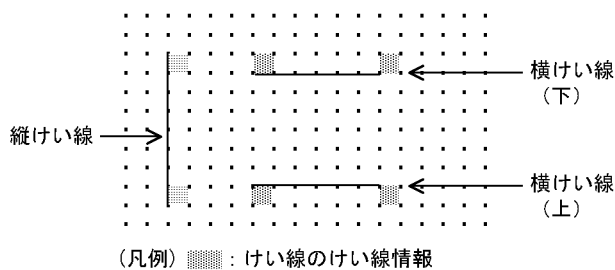
なお、けい線は、AP とのインタフェースを持ちません。そのため、けい線を追加、修正、または削除しても AP に影響はありません。また、けい線はボックスを通過して引いたりすることはできません。なお、CUI 画面で使用できるけい線は、実線だけです。

#### （1）線の位置

CUI 画面の横けい線の位置は、行の「上 / 下」から選べます。縦けい線には位置の種類はありません。

### 3. CUI 画面の作成

図 3-2 けい線の位置



#### (2) 矩形および表けい線

けい線はオブジェクトを区切る以外にも次のように使用できます。

##### (a) 矩形

指定した範囲をけい線で囲むときに使用します。

##### (b) 表けい線の枠

表を作成するときに使用します。表に見出しを付けるか付けないかも指定できます。

##### (c) 表けい線の繰り返し

縦方向、または横方向に等間隔のけい線を引くときに使用します。横方向の繰り返しの間隔は「2行 / 1行 / 自動」から選べます。

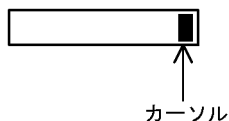
また、けい線は、横けい線の始点、交点、および各行の縦けい線に半角文字で1文字分のけい線情報用の領域を占有します。このため、けい線を画面上に配置する場合には、ほかの項目と重ならないようにします。

なお、けい線は複数行にわたるフィールドの途中の行には引けません。また、フィールドを通過して引くこともできません。

## 3.4.8 カーソル

カーソルが位置づいているオブジェクトは、ユーザが操作できる状態になっています。オブジェクトに対する操作は、必ずマウス、またはキーボードでカーソルを位置づけたあとで行います。

#### (1) 表示形式





## (2) 初期カーソル

各ダイアログのカーソル設定によって、画面を表示したとき、最初にカーソルを位置づけるオブジェクトを設定できます。

## (3) 初期カーソルの AP からの設定

カーソル定数という値を論理マップに代入すると、初期カーソルを位置づけるオブジェクトを AP から設定できます。これらの値が設定されていないと、ドローで定義した初期カーソルの指定が採用されます。カーソル定数については「5.5 定数部の論理マップ生成規則とマッピング規則」を参照してください。

## (4) オブジェクトの遷移条件

- 手動スキップ・右寄せ  
入力データが桁数に満たない場合、[ Tab ] キーで次の入力できる領域にカーソルを移動すると、入力データが右寄せで表示されます。
- 手動スキップ・全桁必須  
一文字でも入力した場合、桁数いっぱいに入力しなければ入力エラーになり、[ Tab ] キーでのカーソル移動ができなくなります。
- 手動スキップ  
桁数いっぱいまで入力し、[ Tab ] キーを押すとカーソルが次の入力できる領域に移動します。桁数を超えるデータを入力するとエラーになります。
- 自動スキップ  
桁数いっぱいまで入力すると、カーソルが次の入力できる領域に自動的に移動します。

### 3.4.9 予約フィールド

OpenTP1 がサポートする予約項目名称を定義するオブジェクトです。論理マップ上にデータを設定していなくても、実行時に予約項目名に応じた内容で画面に出力されます。予約フィールドは OLTP サーバ構成の環境で利用できます。

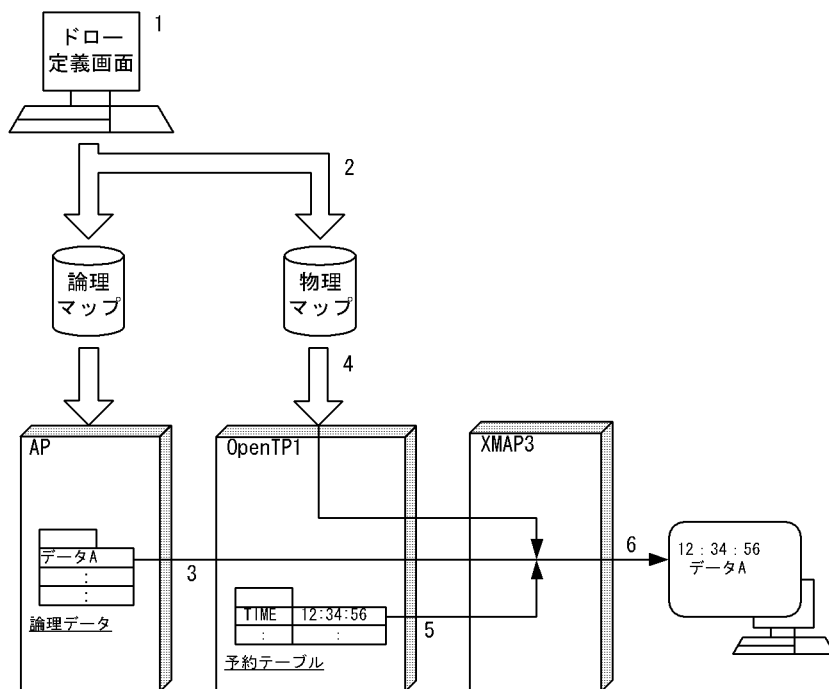
なお、予約フィールドの表示属性の詳細については「付録 A 属性情報一覧」を参照してください。

#### (1) 表示処理

予約フィールドの出力の仕組みを次に示します。

1. ドロー定義画面で予約テキスト（または予約フィールド）「TIME」を定義します。
2. 物理マップ中に、予約項目情報「TIME」が設定されます。
3. AP からの SEND 文によって論理データが指定されます。
4. OpenTP1 に物理マップが読み込みます。
5. OpenTP1 によって予約テーブルが生成され、論理データ、物理マップ、および予約テーブルなどが XMAP3 に渡されます。
6. XMAP3 によって画面にデータが表示されます。

### 3. CUI 画面の作成



#### (2) 予約項目名

予約フィールドに定義する項目名称は OpenTP1 で提供されている予約項目名を指定します。OpenTP1 が提供する予約項目名はドローで定義しておきます。論理マップにデータを設定しなくても、OpenTP1 から生成された日付、時刻などのデータを、項目名に対応したフォーマットで画面上に表示できます。

予約項目名については、マニュアル「OpenTP1 プロトコル TP1/NET/XMAP3 編」を参照してください。

### 3.4.10 次画面

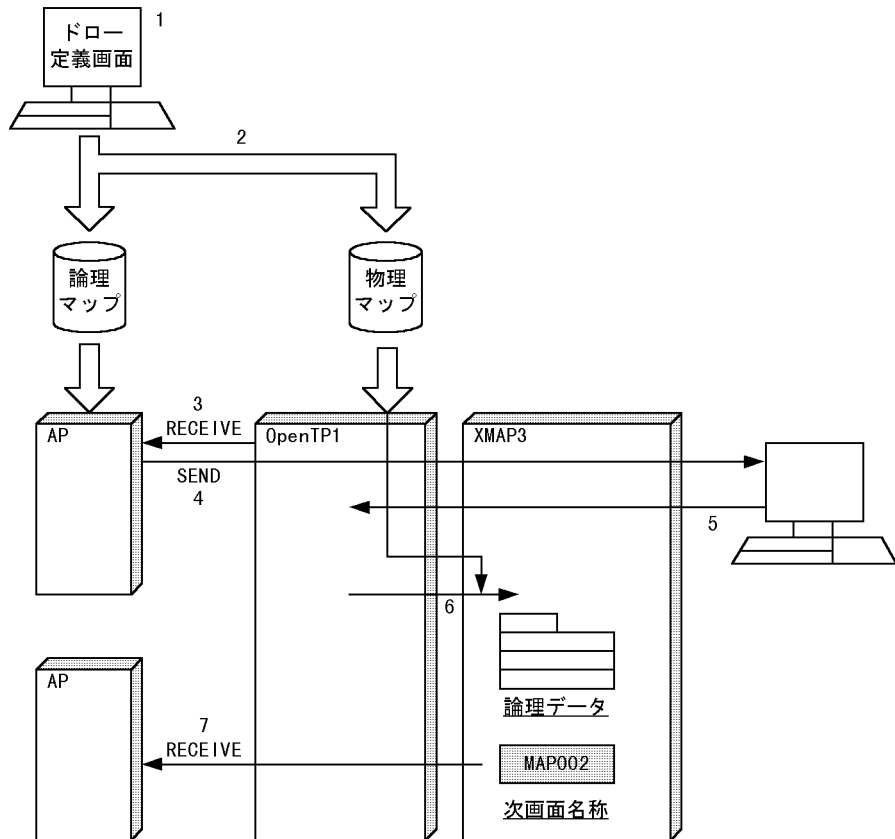
OLTP サーバ構成の環境で、次に表示したい画面を定義するオブジェクトです。通常、ある画面から次に表示したい画面は、AP 上で画面名称を定義しておく必要があります。このようなとき次に表示したい画面をドロー上で定義しておく、AP で画面名称を定義していなくても実行時に、画面名称を取得して表示できます。

#### (1) 表示処理

次画面の表示の仕組みを次に示します。

1. ドロー定義画面で次画面名称「MAP002」を定義します。
2. 物理マップ中に次画面名称「MAP002」が設定されます。
3. OpenTP1 によって AP が起動され、AP から RECEIVE 要求が発行されます。

4. AP から SEND 要求が発行され画面が表示されます。
5. 表示された画面上で入力操作を実行するとイベントが発生します。
6. 発生したイベントに対する物理マップで XMAP3 に RECEIVE 要求され, OpenTP1 に論理データと物理マップ中の次画面名称「MAP002」を返します。
7. OpenTP1 によって, 次のトランザクションに対応する AP が起動されます。このとき, マッピングによって生成された論理データと次画面名称「MAP002」は, RECEIVE 要求で AP に渡されます。



## 3.5 部品の利用

---

あらかじめ登録しておいた部品をレイアウト画面上に配置するときの手順を次に示します。

1. ドローのメニューバーの [ 編集 ] から [ 部品貼り付け ] を選ぶ。  
部品貼り付けダイアログが表示されます。
2. 使用する部品名称を選ぶ。  
部品がレイアウト画面の左上に貼り付けられますので、配置したい位置に移動してください。

なお、ユーザが貼り付けた部品と、すでに定義されているオブジェクトとの関係は次のようになります。

### データ名

貼り付けた部品と既存のオブジェクトとでデータ名が重複している場合、部品で使われているデータ名が XMAP3 の命名規則に従って変更されます。データ名が重複していない場合は、部品のデータ名がそのまま使われます。

- 言語種別が異なる場合  
部品の中に、既存のオブジェクトの言語種別では使用できない文字（ハイフンとアンダーラインは除く）が指定されている場合、XMAP3 の命名規則に従って変更されます。
- マップ名の長さが異なる場合  
マップ名の長さが 6 文字で設定されている場合、データ名として指定できる文字数は 22 文字までですが、マップ名の長さが 7 文字で設定されている場合は、21 文字までです。このため、22 文字のデータ名を、マップ名の長さが 7 桁で設定されているレイアウトに貼り付けると、XMAP3 の命名規則に従って変更されます。

### 通知コード

通知コードの重複はチェックされません。そのため、部品を貼り付けたあと、必要に応じてダイアログを開き、変更する必要があります。

## 3.6 データ名に関するコメントの編集

---

ドローでレイアウトされたオブジェクトには、各オブジェクトの定義ダイアログでオブジェクトに関するコメントを書けます。また、ドローのデータ名編集機能を使用して、レイアウト画面上に定義しているすべてのオブジェクトのコメントを表示し、必要に応じて編集できます。データ名編集機能の詳細は「2.6 データ名の編集」を参照してください。

## 3.7 テスト表示

---

CUI 画面の設計が終了したら、実行時の表示状態を確認するためにテスト表示をします。テスト表示をするときは、[ ツール ] メニューの [ テスト表示 ] を選ぶか、ツールバーから [ テスト表示 ] ボタンを選びます。また、Web ブラウザテスト表示を選択すると、ブラウザを使ったテスト表示ができます。

テスト表示では、データの入力やカーソルの移動などが確認できます。

テスト表示では、画面確定を容易にするために、正しい入力や操作をしないと画面が確定できないような属性を無効にしています。このため、テスト表示では、実際の AP の実行結果と動作が異なることがあります。無効となる属性については、「付録 A 属性情報一覧」を参照してください。

なお、予約フィールドは「X」(項目長分だけ X で埋められた形式) で表示されます。また、次画面の機能をテスト表示することはできません。

# 4

## マウスとキー操作

この章では、XMAP3 で作成した画面でのキーの機能およびマウスの操作について説明します。

---

4.1 有効となるキー

---

4.2 各制御キーの機能

---

4.3 マウス操作

---

4.4 コピー & ペースト操作

---

## 4.1 有効となるキー

### 4.1.1 データ入力キー

#### (1) 入力できる半角文字

ANK コード（英字・数字・カナ文字）を図 4-1 に、このマニュアルで使用している用語が ANK コードの何に該当するかを表 4-1 に示します。なお、(00) ~ (1F) および (7F) は、制御コードであるため、文字コードとして使用できません。

図 4-1 ANK コード

JISコード

列 行	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0			SP	0	@	P	`	p	▲	▲	未定義	一	タ	ミ	▲	▲
1			!	1	A	Q	a	q			。	ア	チ	ム		
2			"	2	B	R	b	r			「	イ	ツ	メ		
3			#	3	C	S	c	s			」	ウ	テ	モ		
4			\$	4	D	T	d	t			・	エ	ト	ヤ		
5			%	5	E	U	e	u			。	オ	ナ	ユ		
6			&	6	F	V	f	v	未定義	未定義	ヲ	カ	ニ	ヨ	未定義	未定義
7			'	7	G	W	g	w	未定義	未定義	ア	キ	ヌ	ラ	未定義	未定義
8			(	8	H	X	h	x	未定義	未定義	イ	ク	ネ	リ	未定義	未定義
9			)	9	I	Y	i	y			ウ	ケ	ノ	ル		
A			*	:	J	Z	j	z			エ	コ	ハ	レ		
B			+	:	K	[	k	{			オ	サ	ヒ	ロ		
C			,	<	L	¥	l				ヤ	シ	フ	ワ		
D			-	=	M	]	m	}			ユ	ス	ヘ	ン		
E			.	>	N	^	n	~			ヨ	セ	。			
F			/	?	O	_	o	~	▼	▼	ツ	ソ	。	▼	▼	▼

表 4-1 用語と ANK コードの関係

用語	該当する ANK コードの範囲
カナ	(A6) <sub>16</sub> ~ (DF) <sub>16</sub> ただし、(B0) <sub>16</sub> ( - ) を除く。
半角	(21) <sub>16</sub> ~ (FE) <sub>16</sub> ただし、(7F) <sub>16</sub> を除く。
英大文字	(41) <sub>16</sub> ~ (5A) <sub>16</sub>
英小文字	(61) <sub>16</sub> ~ (7A) <sub>16</sub>
マイナス	(2D) <sub>16</sub>
長音	(B0) <sub>16</sub>
コンマ	(2C) <sub>16</sub>
数字	(30) <sub>16</sub> ~ (39) <sub>16</sub>



用語	該当する ANK コードの範囲
記号	(21) <sub>16</sub> ~ (2F) <sub>16</sub> , (3A) <sub>16</sub> ~ (3F) <sub>16</sub> , (5B) <sub>16</sub> ~ (60) <sub>16</sub> , (7B) <sub>16</sub> ~ (7E) <sub>16</sub>
アスタリスク	(2A) <sub>16</sub>
漢字	(8140) <sub>16</sub> ~ (F9FC) <sub>16</sub>
空白 (半角)	(20) <sub>16</sub>
空白 (全角, 半角)	(8140) <sub>16</sub> (2020) <sub>16</sub> (20) <sub>16</sub>
空白 (全角だけ)	(8140) <sub>16</sub>

注

詳細は表 4-2 を参照してください。

## (2) 入力できる全角文字

漢字コードを次の表に示します。

表 4-2 漢字コード

	コードの範囲 <sup>1</sup>	使用できる文字
標準漢字領域	(8140) <sub>16</sub> - (EEFC) <sub>16</sub>	JIS X0208 にベンダ固有文字 <sup>2</sup> を含む文字
外字領域	(F040) <sub>16</sub> - (F9FC) <sub>16</sub>	外字 1880 字
ベンダ領域	(FA40) <sub>16</sub> - (FCFC) <sub>16</sub>	標準漢字領域以外の JIS X0208 にベンダ固有文字 <sup>3</sup> を含む文字

注 1

コードの第 1 バイトは (81xx)<sub>16</sub> ~ (9Fxx)<sub>16</sub>, (E0xx)<sub>16</sub> ~ (FCxx)<sub>16</sub> の範囲です。

コードの第 2 バイトは (xx40)<sub>16</sub> ~ (xx7E)<sub>16</sub>, (xx80)<sub>16</sub> ~ (xxFC)<sub>16</sub> の範囲です。

注 2

標準漢字領域のベンダ固有文字とは、Windows 特殊文字 ((8740)<sub>16</sub> ~ (879C)<sub>16</sub> 83 字), NEC 選定 IBM 拡張文字 ((ED40)<sub>16</sub> ~ (EEFC)<sub>16</sub> 374 字) のことです。

注 3

ベンダ領域のベンダ固有文字とは、IBM 選択文字 ((FA40)<sub>16</sub> ~ (FC4B)<sub>16</sub> 388 字) のことです。

### ベンダ固有文字について

ベンダ固有文字は PC 環境に依存します。WS を含めた C/S システム構成で運用する場合、文字コードの一部が WS と PC で異なるため、PC クライアント上で不正な文字が表示される場合があります。このため、WS を含めた C/S システム構成では、ベンダ固有文字を極力使用しないことをお勧めします。WS を含めた C/S システム構成でベンダ固有文字を使用したい場合は、Windows で定義できる外字領域に外字

## 4. マウスとキー操作

として作成，登録してご使用ください。

### (3) 全角文字とスペースコード

全角スペースを入力した場合，特に設定しないと半角スペース 2 文字に変換されて，画面上に表示し，AP に返されます。全角スペースをそのまま全角文字として画面上に表示し，AP に返したいときは，表示・印刷セットアップの「全角スペースコードの扱い」で，「全角スペースをそのまま返す」を設定してください。

### (4) 注意が必要なキー操作

Windows 95，Windows 98，Windows Me の場合に，注意が必要なキー操作を次に示します。

- [ Alt ] + [ F6 ] キーまたは [ Alt ] + [ Shift ] + [ F6 ] キーは Windows の画面重なり制御に割り当てられているため，使用できません。
- [ Shift ] キーを押しながら，テンキーの「+」または「-」を押した場合，OS のキーメッセージ処理の仕様によって，[ Shift ] キーは無視されます。そのため，この場合には，[ Shift ] キーを押さずにテンキーの「+」または「-」を押した場合と同じ動作となります。

## 4.1.2 制御キー

XMAP3 で標準に使用できる，画面を制御するキーについて説明します。XMAP3 では，標準仕様のキーボードから入力されるキーを制御します。キーには，XMAP3 が処理するものと，AP にキーが押されたことを通知するものがあります。

ノート型 PC など，テンキーを持たないキーボードを使用する機種では，表示・印刷セットアップで文字キー部の Enter (リターン) キーを復改キーとして使用するか，送信 (実行) キーとして使用するかを指定できます。運用方法に従って設定してください。

XMAP3 で使用できるキーの種類と機能の一覧を表 4-3 ~ 表 4-7 に示します。なお，表中のフィールドボックスは GUI 画面のフィールドボックスおよび CUI 画面を示します。「入出力フィールド」は GUI 画面のフィールドボックス内の入出力フィールド，および CUI 画面の入出力フィールドを示します。また，「ボックス」は GUI 画面のボックス系オブジェクトを表します。

表 4-3 キーの種類と機能（画面確定キー）

機能名称	キーボード上の標準割り当て	機能	ほかのキーへの割り当て
送信（実行）キー	Enter（テンキー部）（106 / 109）（AX） -（PC-9801）	APに画面の確定を通知する。	
PF1～PF12キー	F1～F12（106 / 109）（AX） PF1～PF10はf・1～f10 PF11, PF12はvf・1, vf・2（PC-9801）		
PF13～PF24キー	Shift + F1～F12（106 / 109）（AX） SHIFT + f・1～f10, vf・1, vf・2（PC-9801）		
PF25～PF36キー <sup>1</sup>	Ctrl+F1～F12（106 / 109）（AX） CTRL+f・1～f・10, vf・1, vf・2（PC-9801）		
PF37～PF48キー <sup>1</sup>	Ctrl+Shift+F1～F12（106 / 109）（AX） CTRL+SHIFT+f・1～f・10, vf・1, vf・2 （PC-9801）		
PF49～PF60キー <sup>1</sup>	Alt+F1～F12（106 / 109）（AX） <sup>2</sup> GRPH+f・1～f・10, vf・1, vf・2（PC-9801）		
PF61～PF72キー <sup>1</sup>	Shift+Alt+F1～F12（106 / 109）（AX） SHIFT+GRPH+f・1～f・10, vf・1, vf・2 （PC-9801）		
PF73～PF84キー <sup>1</sup>	Ctrl+Alt+F1～F12（106 / 109）（AX） CTRL+GRPH+f・1～f・10, vf・1, vf・2 （PC-9801）		
割込キー	Break, Ctrl+Pause, Ctrl+Scroll Lock （106 / 109）（AX） STOP（PC-9801）		×
スクリーン消去キー	Ctrl+Delete（106 / 109）（AX） CTRL+DEL（PC-9801）		×
PA1キー	Page Up（106 / 109）（AX） ROLL DOWN（PC-9801）		
PA2キー	Page Down（106 / 109）（AX） ROLL UP（PC-9801）		
PA3キー	Ctrl + Page Up（106 / 109）（AX） CTRL + ROLL DOWN（PC-9801）		×
無通知コード割り当てキー <sup>3</sup>	Ctrl + I Ctrl + S Ctrl + U		×

（凡例）

：ほかのキーへの割り当てができる。表示・印刷セットアップの「キー割当」で設定する。

×：カスタマイズできない。

#### 4. マウスとキー操作

: ほかのファンクションキー (PF1 ~ PF84) への割り当てができる。表示・印刷セットアップの「ファンクションキー」で設定する。

(106 / 109): 106 / 109 日本語, 101 / 104 英語。

(PC-9801): PC-9801 キーボード。

(AX): AX 仕様キーボード。

##### 注

- ・ PF25 ~ PF84 キーを押した場合は、日本語入力システム (IME) での処理が優先されるので、XMAP3 の画面に対する操作ができなくなることがあります。
- ・ 複数の IME がインストールされている場合、[ Alt ] + [ Shift ] または [ Ctrl ] + [ Shift ] は IME の切り替えキーとして割り当てられます。このとき、[ Alt ] + [ Shift ] + ファンクションキーまたは [ Ctrl ] + [ Shift ] + ファンクションキーを押そうとすると、IME が切り替わってしまうことがあります。使用するファンクションキーと IME の切り替えキーの設定が重複しないようにしてください。

##### 注 1

GUI 固有のファンクションキーです。CUI では使用できません。なお、既存の画面に対して確定キーとして使用する場合は、ユティリティで「セットアップ情報反映」を実行してください。

##### 注 2

- ・ [ Alt ] + [ F4 ] にファンクションキーが割り当てられていない場合で、表示・印刷セットアップでタイトルバーの [ 閉じる ] ボタンが「表示する」に設定されているときに、[ Alt ] + [ F4 ] 押下、タイトルバーの [ × ] ボタンのマウス選択、およびコントロールメニューの「閉じる」メニュー選択をすると、[ Break ] キーと同じ通知コードが AP に返されます。
- ・ [ Alt ] + [ F4 ] にファンクションキーが割り当てられている場合で、表示・印刷セットアップでタイトルバーの [ 閉じる ] ボタンが「表示する」に設定されているときに、[ Alt ] + [ F4 ] 押下すると、割り当てられたファンクションキーに対応する通知コードが AP に返されます。[ × ] ボタンや「閉じる」メニュー選択すると、[ Break ] キーと同じ通知コードが AP に返されます。
- ・ [ Alt ] + [ F4 ] にファンクションキーが割り当てられている場合で、表示・印刷セットアップでタイトルバーの「閉じる」メニューが「表示しない」に設定されているときに、[ Alt ] + [ F4 ] 押下すると、対応する通知コードが AP に返されます。

##### 注 3

XMAP3 では、該当するイベント通知コードなし (空白) で AP に制御が戻ります。

表 4-4 キーの種類と機能 (文字入力キー)

機能名称	キーボード上の標準割り当て	機能	ほかのキーへの割り当て
スペースキー	スペースバー (106 / 109)(PC-9801) (AX)	ボタン、リストボックスなどのオブジェクトでは、選択動作をする。 文字入力時は空白を入力する。	×

##### (凡例)

× : カスタマイズできない。

(106 / 109): 106 / 109 日本語, 101 / 104 英語。

(PC-9801): PC-9801 キーボード。

(AX): AX仕様キーボード。

表 4-5 キーの種類と機能 (フォーカス・カーソル移動キー)

機能名称	キーボード上の標準割り当て	機能	ほかのキーへの割り当て
カーソル移動キー	(106 / 109)(PC-9801) (AX)	カーソルを1コラム移動する。	×
後退キー (バックスペースキー)	BackSpace (106 / 109)(AX) BS (PC-9801)	GUI画面では、前の入力または入出力フィールドにカーソルを移動する。または前のボックスにフォーカスを移動する。	
始点キー	Home (106 / 109)(AX) HOME CLR (PC-9801)	GUI画面では、フィールド始点にカーソルを移動する。フィールドボックス内では、画面始点にカーソルを移動する。	×
復改キー	Enter_復改マーク (106 / 109) 復改マーク (PC-9801) Enter (AX) (文字キー部)	GUI画面では、フィールド始点にカーソルを移動する。フィールドボックス内では、次行以降の最初の入力または入出力フィールドにカーソルを移動する。改行入力属性を指定したフィールドの場合、改行コードを挿入する。	
タブキー	Tab (106 / 109)(AX) TAB (PC-9801)	次の入力テキストまたは入出力フィールドにカーソルを移動する。または次のボックスにフォーカスを移動する。	
バックタブキー	Shift + Tab (106 / 109)(AX) SHIFT + TAB (PC-9801)	前の入力テキストまたは入出力フィールドにカーソルを移動する。または前のボックスにフォーカスを移動する。	×
マイナスタブキー	Ctrl + Tab (106 / 109)(AX) CTRL + TAB (PC-9801)	遷移条件で「手動スキップ・右寄せ」を指定しているフィールドでだけ、右寄せして負(-)記号を付ける。	
スクロールキー	Ctrl + (106 / 109)(AX) CTRL + (PC-9801)	行、列単位に、画面のスクロールをする。	×

(凡例)

:ほかのキーへの割り当てができる。表示・印刷セットアップで設定する。

×:カスタマイズできない。

(106 / 109): 106 / 109 日本語, 101 / 104 英語。

(PC-9801): PC-9801 キーボード。

#### 4. マウスとキー操作

(AX): AX 仕様キーボード。


注※ 復改マーク : 

表 4-6 キーの種類と機能 (文字編集キー)

機能名称	キーボード上の標準割り当て	機能	ほかのキーへの割り当て
挿入キー	Insert (106 / 109)(AX) INS (PC-9801)	文字を挿入できるように半角 1 文字分を空ける。	
削除キー	Delete (106 / 109)(AX) DEL (PC-9801)	入力フィールドまたは入力テキストボックスの 1 文字を消去する。	×
フィールドキー	End (106 / 109)(AX) HELP (PC-9801)	カーソルがある入力フィールドまたは入力テキストボックス内の文字を消去する。	
データキー	Ctrl + End (106 / 109)(AX) CTRL+HELP (PC-9801)	カーソルの位置に関係なく、画面上のすべての入力または入出力フィールドの文字を消去する。	
コピー操作キー	Ctrl+Insert Ctrl+C (106 / 109)(AX) CTRL+INSERT CTRL+C (PC-9801)	マウスで選択した文字列をクリップボードにコピーする。	
ペースト操作キー	Shift+Insert Ctrl+V (106 / 109)(AX) SHIFT+INSERT CTRL+V (PC-9801)	クリップボードからフィールドまたはテキストに文字列を貼り付ける。	

(凡例)

: ほかのキーへの割り当てができる。表示・印刷セットアップで設定する。

× : カスタマイズできない。

(106 / 109): 106 / 109 日本語, 101 / 104 英語。

(PC-9801): PC-9801 キーボード。

(AX): AX 仕様キーボード。

表 4-7 キーの種類と機能（選択キー）

機能名称	キーボード上の標準割り当て	機能	ほかのキーへの割り当て
エスケープキー	Esc (106 / 109)(AX) ESC (PC-9801)	プルダウンメニュー、ポップアップリストが表示されているとき、その表示を閉じる。	×
取消キー	Ctrl+R (106 / 109)(AX) CTRL+R (PC-9801)	入力エラーの表示を「キー OK」表示する。 キーボードのロック状態や入力エラー表示の状態を取り消して、キー操作をできる状態にする。	×
スペースキー	スペースバー (106 / 109) (PC-9801) (AX)	ボタン、リストボックスなどのオブジェクトでは、選択動作をする。 文字入力時は空白を入力する。	×

(凡例)

× : カスタマイズできない。

(106 / 109): 106 / 109 日本語, 101 / 104 英語。

(PC-9801): PC-9801 キーボード。

(AX): AX 仕様キーボード。

### 4.1.3 キーボードのカスタマイズ

標準で機能を割り当てられているキー以外のキーにも、機能を割り当てられます。使用するキーと機能の関係を次の表に示します。機能の割り当ては表示・印刷セットアップで行います。詳細についてはマニュアル「XMAP3 開発・実行ガイド」を参照してください。

表 4-8 機能と割り当てることができるキー

割り当てる機能	割り当てることができるキー	標準割り当てキー
送信（実行）キー	文字側の Enter キー 右 Ctrl キー テンキー側の Enter キー	テンキー側の Enter キー
タブキー	文字側の Enter キー テンキー側の Enter キー + , - , * , / Tab キー	Tab キー
バックタブキー	+ , - , * , / Shift + Tab キー	Shift + Tab キー
PF13 キー	PageDown PageUp Shift + F1 キー	Shift + F1 キー

#### 4. マウスとキー操作

割り当てる機能	割り当てることができるキー	標準割り当てキー
PF14 キー	PageDown PageUp Shift + F2 キー	Shift + F2 キー
フィールドキー	+ , - , * , / End キー	End キー



注

右 Ctrl キーを送信（実行）キーとして割り当てた場合でも、右 Ctrl キーとほかのキーを組み合わせで使用したときは、右 Ctrl キーは通常の Ctrl キーとして動作します。



## 4.2 各制御キーの機能

### 4.2.1 キーの機能

画面を制御するキーの、標準で使用できる機能について説明します。説明中の  および  は、カーソルを示します。また、各オブジェクトでキーを押したときのスクロール量は次のようになります。

オブジェクト	キー		
	[ ] [ ]	[ ] [ ]	[ Ctrl ] + [ ]
フィールドボックス	1 列	1 行	1 行 / 1 列
リストボックス	スクロールしない	1 行	1 行 / 1 列
コンボボックス	スクロールしない	1 行	スクロールしない
ポップアップウィンドウ	スクロールしない	スクロールしない	スクロールしない

#### (1) フォーカス・カーソル移動キー

##### (a) カーソル移動キー ([ ] [ ] [ ] [ ])

キーを 1 回押すと、カーソルが 1 文字分移動します。

CUI 画面およびフィールドボックス内でのフィールドでのカーソルの移動には次に示す三つの動作があります。動作は、表示・印刷セットアップで選択します。

表示・印刷セットアップで設定できる動作の項目を次に示します。なお、表示・印刷セットアップの詳細については、マニュアル「XMAP3 開発・実行ガイド」を参照してください。

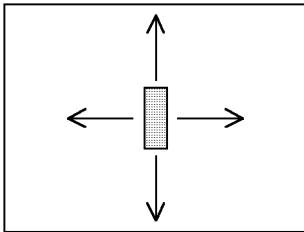
#### 注

フィールドボックス内で上下方向にカーソルを移動する場合、ワンタッチクリアが設定されているフィールドの先頭以外の位置を通過すると、カーソルの列位置が先頭に移動します。

#### 《全ての位置》

- フィールドの内外に関係なく、押したキーの向きにカーソルが移動します。

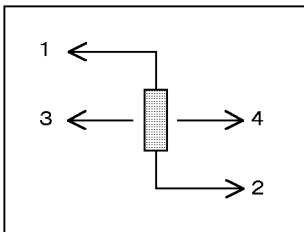
#### 4. マウスとキー操作



- カーソルが画面終点にあるとき [ ] を押すと、カーソルは画面始点に移動します。
- カーソルが画面始点にあるとき [ ] を押すと、カーソルは画面終点に移動します。

##### 《全てのフィールド》

- 押したキーの向きにある、不活性または入力できる状態の入出力フィールドの先頭にカーソルを移動します。



1. [ ] は1文字分上の行から始点に向けて、フィールドをサーチします。
2. [ ] は1文字分下の行から終点に向けて、フィールドをサーチします。
3. [ ] は1文字分左の位置から始点に向けて、フィールドをサーチします。
4. [ ] は1文字分右の位置から終点に向けて、フィールドをサーチします。

- 入出力フィールドがないときは、カーソルは移動しません。

##### 《活性フィールド》

- 押したキーの向きにある、入力できる状態の入出力フィールドの先頭にカーソルを移動します。
- 押したキーの向きについては、《全てのフィールド》の場合と同じです。
- 入力できる状態の入出力フィールドがないときは、カーソルは移動しません。

表 4-9 GUI オブジェクトのボックス内のフィールド間のフォーカス・カーソル移動

移動キー オブジェクト	[ ]	[ ]	[ ]	[ ]
ラジオボタン	前進	前進	後退	後退
チェックボタン	前	前	後	後
プッシュボタン	前	前	後	後
ポップアップ	該当しない <sup>1</sup>	下	該当しない <sup>1</sup>	上
コンボボックス <sup>2</sup>	該当しない	下	該当しない	上
リストボックス	該当しない	下	該当しない	上
スピンボックス	該当しない	ダウンボタンと同じ(値の減少)	該当しない	アップボタンと同じ(値の増加)

## 注 1

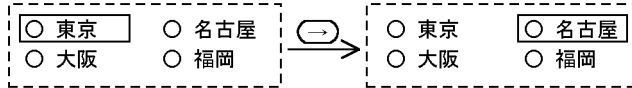
カスケードメニューおよびリストを表示している場合は、下位のメニューおよびリストの表示（前進）/非表示（後退）になります。

## 注 2

キー入力コンボボックスおよび表示・印刷セットアップでカーソルキーでのメニュー表示が設定されているメニュー選択コンボボックスでは、[ ] [ ] でメニューを表示します。

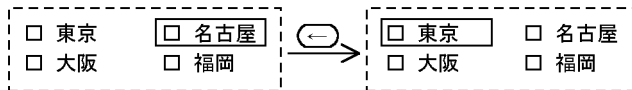
- ・次、または下のボタンにフォーカス（入力権）を移動させる移動キー。

[ ] [ ]



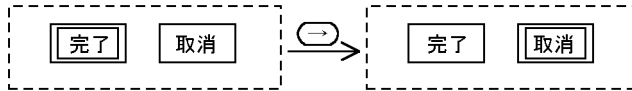
- ・一つ前、または上のボタンにフォーカス（入力権）を移動させる移動キー。

[ ] [ ]



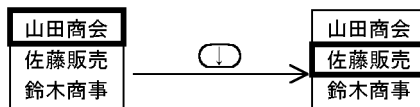
- ・次、または下のボタンにフォーカス（入力権）を移動させる移動キー。

[ ] [ ]



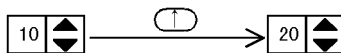
- ・次の項目（リスト）にフォーカス（入力権）を移動させる移動キー。

[ ]



- ・スピンボックスの値をアップダウンさせるキー。

[ ] [ ]



## (b) 後退キー ([ Back Space ])

カーソルが前方に移動します。

表示・印刷セットアップでの [ Back Space ] キーの指定に対応して動作が次のように異なります。

#### 4. マウスとキー操作

- 直前の項目：フィールドの先頭にカーソルを位置づける。  
フィールドの先頭では、前の位置にあるフィールドにフォーカスは移動します。

住所 (□	)
氏名 (↑————□)	年齢 ( )
趣味 (	)

右詰めで表示されている「金額」または「数値」のテキストボックス（またはフィールド）では、右端が先頭位置になります。

一つ前の入力できる状態の入出力テキストボックス（または入出力フィールド）の先頭にカーソルは移動します。

- 1文字削除：直前の文字を削除する。

氏名 (日立△△美花子 )
---------------

↓

氏名 (日立△△花子 )
--------------

- ヌル置換：直前の文字を削除して、ヌル(00)<sub>16</sub>に置き換える。  
直前の文字が全角文字の場合、2個のヌル(00)<sub>16</sub>に置き換えます。

氏名 (日立△△美花子 )
---------------

↓

氏名 (日立△△(00) <sub>16</sub> (00) <sub>16</sub> 花子 )
--

フィールドボックスの改行入力属性を指定したフィールドでは、直前の文字が半角文字の場合に1個の半角スペースに置き換え、直前の文字が全角文字の場合に2個の半角スペースに置き換えます。

#### (c) 始点キー ([ Home ])

カーソルのある入力できる状態の入出力テキストボックスの先頭に、カーソルは移動します。

CUI画面およびフィールドボックスでは、画面またはボックス内の先頭のフィールドに移動します。

リスト形式で表示しているポップアップメニューでは、第1レベルのメニュー項目にフォーカスが移動します。

住所 (□←————□)
--------------

- 入力できる状態の入出力フィールドがないときは、カーソルは画面始点に移動します。

## (d) 復改キー ([ Enter ])

次行の、最初の入力できる状態の入出力フィールドの先頭に、カーソルは移動します。

住所 (		)
氏名 (		)
趣味 (		)

↑

- 入力できる状態の入出力フィールドがないときは、カーソルは移動しません。
- 入力できる状態の入出力フィールドが複数行にわたっているときは、次の行の先頭に移動します。改行入力属性を指定した場合は、カーソルの左側に改行コードを挿入します。

1	2	3	4	5	6	7	8
9	0	1	2	↓			
3	4	5					

↑

- 入力できる状態の入出力テキストボックスでは、カーソルはそのテキストの先頭に移動します。

## (e) タブキー ([ Tab ])

ボックス中のカーソル移動

次の入力できる状態の入出力テキストボックス（または入出力フィールド）の先頭に、カーソルが移動します。

住所 (		)
氏名 (		)
趣味 (		)

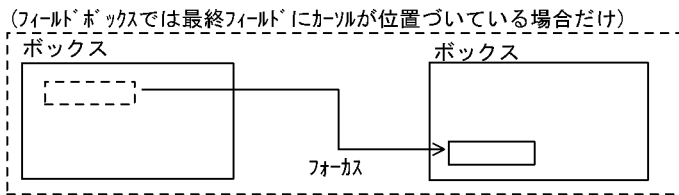
↑

- 入力できる状態の入出力テキストボックス（または入出力フィールド）がないときは、カーソルは移動しません。
- 画面終点までに、ほかに入力できる状態の入出力テキストボックス（または入出力フィールド）がないときは、カーソルは画面始点に最も近い入力できる状態の入出力テキストボックス（または入出力フィールド）に移動します。

#### 4. マウスとキー操作

##### ボックス間のフォーカス（入力権）移動

次のボックスにフォーカスを移動します。



フォーカスが移動するボックスは、次のとおりです。

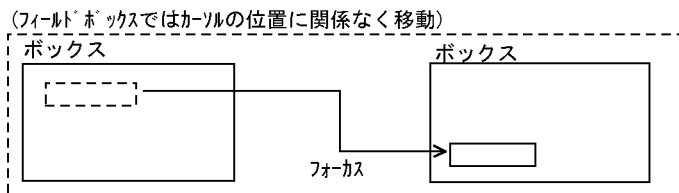
- フィールドボックス
- 入出力テキストボックス
- ポップアップテキスト
- コンボボックス
- スピンボックス
- リストボックス
- ボタンボックス（ラジオボタン、チェックボタン、プッシュボタン）です。

##### (f) シフト+復改 ([ Shift ] + [ Enter ])

次のボックスにフォーカスを移動します。

送信動作は行いません。

この動作は、XMAP3の表示・印刷サーバ環境定義ファイルでの動作を「復改」から「送信」に変更した場合でも有効です。



##### (g) バックタブキー ([ Shift ] + [ Tab ])

一つ前の入力できる状態の入出力テキストボックス（または入出力フィールド）の先頭への移動

- 入力できる状態の入出力テキストボックス（または入出力フィールド）がないときは、カーソルは移動しません。

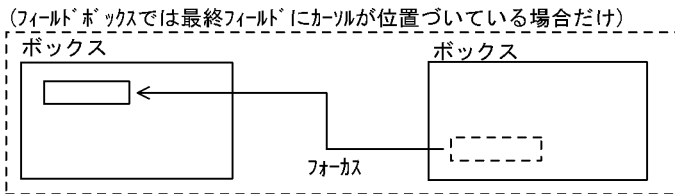
- カーソルが入力できる状態の入出力テキストボックス（または入出力フィールド）の途中にある場合は、カーソルのある入力できる状態の入出力テキストボックス（または入出力フィールド）の先頭に戻ります。

住所 ( █ ←————— █ )
氏名 (                    ) 年齢 (                    )
趣味 (                    )

- 画面始点までにほかに入力できる状態の入出力テキストボックス（または入出力フィールド）がないときは、カーソルは画面終点に最も近い、入力できる状態の入出力テキストボックス（または入出力フィールド）の先頭に移動します。

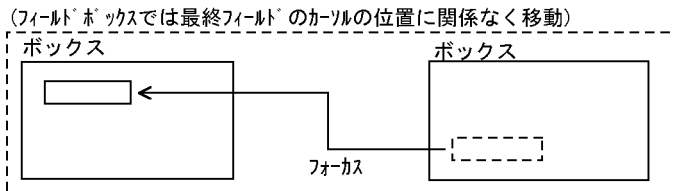
住所 ( █ )
氏名 ( ↓                    ) 年齢 (                    )
趣味 ( █ )

#### 一つ前のボックスへのフォーカス移動



#### (h) コントロール+復改 ([ Ctrl ] + [ Enter ])

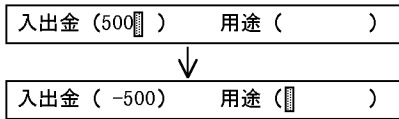
一つ前のボックスにフォーカスを移動します。送信動作は行いません。この動作は、XMAP3 の表示・印刷サーバ環境定義ファイルでの動作を「復改」から「送信」に変更した場合でも有効です。



#### 4. マウスとキー操作

##### マイナスタブキー ([Ctrl] + [Tab])

属性が「右寄せ」に設定されている入出力テキストボックス（または入出力フィールド）の数字を右寄せして、負（-）記号を付けます。カーソルは次の入力できる状態の入出力テキストボックス（または入出力フィールド）の先頭に移動します。



- フィールドに負（-）記号を付ける空きがないときは、入力エラーとなります。
- 入力できないテキストボックス（またはフィールド）にカーソルがあるときは、次の入力できる状態の入出力テキストボックス（または入出力フィールド）の先頭にカーソルを移動します。
- 右寄せの属性以外では、右寄せしたり負記号を付けたりしないで、次の入力できる状態の入出力テキストボックス（または入出力フィールド）の先頭にカーソルを移動します。
- フィールドにデータがない場合、負（-）記号を付け、カーソルは次の入力できる状態の入出力テキストボックス（または入出力フィールド）の先頭にカーソルを移動します。

#### (2) 画面の移動に使用するキー

##### (a) スクロールキー ([Ctrl] + [ ] [ ] [ ] [ ])

画面上に表示されていないプレーン領域をスクロールし、見えるようにできます。

対象となるオブジェクトを次の表に示します。

表 4-10 画面の移動に使用するキー

オブジェクト		[Ctrl] + [ ]	[Ctrl] + [ ]	[Ctrl] + [ ]	[Ctrl] + [ ]
マウス選択 オブジェクト	フィールド ボックス	下方向にスク ロール	上方向にスク ロール	右方向にスク ロール	左方向にスク ロール
	リスト ボックス	最下行にスク ロール	最上行にスク ロール	最右行にスク ロール	最左行にスク ロール
文字入力オブジェクト		無効			

#### 注

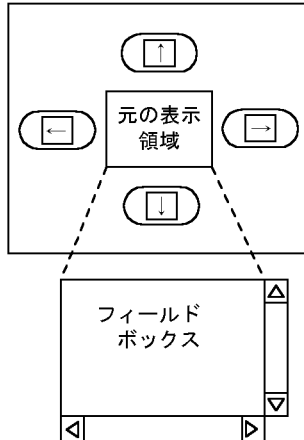
ポップアップで表示されるリストでは、[Ctrl] + [ ] または [Ctrl] + [ ] のときは、水平方向に半角 1 文字分スクロールします。



### フィールドボックス

押したキーの向きに、指定した行、列分画面の表示が移動します。

フィールドボックスのプレーン領域



### リストボックス

押したキーの向きに、プレーンの端までリストボックスが移動します。

## (3) 文字編集キー

文字入力できるオブジェクトにだけ使用できます。

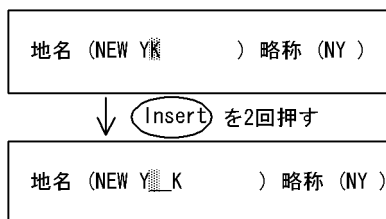
### (a) 挿入キー ([ Insert ])

文字入力できるオブジェクトにだけ使用できます。

カーソル以降の文字が右にずれ、カーソル位置から押した数だけ「空き」ができます。

空きに文字を入力してください。

なお、表示・印刷セットアップの「挿入/上書きモードのデフォルト設定」で「挿入モード」を指定している場合、挿入キーを使用しなくても、入力した文字が自動的に挿入されます。



(凡例) \_ : ヌル ( (00)<sub>16</sub> )

- 入力できない状態のテキストボックス (またはフィールド) には、文字の挿入はでき

#### 4. マウスとキー操作

ません。

- 入力・表示したデータを AP へ返す状態になります。
- 日本語（漢字）フィールドは [ Insert ] を 1 回押すと 2 カラムの「空き」ができます。
- 入出力テキスト・フィールドで、使用目的が「数値」のとき（ただし、詳細目的が「数字・数字記号」の場合は除く）、[ Insert ] を押すとカーソル位置に 0（ゼロ）が入力されます。
- 入出力テキスト・フィールドで、使用目的が「数値」、詳細目的が「数字・数字記号」のとき、[ Insert ] を押すとカーソル位置にヌル（ $(00)_{16}$ ）が入力されます。

地名（東京 名麿 ）略称（T N）

↓ (Insert) を1回押す

地名（東京 名屋 ）略称（T N）

（凡例） \_ : ヌル（ $(00)_{16}$ ）

- フィールドボックスの改行入力属性を指定したフィールドでは、半角スペースに置き換えます。

《挿入キー仕様を「NULL まで」とした場合》表示・印刷セットアップで変更した場合  
[ Insert ] で空きを作るとフィールドから文字があふれてしまう場合は、挿入できません。

地名（ NW YORK ）

↓  
地名（ N YORK ）

地名（ TOKYOYOKOHAMA ）

↓  
入力エラー

《挿入キー仕様を「フィールドの終わりまで」とした場合》標準仕様の場合

[ Insert ] で空きを作るとフィールドから文字があふれてしまう場合は、あふれた文字は消去されます。

地名（ NW YORK ）

↓  
地名（ N YORK ）

地名（ TOKYOYOKOHAMA ）

↓  
地名（ TOKYOYOKOHAM ）

Aが消える

## (b) 削除キー ([ Delete ])

カーソル位置の1文字が消え、カーソル以降の文字列が1文字分左に詰まります。全角文字のときは、2カラム分左に詰まります。



(凡例) \_ : ヌル (00)<sub>16</sub>

- [ Delete ] を押してもカーソルは移動しません。
- 文字を消去してできた領域には、ヌル (00)<sub>16</sub> が入ります。
- 入力できない状態の入出力テキストボックス (または入出力フィールド) の文字は、削除キーで消去できません。
- 入力・表示したデータを AP へ返す状態 (転送状態) になります。
- ヌル (00)<sub>16</sub> は AP へは返されません。詰めた型となります。

埋字指定があれば桁寄せ向きに従って桁寄せされ、埋字が入られます。

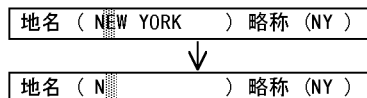
## 注

カーソル以降の文字を削除する場合は、フィールドキーを使用します。オプションでカーソルのあるフィールド全体を消去することもできます。

## (c) フィールドキー ([ End ])

文字が入力できるオブジェクト

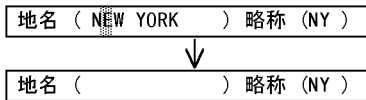
- カーソル以降の消去 (標準仕様)  
入力できる状態の入出力テキストボックス (または入出力フィールド) 内で、カーソル位置からフィールドの終わりまでの文字を消去します。



- 入力できない状態のテキストボックス (またはフィールド) の文字は、[ End ] で消去できません。
- 入力・表示したデータを AP へ返す状態 (転送状態) になります。

#### 4. マウスとキー操作

カーソルのあるフィールド全体の消去《表示・印刷セットアップのオプション》



マウス選択できるオブジェクト

オブジェクト	選択済	未選択	動作
ラジオボタン	オフ	オフ	選択状態を解除
チェックボタン			
プッシュボタン	-	-	-
単一リストボックス	オフ	オフ	選択状態を解除
複数リストボックス			
スピンボックス	最小値に戻す	最小値に戻す	選択済み状態（AP に値を返す）
ポップアップメニュー 1	クリア	-	選択状態を解除
コンボボックス 2	メニュー先頭の値	メニュー先頭の値	選択状態を解除
トグルフィールド	オフ	オフ	選択状態を解除
メニューバー (プルダウン、カスケード)	-	-	-

(凡例)

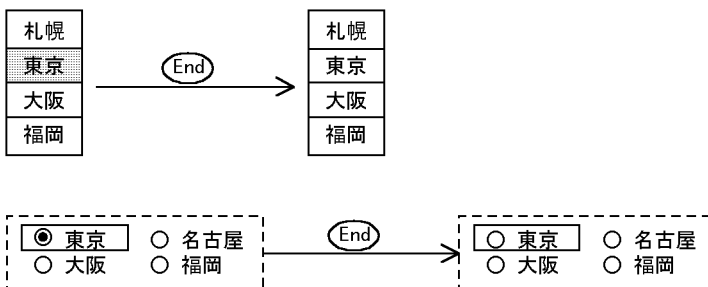
- : 該当しない。

注 1

ポップアップメニューが表示されているときは、入出力テキストボックスが全桁クリアされません。このとき、表示・印刷セットアップで「カーソル以降をクリア」を設定していても無視されます。

注 2

キー入力コンボボックスは、入出力テキストボックスと同じ動作になります。



## (d) データキー ([ Ctrl ] + [ End ])

## 文字入力できるオブジェクト

画面上の入力できる状態の入出力テキストボックス（または入出力フィールド）の文字を、カーソルの位置にかかわらずすべて消去し、始点に最も近い入出力テキストボックス（または入出力フィールド）の先頭に、カーソルを移動します。

住所（東京都 新宿区	）
氏名（山田 太郎	） 年齢（
趣味（	）



住所（	）
氏名（	） 年齢（
趣味（	）

- 入力できるフィールド（フィールドボックス内のフィールド，または入出力テキストボックス）内の文字を消去します（すべての入力できるフィールドを，再入力したデータだけ AP に返す状態にします）。
- ボタンボックスのすべての有効ボタンをオフ状態にします。

## マウス選択できるオブジェクト

オブジェクト	選択済	未選択	動作
ラジオボタン	オフ	オフ	すべての選択状態を解除
チェックボタン			
プッシュボタン	-	-	-
単一リストボックス	オフ	オフ	すべての選択状態を解除
複数リストボックス			
スピンボックス	最小値に戻す	最小値に戻す	すべての選択状態を解除 <sup>1</sup>
ポップアップメニュー <sub>2</sub>	クリア	-	すべての選択状態を解除
コンボボックス <sup>3</sup>	メニュー先頭の値	メニュー先頭の値	すべての選択状態を解除
トグルフィールド	オフ	オフ	すべての選択状態を解除
メニューバー (プルダウン, カスケード)	-	-	-

## (凡例)

- : 該当しない。

## 注 1

表示・印刷セットアップでのデータキーの動作設定に従います。  
未入力状態：AP に値を返さない。

#### 4. マウスとキー操作

ヌル (00)<sub>16</sub> クリア状態：AP に値を返す。

注 2

ポップアップメニューが表示されているときは、入出力テキストボックスが全桁クリアされま  
す。

注 3

キー入力コンボボックスは、入出力テキストボックスと同じ動作になります。

(e) コピー・ペースト操作キー ([ Ctrl ] + [ Insert ] / [ Shift ] + [ Insert ] または [ Ctrl ]  
+ [ C ] / [ Ctrl ] + [ V ])

- 選択されたテキストをクリップボードにコピーします。
- クリップボードのテキストをフォーカスのあるテキストボックスにペーストします。

(4) エスケープキー ([ Esc ])

(a) 文字入力できるオブジェクト

文字入力できるオブジェクトに対しては、無効です。

(b) マウス選択対象となるオブジェクト

オブジェクト	選択済	未選択	動作
ラジオボタン	-	-	-
チェックボタン	-	-	-
プッシュボタン	-	-	-
単一リストボックス	-	-	-
複数リストボックス	-	-	-
スピンボックス	-	-	-
ポップアップメニュー	メニュークローズ	-	メニューのクローズ
コンボボックス	-	-	メニューのクローズ
トグルフィールド	-	-	-
メニューバー (プルダウン, カスケード)	-	-	-

(凡例)

- : 該当しない。

コンボボックスまたはポップアップテキスト・フィールドでのエスケープキー  
コンボボックスまたはポップアップテキスト・フィールドのメニュー表示を消去しま  
す。メニュー消去後の動作は、「カーソルが次のフィールドへ移動する」または  
「フォーカスが次のボックスに移動する」のどちらかになります。

(5) スペースキー (スペースバー)

(a) 文字入力テキスト・フィールドの場合

文字入力テキストおよびフィールドでは、スペース文字として扱われます。

## (b) マウス選択オブジェクト

オブジェクト	選択済	未選択	動作
ラジオボタン	オンのまま	オン	項目選択
チェックボタン	オフ	オン	オン / オフの切り替え
プッシュボタン	-	オン	画面選択
単一リストボックス	オフ	オン	項目選択
複数リストボックス			
スピンボックス	-	-	-
ポップアップメニュー	オフ <sup>1</sup>	オン	項目選択
コンボボックス <sup>2</sup>	オフ	オン	項目選択
トグルフィールド	オフ	オン	オン / オフの切り替え
メニューバー (プルダウン, カスケード)	-	オン	画面確定

(凡例)

- : 該当しない。

注 1

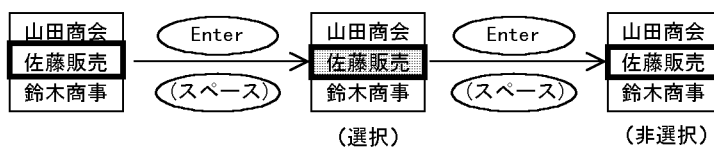
ポップアップで、階層構造のメニュー項目を表示している場合は、下位レベルのメニュー項目(カスケードメニューや第2～第8レベルのリスト)の表示になります。

注 2

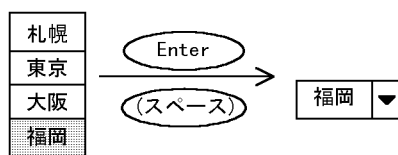
キー入力コンボボックスは、入出力テキストボックスと同じ動作になります。

スペースまたは復改キー ([スペース], [Enter])

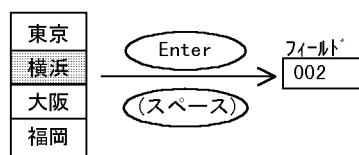
フォーカス(入力権)のある項目(リスト)に対応した値が選択, または非選択となります。



コンボのとき



ポップアップのとき



注

属性の指定によって, リストを選択したあとに, 次を示すどちらかの動作を指定できます。

1. フォーカスが次の項目(リスト)へ移動する

#### 4. マウスとキー操作

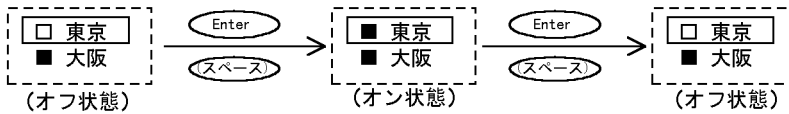
複数選択で、「選択後次の項目にフォーカスを移す」指定をした場合。

2. フォーカスが次のボックスに移動する

単一選択で、「選択後次のボックスにフォーカスを移す」指定をした場合。

スペース, または復改キー ([ スペース ], [ Enter ])

フォーカス (入力権) のあるボタンの状態が, 交互 (オン / オフ) に変わります。



#### (6) 画面確定キー

画面が送信できる状態で, キーが押されると送信要求の発生を AP に通知します。

なお, OLTP サーバ構成の環境で, OpenTP1 上で「次メッセージ要求キー」として割り当てられているキーのイベントは AP に渡りません。

(a) 送信 (実行) キー (テンキー側の [ Enter ])

送信できる状態であるかどうかのチェック (ウィンドウ内のフィールドボックス, 入力テキストボックス, リストボックス, ラジオボタンボックスについて) をして, 送信のイベント通知コードを AP に返します。

(b) PF1 ~ PF12 キー

([ F1 ] ~ [ F12 ])

(c) PF13 ~ PF24 キー

([ Shift ] + [ F1 ] ~ [ F12 ])

(d) PF25 ~ PF36 キー

([ Ctrl ] + [ F1 ] ~ [ F12 ])

(e) PF37 ~ PF48 キー

([ Ctrl ] + [ Shift ] + [ F1 ] ~ [ F12 ])

(f) PF49 ~ PF60 キー

([ Alt ] + [ F1 ] ~ [ F12 ])

(g) PF61 ~ PF72 キー

([ Alt ] + [ Shift ] + [ F1 ] ~ [ F12 ])



(h) PF73 ~ PF84 キー

( [ Ctrl ] + [ Alt ] + [ F1 ] ~ [ F12 ] )

(i) PA1 ~ PA3 キー

( [ PageUp ] , [ PageDown ] , [ Ctrl + PageUp ] )

(j) 割込キー

( [ Break ] , [ Ctrl ] + [ Pause ] )

(k) スクリーン消去キー

( [ Ctrl ] + [ Delete ] )

XMAP3 で入力エラーとなるキー

オンライン端末で使用するキーです。XMAP3 では押しても無効です。

- SRQ キー ( [ Ctrl ] + [ Q ] )
- テストキー ( [ Ctrl ] + [ T ] )

使用できないキー

Windows 95 , Windows 98 , Windows Me では , [ Alt ] + [ F6 ] キーまたは [ Alt ] + [ Shift ] + [ F6 ] キーは Windows の画面重なり制御に割り当てられているため , 使用できません。

(7) 取消キー ( [ Ctrl ] + [ R ] )

- オペレータインジケータの入力エラー表示を解除します。オペレータインジケータが表示するメッセージについてはマニュアル「XMAP3 開発・実行ガイド」を参照してください。
- キーボードのロック状態を解除します。

## 4.2.2 キーでのデータ入力と編集

(1) 一般規則

キーでデータを入力したり , 入力したデータを編集する場合の , 一般的な動きを次に示します。

(a) 文字を入力する場合

キーでの文字入力は左詰めです。

A

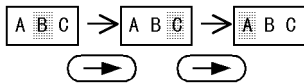
↓ [ B ] [ C ] をキー入力

ABC

#### 4. マウスとキー操作

##### (b) カーソルを移動する場合

カーソルはカーソル移動キーを押すと、カーソル移動キーの方向に一文字ずつ移動します。フィールドの最後までカーソルが移動したあとで、さらにカーソル移動キーを押すと、フィールドの先頭にカーソルが戻ります。



##### (c) オブジェクトの属性によってフォーカス・カーソル移動が変わる場合

ポップアップメニューは、使用目的が自動スキップか手動スキップかでフォーカス・カーソルの移動が変わります。

使用目的が「自動スキップ」になっている場合は、フォーカスが位置づくとき自動的にポップアップメニューが表示されます。ポップアップのメニュー項目をマウスまたはキーで選択すると、次の項目にフォーカスが移ります。

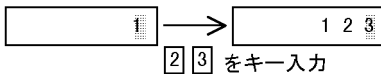
使用目的が「手動スキップ」になっている場合は、ポップアップのメニュー項目を選択しても、フォーカスは次の項目へ移りません。「手動スキップ」のときは、[ Tab ] キーで次の項目へ移動させます。

## (2) 数値入力

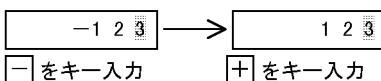
### (a) データ入力

数値データは右詰めで入力されます。

#### ●データの入力

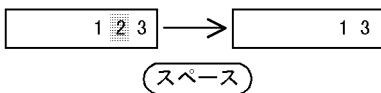


#### ●記号の入力



### (b) データ編集

データを削除する場合は、削除したい数字にカーソルを位置づけ、[ スペース ] を押しします。



使用目的が「数値」、詳細目的が「数字・数字記号」の入出力テキスト・フィールドで

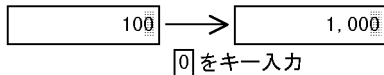
は、ヌル ( 0x00 ) が入力されます。

### (3) 金額入力

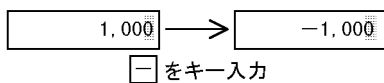
#### (a) データ入力

金額データは右詰めめで入力され、3桁目ごとに「,」が打たれます。

##### ●データの入力

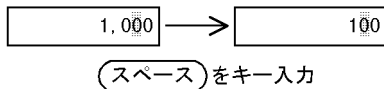


##### ●記号の入力



#### (b) データ編集

データを削除する場合は、削除したい数字にカーソルを位置づけ、[スペース]を押します。編集前に打たれた「,」の位置は自動的に変更されます。



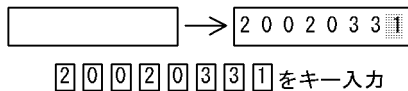
### (4) 日付入力

ここでは西暦の日付を入力する例を示します。和暦の日付の入力、および時刻の入力も同様です。

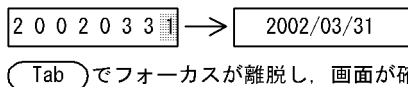
#### (a) データ入力

表示したい年月日を入力します。[Tab]などでフィールドからフォーカスが離脱し画面が確定すると、ドロワー時に設定した形式で表示されます。

##### ●データの入力



##### ●画面の確定



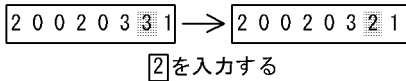
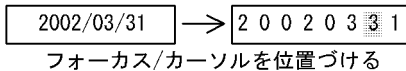
#### (b) データ編集

一度確定した日付のフィールドにフォーカス・カーソルを位置づけると、データを編集

#### 4. マウスとキー操作

できる状態になります。

##### ●編集状態に戻す



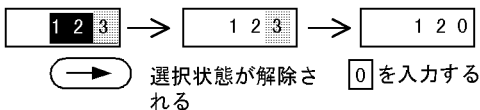
#### (5) ワンタッチクリア

全体を選択状態にしておくことで、データの一括書き換えができます。

##### (a) ワンタッチクリア

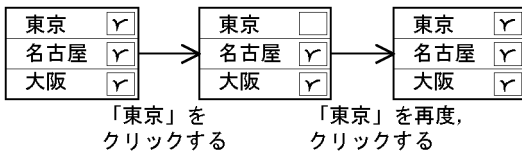


##### (b) ワンタッチクリアを解除して、一部をクリア



#### (6) トグルフィールド

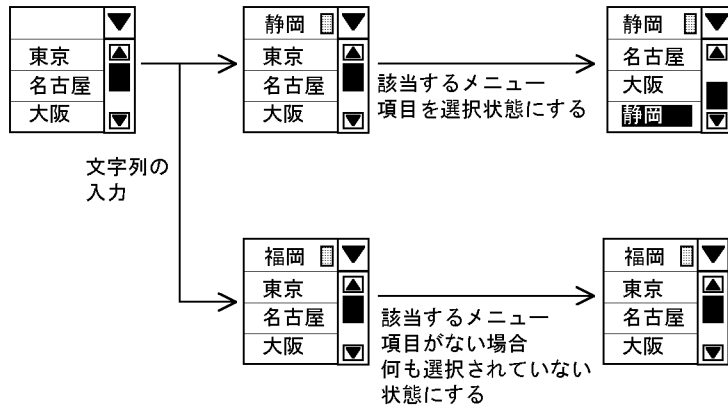
トグルフィールドの「オン/オフ」の切り替えをマウスでします。



#### (7) キー入力コンボボックス

メニューが開いている状態で入出力ボックスに文字列を入力すると、入力した文字列と同じ通知コードを持つメニュー項目がある場合は、そのメニュー項目が見えるようにスクロールし、そのメニュー項目を選択状態にします。

同じ通知コードを持つメニュー項目がない場合は、どのメニュー項目も選択されていない状態にします。

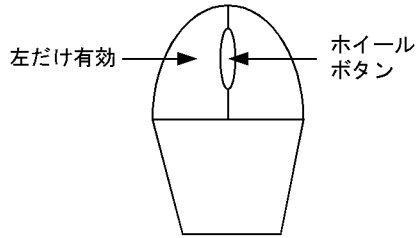


## 4.3 マウス操作

---





### 4.3.1 有効となるボタン

マウスのボタンは、左側だけを使用します。なお、このマニュアルでマウスを指すときは、右手用のマウスを想定しています。また、ホイールマウスのホイールボタンを利用できます。



## 4.3.2 シングルクリックとダブルクリック

画面上のオブジェクトとマウスでのクリックによる動作

画面	クリックする対象	クリックと動作など
GUI/CUI/ フィールドボ ックス共通	タイトルバーのリサイズ 	表示サイズを変える。
	タイトルバーのアイコン化 	アイコンにする。
	タイトルバーのクローズ 	クリックすると割込キーと 同じイベントを AP に通知 する (オプション)。
	コントロールメニューの表示	システムメニューを表示す る。
GUI	ポップアップのメニュー コンボのメニュー リストのメニュー 	メニュー項目のオンとオフ を切り替える (単一リストボックスでリ ストのメニューをダブルク リックすると、送信キーを 押したときと同じイベント を AP に通知する)。
	ラジオ/チェックボタン	ボタンのオンとオフを切り 替える。
	入出力テキスト	カーソルを位置づける。
	日付・時刻テキスト (入出力)	ドラッグして全選択にす る。
	プッシュボタン タイトルバーのメニュー コンボのボタン	押す (オン)。
	スピンボックス	数字データの増減をする。
フィールドボ ックス	入出力フィールド	カーソルを位置づける。
	日付・時刻フィールド (入出力)	
	トグルフィールド	トグルフィールドのオン とオフを切り替える。
	ポップアップのメニュー	メニュー項目のオンとオフ を切り替える。
CUI 画面	入出力フィールド	カーソルを位置づける。

注 ユーザ定義ウィンドウアイコンを使用している場合には、指定したウィンドウアイコンが表示  
されます。ユーザ定義ウィンドウアイコンについては、マニュアル「XMAP3 開発・実行ガイド」  
を参照してください。

## 4.3.3 マウスでのクリック

### (1) ボックス間のフォーカス移動

フォーカス (入力権) のあるボタンをクリックすると、ボタンの状態が交互 (オン/オ

#### 4. マウスとキー操作

フ) に変わります。

フォーカス（入力権）のないボタンをクリックすると、クリックしたボタンにフォーカスが移動し、ボタンの状態がオンまたはオフに変わります。

マウスでボックスの上をクリックすると、クリックしたボックスにフォーカスが移動します。

#### (2) リストボックス内のフォーカス移動

マウスの左ボタンをクリックすると、フォーカス（入力権）の有無に関係なくクリックされた項目（リスト）に割り当てられた値をフィールドに展開します。

フィールドにデータを展開すると、タブキーを押したときと同じ動作をします。なお、このとき属性で指定した「選択必須（選択しないときはエラーにする）」、および「自動送信（選択直後 AP へ制御を渡す）」が有効になります。

リストを選択したあとの動作を、属性の設定によって指定できます。指定できる動作は次の二つです。

1. フォーカスが次の項目（リスト）へ移動する  
複数選択で、「自動スキップ（選択したら次のフィールドにフォーカスを移す）」指定をした場合。
2. フォーカスが次のボックスに移動する  
単一選択で、「自動スキップ（選択したら次のボックスにフォーカスを移す）」指定をした場合。

#### (3) コンボボックスのフォーカス移動

マウスの左ボタンをクリックすると、フォーカスの有無に関係なくクリックされたメニューに対応する値をコンボボックスに展開します。

注

コンボボックスに値を展開すると、タブキーを押したときと同じ動作をします。なお、このとき属性で指定した「選択済み（選択しなくても通知コードを AP に返す）」、および「自動送信（選択直後に AP へ制御を渡す）」が有効になります。



#### (4) 各オブジェクトでのスクロール量

各オブジェクトでマウスをクリックしたりドラッグしたときのスクロール量を次に示します。

オブジェクト	マウス				
	スクロールバーのバー	スクロールバーのバー以外の部分		スクロールバーのボタン部分	
	縦または横	縦	横	縦	横
フィールドボックス	ドラッグ分	1 ページ	1 ページ	1 行	1 列
リスト	ドラッグ分	1 ページ	1 ページ	1 行	1 列
コンボ	ドラッグ分	1 ページ	スクロールしない	1 行	スクロールしない
ポップアップ	ドラッグ分	1 ページ	スクロールしない	1 行	スクロールしない
ウィンドウ	ドラッグ分	1 ページ	1 ページ	1 行	1 列

#### 注

メニューをリスト表示しているポップアップでは、次のようになります。

スクロールバーのバー以外の部分：1 ページ



スクロールバーのボタン部分：半角 1 文字分

### 4.3.4 ホイールマウスでの操作

ホイールマウスでは、スクロール、コンボボックスメニュー選択、およびスピンボックス値を設定できます。動作および対象オブジェクトについて説明します。

#### (1) スクロール

##### (a) 動作

- 垂直スクロールバーがあるオブジェクトに対してホイールマウスを奥に回転させると上に、手前に回転させると下にスクロールします。
- ホイールマウスの 1 メモリ回転分が、スクロールバーの「」、「」ボタンの 1 回の押下に対応します。なお、フィールドボックスでは、このときのスクロール量は、表示・印刷セットアップで設定できます。
- 垂直スクロールバーが表示されない状態では、ホイールマウスによるスクロールはできません。
- 水平スクロールには対応しません。

##### (b) 対象オブジェクト

ホイールマウスのスクロールの対象オブジェクトは、次のとおりです。

- フィールドボックス

## 4. マウスとキー操作

- リストボックス
- コンボボックスのメニュー部分
- キー入力コンボボックスのメニュー部分
- ポップアップテキストのメニュー部分（標準，リスト - ポップアップ型，リスト - ダイアログ型）
- ポップアップフィールドのメニュー部分
- 出力グラフィック
- GUI 画面
- CUI 画面

### 注

スクロールさせる場合は，フォーカスを持つオブジェクトがない画面で操作するか，ボタン系オブジェクトにフォーカスを置いて操作する必要があります。

その際，出力グラフィックをスクロールさせる場合は出力グラフィックを，GUI 画面をスクロールさせる場合は GUI 画面のオブジェクトがない部分をクリックすると，それぞれをスクロール対象に変更できます。

## (2) コンボボックスメニュー選択

### (a) 動作

- メニューの閉じた状態のコンボボックスに対して，ホイールマウスを奥に回転させると上の，手前に回転させると下のメニューを選択します。
- ホイールマウスの 1 メモリ回転分が，メニュー一つの移動に相当します。
- ホイールマウスの回転操作では，自動送信属性の有無にかかわらず，自動送信されません。フォーカス遷移時は，自動送信属性が有効になります。
- ホイールマウスの回転操作では，自動スキップ属性の有無にかかわらず，自動スキップしません。

### (b) 対象オブジェクト

コンボボックスメニュー選択の対象オブジェクトは，次のとおりです。

- 固定コンボボックス
- 可変コンボボックス
- キー入力固定コンボボックス
- キー入力可変コンボボックス

## (3) スピンボックス値設定

### (a) 動作

- スピンボックスに対して，ホイールマウスを奥に回転させると増分値加算，手前に回転させると増分値減算をします。
- ホイールマウスの 1 メモリ回転分が，増分値で設定した値分の変更に相当します。
- マウスホイールの回転操作では，自動送信属性の有無にかかわらず，自動送信されません。フォーカス遷移時は自動送信属性が有効になります。
- マウスホイールの回転操作では，自動スキップ属性の有無にかかわらず，自動スキップ

プされません。

(b) 対象オブジェクト

スピンボックス

(4) 注意事項

マウスによってはコントロールパネルなどで、ホイールマウス操作時の動作をカスタマイズできるものがあります。ホイールマウス操作時の動作をカスタマイズした場合は、カスタマイズした内容に依存した動作となり、通常の動作をするとは限りません。

## 4.4 コピー & ペースト操作

XMAP3 では、コピー（複写）&ペースト（貼り付け）ができます。

コピー & ペーストには、XMAP3 の表示サービスが同じである画面間のできる場合と、表示サービスが異なっている画面間（XMAP3 以外の画面との間も含む）でもできる場合があります。GUI 機能の画面のコピー & ペースト機能の範囲を次の表に示します。

表 4-11 コピー & ペースト機能範囲

画面	コピー & ペースト範囲	
	表示サービスが同じ画面間	表示サービスが異なる画面間
GUI 機能	C/S システム構成でのコピー & ペーストには、XMAP3 が持っているクリップボードを使用する方法と、OS が持っているクリップボードを使用する方法がある。 ただし、スタンドアロン構成でのコピー & ペーストは、OS が持つクリップボードを使用する方法だけとなる。	OS が持っているクリップボードを使用するコピー & ペースト。

以降の項で、XMAP3 でのコピーの対象、ペーストの対象、およびクリップボードを使用したコピー & ペースト操作について説明します。

### 4.4.1 コピーの対象

XMAP3 では次に示すデータをコピーします。

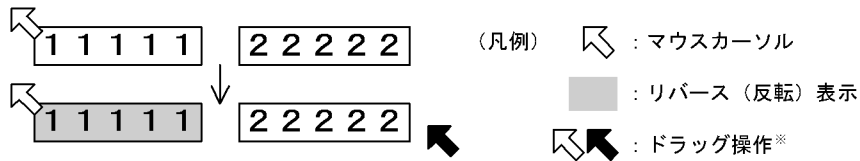
- フィールドボックスの 1 フィールド
- フィールドボックスの複数フィールド
- 入出力テキスト
- リストボックスの 1 リスト（項目）

コピーするデータは必ず表示されていなければなりません。CUI 画面で、データがない部分に対してコピーの操作をした場合は、オペレータインジケータに「X コピー」が表示されます。

コピーの対象となるデータについて (1) ~ (5) で説明します。

#### (1) フィールドボックスの 1 フィールド

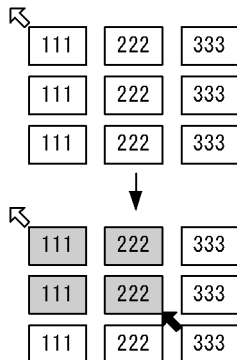
文字データだけがコピーの対象となります。属性（拡大、色など）はコピーの対象にはなりません。なお、表示・印刷セットアップの「キー操作 2」タブの「コピー & ペーストモード」で「単一フィールド」が選択されている場合、複数のフィールドに対しマウスでドラッグ操作をした場合には、一つのフィールドだけが反転表示されます。



注※ コピーしたい文字データの先頭にマウスカーソルを置き (マウスカーソル), マウスの左ボタンを押したままコピーしたい文字データの末尾までマウスカーソルを移動したあと (ドラッグ操作\*), マウスの左ボタンを離す動作

## (2) フィールドボックスの複数フィールド

表示・印刷セットアップの「キー操作 2」タブの「コピー & ペーストモード」で「複数フィールド」が選択されている場合、文字だけがコピーの対象となり、属性（拡大、色など）はコピーの対象なりません。表形式で表示されている複数のフィールドに対してマウスをドラッグさせると、ドラッグした矩形の領域内にあるフィールドがコピーの対象となるため、反転表示されます。



これらのフィールドをコピーすると、フィールド間に区切り文字が付けられた形式で、クリップボードに複写されます。単一行内のフィールド間の区切り文字は、タブ (09)<sub>16</sub> になります。また、コピー範囲が複数行にわたる場合は、行間の区切り文字は改行コード CRLF(0D0A)<sub>16</sub> になります。この形式は、表計算ソフトなどで矩形領域を選択してコピーをするときの、一般的な形式と同じです。

なお、この機能を利用できるのは、OS のクリップボードを使用してコピーをするときだけです。

また、プレーン領域がフィールドボックスよりも大きい場合、ボックスの枠内に表示されている文字列だけがコピーできます。隠れたプレーン領域にある文字列をコピーすることはできません。

「複数フィールド」を選択している場合、複数行にわたるフィールドはコピー & ペーストの対象にできません。複数行にわたるフィールドを対象とするときは、表示・印刷セットアップの「キー操作 2」タブの「コピー & ペーストモード」で「単一フィールド」を

#### 4. マウスとキー操作

選択してください。

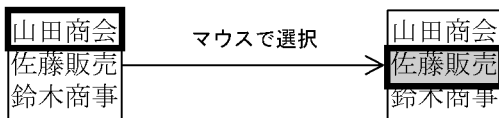
##### (3) 入出力テキスト

表示・印刷セットアップの指定にかかわらず、単一フィールドだけがコピーの対象となります。

また、文字データだけがコピーの対象となり、属性（拡大、色など）はコピーの対象となりません。表形式で表示されている複数のフィールドに対してマウスをドラッグさせると、一つのフィールドだけがコピーの対象となり、反転表示されます。

##### (4) リストボックスの1リスト（項目）

リストボックスでは、マウスで選択操作をしたリスト（項目）がコピーの対象となります。なお、マウスでのドラッグ操作はできません。



### 4.4.2 ペーストの対象

XMAP3 でのペーストの対象について説明します。また、ペーストの動作についてもあわせて説明します。

XMAP3 でのペーストの対象と動作は、表示・印刷セットアップの「キー操作 2」タブの「コピー & ペーストモード」で「単一フィールド」を選択している場合と、「複数フィールド」を選択している場合で異なります。それぞれの場合のペーストの対象と動作について説明します。

#### (1) 「単一フィールド」を選択している場合

「単一フィールド」を選択している場合、コピーしたデータを入出力テキストボックスおよびフィールドボックスの入力できる状態のフィールドにペーストします。ペーストはカレントカーソル位置を基準に実行されます。

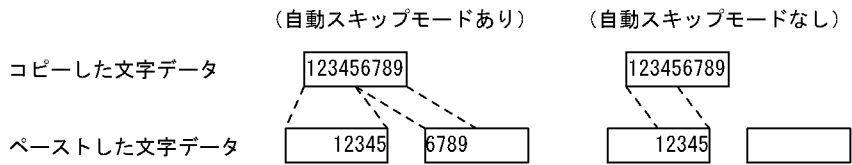
なお、ペーストの動作は、ペーストするフィールドの属性と、表示・印刷セットアップの「キー操作 2」タブの「コピー & ペーストモード」での設定によって異なります。

ペーストするフィールドの属性とペーストの動作

ペーストするフィールドの属性が自動スキップモードの場合、ペーストデータがペースト先のフィールドより長いとき、次にフォーカス遷移するフィールドに順次残りのデータがペーストされます。次にフォーカス遷移するフィールドの属性が文字入力不可の場合、残りのデータは切り捨てられます。

自動スキップモードのフィールドへのペーストの例を次の図に示します。

図 4-2 自動スキップモードのフィールドへのペーストの例



## 表示・印刷セットアップの設定とペーストの動作

表示・印刷セットアップの「キー操作 2」タブの「コピー & ペーストモード」で「単一フィールド」を選択して、「タブ/改行コードをペーストする」を選択していない場合、コピーしたデータに含まれるタブ (09)<sub>16</sub> および改行コード ((0D0A)<sub>16</sub> または (0A)<sub>16</sub>) が削除されて、前後の文字と連結した状態でペーストされます。

## 参考

表示・印刷セットアップの「キー操作 2」タブの「コピー & ペーストモード」で「複数フィールド」を選択している場合、タブ (09)<sub>16</sub> および改行コード ((0D0A)<sub>16</sub> または (0A)<sub>16</sub>) はペーストされません。

「タブ/改行コードをペーストする」を選択していない場合のペーストの例を次の図に示します。

#### 4. マウスとキー操作

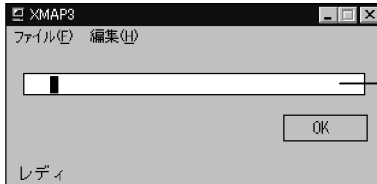
図 4-3 「タブ/改行コードをペーストする」を選択していない場合のペーストの例

《ペーストデータ》

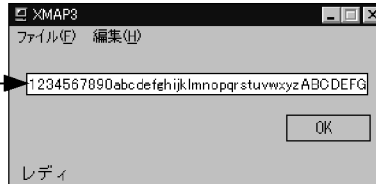
```
1234567890<TAB>abcdefghi jk lmnopqrstuvwxyz<改行>  
ABCDEFGHIJKLMNopqrstuvwxyz
```

●単一のフィールドにペーストする場合の例

(ペースト前)

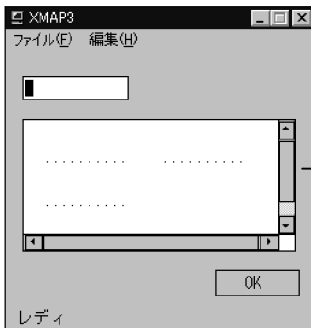


(ペースト後)

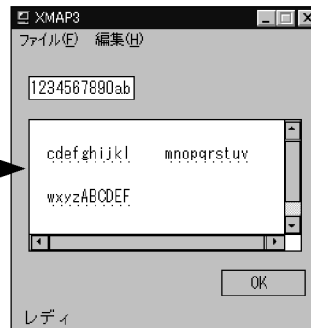


●複数のフィールドにまたがってペーストする場合の例

(ペースト前)



(ペースト後)



(凡例) <TAB> : タブ (09)<sub>16</sub> <改行> : 改行コード ((0D0A)<sub>16</sub> または (0A)<sub>16</sub>)

なお、「タブ/改行コードをペーストする」を選択していない場合、日付や時刻を表すデータを表示するフィールドに、タブ(09)<sub>16</sub> または改行コード((0D0A)<sub>16</sub> または(0A)<sub>16</sub>) で区切られたデータをペーストすることもできます。時刻フィールドへのペーストの例を次の図に示します。



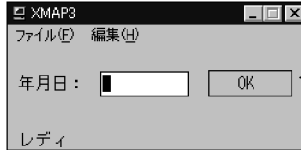
図 4-4 時刻フィールドへのペーストの例

《ペーストデータ》

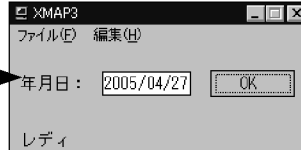
2005<TAB>04<TAB>27<改行>

● 「タブ/改行コードをペーストする」を選択していない場合の例

(ペースト前)

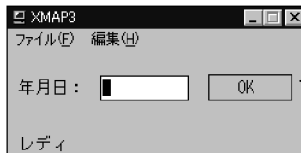


(ペースト後)

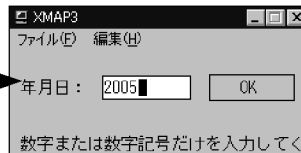


● 「タブ/改行コードをペーストする」を選択している場合の例

(ペースト前)



(ペースト後)



(凡例) <TAB> : タブ (09)<sub>16</sub> <改行> : 改行コード ((0D0A)<sub>16</sub> または (0A)<sub>16</sub>)

## (2) 「複数フィールド」を選択している場合

「複数フィールド」を選択している場合、コピーしたデータをフィールドボックスの入力できる状態のフィールドにペーストします。コピーした領域の行列数またはフィールドの長さよりも大きい目の領域をペーストの対象領域にすると、コピーした内容を確実にペーストできます。

「複数フィールド」を選択している場合、複数行にわたるフィールドはコピー & ペーストの対象にできません。複数行にわたるフィールドを対象とするときは、表示・印刷セットアップの「キー操作 2」タブの「コピー & ペーストモード」で「単一フィールド」を選択してください。

次に、特殊なペーストについて説明します。

### 保護されているフィールドの場合

保護されている可変フィールドや、固定フィールドがある場合、そのフィールドをスキップして次のフィールドにデータをペーストします。

### コピーデータに区切りコードがある場合

コピーデータ中にフィールド区切りコードがある場合、区切りコード以降のデータは次のフィールドにペーストされます。

### コピーデータに改行コードがある場合

コピーデータ中に改行コードがある場合、改行コード以降のデータは改行され、次の行のフィールドにペーストされます。ペーストと同時にカーソル位置も移動されます。なお、改行コードとして CRLF(0D0A)<sub>16</sub> を使用した場合と、LF(0A)<sub>16</sub> を使用した場

#### 4. マウスとキー操作

合のどちらでも同様に動作します。

ペースト前・表示画面 (CUIBOX)

111111	222222	333333
AAAAAA	BBBBBB	CCCCCC
777777	IIIIII	ウウウウ

非保護フィールド :

カーソル位置 :

コピーデータ

444444	区切	555555	改行
DDDDDD	区切	EEEEEE	改行
かかかか	区切	#####	改行

ペースト後・表示画面 (CUIBOX)

444444	555555	333333
DDDDDD	EEEEEE	CCCCCC
かかかか	#####	ウウウウ

対象フィールド :

カーソル位置 :

##### 自動送信動作によって中断される場合

ペースト中に自動送信動作が実行された場合は、ペーストは中断され、残りのデータは破棄されます。自動送信動作は、フィールドが自動入力フィールドである場合、コピーデータをペーストする前に、次のフィールドに移るとき実行されます。

##### 入力論理チェックによって中断される場合

ペースト中に入力論理チェックが実行された場合は、ペーストは中断され、残りのデータは破棄されます。入力論理チェックには、「桁あふれチェック」「全桁チェック」「モジュラスチェック」「右寄せチェック」「フィールドの有効文字指定と入力データとの整合性チェック」などがあります。

##### フィールドにデータが収まらない場合

データがフィールドに収まらない場合は、収まらない分を次のフィールドにペーストします。

##### データ長がフィールド長より短い場合

データ長がフィールド長より短い場合は、ペーストは中断されます。この場合、ヌル(00)<sub>16</sub>を埋めてもエラーになります。

##### ペースト中に後続のフィールドがなくなった場合


画面の最後の対象フィールドにペースト中、フィールドが終了した場合、残りのデータは画面の最初の対象フィールドにペーストされます。ただし、そのフィールドにすでにデータがペーストされている場合は、ペーストが中断されます。

#### 注

矩形領域をペーストする場合、ペースト前の XMAP3 画面のカーソルがフィールドの途中で位置づいていても、フィールドの先頭からペーストされます。

ペースト前・表示画面 (CUIBOX)

111	222
333	444
555	666


カーソル位置 : 

コピーデータ

abcd	区切	efgh	改行
123	区切	456	改行

ペースト後・表示画面 (CUIBOX)

abcd	efgh
123	456
555	666

カーソル位置 : 

### 4.4.3 クリップボードを使用したコピー & ペースト操作

コピー & ペーストは、フォーカスのあるビューポートで操作できます。

XMAP3 表示サービスが同じ画面間かどうかにかかわらず、キーによるコピー & ペースト操作 (OS のクリップボードを使用したコピー & ペースト) ができます。Windows 上で動作する一般的な AP と操作を同じにするには、キーによるコピー & ペーストを使用することをお勧めします。

次に説明している、マウスによるコピー & ペースト操作 (XMAP3 が持つクリップボードを使用した操作) は、表示・印刷セットアップの設定で「マウスによるコピー & ペーストを使用する」をチェックすると、使用できます。

XMAP3 画面でのコピー

1. コピーしたい文字データの先頭にカーソルを位置づけ、ダブルクリックするか、またはマウスの左ボタンを押したまま文字データの末尾までカーソルを移動させます。
2. コピー対象の文字列が反転している状態で、コピー機能を割り当てているキーを押すか、メニューバーの [コピー] を選びます。  
クリップボードにデータが複写されます。  
なお、コピー機能に使用するキーは、([Ctrl] + [Insert]) または ([Ctrl] + [C]) です。表示・印刷セットアップで、どちらにコピー機能を割り当てるかを指定します。

GUI 画面では、メニューバーにコピー機能を指定できるので、そのメニューを選択しても同じ動作になります。

#### 4. マウスとキー操作

##### XMAP3 画面でのペースト

1. 文字列をペーストしたい位置に、マウスの左ボタンでカーソルを位置づけます。
2. その状態でペースト機能を割り当てているキーを押すか、メニューバーから [ 貼り付け ] を選ぶと、クリップボードのデータがテキストまたはフィールドに貼り付けられます。なお、ペースト機能に使用するキーは ([ Shift ] + [ Insert ]) または ([ Ctrl ] + [ V ]) です。表示・印刷セットアップで、どちらにペースト機能を割り当てておくかを指定します。GUI 画面では、メニューバーにペースト機能を指定できるので、そのメニューを選択しても同じ動作になります。

##### 注

表示・印刷セットアップで「単一フィールド」および「タブ/改行コードをペーストする」が指定されている場合、使用目的が「日本語」、「カナ」または「英数」の入出力テキスト・フィールドに、XMAP3 画面以外からデータをペーストすると、「タブ文字」および「改行文字」もペーストされます。ただし、改行文字として  $\text{CRLF}(\text{0D0A})_{16}$  をペーストした場合は  $\text{LF}(\text{0A})_{16}$  としてペーストされます。このとき、これらの文字は空白表示されます。

なお、次に示す場合は、「タブ文字」および「改行文字」はペーストされません。

- 入出力テキスト・フィールドの使用目的が「日本語」、「カナ」または「英数」以外の場合
- 表示・印刷セットアップで「単一フィールド」が指定されていて、「タブ/改行コードをペーストする」が指定されていない場合
- 表示・印刷セットアップで「複数フィールド」が指定されている場合

# 5

## 論理マップ生成規則とマッピング規則

GUI 画面，または CUI 画面をレイアウトして保存すると，論理マップが生成されます。この章では，生成された論理マップとマッピング規則の関係について説明します。

---

5.1 論理マップとは

---

5.2 固定部の論理マップ生成規則とマッピング規則

---

5.3 可変部の論理マップ生成規則とマッピング規則

---

5.4 可変部の論理マップ生成規則とマッピング規則（GUI 画面固有）

---

5.5 定数部の論理マップ生成規則とマッピング規則

---

5.6 予約テキスト・フィールドのマッピング規則

---

## 5.1 論理マップとは

### 5.1.1 論理マップの構成

論理マップとは、AP とディスプレイ、またはプリンタとの間で受け渡されるデータを格納する領域のことで、COBOL の登録集原文または C 言語のヘッダファイルとして記述されています。

論理マップには、出力論理マップと入力論理マップがあります。

#### 出力論理マップ

AP からディスプレイ、またはプリンタに出力するデータを格納する。

#### 入力論理マップ

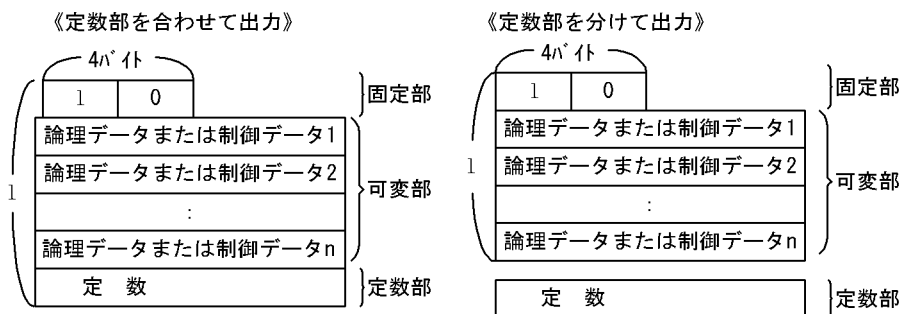
ディスプレイから表示画面に入力されたデータを格納する。

この出力論理マップと入力論理マップは、それぞれ固定部、可変部で構成されます。

また、ドロースेटアップの論理マップ属性ダイアログで「定数部への論理マップ長出力」か「定数部の別ファイル出力」を指定したときだけ、定数部が生成されます。「定数部の別ファイル出力」を選んだときは、定数部だけが別ファイルに出力されます。

論理マップの構成を次の図に示します。

図 5-1 論理マップの構成



(凡例) 1 : 論理マップ長を示します。

#### 論理マップ固定部

論理マップ固定部は、論理マップの先頭に必ず生成される領域です。この領域には、論理マップ全体の長さである論理マップ長を格納します。

#### 論理マップ可変部

論理マップ可変部は、実際に AP とディスプレイ、またはプリンタとの間で受け渡されるデータを格納する領域です。この領域に格納されるデータには、次に示す論理データと制御データがあります。

- 論理データ  
AP とディスプレイ，またはプリンタ上の可変項目やメニューなどの各項目（オブジェクト）との間で実際に受け渡されるデータです。
- 制御データ  
ディスプレイでカーソルの位置づけや項目の表示属性を動的に変更するときなどのように，表示制御を目的としたデータです。

#### 定数部

定数部は，AP を作成するときに制御項目に代入，または参照するカーソル定数やフォーカス定数など定数をテーブル化したものです。

なお，表示属性を変更する標準の修飾名はあらかじめ登録集原文として，X3MODTBL が用意されていますので，このテーブルを使って制御項目に代入します。変更した場合，この定数テーブルを基にユーザ責任で追加，変更が必要になります。

### 5.1.2 論理マップ生成規則で使用する用語

ここでは，論理マップ生成規則で使っている用語と定義方法について説明します。用語と定義方法の対応を次の表に示します。

表 5-1 論理マップ生成規則で使う用語と定義方法の対応

用語	説明	定義方法
再定義名	定義済みのマップを別定義として使用する際の名称。	画面属性ダイアログの [ 再定義名 ... ] ボタンで表示される論理マップの再定義名ダイアログで指定する。
埋字	入出力テキスト（フィールド）の場合 画面入力または AP から出力要求されたデータが，ドローで定義した項目の長さより短いときに埋められるデータ。 出力テキスト（フィールド）の場合 AP から出力要求されたデータが，ドローで定義した項目の長さより短いときに埋められるデータ。	初期値は，ドローセットアップの使用目的および詳細目の別データ型（入出力テキスト・フィールド） / （出力テキスト・フィールド）ダイアログで指定する。また，各項目のダイアログでも指定できる。
桁寄せ	テキストやフィールドに表示されるデータを右または左のどちらかに寄せるための項目。	
出力カーソル項目（2進）， 論理カーソル項目， 入力カーソル項目（2進）， 出力フォーカス項目， 入力フォーカス項目	画面の入力などを促すための位置を示す項目。GUI 画面のテキストボックスではフォーカスおよびカーソルで指定し，それ以外のオブジェクトはフォーカスで指定する。CUI 画面ではカーソルで指定する。	ドローセットアップのカーソルとフォーカスダイアログで指定する。
イベント通知コード	AP に画面からのオペレーション（画面が確定したこと）を通知するためのコード。AP と XMAP3 システム間のインタフェース情報となる。	ドローセットアップのイベント通知コードダイアログで「データ名」，「通知コードの長さ」，および「通知コード」を指定する。

## 5. 論理マップ生成規則とマッピング規則

用語	説明	定義方法
初期クリア文字	入力テキスト（フィールド）に対して何も入力しなかったとき、論理項目を何でクリアするかの文字。標準では、埋字と同じ文字が仮定される。	ドローセットアップの論理マップ属性ダイアログで指定する。
データ消去通知文字	表示データを [ End ] キー、または [ HELP ] キーで消去したり、(00) <sub>16</sub> のデータを受信したりしたときに、XMAP3 が AP に通知する値。	
コードエラー通知文字	ディスプレイからの入力コードエラーを検出したことを XMAP3 が AP に通知する値。	
下位項目	一つのオブジェクトの入力または出力領域を、複数の領域に分割するための機能。	各オブジェクトの定義ダイアログで指定する。下位項目はデータ型が「文字 ( XX )」のときだけ指定できる。

### 5.1.3 生成される標準のデータ名

XMAP3 が生成する論理マップは大きく次のようになります。

マップ名 + G : 出力論理マップ全体をクリアするときに使用

マップ名 + K : 入力論理マップ全体をクリアするときに使用

マップ名 + I : 入力データ参照用

マップ名 + O : 出力データ代入用

マップ名 + H : 入力データチェック用 ( スペース , HIGH-VALUE , LOW-VALUE , ほか )

マップ名 + A : 表示属性の動的変更制御項目代入用

次に、XMAP3 が標準に生成するデータ名の一覧を次の表に示します。

表 5-2 生成される標準のデータ名一覧

対象画面	対象オブジェクト	適用物	データ名	
			入力系データ名	出力系データ名
GUI画面	-	マップクリア項目	マップ名 K	マップ名 G
		ウィンドウ制御	-	マップ名 -CNTRLO
		ウィンドウ位置制御	-	マップ名 -WINDOWW
		イベント通知コード	マップ名 -INCI	-



対象画面	対象オブジェクト	適用物		データ名	
				入力系データ名	出力系データ名
	行列(2進)カーソル	行	1個目	マップ名・INCURSN	マップ名・OUTCURSL
			2個目以降	マップ名・INCURSnN	マップ名・OUTCURSnL
		列	1個目	マップ名・INCURSM	マップ名・OUTCURSC
			2個目以降	マップ名・INCURSnM	マップ名・OUTCURSnC
		論理カーソル	1個目	マップ名・INCURS-LOCI	マップ名・OUTCURS-LOCO
			2個目以降	マップ名・INCURSnI	マップ名・OUTCURSnO
		フォーカス制御		マップ名・INFOCUS-I	マップ名・OUTFOCUS-O
		隠しフィールド		マップ名・TRAN-I	-
	確定キーの制御		-	マップ名・INCO	
	下位項目 <sup>1</sup>		マップ名・FIELDnnnn- <i>nnn</i> -I	マップ名・FIELDnnnn- <i>nnn</i> -O	
	メニューバー	メニューバー		イベント通知コードに対応	-
		ブルダウン			
		カスケード			
メニュー制御項目		-	マップ名・MENUnnnn-A		
ポップアップメニュー	メニューリスト		-	-	
	可変用論理テーブル		-	マップ名・POPUPnnnn-O	
	可変メニューラベル		-	マップ名・POPUP-LABELnnnn-O	
	可変メニューコード		-	マップ名・POPUP-CODEnnnn-O	
	可変メニュー選択ラベル		-	マップ名・POPUP-TEXTnnnn-O	

5. 論理マップ生成規則とマッピング規則

対象画面	対象オブジェクト	適用物	データ名	
			入力系データ名	出力系データ名
	可変メニューアクセスキー			マップ名 - POPUP-KEYnnnnn-O
	ボックス	マップ名 - FIELDnnnnn-I		マップ名 - FIELDnnnnn-O
	制御項目	-		マップ名 - FIELDnnnnn-A
	入力チェック項目	マップ名 - FIELDnnnnn-H		-
	ポップアップメニューファイル	-		マップ名 - POPFILEnnnnn-O
固定テキスト	ボックス	-		-
	制御項目	-		マップ名 - FIELDnnnnn-A
出力テキスト	ボックス	-		マップ名 - FIELDnnnnn-O
	制御項目	-		マップ名 - FIELDnnnnn-A
入出力テキスト	ボックス	マップ名 - FIELDnnnnn-I		マップ名 - FIELDnnnnn-O
	制御項目	-		マップ名 - FIELDnnnnn-A
	入力チェック項目	マップ名 - FIELDnnnnn-H		-
プッシュボタン	ボックス	イベント通知コードに対応		-
	ラベルテキスト項目	-		マップ名 - BUTTONnnnnn-O
	ボタン制御項目			マップ名 - BUTTONnnnnn-A
ラジオボタン	ボックス単位通知項目	マップ名 - FIELDnnnnn-I		-
	可変用論理テーブル	-		マップ名 - RADIOnnnnn-O
	可変ボタンラベル			マップ名 - RADIO-LABELnnnnn-O
	可変ボタンコード			マップ名 - RADIO-CODEnnnnn-O

対象画面	対象オブジェクト	適用物	データ名	
			入力系データ名	出力系データ名
		固定ボタン制御項目		マップ名 - RADIOnnnn-A
		可変ボタン制御項目		マップ名 - FIELDnnnn-A
		入力チェック項目	マップ名 - FIELDnnnn-H	-
チェックボタン		ボタン単位通知項目	マップ名 - FIELDnnnn-I	-
		可変用論理テーブル	-	マップ名 - CHECKnnnn-O
		可変ボタンラベル		マップ名 - CHECK-LABELnnnn-O
		可変ボタンコード		マップ名 - CHECK-CODEnnnn-O
		ボタン制御項目		マップ名 - FIELDnnnn-A
		入力チェック項目	マップ名 - FIELDnnnn-H	-
リストボックス		ボックス単位通知項目 <sup>2</sup>	マップ名 - FIELDnnnn-I	-
		可変用論理テーブル	-	マップ名 - LISTnnnn-O
		可変リストラベル		マップ名 - LIST-LABELnnnn-O
		可変リストコード		マップ名 - LIST-CODEnnnn-O
		リスト制御項目		マップ名 - FIELDnnnn-A
		入力チェック項目	マップ名 - FIELDnnnn-H	-
コンボボックス		ボックス	マップ名 - FIELDnnnn-I	マップ名 - FIELDnnnn-O
		可変用論理テーブル	-	マップ名 - POPUPnnnn-O
		可変メニューラベル		マップ名 - POPUP-LABELnnnn-O
		可変メニューコード		マップ名 - POPUP-CODEnnnn-O

5. 論理マップ生成規則とマッピング規則

対象画面	対象オブジェクト	適用物	データ名		
			入力系データ名	出力系データ名	
		制御項目		マップ名 - FIELDnnnn-A	
		入力チェック項目	マップ名 - FIELDnnnn-H	-	
	グラフィック	出力グラフィック	-	マップ名 - FIELDnnnn-O	
	スピンボックス	ボックス	マップ名 - FIELDnnnn-I	マップ名 - FIELDnnnn-O	
		制御項目	-	マップ名 - FIELDnnnn-A	
	出力時刻	ボックス	-	マップ名 - FIELDnnnn-O	
		制御項目	-	マップ名 - FIELDnnnn-A	
	出力日付	ボックス	-	マップ名 - FIELDnnnn-O	
		制御項目	-	マップ名 - FIELDnnnn-A	
	入出力時刻	ボックス	マップ名 - FIELDnnnn-I	マップ名 - FIELDnnnn-O	
		制御項目	-	マップ名 - FIELDnnnn-A	
	入出力日付	ボックス	マップ名 - FIELDnnnn-I	マップ名 - FIELDnnnn-O	
		制御項目	-	マップ名 - FIELDnnnn-A	
	FB 3	固定フィールド	固定項目	-	-
		出力フィールド	出力項目		マップ名 - FIELDnnnn-O
			出力項目制御		マップ名 - FIELDnnnn-A
入出力フィールド		入出力項目	マップ名 - FIELDnnnn-I	マップ名 - FIELDnnnn-O	
		入出力項目制御	-	マップ名 - FIELDnnnn-A	
		入力チェック項目	マップ名 - FIELDnnnn-H	-	

対象画面	対象オブジェクト	適用物	データ名	
			入力系データ名	出力系データ名
	トグルフィールド	ボックス	マップ名・FIELDnnnn-I	-
		制御項目	-	マップ名・FIELDnnnn-A
		入力チェック項目	マップ名・FIELDnnnn-H	-

(凡例)

- : 該当しない。

注

この表のデータ名は、COBOL の場合です。C 言語の場合は、- (ハイフン) が \_ (アンダーバー) になります。

注 1

下位項目は、テキストおよびフィールドで、各オブジェクトの定義ダイアログから設定した場合に生成されます。

注 2

単一選択リストボックスは、ボックス単位の通知項目で、複数選択リストボックスではリスト単位の通知項目となります。この表では、単一選択リストボックスで記述しています。

注 3

FB : フィールドボックス

#### 5.1.4 ターゲットでの論理マップの違い

XMAP3 では、ドロースेटアップのターゲットの指定によって、生成する論理マップが異なります。

ドロースेटアップの「運用管理者用の設定」で、ターゲットに「Windows 用の画面・帳票開発」を指定した場合には、生成する論理マップの展開形式として、リトルエンディアン用、またはビッグエンディアン用のどちらかを設定できます。

ターゲットに「Windows 用の画面・帳票開発」以外を指定した場合には、それぞれのターゲットに応じたエンディアンで論理マップを生成します。

#### 5.1.5 マッピングオプション

マッピングオプションは、AP 作成時に次に示す個所に指定します。

- COBOL の SEND / RECEIVE / TRANSCEIVE 文を用いる場合 : 通信記述項の MAPPING MODE 句
- COBOL の CALL インタフェースを用いる場合 : マッピングインタフェース領域

5. 論理マップ生成規則とマッピング規則

(XMAP-MDO)

- C 言語：マッピングインタフェース領域 (XMAP\_MDO)

マッピングオプションは次の項目で有効になります。

- 可変項目 (入出力テキスト (フィールド))
- 項目属性
- ポップアップ, テキストボックス, コンボボックス, スピンボックス
- プッシュボタン
- ラジオボタン, チェックボタン, リストボックス, トグルフィールド
- 出力グラフィック
- 入出力日付テキスト (フィールド), 入出力時刻テキスト (フィールド)

マッピングオプションの説明を次の表に示します。

表 5-3 マッピングオプション

目的	マッピングオプション	MAPPING MODE 句の指定値 (COBOL)	マッピングインタフェース領域 (COBOL)	マッピングインタフェース領域 (C 言語)	説明
同じ画面に対して 2 回目以降の表示方法	1 回目または 2 回目でユーザデータを表示したいとき	空白, または 0	XMAP-MDO- MAPFLD	XMAP_MDO- MAPFLD	論理項目, または制御項目に指定したデータを使って画面を表示する。ただし, 表示属性の変更指定をしない項目属性については, 画面定義時に指定した表示属性となる。同一画面をすべて書き換える (初期化) ときに使用する。
	AP からのデータだけで部分書き換えをしたいとき	2	XMAP-MDO- LOGFLD	XMAP_MDO- LOGFLD	論理項目, または制御項目に指定したデータを使って画面を表示する。ただし, 表示属性の変更指定をしない項目属性については, 直前の画面の表示属性になる。一般的には, 3 回目以降の表示で, 入力した状態をそのままにして部分書き換えをしたいときに使用する。
1 回目 (初期) 表示用出力するユーザデータなし	物理マップだけ	3	XMAP-MDO- PHFLD	XMAP_MDO- PHFLD	論理項目, または制御項目に指定したデータを無視して, 画面定義時の情報だけで画面を表示する。初期表示したいときに使用する (メニュー表示などに使用する)。

### 5.1.6 使用目的 / 詳細目的とデータ型

入出力テキスト（フィールド）、出力テキスト（フィールド）およびキー入力コンボボックスの使用・詳細目的とデータ型の関係を表 5-4 および表 5-5 に示します。データ型は、COBOL のピクチャ句で指定するデータの型です。なお、CUI 画面では詳細目的は指定しません。CUI 画面の場合は、「詳細目的」の列を無視してください。

表 5-4 使用目的および詳細目的とデータ型の関係（入出力テキスト・フィールド、キー入力コンボボックス）

使用目的	詳細目的 (GUI だけ)	データ型 <sup>1</sup>		
		COBOL		
		入力データ型	出力データ型	
数字 <sup>2</sup>	数字・ 数字記号 <sup>2</sup>	文字 (XX) <sup>2</sup>	文字 (XX)	
		99999 <sup>2</sup>	99999	
		99999	ZZZZ9	
		その他のピクチャ	その他のピクチャ	
	モジュラス	文字 (XX)	文字 (XX)	
		99999	99999	
		99999	ZZZZ9	
		その他のピクチャ	その他のピクチャ	
	数字	文字 (XX)	文字 (XX)	
		99999	99999	
金額	数字	文字 (XX)	文字 (XX)	
		99999	99999	
	小数点付き	999V9	999.9	
		99V99	99.99	
		9V999	9.999	
		V9999	.9999	
	符号付き	S99999	-99999	
	符号・小数点付き	S999V9	-999.9	
		S99V99	-99.99	
		S9V999	-9.999	
		SV9999	-.9999	
	数値	数字	文字 (XX)	文字 (XX)
			99999	99999
		小数点付き	999V9	999.9

5. 論理マップ生成規則とマッピング規則

使用目的	詳細目的 ( GUI だけ )	データ型 <sup>1</sup>	
		COBOL	
		入力データ型	出力データ型
		99V99	99.99
		9V999	9.999
		V9999	.9999
	符号付き	S99999	-99999
	符号・小数点付き	S999V9	-999.9
		S99V99	-99.99
		S9V999	-9.999
		SV9999	-.9999
数字・数字記号	文字 ( XX )	文字 ( XX )	
カナ <sup>2</sup>	カナ・半角 <sup>2</sup>	文字 ( XX )	文字 ( XX )
	カナ		
	カナ・英大・数		
	カナ・英大・数・マイナス		
	カナ・英大・数・長音		
	カナ・英大・数・マイナス・長音		
	カナ・数		
	カナ・数・マイナス		
	カナ・数・長音		
	カナ・数・マイナス・長音		
英数 <sup>2</sup>	半角 <sup>2</sup>	文字 ( XX )	文字 ( XX )
	英大・カナ・数・記号		
	英大		
	アスタリスク・英大		
	英大・英小		
	英大・数・マイナス		
	英大・数		
	英大・英小・コンマ		
	数・マイナス		
日本語 <sup>2</sup>	混在 ( 全角・半角 ) <sup>2</sup>	文字 ( XX )	文字 ( XX )
	漢字 <sup>2</sup>	文字 ( XX )	文字 ( XX )
		漢字 ( NN )	漢字 ( NN )



使用目的	詳細目的 (GUI だけ)	データ型 <sup>1</sup>	
		COBOL	
		入力データ型	出力データ型
		漢字 (XX) <sup>3</sup>	漢字 (XX) <sup>3</sup>
パスワード	半角	文字 (XX)	文字 (XX)
	英大		
	英大・英小		
	英大・数		
MCR	半角	文字 (XX)	出力なし
	英大		
	英大・英小		
	英大・数		

注 1

C 言語の場合は、データ型「char」だけ指定できます。

注 2

キー入力コンボボックスで指定できる使用目的、詳細目的およびデータ型です。

注 3

CUI 画面の入出力フィールドの場合にだけ指定できます。

表 5-5 使用目的および詳細目的とデータ型の関係 (出力テキスト・フィールド)

使用目的	詳細目的 (GUI だけ)	データ型		
		COBOL		
		入力データ型	出力データ型	
数字	数字・ 数字記号	文字 (XX)	文字 (XX)	
		99999	99999	
		99999	ZZZZ9	
		その他のピクチャ	その他のピクチャ	
	モジュール	モジュール	文字 (XX)	文字 (XX)
			99999	99999
			99999	ZZZZ9
			その他のピクチャ	その他のピクチャ
	数字	数字	文字 (XX)	文字 (XX)
			99999	99999
	英数	半角	文字 (XX)	文字 (XX)

## 5. 論理マップ生成規則とマッピング規則

使用目的	詳細目的 (GUI だけ)	データ型	
		COBOL	
		入力データ型	出力データ型
	英大・カナ・数・記号		
	英大		
	アスタリスク・英大		
	英大・英小		
	英大・数・マイナス		
	英大・数		
	英大・英小・コンマ		
	数・マイナス		
日本語	混在 (全角・半角)	文字 (XX)	文字 (XX)
	漢字	文字 (XX)	文字 (XX)
		漢字 (NN)	漢字 (NN)
		漢字 (XX)	漢字 (XX)

### 注

C 言語の場合は、データ型「char」だけ指定できます。

入出力テキスト・フィールドの使用目的および詳細目的とデータ型の標準値は、ドローセットアップの使用目的別データ型（入出力テキスト・フィールド）、使用目的別データ型（出力テキスト・フィールド）で変更することもできます。ただし、表 5-4、表 5-5 以外の組み合わせは指定できません。なお、キー入力コンボボックスのデータ型は変更できません。

### 5.1.7 ドローセットアップとの関係

ドローセットアップは、画面・帳票定義に関する標準設定値の変更のためのものです。ここで設定した内容が生成されるマップの形式に反映されるため、必ず画面・帳票定義を始める前に済ませておく必要があります。

定義した後に変更する場合は、必ず「セットアップ情報反映」を使用してください。

なお、ドローセットアップの詳細については、マニュアル「XMAP3 開発・実行ガイド」を参照してください。

### 5.1.8 下位項目を指定したときの扱い

下位項目は上位項目のデータ型が「文字 (XX)」のときだけ指定できます。下位項目の指定では、下位項目のデータ長とデータ型を指定できます。

## 《COBOL》

```
{02|03|04} マップ名-FIELDnnnn-nnn-{0|1}. ……オブジェクトに対応する上位項目
  {03|04|05} マップ名-FIELDnnnn-nnn-{0|1} ……} 下位項目
  {03|04|05}      :
```

## 《C 言語》

```
struct {
  unsigned char マップ名_FIELDnnnn_nnn_{0|1}[長さ];
  :
```

## 5.1.9 フレームを使用したときの扱い

オブジェクトに対応するデータ項目をまとめて集団項目にします。

## 《COBOL》

```
{02|03} マップ名-FRAMEnnnn-{0|1} [OCCURS n].
  {04|05} マップ名……………オブジェクトに対応する項目
```

## 《C 言語》

```
struct {
  オブジェクトに対応する項目
  …} マップ名_FRAMEnnnn_{0|1} [[n]] ;
```

## 5.2 固定部の論理マップ生成規則とマッピング規則

ここでは、固定部の入出力に関する論理マップ生成規則とマッピング規則について説明します。固定部の論理マップ生成規則とマッピング規則は、再定義名の有無によって内容が変わります。

固定部のマップに関する定義

固定部のマップに関する定義は、再定義名の指定によって変わってきます。

再定義名は、画面属性ダイアログ（GUI画面の場合）の[再定義名...]ボタンで表示される論理マップの再定義名ダイアログで指定します。

### 5.2.1 固定部の出力論理マップ生成規則とマッピング規則

固定部の出力論理マップ生成規則とマッピング規則を次に示します。

#### (1) 出力論理マップ生成規則

##### (a) 再定義名を指定した場合

《COBOL》

論理マップ可変部の集団項目化を指定した場合

- Windows リトルエンディアン用，Linux 用

```
01 マップ名0 REDEFINES 出力再定義名.  
02 マップ名L PIC S9(4) COMP-5.  
02 マップ名Z PIC S9(4) COMP.  
02 マップ名G.
```

- Windows ビッグエンディアン用，AIX / HP-UX / Solaris 用

```
01 マップ名0 REDEFINES 出力再定義名.  
02 マップ名L PIC S9(4) COMP.  
02 マップ名Z PIC S9(4) COMP.  
02 マップ名G.
```

論理マップ可変部の集団項目化を指定していない場合

- Windows リトルエンディアン用，Linux 用

```

01 マップ名0 REDEFINES 出力再定義名.
02 マップ名L PIC S9(4) COMP-5.
02 マップ名Z PIC S9(4) COMP.

```

- Windows ビッグエンディアン用 , AIX / HP-UX / Solaris 用

```

01 マップ名0 REDEFINES 出力再定義名.
02 マップ名L PIC S9(4) COMP.
02 マップ名Z PIC S9(4) COMP.

```

### 《C 言語》

C 言語の場合 , 集団項目化の指定は無視されます。

- Windows リトルエンディアン用 , AIX / HP-UX / Solaris / Linux 用

```

struct マップ名0 {
    short      マップ名L;
    unsigned char マップ名Z[2];
    :
} 出力再定義名;

```

#### 注

short の大きさは , 2 バイトとして展開されます。

- Windows ビッグエンディアン用

```

struct マップ名0 {
    unsigned char マップ名L[2];
    unsigned char マップ名Z[2];
    :
} 出力再定義名;

```

### (b) 再定義名を指定しない場合

#### 《COBOL》

論理マップ可変部の集団項目化を指定した場合

- Windows リトルエンディアン用 , Linux 用

## 5. 論理マップ生成規則とマッピング規則

```
01 マップ名0.  
02 マップ名L PIC S9(4) COMP-5 [VALUE+長さ] .  
02 マップ名Z PIC S9(4) COMP [VALUE+0] .  
02 マップ名G.
```

- Windows ビッグエンディアン用 , AIX / HP-UX / Solaris 用  
< 論理マップ長が 10,000 バイト未満の場合 >

```
01 マップ名0.  
02 マップ名L PIC S9(4) COMP [VALUE+長さ] .  
02 マップ名Z PIC S9(4) COMP [VALUE+0] .  
02 マップ名G.
```

< 論理マップ長が 10,000 バイト以上の場合 >

```
01 マップ名0.  
02 マップ名Q PIC X(2) VALUE X'長さ'.  
02 マップ名L REDEFINES マップ名Q PIC S9(4) COMP.  
02 マップ名Z PIC S9(4) COMP VALUE +0.  
02 マップ名G.
```

論理マップ可変部の集団項目化を指定していない場合

- Windows リトルエンディアン用 , Linux 用

```
01 マップ名0.  
02 マップ名L PIC S9(4) COMP-5 [VALUE+長さ] .  
02 マップ名Z PIC S9(4) COMP [VALUE+0] .
```

- Windows ビッグエンディアン用 , AIX / HP-UX / Solaris 用  
< 論理マップ長が 10,000 バイト未満の場合 >

```
01 マップ名0.  
02 マップ名L PIC S9(4) COMP [VALUE+長さ] .  
02 マップ名Z PIC S9(4) COMP [VALUE+0] .
```

< 論理マップ長が 10,000 バイト以上の場合 >

```
01 マップ名0.  
02 マップ名Q PIC X(2) VALUE X'長さ'.  
02 マップ名L REDEFINES マップ名Q PIC S9(4) COMP.  
02 マップ名Z PIC S9(4) COMP VALUE +0.
```

## 《C言語》

C言語の場合，集団項目化の指定は無視されます。

- Windows リトルエンディアン用，AIX / HP-UX / Solaris / Linux 用

```
struct {
    short      マップ名L;
    unsigned char マップ名Z[2];
    :
} マップ名0 [= {長さ, {0x00, 0x00}}] ;
```

## 注

short の大きさは，2 バイトとして展開されます。

- Windows ビッグエンディアン用

```
struct {
    unsigned char マップ名L[2];
    unsigned char マップ名Z[2];
    :
} マップ名0 [= {{0xnn, 0xnn}, {0x00, 0x00}}] ;
```

## (2) マッピング規則

## (a) 再定義名を指定した場合

論理マップ長部（マップ名Lで生成される項目）に代入した内容とその結果

論理マップ長部に代入した内容	結果
展開した論理マップより小さい値	代入された論理マップ長が示す範囲のデータを使って画面・帳票出力のデータとして使用される。範囲外の部分にはデータが無コードが代入されているものとして扱う。
展開した論理マップ長と等しい値	論理マップ内のデータを，画面・帳票出力のデータとして使用する。
展開した論理マップより大きい値	マップ生成機能が展開した論理マップの範囲までのデータを，画面・帳票出力のデータとして使用する。それ以外の部分は無視される。

## (b) 再定義名を指定しない場合

論理マップ長部（マップ名Lで生成される項目）に代入した内容とその結果は，再定義名を指定した場合と同様になります。

## 5.2.2 固定部の入力論理マップ生成規則とマッピング規則

固定部の入力論理マップ生成規則とマッピング規則を次に示します。

## 5. 論理マップ生成規則とマッピング規則

### (1) 入力論理マップ生成規則

#### (a) 再定義名を指定した場合

##### 《COBOL》

論理マップ可変部の集団項目化を指定した場合

- Windows リトルエンディアン用, Linux 用

```
01 マップ名I REDEFINES入力再定義名.  
02 マップ名S PIC S9(4) COMP-5.  
02 マップ名0 PIC S9(4) COMP.  
02 マップ名K.
```

- Windows ビッグエンディアン用, AIX / HP-UX / Solaris 用

```
01 マップ名I REDEFINES入力再定義名.  
02 マップ名S PIC S9(4) COMP.  
02 マップ名0 PIC S9(4) COMP.  
02 マップ名K.
```

論理マップ可変部の集団項目化を指定していない場合

- Windows リトルエンディアン用, Linux 用

```
01 マップ名I REDEFINES入力再定義名.  
02 マップ名S PIC S9(4) COMP-5.  
02 マップ名0 PIC S9(4) COMP.
```

- Windows ビッグエンディアン用, AIX / HP-UX / Solaris 用

```
01 マップ名I REDEFINES入力再定義名.  
02 マップ名S PIC S9(4) COMP.  
02 マップ名0 PIC S9(4) COMP.
```

##### 《C 言語》

C 言語の場合, 集団項目化の指定は無視されます。

- Windows リトルエンディアン用, AIX / HP-UX / Solaris / Linux 用

```
struct マップ名I {  
    short      マップ名S;  
    unsigned char マップ名0[2];  
    :  
}  
入力再定義名 ;
```



## 注

short の大きさは、2 バイトとして展開されます。

- Windows ビッグエンディアン用

```
struct マップ名I {
    unsigned char マップ名S[2];
    unsigned char マップ名O[2];
    :
} 入力再定義名;
```

## (b) 再定義名を指定しない場合

## 《COBOL》

論理マップ可変部の集団項目化を指定した場合

- Windows リトルエンディアン用, Linux 用

```
01 マップ名I.
02 マップ名S PIC S9(4) COMP-5 [VALUE +長さ].
02 マップ名O PIC S9(4) COMP [VALUE +0].
02 マップ名K.
```

- Windows ビッグエンディアン用, AIX / HP-UX / Solaris 用  
<論理マップ長が 10,000 バイト未満の場合>

```
01 マップ名I.
02 マップ名S PIC S9(4) COMP [VALUE +長さ].
02 マップ名O PIC S9(4) COMP [VALUE +0].
02 マップ名K.
```

<論理マップ長が 10,000 バイト以上の場合>

```
01 マップ名I.
02 マップ名F PIC X(2) VALUE X'長さ'.
02 マップ名S REDEFINES マップ名Q PIC S9(4) COMP.
02 マップ名O PIC S9(4) COMP VALUE +0.
02 マップ名K.
```

論理マップ可変部の集団項目化を指定していない場合

- Windows リトルエンディアン用, Linux 用

```
01 マップ名I.
02 マップ名S PIC S9(4) COMP-5 [VALUE+長さ].
02 マップ名O PIC S9(4) COMP [VALUE+0].
```

## 5. 論理マップ生成規則とマッピング規則

- Windows ビッグエンディアン用, AIX / HP-UX / Solaris 用  
< 論理マップ長が 10,000 バイト未満の場合 >

```
01 マップ名I.  
02 マップ名S PIC S9(4) COMP [VALUE+長さ].  
02 マップ名O PIC S9(4) COMP [VALUE+0].
```

< 論理マップ長が 10,000 バイト以上の場合 >

```
01 マップ名I.  
02 マップ名F PIC X(2) VALUE X'長さ'.  
02 マップ名S REDEFINES マップ名O PIC S9(4) COMP.  
02 マップ名O PIC S9(4) COMP VALUE +0.
```

### 《C 言語》

C 言語の場合, 集団項目化の指定は無視されます。

- Windows リトルエンディアン用, AIX / HP-UX / Solaris / Linux 用

```
struct {  
    short      マップ名S;  
    unsigned char マップ名O[2];  
    :  
} マップ名I [= {長さ, {0x00, 0x00}} ] ;
```

注

short の大きさは, 2 バイトとして展開されます。

- Windows ビッグエンディアン用

```
struct {  
    unsigned char マップ名S[2];  
    unsigned char マップ名O[2];  
    :  
} マップ名I [= { {0xnn, 0xnn}, {0x00, 0x00} } ] ;
```

## (2) マッピング規則

### (a) 再定義名を指定した場合

論理マップ長部（マップ名 S で生成される項目）に代入した内容とその結果

論理マップ長部に代入した内容	結果
展開した論理マップより小さい値	入力マッピング時、論理マップ長部はチェックされない（マッピング制御機能が入力論理マップ長の値を代入する）ため、ほかの領域を壊すことがある。必ずマップ長分の領域を用意する必要がある。
展開した論理マップ長と等しい値	指定された論理マップの範囲に入力データが代入される。
展開した論理マップより大きい値	マップ生成機能が展開した入力論理マップの範囲に入力データを代入し、それ以外の部分は無視する（データは代入されない）。論理マップ長部には、マッピング制御機能が入力論理マップ長の値を代入する。

### (b) 再定義名を指定しない場合

論理マップ長部（マップ名 S で生成される項目）に代入した内容とその結果は、再定義名を指定した場合と同様になります。

## 5.3 可変部の論理マップ生成規則とマッピング規則

ここでは、可変部の入出力に関する論理マップ生成規則とマッピング規則について説明します。この節では、COBOLの数字項目の記述を次のようにしています。

- PIC 9 (長さ) と表記している項目  
実際に生成される論理マップは、PIC 99999 のように 9 が長さ分生成されます。

なお、論理項目に指定する文字データとして、ヌル (00)<sub>16</sub> およびデータ有無コード (1F)<sub>16</sub> を除く制御コード (JIS : (00)<sub>16</sub> ~ (1F)<sub>16</sub> , (7F)<sub>16</sub>) は指定できません。

### 5.3.1 ウィンドウ表示

#### (1) ウィンドウ表示の制御情報

制御情報の標準値

- ウィンドウ制御項目データ名 : マップ名-CNTRLO  
データ名は、ドロースettingsの「ドロ-の設定」から表示する、表示属性の動的変更ダイアログの「データ名」タブで変更できます。
- 長さ : 2  
長さは、ドロースettingsの「運用管理者用の設定」の [ 修飾名の設定 ... ] ボタンをクリックして表示される、修飾名の設定ダイアログで変更できます。変更するときには、「動的変更の種別」から「ウィンドウ属性」を選択し、「修飾名長」で長さを設定します。

#### (2) 論理マップ生成規則

ウィンドウ表示制御情報の論理マップ生成例を次に示します。

《COBOL》

```
{02|03} マップ名-CNTRLO PIC X(長さ).
```

《C 言語》

```
unsigned char マップ名_CNTRLO [長さ];
```

#### (3) マッピング規則

ウィンドウ表示制御情報のマッピング規則について説明します。

## AP がウィンドウ表示制御情報のデータ名の領域に代入した内容とその結果

AP がウィンドウ表示制御情報のデータ名の領域に代入した内容	結果
修飾名と同じ	ウィンドウ表示制御情報の修飾名に対応するウィンドウ制御情報を使って画面が表示される。
データの先頭またはすべてがデータ有無コード，または修飾名以外（上記以外）	標準のウィンドウ表示制御情報を使って画面が表示される。画面属性ダイアログの表示形態が「一部上書」，または「自動」の場合は，直前の表示画面と同じ画面のとき「一部上書」で，異なる画面のとき「全面書換」で表示する。

## 注

修飾名と修飾名に対応したウィンドウ表示制御情報は，ドロースेटアップの「表示属性の動的変更」の「ウィンドウ属性」タブで変更できます。

## 5.3.2 出力カーソル制御

## (1) 出力カーソル項目の定義

行列（2進）カーソルの標準値

- 出力カーソル項目データ名：(行) マップ名-OUTCURSL,(列) マップ名-OUTCURSC
- 長さ：4

論理カーソルの標準値

- 論理カーソル項目データ名： マップ名-OUTCURS-LOCO
- 長さ：2

データ名は，ドロースेटアップの「カーソルとフォーカス」で変更できます。

## (2) 論理マップ生成規則

出力カーソル制御の論理マップ生成例を次に示します。GUI画面のフォーカス・カーソル制御については，「5.4.8 フォーカス・カーソル位置の論理マップ生成規則とマッピング規則」を参照してください。

## (a) 行列（2進）カーソルを指定した場合

《COBOL》

- Windows リトルエンディアン用，Linux 用

```
{02|03} マップ名-OUTCURSL PIC S9(4) COMP-5.
{02|03} マップ名-OUTCURSC PIC S9(4) COMP-5.
```

## 5. 論理マップ生成規則とマッピング規則

- Windows ビッグエンディアン用 , AIX / HP-UX / Solaris 用

```
{02|03} マップ名-OUTCURSL PIC S9(4) COMP.
{02|03} マップ名-OUTCURSC PIC S9(4) COMP.
```

### 《C 言語》

```
unsigned char マップ名_OUTCURSL [2];
unsigned char マップ名_OUTCURSC [2];
```

#### (b) 論理カーソルを指定した場合

論理カーソル定数については「5.5.1(2)(b) カーソル定数の出力の指定」を参照してください。

### 《COBOL》

```
{02|03} マップ名-OUTCURS-LOC0 PIC X(2).
```

### 《C 言語》

```
unsigned char マップ名_OUTCURS_LOC0 [2];
```

## (3) マッピング規則

出力カーソル制御のマッピング規則について説明します。GUI 画面のフォーカス・カーソル制御については、「5.4.8 フォーカス・カーソル位置の論理マップ生成規則とマッピング規則」を参照してください。

#### (a) 行列 (2 進) カーソルを指定した場合

AP がカーソル項目データ名の領域に代入した内容とその結果

AP がカーソル項目データ名の領域に代入した内容	結果
不正なカーソル位置 , (00) <sub>16</sub> クリア , またはデータ有無 コードクリア	《ドローの定義画面で初期カーソル位置の指定がある》 初期カーソルを指定したフィールドの先頭にカーソルが位置づく。 《初期カーソル位置の指定がない》 先頭の行列位置にある入力できるフィールドにカーソルが位置づく。カーソルはフィールドの先頭に位置づく。
正しいカーソル位置 (1 行 / 1 列 - 画面サイズ)	出力カーソル項目データ名の領域に代入された値 (行 / 列位置) にカーソルが位置づく。

注

例えば、行 / 列が画面の大きさを超えている場合が該当します。

(b) 論理カーソルを指定した場合

AP がカーソル項目データ名の領域に代入した内容とその結果

AP がカーソル項目データ名の領域に代入した内容	結果
(00) <sub>16</sub> クリア, または マップ生成機能が生成するカーソル定数以外の値	《ドローの定義画面で初期カーソル位置の指定がある》 初期カーソルを指定したフィールドの先頭にカーソルが位置づく。 《初期カーソル位置の指定がない》 先頭の行列位置にある入力できるフィールドにカーソルが位置づく。カーソルはフィールドの先頭に位置づく。
マップ生成機能が生成するカーソル定数	カーソル定数に対応したフィールドの先頭にカーソルが位置づく。

### 5.3.3 入力カーソル制御

#### (1) 入力カーソル項目の定義

行列 (2 進) カーソルの標準値

- 入力カーソル項目データ名 : (行) マップ名 -INCURSN, (列) マップ名 -INCURSM

- 長さ : 4

データ名は、ドローセットアップの「カーソルとフォーカス」で変更できます。

論理カーソルの標準値

- 論理カーソル項目データ名 : マップ名 -INCURS-LOCI

- 長さ : 2

データ名は、ドローセットアップの「カーソルとフォーカス」で変更できます。

#### (2) 論理マップ生成規則とマッピング規則

入力カーソル制御の論理マップ生成例とマッピング規則について説明します。GUI 画面のフォーカス・カーソル制御については、「5.4.8 フォーカス・カーソル位置の論理マップ生成規則とマッピング規則」を参照してください。

##### (a) 行列 (2 進) カーソルを指定した場合

《COBOL》

- Windows リトルエンディアン用, Linux 用

[02 03] マップ名-INCURSN PIC S9(4) COMP-5.
[02 03] マップ名-INCURSM PIC S9(4) COMP-5.

## 5. 論理マップ生成規則とマッピング規則

- Windows ビッグエンディアン用, AIX / HP-UX / Solaris 用

```
{02|03} マップ名-INCURSN PIC S9(4) COMP.
{02|03} マップ名-INCURSM PIC S9(4) COMP.
```

### 《C 言語》

```
unsigned char マップ名_INCURSN[2];
unsigned char マップ名_INCURSM[2];
```

#### (b) 論理カーソルを指定した場合

論理カーソル定数については、「5.5.1(2)(b) カーソル定数の出力の指定」を参照してください。

### 《COBOL》

```
03 マップ名-INCURS-LOGI PIC X(2).
```

### 《C 言語》

```
unsigned char マップ名_INCURS_LOGI[2];
```

## (3) マッピング規則

入力カーソル制御のマッピング規則について説明します。

#### (a) 行列 (2 進) カーソルを指定した場合

AP のカーソル項目データ名の領域に代入される内容

画面送信時の入力カーソル位置 (行 / 列位置) が行, 列それぞれのデータ名に代入されます。

#### (b) 論理カーソルを指定した場合

AP のカーソル項目データ名の領域に代入される内容

入力操作	結果
画面送信時のカーソル位置が入力・入出力フィールド内に収まっていない。	(00) <sub>16</sub> を代入する。
画面送信時のカーソル位置が入力・入出力フィールド内に収まっている。	該当するフィールドに対応する論理カーソル定数を代入する。



## 注

入力単位が「フィールド単位」または「イベント単位」の場合、キー操作をしたオブジェクトのカーソル定数を代入します。

### 5.3.4 イベント通知コード

#### (1) イベント通知コードの定義

イベント通知項目は、通常マップ生成を実行すると自動的に生成されます。

イベント通知項目の標準値

- イベント通知項目：マップ名-INCI
- 長さ（標準値）：4

標準値は、ドローセットアップのイベント通知コードの「通知コードのデータ名」および「通知コードの長さ」で変更できます。

#### (2) 論理マップ生成規則

イベント通知項目の論理マップ生成規則について説明します。

《COBOL》

```
[02|03] マップ名-INCI PIC X(4)
```

《C言語》

```
unsigned char マップ名_INCI[4];
```

#### (3) マッピング規則

イベント通知項目のマッピング規則について説明します。

入力操作と結果（画面単位 / フィールド単位の場合）

入力操作	結果
指定されたイベント通知コードと対応するキーが入力された。	イベント通知項目の領域にイベント通知コードを代入する。余白がある場合、余白は空白（(20) <sub>16</sub> ）で埋める。
指定されたイベント通知コードと対応しないキーが入力された。またはMCR入力によって自動送信された。	初期クリア文字に従う。初期クリア文字が埋字の場合、イベント通知項目は空白（(20) <sub>16</sub> ）で埋める。

## 注

- イベント通知コードは、ドローセットアップのイベント通知コードダイアログで変更できます。フィールドの全桁入力による自動送信の場合、送信キーが入力されたものとして動作します。フィールド単位、または自動送信属性が定義された項目に対してMCR入力するとき、項目長よりも長いデータを入力した場合は、MCR入力による自動送信ではなくフィールド離脱による自

## 5. 論理マップ生成規則とマッピング規則

動送信の扱いとなります。

- ターゲットの文字コードが EUC の場合、半角カナは 2 バイトの領域を必要とします。そのため、イベント通知コードの長さが半角カナの上位バイト (1 バイト目) までしか設定できない場合には、イベント通知コードの末尾に不当な文字コードが設定されます。

入力操作と結果 (イベント単位の場合)

入力操作	結果
マウスまたはキー操作でのフィールド離脱で、イベントが発生した。	イベント通知項目に、イベント定数を代入する。繰り返しを指定している項目の場合、繰り返しの先頭項目と同じイベント定数を代入する。イベント定数については「5.5.2 イベント定数の論理マップ生成規則」を参照のこと。
指定されたイベント通知コードと対応するキーが入力された。	イベント通知項目の領域にイベント通知コードを代入する。余白がある場合、余白は空白 (20 <sub>16</sub> ) で埋める。
指定されたイベント通知コードと対応しないキーが入力された。または MCR 入力によって自動送信された。	初期クリア文字に従う。初期クリア文字が埋字の場合、イベント通知項目は空白 (20 <sub>16</sub> ) で埋める。

注

MCR 入力するとき、項目長よりも長いデータを入力した場合は、MCR 入力による自動送信ではなくフィールド離脱による自動送信の扱いとなります。

### 5.3.5 可変項目の論理マップ生成規則とマッピング規則

#### (1) 可変項目の定義

GUI 画面と GUI 画面内のフィールドボックスでは、各項目の名称が次の表のように異なります。

表 5-6 可変項目の名称の違い

可変項目	GUI 画面	フィールドボックス
出力項目	出力テキストボックス	出力フィールド
入出力項目	入出力テキストボックス	入出力フィールド

各項目の名称は、それぞれの画面での名称に置き換えてください。

出力項目

出力テキストボックス、または出力フィールドに対して生成されます。

- 出力項目 : マップ名-FIELDnnnn-O
  - 繰り返し回数 : 縦、または横の反復回数 (出力フィールドだけ)
  - 長さ : 論理項目の長さ。指定がないときはフィールドのデータ長
  - データ型 : テキストボックス、フィールドのデータ型で設定
  - 使用目的 : テキストボックス、フィールドの使用目的で設定
- 出力項目の使用目的を次の表に示します。

表 5-7 出力項目の使用目的と使用できる項目

使用目的	使用できる項目
数字	文字, 数字編集項目
英数	文字
日本語	文字, 漢字

#### 入力項目

入出力テキストボックス, または入出力フィールドに対して生成されます。次に示す入力項目以外の定義内容は「出力項目」と同じになります。

- 入力項目 : マップ名-FIELDnnnn-I

入力項目の使用目的を次の表に示します。

表 5-8 入力項目の使用目的と使用できる項目

使用目的	使用できる項目	使用目的	使用できる項目
数字	文字, 数字編集項目	英数	文字
金額	文字編集項目	日本語	文字, 漢字
数値	文字編集項目	MCR	文字
カナ	文字	パスワード	文字

#### 入出力項目

入出力テキストボックス, または入出力フィールドに対して生成されます。論理マップは, 入力論理マップと出力論理マップに分けて出力されます。定義内容は出力項目と同じになります。

### (2) 可変項目の出力論理マップ生成規則とマッピング規則

可変項目の出力論理マップ生成規則と論理マップ生成規則について説明します。

#### (a) 出力論理マップ生成規則

##### 初期値を設定しない場合

《COBOL》

COBOL の可変項目の生成規則は, 使用目的によってデータ型が変わります。

- 数字項目の場合

```
{02|03} マップ名-FIELDnnnn-0 PIC 9(長さ) [OCCURS 回数] .
```

- 文字項目の場合

```
{02|03} マップ名-FIELDnnnn-0 PIC X(長さ) [OCCURS 回数] .
```

5. 論理マップ生成規則とマッピング規則

- 漢字項目の場合

{02|03} マップ名-FIELDnnnn-0 PIC N(長さ÷2) [OCCURS 回数] .

- 数字編集項目の場合

{02|03} マップ名-FIELDnnnn-0 PIC 編集文字 [OCCURS 回数] .

《C 言語》

unsigned char マップ名\_FIELDnnnn\_0 [[回数]] [長さ];

- C 言語使用時は、数字編集項目は指定できません。
- 日本語では、「桁寄せ」は左寄せになります。

初期値を指定した場合

初期値を指定した場合の論理マップ生成例は初期値を指定しないときと同じです。

(b) マッピング規則

初期値を設定しない場合

- AP が出力項目データ名の領域に代入した内容と結果

AP が出力項目データ名の領域に代入した内容	マッピングオプション	結果
すべてデータの場合	マージ, 論理マップ	代入されたデータを表示する。
先頭 1 文字がデータ有無コード		データ有無コード (1F) <sub>16</sub> を設定した場合、例として X'1F414234' または X'1F1F1F' のようなデータが該当する。このとき、表示形態が「全面書換」か「一部上書」によって次のように異なる。 全面書換の場合：何も表示しない。 一部上書の場合：直前に表示したデータのまま表示する。
データの後半にデータ有無コード		データ有無コード (1F) <sub>16</sub> を仮定した場合、例として X'4142431F1F' のようなデータが該当する。この場合、「桁寄せ」の指定に従って桁寄せをし、指定した埋字を埋めて表示する。埋字と桁寄せのマッピング規則については、「5.3.9 埋字と桁寄せに関するマッピング規則（出力フィールド）」を参照のこと。
-	物理マップ	データを表示しない。

(凡例)

- : 該当しない。

## 初期値を指定した場合

- AP が出力項目データ名の領域に代入した内容と結果

AP が出力項目データ名の領域に代入した内容	マッピングオプション	結果
先頭 1 文字がデータ有無コード	マージ, 論理マップ	《先頭 1 文字がデータ有無コードのときの結果》を参照。
すべてデータ		代入されたデータを表示する。
データの後半にデータ有無コード		この場合、「桁寄せ」の指定に従って桁寄せをし、指定した埋字を埋めて表示する（数字編集項目の場合は、編集文字に従って編集してから表示）。埋字と桁寄せのマッピング規則については、「5.3.9 埋字と桁寄せに関するマッピング規則（出力フィールド）」を参照のこと。
-	物理マップ	《先頭 1 文字がデータ有無コードのときの結果》を参照。

(凡例)

- : 該当しない。

## 《先頭 1 文字がデータ有無コードのときの結果》

初期値の内容	結果
すべて空白 (20) <sub>16</sub> )	項目全体を空白で表示する。
すべて 0 (30) <sub>16</sub> )	項目全体を 0 で表示する。
すべてヌル (00) <sub>16</sub> )	項目をヌルクリアして表示する。
繰り返し文字指定	項目全体を繰り返し文字で表示する。
初期値の長さや画面の表示長とが等しい	初期値を項目に表示する。
初期値の長さが画面の表示長より短い	定義で指定した「桁寄せ」、および埋字に従って表示する。埋字抑止を指定した場合は、左寄せで初期値を項目に表示する。
初期値の長さが画面の表示長より長い	「桁寄せ」に従って桁寄せし、余りを切り捨てて項目を表示する。

## (3) 可変項目の入力論理マップ生成規則とマッピング規則

入出力フィールド・テキストを指定した場合の可変項目の入力論理マップ生成規則と論理マップ生成規則について説明します。

## (a) 入力論理マップ生成規則

## 《COBOL》

COBOL の可変項目の生成規則は、使用目的によってデータ型が変わります。

5. 論理マップ生成規則とマッピング規則

論理マップ可変部の集団項目化を指定した場合

• 数字項目の場合

03 マップ名-FIELDnnnn-H [OCCURS 回数].  
04 マップ名-FIELDnnnn-I PIC 編集文字.

• 文字項目の場合

03 マップ名-FIELDnnnn-H [OCCURS 回数].  
04 マップ名-FIELDnnnn-I PIC X(長さ).

• 漢字項目の場合

03 マップ名-FIELDnnnn-H [OCCURS 回数].  
04 マップ名-FIELDnnnn-I PIC N(長さ÷2).

論理マップ可変部の集団項目化を指定していない場合

• 数字項目の場合

02 マップ名-FIELDnnnn-I PIC 9(長さ) [OCCURS 回数].

• 文字項目の場合

02 マップ名-FIELDnnnn-I PIC X(長さ) [OCCURS 回数].

• 漢字項目の場合

02 マップ名-FIELDnnnn-I PIC N(長さ÷2) [OCCURS 回数].

《C 言語》

unsigned char マップ名\_FIELDnnnn\_I [[回数]] [長さ];

(b) マッピング規則

入力操作と入力項目データ名の領域の内容

《使用目的が日本語以外の場合》

入力操作	結果 (入力項目データ名の領域の内容)
半角のデータを入力	データを入力項目データ名の領域に代入する。
全角, または半角 / 全角のデータを入力	コードエラー通知文字に従ってコードエラー通知文字で入力項目データ名の領域をクリアする。コードエラーを検出したときのマッピング規則の詳細については、「5.3.13 コードエラーを検出したときのマッピング規則」を参照のこと。

入力操作	結果（入力項目データ名の領域の内容）
フィールドキーを押す （すべて(00) <sub>16</sub> のデータ 入力）	データ消去通知文字に従って入力項目データ名の領域をクリアする。 フィールドキーを押したときのマッピング規則については、「5.3.10 フィールドキーを押したときのマッピング規則」を参照のこと。
入力操作なし	（初期値を指定している） 初期値で指定した定数を入力項目データ名の領域に代入する。 （初期値を指定していない） 初期値クリア文字に従って入力データ名の領域をクリアする。入力操 作がなかったときのマッピング規則の詳細については、「5.3.12 入力 操作がなかったときのマッピング規則」を参照のこと。

## 注

カナのときだけ該当します。数字または英数字のときは、キー入力時点で入力できません。

## 《使用目的が日本語かつデータ型が漢字の場合》

入力操作	結果（入力項目データ名の領域の内容）
全角のデータを入力ま たは半角スペースを入力	データを入力項目データ名の領域に代入する。
フィールドキーを押す （すべて(00) <sub>16</sub> のデータ 入力）	データ消去通知文字に従って入力項目データ名の領域をクリアする。 フィールドキーを押したときのマッピング規則については、「5.3.10 フィールドキーを押したときのマッピング規則」を参照のこと。
入力操作なし	（初期値を指定している） 初期値で指定した定数を入力項目データ名の領域に代入する。 （初期値を指定していない） 初期値クリア文字に従って入力データ名の領域をクリアする。入力操 作がなかったときのマッピング規則の詳細については、「5.3.12 入力 操作がなかったときのマッピング規則」を参照のこと。

## 《使用目的が日本語かつデータ型が文字の場合》

入力操作	結果（入力項目データ名の領域の内容）
(00) <sub>16</sub> 以外のデータを入 力	データを入力項目データ名の領域に代入する。
フィールドキーを押す （すべて(00) <sub>16</sub> のデータ 入力）	データ消去通知文字に従って入力項目データ名の領域をクリアする。 フィールドキーを押したときのマッピング規則については、「5.3.10 フィールドキーを押したときのマッピング規則」を参照のこと。
入力操作なし	（初期値を指定している） 初期値で指定した定数を入力項目データ名の領域に代入する。 （初期値を指定していない） 初期値クリア文字に従って入力データ名の領域をクリアする。入力操 作がなかったときのマッピング規則の詳細については、「5.3.12 入力 操作がなかったときのマッピング規則」を参照のこと。

### 5.3.6 入力数字編集項目の論理マップ生成規則とマッピング規則

入力数字編集項目の論理マップ生成規則とマッピング規則について説明します。

#### (1) 入力数字編集項目の定義

入出力テキストボックス、または入出力フィールドで、データ型が数値、金額のフィールドに対して生成されます。

- 入力項目 : マップ名-FIELDnnnn-I  
数字編集文字列: データ型で指定できる数字編集文字列は次のとおり。「9」、「V」、「S」
- 繰り返し回数 : 縦、または横の反復回数 (入出力フィールドだけ)

#### (2) 論理マップ生成規則

《COBOL》

```
{02|03} マップ名-FIELDnnnn-H [OCCURS 回数].
{03|04} マップ名-FIELDnnnn-I PIC 編集文字.
```

#### (3) マッピング規則

入力操作と結果

入力操作	結果
フィールドキーを押した場合	データ消去通知文字に従って入力項目データ名の領域をクリアする。データ消去通知文字が埋字として指定してあるときは、0 ((30) <sub>16</sub> ) でクリアする。
入力操作がない、またはENDキーを押した場合	(初期値が指定してある場合) 初期値で指定した定数を入力項目データ名の領域に代入する。 (初期値が指定していない場合) 初期クリア文字が埋字として指定してあるときは、0 ((30) <sub>16</sub> ) でクリアする。
全桁に空白を入力した場合	空白 ((20) <sub>16</sub> ) で入力項目データ名の領域をクリアする。また、NULLと空白を混在している場合も、すべて空白となる。
数字項目として正しいデータを入力した場合	数字編集項目の指定に従って入力項目データ名の領域に値を代入する。
数字項目として不正なデータを入力した場合	入力チェックを指定した場合、入力時にエラーとなる。入力チェックを指定していない場合、コードエラー通知文字に従って入力項目データ名の領域をクリアする。コードエラー通知文字の標準はHIGH (X'FF') である。

注

不正なデータとは次のとおりです。



- 数字入力として不正な場合
  - コンマ, ペリオド, 符号, ¥ が連続している。
  - 数字の間に空白がある。
  - 数字 (0 ~ 9) が一つもない。
  - \* の直後が数字 (0 ~ 9) でない。
- ピクチャと一致しない場合
  - オーバフロー, アンダフロー。
  - V なしのピクチャに対し, 小数点がある。
  - S なしのピクチャに対し, マイナス符号がある。
- 「入力済み」属性を指定して, 数字編集文字以外や桁数が合わないデータを表示して画面確定している場合

### 5.3.7 制御項目の論理マップ生成規則とマッピング規則

制御項目の論理マップ生成規則とマッピング規則について説明します。制御項目は, 入出力テキストボックスダイアログ, または入出力フィールドダイアログで, 出力だけに関する制御項目は出力テキストボックスダイアログ, または出力フィールドダイアログで「動的変更 (AP から表示属性を変更する)」を選んだときに生成されます。

#### (1) 制御項目の定義

- 制御項目のデータ名: マップ名-FIELDnnnn-A  
データ名は, 各オブジェクトのダイアログの「データ名」で変更できます。
- 長さ: 2  
長さは, ドローセットアップの「運用管理者用の設定」の [ 修飾名の設定 ] ボタンをクリックして表示される, 修飾名の設定ダイアログで変更できます。変更するときには, 「動的変更の種別」から「キャラクタコントロール (キーエントリ / 選択エントリ)」を選択し, 「修飾名長」で長さを設定します。
- 繰り返し: フィールドの繰り返しと同じ

#### (2) 論理マップ生成規則

《COBOL》

```
[02|03] マップ名-FIELDnnnn-A PIC X(長さ) [OCCURS 回数] .
```

《C 言語》

```
unsigned char マップ名_FIELDnnnn_A [[回数]] [長さ];
```

### (3) マッピング規則

AP が制御項目データ名の領域に代入した内容と結果

《上記ダイアログで「動的変更 ( AP から表示属性を変更する )」を選んだ場合》

AP が制御項目データ名の領域に代入した内容	マッピングオプション	結果
修飾名と同じ	マージ	修飾名に対応する表示属性を使って項目を表示する。
先頭にデータ有無コード、または修飾名以外 ( 上記以外 )		標準の属性を使って項目を表示する。
修飾名と同じ	論理マップ	修飾名に対応する属性を使って項目を表示する。
先頭にデータ有無コード、または修飾名以外 ( 上記以外 )		( 表示形態が「全面書換」の場合 ) 標準の属性を使って表示する。 ( 表示形態が「一部上書」で直前に表示したマップ名と同じ場合 ) 直前の画面の表示属性を変更しないでデータだけを書き換える。 ( 表示形態が「一部上書」で直前に表示したマップ名と異なる場合 ) 標準の属性を使って表示する。
-	物理マップ	標準の属性を使って項目を表示する。

( 凡例 )

- : 該当しない。

注

修飾名と修飾名に対応したテキスト・フィールド変更属性情報は、ドロースेटアップの「表示属性の動的変更」の「キャラクタコントロール」タブで変更できます。

《上記ダイアログで「動的変更 ( AP から表示属性を変更する )」を選ばない場合》

標準の属性を使って項目を表示します。

## 5.3.8 入力バイト数格納項目の論理マップ生成規則とマッピング規則

入力バイト数格納項目の論理マップ生成規則とマッピング規則について説明します。ドロースेटアップの論理マップ属性で「入力データ長格納領域の生成」を選んだ場合対象になります。対象となるオブジェクトは、入出力テキスト ( フィールド ) になります。

### (1) 入力バイト数格納項目の定義

ドロースेटアップの論理マップ属性で展開方式の「入力データ長格納領域の生成」を選んだときに生成されます。

## (2) 論理マップ生成規則

## 《COBOL》

- Windows リトルエンディアン用, Linux 用

```
[02|03] マップ名-FIELDnnnn-L PIC S9(4) COMP-5 [OCCURS 回数] .
```

- Windows ビッグエンディアン用, AIX / HP-UX / Solaris 用

```
[02|03] マップ名-FIELDnnnn-L PIC S9(4) COMP [OCCURS 回数] .
```

## 《C 言語》

```
unsigned char マップ名_FIELDnnnn_L [[回数]] [2];
```

## (3) マッピング規則

入力操作と結果 (制御項目の内容)

《使用目的がカナ, 英数の場合》

入力操作	結果 (制御項目の内容)
半角のデータを入力	入力データ長を制御項目に代入する (埋字を含まない)。
全角または半角 / 全角混在のデータを入力, またはフィールドキーを押す。	(00) <sub>16</sub> を制御項目に代入する。
入力操作なし	入出力テキストボックス・フィールドダイアログで「入力済み (入力しなくても表示データを AP に返す)」を選び, 画面出力データが半角のときは画面上のデータ長を制御項目に代入する。それ以外は (00) <sub>16</sub> を代入する。

《使用目的が日本語でデータ型が漢字の場合》

入力操作	結果 (制御項目の内容)
フィールドキーを押す	(00) <sub>16</sub> を制御項目に代入する。
全角のデータを入力	入力データ長 (バイト数) を制御項目に代入する (埋字を含まない)。
入力操作なし	入出力テキストボックス・フィールドダイアログで「入力済み (入力しなくても表示データを AP に返す)」を選び, 画面出力データが全角のときは画面上のデータ長を制御項目に代入する。それ以外は (00) <sub>16</sub> を代入する。

## 5. 論理マップ生成規則とマッピング規則

### 《使用目的が日本語でデータ型が文字の場合》

入力操作	結果（制御項目の内容）
(00) <sub>16</sub> 以外のデータを入力	入力データ長を制御項目に代入する（埋字を含まない）。
フィールドキーを押す	(00) <sub>16</sub> を制御項目に代入する。
入力操作なし	入出力テキストボックス・フィールドダイアログで「入力済み（入力しなくても表示データを AP に返す）」を選んだ場合、画面上のデータ長を制御項目に代入する。それ以外は (00) <sub>16</sub> を代入する。

### 《使用目的が数字，数値，金額の場合》

入力操作	結果（制御項目の内容）
数字項目として正しいデータを入力	論理項目長を制御項目に代入する。
数字項目として不正なデータを入力，フィールドキーを押す，入力操作なし，または空白を入力。	(00) <sub>16</sub> を制御項目に代入する。

## 5.3.9 埋字と桁寄せに関するマッピング規則（出力フィールド）

出力テキストボックス，または出力フィールドの埋字と桁寄せに関するマッピング規則について説明します。

定義	結果（画面表示結果）
埋字に「埋めない」以外を指定	桁寄せ向きに従って桁寄せし，埋字に指定した文字で埋める。
埋字に「埋めない」を指定	初期値，および出力論理データは，桁寄せ向きに関係なく左寄せし，埋字を代入しない。ただし，埋字を代入しないことで画面上の次の項目位置はドロ画面で指定した位置に表示され，詰まることはない。表示形態が「全面書換」の場合，ヌルが入り，「一部上書」の場合，前画面の属性をそのまま引き継ぎデータだけを書き換える。

注

使用目的が日本語の場合，左寄せ固定になります。

## 5.3.10 フィールドキーを押したときのマッピング規則

入出力テキストボックス・フィールドに対して，フィールドキーを押したときのマッピング規則について説明します。フィールドキーに関するマッピング規則は，ドロセットアップの論理マップ属性にある「データ消去通知文字」と「初期クリア文字」の指定で結果が異なります。

また，次の操作をした場合，フィールドキーを押したときと同じ扱いになります。

- (00)<sub>16</sub> でクリアされていて、入出力テキストボックス・フィールドダイアログで「入力済み（入力しなくても表示データを AP に返す）」を指定しているフィールド
  - ウィンドウ表示制御の表示形態が「全面書換」でデータを何も表示しないフィールド
- 《「データ消去通知文字」が「なし」の場合》

初期クリア文字	各フィールドの初期値	結果（入出力項目データ名の領域の内容）
なし	あり	初期値を設定する。
	なし	マッピング制御機能は論理マップに何もセットしないため、マッピング制御機能に渡された時点の論理マップの内容のままになる。
埋字	あり	初期値を設定する。
	なし	各項目の埋字を設定する。
その他 (文字指定)	あり	初期値を設定する。
	なし	初期クリア文字を設定する。

《「データ消去通知文字」が「埋字」の場合》  
各項目の埋字を設定します。

《「データ消去通知文字」がその他の場合》  
データ消去通知文字を設定します。

### 5.3.11 データキーを押したときのマッピング規則

入出力テキストボックス・フィールドに対して、データキーを押したときのマッピング規則について説明します。データキーに関するマッピング規則は、ドロースेटアップの論理マップ属性にある「データ消去通知文字」と「初期クリア文字」の指定および表示・印刷セットアップの「キー操作」タブの「データキーの動作」の指定で結果が異なります。

《「データキーの動作」が「未入力状態」の場合》

初期クリア文字	各フィールドの初期値	結果（入出力項目データ名の領域の内容）
なし	あり	初期値を設定する。
	なし	マッピング制御機能は論理マップに何もセットしないため、マッピング制御機能に渡された時点の論理マップの内容のままになる。
埋字	あり	初期値を設定する。
	なし	各項目の埋字を設定する。
その他 (文字指定)	あり	初期値を設定する。
	なし	初期クリア文字を設定する。

## 5. 論理マップ生成規則とマッピング規則

### 《「データキーの動作」が「ヌルクリア状態」の場合》

各項目に対してフィールドキーを押したときと同じ扱いになります。フィールドキーを押したときのマッピング規則については、「5.3.10 フィールドキーを押したときのマッピング規則」を参照してください。

## 5.3.12 入力操作がなかったときのマッピング規則

入力操作がなかったときのマッピング規則について説明します。なお、入力操作をしていない場合でも、入力済み属性が有効となる画面を表示すると、表示データを入力した場合と同じ動作となります。

### 《「初期クリア文字」が「なし」の場合》

各フィールドの初期値	結果（入出力項目データ名の領域の内容）
あり	初期値を設定する。
なし	マッピング制御機能は論理マップに何もセットしないため、マッピング制御機能に渡された時点の論理マップの内容のままになる。

### 《「初期クリア文字」が「埋字」の場合》

各フィールドの初期値	結果（入出力項目データ名の領域の内容）
あり	初期値を設定する。
なし	各項目の埋字を設定する。

### 《「初期クリア文字」がその他（文字指定）の場合》

各フィールドの初期値	結果（入出力項目データ名の領域の内容）
あり	初期値を設定する。
なし	初期クリア文字を設定する。

## 5.3.13 コードエラーを検出したときのマッピング規則

コードエラーを検出したときのマッピング規則について説明します。コードエラーを検出したときのマッピング規則は、ドロースेटアップの論理マップ属性にある「初期クリア文字」と「エラー通知文字」の指定で結果が異なります。

### 《「初期クリア文字」が「なし」の場合》

エラー通知文字	結果（入出力項目データ名の領域の内容）
なし	マッピング制御機能は論理マップ中に何もセットしないため、マッピング制御機能に渡された時点の論理マップの内容のままになる。
埋字	各項目の埋字を設定する。
その他（文字指定）	エラー通知文字を設定する。

## 《「初期クリア文字」が「埋字」の場合》

エラー通知文字	結果（入出力項目データ名の領域の内容）
なし	各項目の埋字を設定する。
埋字	各項目の埋字を設定する。
その他（文字指定）	コードエラー通知文字を設定する。

## 《「初期クリア文字」がその他（文字指定）の場合》

コードエラー通知文字	結果（入出力項目データ名の領域の内容）
なし	初期クリア文字を設定する。
埋字	各項目の埋字を設定する。
その他（文字指定）	コードエラー通知文字を設定する。

### 5.3.14 埋字と桁寄せに関するマッピング規則（入力フィールド）

入出力テキストボックス・フィールドの埋字と桁寄せに関するマッピング規則について説明します。

定義	結果（入出力項目データ名の領域）
埋字に「埋めない」以外を指定	桁寄せ向きに従って桁寄せし，埋字に指定した文字で埋める。
埋字に「埋めない」を指定	初期値，および入力論理データは，桁寄せ向きに従って桁寄せし，埋字を代入しないで初期クリア文字が入る。

注

使用目的が日本語の場合，左寄せ固定になります。

### 5.3.15 MCR 入力のマッピング規則

MCR 入力のマッピング規則について説明します。論理マップ生成規則については，「5.3.5(3) 可変項目の入力論理マップ生成規則とマッピング規則」を参照してください。MCR は入出力テキストボックス・フィールドの使用目的を MCR にしておく必要があります。

## 5. 論理マップ生成規則とマッピング規則

### 《使用目的が MCR の場合》

入力操作	結果（入力項目データ名領域）
半角のカードデータを入力	データを論理項目に代入する。カードデータ長が論理項目長より短いときは桁寄せ向きに従って桁寄せし、埋字指定の内容で埋字を代入する。
入力操作なし、またはキー入力データを入力	初期値で指定した定数を論理項目に代入する。初期値が指定されていない場合、初期クリア文字に従って論理項目をクリアする。

#### 注

ID コードが「>」のカードデータの場合になります。「>」以外の ID コードを使った場合は、入力操作なしと同じ動作となります。また、「>」だけを入力した場合は、データ消去通知文字に従って論理項目をクリアします。

### 《使用目的が MCR 以外の場合》

可変項目の場合と同じです。「5.3.5(3) 可変項目の入力論理マップ生成規則とマッピング規則」を参照してください。

## 5.3.16 隠しフィールドの論理マップ生成規則とマッピング規則

### (1) 隠しフィールドの定義

隠しフィールドは、画面属性ダイアログで隠しフィールドを設定すると生成されます。

### (2) 論理マップ生成規則

隠しフィールドの論理マップ生成規則について説明します。

#### 《COBOL》

```
{02|03} マップ名-TRAN-I PIC X(9).
```

#### 《C 言語》

```
unsigned char マップ名_TRAN_I[9];
```

### (3) マッピング規則

#### 入力操作と結果

入力操作	結果
画面からの入力	隠しフィールドデータ名の領域に、画面定義時に設定した値が左寄せで入り、残りをスペースで埋字する。



### 5.3.17 確定キーの制御情報

#### (1) 制御情報の標準値

- 確定キーの制御データ名：マップ名-INCO  
データ名は、ドロースettingsアップの「ドロ-の設定」から表示する、表示属性の動的変更ダイアログの「データ名」タブで変更できます。
- 長さ：2  
長さは、ドロースettingsアップの「運用管理者用の設定」の[修飾名の設定...]ボタンをクリックして表示される、修飾名の設定ダイアログで変更できます。変更するときには、「動的変更の種別」から「確定キー属性」を選択し、「修飾名長」で長さを設定します。

#### (2) 論理マップ生成規則

確定キーの制御情報の論理マップ生成例について説明します。

《COBOL》

```
{02|03} マップ名-INCO PIC X(長さ).
```

《C 言語》

```
unsigned char マップ名_INCO[長さ];
```

#### (3) マッピング規則

確定キーの制御情報のマッピング規則について説明します。

AP が確定キーの制御情報のデータ名の領域に代入した内容とその結果

AP が確定キーの制御情報のデータ名の領域に代入した内容	結果
修飾名と同じ	確定キーの制御情報の修飾名に対応するウィンドウ制御情報を使って画面が表示される。ただし、標準属性が「キー押下無効」に設定されているキーは常にキー操作が無効となる。
データの先頭またはすべてがデータ有無コード、または修飾名以外（上記以外）	標準の確定キーの制御情報を使って画面が表示される。次に示す条件の場合、直前の表示画面と同じ画面のときは同じキー制御情報を使って表示し、直前の表示画面と異なるときは、標準のキー制御情報を使って表示する。 <ul style="list-style-type: none"> <li>画面属性ダイアログの表示形態が「一部上書」で、かつマッピングオプションで論理マップを指定した場合。</li> <li>画面属性ダイアログの表示形態が「自動」で、かつマッピングオプションで論理マップを指定した場合。</li> </ul>

注

修飾名と修飾名に対応した確定キー制御情報は、ドロースettingsアップの「表示属性の動的変更」の「確定キー属性」タブで変更できます。

### 5.3.18 下位項目の論理マップ生成規則とマッピング規則

#### (1) 下位項目の定義

下位項目は主にテキスト、フィールドおよびラベルに対して、データ型が「文字 (XX)」のときに指定します。各オブジェクトの定義ダイアログで設定すると、生成されます。

#### (2) 下位項目の出力論理マップ生成規則とマッピング規則

下位項目の出力論理マップ生成規則について説明します。

##### (a) 出力論理マップ生成規則

《COBOL》

- 数字項目の場合

```
04 マップ名-FIELDnnnn-nnn-0 PIC 9(長さ).
```

- 文字項目の場合

```
04 マップ名-FIELDnnnn-nnn-0 PIC X(長さ).
```

- 1文字の反復の場合

```
04 マップ名-FIELDnnnn-nnn-0 PIC X(1) OCCURS 回数.
```

《C 言語》

```
unsigned char マップ名_FIELDnnnn_nnn_0 [ [回数] ] [長さ] ;
```

##### (b) マッピング規則

出力結果

上位項目単位にマッピングします。下位項目を二つ定義し、二つ目の下位項目にデータ有無コード以外のデータを設定し、一つ目の下位項目にデータ有無コードを設定しても、埋字の対象にはなりません。

#### (3) 下位項目の入力論理マップ生成規則とマッピング規則

下位項目の入力論理マップ生成規則について説明します。

##### (a) 入力論理マップ生成規則

《COBOL》

- 数字項目の場合

```
04 マップ名-FIELDnnnn-nnn-I PIC 9(長さ).
```

- 文字項目の場合

04 マップ名-FIELDnnnn-*nnn*-I PIC X(*長さ*).

- 1文字の反復の場合

04 マップ名-FIELDnnnn-*nnn*-I PIC X(1) OCCURS *回数*.

《C言語》

unsigned char マップ名\_FIELDnnnn\_*nnn*\_I [ [*回数*] ] [*長さ*] ;

(b) マッピング規則

入力結果

上位項目単位にマッピングします。

## 5.4 可変部の論理マップ生成規則とマッピング規則 (GUI 画面固有)

ここでは、GUI 画面固有の論理マップ生成規則について説明します。なお、フィールドボックス内の可変項目の論理マップ生成規則については、「5.3 可変部の論理マップ生成規則とマッピング規則」を参照してください。なお、この節では、COBOLの数字項目の記述を次のようにしています。

- PIC 9 (長さ) と表記している項目  
実際に生成される論理マップは、PIC 99999 のように 9 が長さ分生成されます。

### 5.4.1 ポップアップメニューの論理マップ生成規則とマッピング規則

#### (1) ポップアップメニューの定義

ポップアップメニューには、データを入力するテキストボックスと入力するデータを表示するメニューがあります。テキストボックスをポップアップフィールドといい、表示するメニューをポップアップメニューといいます。

##### ポップアップフィールドの定義

可変ポップアップに対して生成されます。定義については、「5.3.5 可変項目の論理マップ生成規則とマッピング規則」、および「5.3.7 制御項目の論理マップ生成規則とマッピング規則」と同じです。

##### ポップアップメニューの定義

可変ポップアップの「メニュー」で指定した内容で生成されます。

- 可変用論理テーブル：マップ名・可変用論理テーブル名の指定値 (標準のデータ名：マップ名・POPUPnnnn-O)
- 回数：繰り返し回数の指定値
- 可変メニューラベル：マップ名・可変メニューラベル名の指定値 (標準のデータ名：マップ名・POPUP-LABELnnnn-O)
- データ型：データ型の指定値
- 長さ：ラベルのデータ長の指定値
- 可変メニューコード：マップ名・可変メニューコード名の指定値 (標準のデータ名：マップ名・POPUP-CODEnnnn-O)
- コードの長さ：コード項目のデータ長の指定値
- 可変メニュー選択ラベル：マップ名・可変メニュー選択ラベル名の指定値 (標準のデータ名：マップ名・POPUP-TEXTnnnn-O)
- 選択ラベルのデータ型：選択ラベルのデータ型の指定値
- 選択ラベルの長さ：選択ラベルのデータ長の指定値
- 可変メニューアクセスキー：マップ・可変メニューアクセスキー名の指定値 (標準

のデータ名：マップ名・POPUP-KEYnnnn-O)

- メニューデータファイル：マップ名・メニューデータファイル項目名の指定値（標準のデータ名：マップ名・POPUP-FILEnnnn-O)
- データ型：文字 (XX)
- 長さ：ファイル名指定：12  
フルパス 64：64  
フルパス 128：128  
フルパス 259：259

また、使用目的は次のとおりです。なお、可変メニューラベル、可変メニュー選択ラベルには、使用目的に関係なく、文字、漢字、数字から選択できます。

使用目的	使用できる項目
POP (手動・数字)	文字
POP (手動・カナ)	文字
POP (手動・英数)	文字
POP (手動・日本語)	文字、漢字
POP (自動・英数)	文字
POP (自動・日本語)	文字、漢字

## (2) ポップアップフィールドの論理マップ生成規則とマッピング規則

ポップアップフィールドで生成される論理マップには、項目属性、出力、入力、の三つの論理マップがあります。

### (a) ポップアップ項目属性の論理マップ生成規則とマッピング規則

ポップアップ項目属性の論理マップ生成規則とマッピング規則について説明します。

論理マップ生成規則

- 「動的変更 (AP から表示属性を変更する)」を指定した場合  
《COBOL》

```
[02|03] マップ名-FIELDnnnn-A PIC X(長さ).
```

《C 言語》

```
unsigned char マップ名_FIELDnnnn_A[長さ];
```

- 「動的変更 (AP から表示属性を変更する)」を指定しない場合  
定義に関係なく標準の属性を使って表示します。

5. 論理マップ生成規則とマッピング規則

マッピング規則

- AP が制御項目に代入した内容と表示結果

AP が制御項目に代入した内容	マッピングオプション	表示結果
ドロースेटアップで指定した修飾名と同じ修飾名を指定	マージ	ポップアップ項目の修飾名に対応する属性を使って表示する。
先頭またはすべてにデータ有無コード, または定義で指定していない修飾名を指定		標準の属性を使って表示する。
ドロースेटアップで指定した修飾名と同じ修飾名を指定	論理マップ	ポップアップ項目の修飾名に対応する属性を使って表示する。
先頭またはすべてにデータ有無コード, または定義で指定していない修飾名を指定		画面属性の表示形態によって次のように変わる。 <ul style="list-style-type: none"> <li>• 「全面書換」の場合 標準の属性を使って表示する。</li> <li>• 「一部上書」で直前に表示したマップと同じ場合 前回の属性のまま表示する。</li> <li>• 「一部上書」で直前に表示したマップと異なる場合 標準の属性を使って表示する。</li> </ul>
-	物理マップ	標準の属性を使って表示する。

(凡例)

- : 該当しない。

(b) ポップアップ項目の出力論理マップ生成規則とマッピング規則

ポップアップ項目の出力論理マップ生成規則とマッピング規則について説明します。

出力論理マップ生成規則

《COBOL》

```
{02|03} マップ名-FIELDnnnn-0 PIC X(長さ) [OCCURS 回数].
```

《C 言語》

```
unsigned char マップ名_FIELDnnnn_0 [[回数]] [長さ];
```

マッピング規則

- AP が論理項目に代入した内容と表示結果

《初期値を指定しない場合》

AP が論理項目に代入した内容	マッピングオプション	表示結果
すべてデータ	マージ,	代入されたデータを表示する。
すべて (00) <sub>16</sub> のデータ	論理マップ	ヌルクリアして表示する。

AP が論理項目に代入した内容	マッピングオプション	表示結果
すべて (20) <sub>16</sub> のデータ		空白で表示する。
先頭 1 文字がデータ有無コード		データ有無コード (1F) <sub>16</sub> を設定した場合、例として X'1F414234' または X'1F1F1F' のようなデータが該当する。このとき、画面属性の表示形態によって次のように変わる。 <ul style="list-style-type: none"> <li>・「全面書換」の場合 ヌルクリアして表示する。</li> <li>・「一部上書」で直前に表示したマップと同じ場合 前回のデータのまま表示する。</li> <li>・「一部上書」で直前に表示したマップと異なる場合 ヌルクリアして表示する。</li> </ul>
データの後半にデータ有無コード		データ有無コード (1F) <sub>16</sub> を仮定した場合、例として X'4142431F1F' のようなデータが該当する。論理項目の桁寄せ向きと、指定した埋字の内容に従って表示する。
画面の表示長より長いデータ		論理項目の桁寄せ向きに従って桁寄せし、余りを切り捨てて表示する。
-	物理マップ	何も表示しない。

( 凡例 )

- : 該当しない。

《初期値を指定した場合》

AP が論理項目に代入した内容	マッピングオプション	表示結果
先頭 1 文字がデータ有無コード	マージ, 論理マップ	《先頭 1 文字がデータ有無コードのときの結果》を参照のこと。
すべてのデータ		代入されたデータを表示する。
データの後半にデータ有無コード		データ有無コード (1F) <sub>16</sub> を仮定した場合、例として X'4142431F1F' のようなデータが該当する。論理項目の向きと、指定した埋字の内容に従って表示する。
-	物理マップ	《先頭 1 文字がデータ有無コードのときの結果》を参照のこと。

( 凡例 )

- : 該当しない。

《先頭 1 文字がデータ有無コードのときの結果》

初期値の内容	結果
すべて空白 ( (20) <sub>16</sub> )	項目全体を空白で表示する。
すべて 0 ( (30) <sub>16</sub> )	項目全体を 0 で表示する。

## 5. 論理マップ生成規則とマッピング規則

初期値の内容	結果
すべてヌル (00) <sub>16</sub> )	項目全体をヌルクリアして表示する。
繰り返し文字指定	項目全体を繰り返し文字で表示する。
初期値の長さ画面の表示長が等しい	初期値を項目に表示する。
初期値の長さが画面の表示長より短い	定義で指定した「桁寄せ」、および埋字に従って表示する。埋字なしを指定した場合は、左寄せで初期値を項目に表示する。
初期値の長さが画面の表示長より長い	「桁寄せ」に従って桁寄せし、余りを切り捨てて項目を表示する。

### (c) ポップアップ項目の入力論理マップ生成規則とマッピング規則

ポップアップ項目の入力論理マップ生成規則とマッピング規則について説明します。

#### 入力論理マップ生成規則

##### 《COBOL》

- 論理マップ可変部の集団項目化を指定した場合

```
03 マップ名-FIELDnnnn-H [OCCURS 回数] .
04 マップ名-FIELDnnnn-I PIC {X(長さ)|N(長さ÷2)}.
```

- 論理マップ可変部の集団項目化を指定しない場合

```
02 マップ名-FIELDnnnn-I PIC {X(長さ)|N(長さ÷2)} [OCCURS 回数] .
```

##### 《C言語》

```
unsigned char マップ名_FIELDnnnn_I [[回数]] [長さ];
```

#### マッピング規則

- 入力操作と結果（論理項目の内容）  
《自動ポップアップの場合》

入力操作	結果（論理項目の内容）
ポップアップ項目にカーソルを位置づけ表示されたポップアップメニューの中からメニューを選ぶ	選んだメニュー（フォーカスを位置づけたラベル）に対応した定数を論理項目に代入する。
ポップアップ項目にカーソルを位置づけ表示されたポップアップメニューの中からメニューを選ばない、またはポップアップ項目にカーソルを位置づけけない	<ul style="list-style-type: none"> <li>論理項目の初期値を指定している場合、指定した定数を論理項目に代入する。</li> <li>論理項目の初期値を指定していない場合、初期クリア文字の指定に従って論理項目をクリアする。</li> </ul>



入力操作	結果（論理項目の内容）
メニューのアクセスキー項目に指定されているキーを入力	<ul style="list-style-type: none"> <li>アクセスキーに対応した通知コードを、論理項目に代入する。</li> </ul>
メニューのアクセスキー項目に指定されていないキーを入力	<ul style="list-style-type: none"> <li>論理項目の初期値を指定している場合、初期値を論理項目に代入する。</li> <li>論理項目の初期値を指定していない場合、初期クリア文字の指定に従って論理項目をクリアする。</li> </ul>

## 注

「メニューデータをファイルで指定する」を指定している場合、使用目的が日本語以外で、かつ通知コードが全角を含む場合には、エラー通知文字が返ります。

## 《手動ポップアップの場合》

入力操作	結果（論理項目の内容）
ポップアップ項目にカーソルを位置づけ表示されたポップアップメニューの中からメニューを選ぶ	選んだメニュー（フォーカスを位置づけたラベル）に対応した定数を論理項目に代入する <sup>1</sup> 。
ポップアップ項目にカーソルを位置づけ表示されたポップアップメニューの中からメニューを選ばない	<ul style="list-style-type: none"> <li>論理項目の初期値を指定している場合、指定した定数を論理項目に代入する。</li> <li>論理項目の初期値を指定していない場合、初期クリア文字の指定に従って論理項目をクリアする。</li> </ul>
半角のデータを入力 <sup>2</sup>	データを論理項目に代入する。
全角、または半角 / 全角混在のデータを入力 <sup>2</sup>	
フィールドキーを押す（すべて (00) <sub>16</sub> ）データ入力 <sup>2</sup>	データ消去通知文字に従って論理項目をクリアする。
ポップアップ項目にカーソルを位置づけない	<ul style="list-style-type: none"> <li>論理項目の初期値を指定している場合、指定した定数を論理項目に代入する。</li> <li>論理項目の初期値を指定していない場合、初期クリア文字の指定に従って論理項目をクリアする。</li> </ul>
メニューのアクセスキー項目に指定されているキーを入力	アクセスキーに対応した通知コードを、論理項目に代入する <sup>1</sup> 。
メニューのアクセスキー項目に指定されていないキーを入力	<ul style="list-style-type: none"> <li>論理項目の初期値を指定している場合、初期値を論理項目に代入する。</li> <li>論理項目の初期値を指定していない場合、初期クリア文字の指定に従って論理項目をクリアする。</li> </ul>

## 注 1

「メニューデータをファイルで指定する」を指定している場合、使用目的が日本語以外で、かつ通知コードが全角を含む場合には、エラー通知文字が返ります。

## 注 2

ポップアップ項目にカーソルを位置づけてデータを入力する場合は該当します。

(3) ポップアップメニューの論理マップ生成規則とマッピング規則

(a) メニュー属性のマッピング規則

標準の属性を使ってメニューを表示します。

(b) ポップアップメニューラベルの論理マップ生成規則とマッピング規則

ポップアップメニューラベルの論理マップ生成規則とマッピング規則について説明します。

ラベルの論理マップ生成規則

- 可変用論理テーブル名を指定した場合  
《COBOL》

```

{02|03} マップ名-POPUPnnnn-0 [OCCURS 回数].
{03|04} マップ名-POPUP-CODEnnnn-0 PIC {X(長さ)|N(長さ÷2)}.
{03|04} マップ名-POPUP-LABELnnnn-0 PIC {X(長さ)|N(長さ÷2)|編集文字}.
{03|04} マップ名-POPUP-KEYnnnn-0 PIC X(長さ). .....※1
{03|04} マップ名-POPUP-TEXTnnnn-0 PIC {X(長さ)|N(長さ÷2)|編集文字}. ...※2
    
```

《C 言語》

```

struct {
    unsigned char マップ名_POPUP_CODEnnnn-0[長さ];
    unsigned char マップ名_POPUP_LABELnnnn-0[長さ];
    unsigned char マップ名_POPUP_KEYnnnn-0[長さ]; .....※1
    unsigned char マップ名_POPUP_TEXTnnnn-0[長さ]; .....※2
} マップ名_POPUPnnnn_0 [[回数]] ;
    
```

注 1  
アクセスキーを指定すると展開されます。

注 2  
選択ラベルと通知コードを同じ値にしない場合に展開されます。

マッピング規則

- AP が論理項目に代入した内容と表示結果  
《ラベル項目を指定した場合》

AP が論理項目に代入した内容	マッピングオプション	表示結果
すべてデータ	マージ,	代入されたデータを表示する。
すべて (00) <sub>16</sub> <sup>1</sup>	論理マップ	ヌルクリアして表示する。
すべて (20) <sub>16</sub> <sup>1</sup>		空白で表示する。

AP が論理項目に代入した内容	マッピングオプション	表示結果
先頭 1 文字がデータ有無コード <sup>2</sup>		データ有無コード (1F) <sub>16</sub> を設定した場合、例として X'1F414234' または X'1F1F1F' のようなデータが該当する。このとき、画面属性の表示形態によって次のように変わる。 <ul style="list-style-type: none"> <li>「全面書換」の場合 コード項目のデータがあるときは、コード項目のデータを表示する。コード項目のデータがないときは、何も表示しない。</li> <li>「一部上書」の場合 前回のデータのまま表示する。</li> </ul>
データの後半にデータ有無コード		データ有無コード (1F) <sub>16</sub> を仮定した場合、例として X'4142431F1F' のようなデータが該当する。指定したラベル項目の桁寄せ向き、および埋字の内容に従って表示する。
画面の表示長より長いデータ		ラベル項目の桁寄せ向きに従って桁寄せし、余りを切り捨てて表示する。
-	物理マップ	画面属性の表示形態によって次のように変わる。 <ul style="list-style-type: none"> <li>「全面書換」の場合 何も表示しない。</li> <li>「一部上書」の場合 前回のデータのまま表示する。</li> </ul>

(凡例)

- : 該当しない。

注 1

メニューの後半に該当する項目がある場合は、項目は圧縮されます。

注 2

ラベル項目のデータ型とコード項目のデータ型が同じ場合だけ、ラベル項目にデータ有無コードを設定できます。

《コード項目を指定した場合》

AP が論理項目に代入した内容	マッピングオプション	表示結果
すべてデータ	マージ,	代入されたデータをコードとして扱う。
すべて (00) <sub>16</sub>	論理マップ	ヌルクリアをコードとして扱う。
すべて (20) <sub>16</sub>		空白をコードとして扱う。
先頭 1 文字がデータ有無コード		データ有無コード (1F) <sub>16</sub> を設定した場合、例として X'1F414234' または X'1F1F1F' のようなデータが該当する。このとき、画面属性の表示形態によって次のように変わる。 <ul style="list-style-type: none"> <li>「全面書換」の場合 コードなし、とする。</li> <li>「一部上書」の場合 前回のコードをそのまま扱う。</li> </ul>

5. 論理マップ生成規則とマッピング規則

AP が論理項目に代入した内容	マッピングオプション	表示結果
データの後半にデータ有無コード		データ有無コード (1F) <sub>16</sub> を仮定した場合、例として X'4142431F1F' のようなデータが該当する。指定されたデータをコードとして扱う。
-	物理マップ	画面属性の表示形態によって次のように変わる。 <ul style="list-style-type: none"> <li>・「全面書換」の場合 何も表示しない。</li> <li>・「一部上書」の場合 前回のコードをそのまま扱う。</li> </ul>

(凡例)

- : 該当しない。

注

選択されたメニュー項目に選択ラベルの指定がない場合、対応するテキストまたはフィールドの桁寄せ、および埋字されたデータが入力時に論理項目に代入されます。選択ラベルの指定がある場合は、コードの桁寄せ、および埋字されたデータが入力時に論理項目に代入されます。ドローで定義した場合、対応するテキストまたはフィールドとコードの桁寄せ、および埋字は同じになります。

《選択ラベル項目を指定した場合》

AP が論理項目に代入した内容	マッピングオプション	表示結果
すべてデータ	マージ,	代入されたデータを表示する。
すべて (00) <sub>16</sub>	論理マップ	ヌルクリアして表示する。
すべて (20) <sub>16</sub>		空白で表示する。
先頭 1 文字がデータ有無コード		データ有無コード (1F) <sub>16</sub> を設定した場合、例として X'1F414234' または X'1F1F1F' のようなデータが該当する。このとき、画面属性の表示形態によって次のように変わる。 <ul style="list-style-type: none"> <li>・「全面書換」の場合 コード項目のデータがあるときは、コード項目のデータを表示する。コード項目のデータがないときは、何も表示しない。</li> <li>・「一部上書」の場合 前回のデータのまま表示する。</li> </ul>
データの後半にデータ有無コード		データ有無コード (1F) <sub>16</sub> を仮定した場合、例として X'4142431F1F' のようなデータが該当する。指定された選択ラベル項目の桁寄せ向き、および埋字の内容に従って表示する。
画面の表示長より長いデータ		選択ラベル項目の桁寄せ向きに従って桁寄せし、余りを切り捨てて表示する。
-	物理マップ	画面属性の表示形態によって次のように変わる。 <ul style="list-style-type: none"> <li>・「全面書換」の場合 何も表示しない。</li> <li>・「一部上書」の場合 前回のデータのまま表示する。</li> </ul>

(凡例)

- : 該当しない。

注

選択ラベル項目のデータ型と、コード項目のデータ型が同じ場合だけ、選択ラベル項目にデータ有無コードを設定できます。

《アクセスキーを指定した場合》

AP が論理項目に代入した内容	マッピングオプション	表示結果
すべてデータ	マージ, 論理マップ	<ul style="list-style-type: none"> <li>指定されたデータがすべて半角英数の場合、指定されたデータがアクセスキーとして有効となる。</li> <li>指定されたデータに半角英数以外の文字が含まれる場合、指定された項目のアクセスキーは無効となる。</li> </ul>
先頭 1 文字がデータ有無コード		<p>データ有無コード (1F)<sub>16</sub> を設定した場合、例として X'1F414234' または X'1F1F1F' のようなデータが該当する。このとき、画面属性の表示形態によって次のように変わる。</p> <ul style="list-style-type: none"> <li>「全面書換」の場合 対応するアクセスキーなし、となる。</li> <li>「一部上書」の場合 前回のアクセスキーがそのまま有効となる。</li> </ul>
データの後半にデータ有無コード		<p>データ有無コード (1F)<sub>16</sub> を仮定した場合、例として X'4142431F1F' のようなデータが該当する。指定された項目のアクセスキーは無効となる。</p>
-	物理マップ	<p>画面属性の表示形態によって次のように変わる。</p> <ul style="list-style-type: none"> <li>「全面書換」の場合 対応するアクセスキーなし、となる。</li> <li>「一部上書」の場合 前回のアクセスキーがそのまま有効となる。</li> </ul>

(凡例)

- : 該当しない。

(c) ポップアップメニューファイル使用時の論理マップ生成規則とマッピング規則

ポップアップメニューファイルを使用したときの、論理マップ生成規則とマッピング規則について説明します。

出力論理マップ生成規則

《COBOL》

[02|03] マップ名-POPUP-FILEnnnn-0 PIC X(長さ).

《C 言語》

unsigned char マップ名\_ POPUP\_FILEnnnn-0[長さ];

## 5. 論理マップ生成規則とマッピング規則

### マッピング規則

- AP が論理項目に代入した内容と表示結果

AP が論理項目に代入した内容	マッピングオプション	表示結果
すべてデータ	マージ, 論理マップ	(指定されたファイルが存在した場合) 指定されたファイルを読み込み、メニューを表示する。 (指定されたファイルが存在しない場合) メニューを表示しない。
先頭 1 文字がデータ有無コード		データ有無コード (1F) <sub>16</sub> を設定した場合、例として X'1F414234' または X'1F1F1F' のようなデータが該当する。このとき、画面属性の表示形態によって次のように変わる。 <ul style="list-style-type: none"> <li>• 「全面書換」の場合 メニューを表示しない。</li> <li>• 「一部上書」の場合 前回のメニューデータをそのまま表示する。</li> </ul>
データの後半にデータ有無コード		データ有無コード (1F) <sub>16</sub> を仮定した場合、例として X'4142431F1F' のようなデータが該当する。 (指定されたファイルが存在した場合) 指定されたファイルを読み込み、メニューを表示する。 (指定されたファイルが存在しない場合) メニューを表示しない。

#### 注

- 1 階層の空のメニューを表示します。

### 入力論理マップ生成規則

#### 《COBOL》

- 論理マップ可変部の集団項目化を指定した場合

```
03 マップ名-FIELDnnnn-H [OCCURS 回数].
04 マップ名-FIELDnnnn-I PIC {X(長さ)|N(長さ÷2)}.
```

- 論理マップ可変部の集団項目化を指定しない場合

```
02 マップ名-FIELDnnnn-I PIC {X(長さ)|N(長さ÷2)} [OCCURS 回数].
```

#### 《C 言語》

```
unsigned char マップ名_FIELDnnnn_I [[回数]] [長さ];
```

### 入力論理マップのマッピング規則

- 入力操作と結果 (論理項目の内容)  
ポップアップメニューと同様です。詳細については、「5.4.1(2)(c) ポップアップ

項目の入力論理マップ生成規則とマッピング規則」を参照してください。

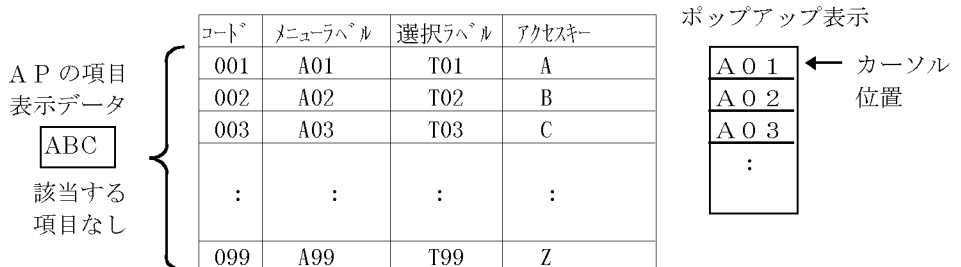
#### (d) ポップアップメニューの表示

ポップアップメニューの表示に関するマッピング規則について説明します。項目の表示データは対応するポップアップメニューコードの値によって表示結果が異なります。

メニューコード内に項目表示データの値があり、項目長とメニューコード長が等しい先頭のメニューから表示し、項目表示データと同じメニューコードに対応するメニューに、カーソルを位置づけます。



メニューコード内に項目表示データの値がないか、項目長とメニューコード長が等しくない先頭メニューを表示し、カーソルを位置づけます。



## 5.4.2 メニューバーの論理マップ生成規則

### (1) メニュー制御項目の定義

メニュー定義ダイアログで「動的変更 (AP から表示属性を変更する)」を選び、制御項目名を指定することで生成されます。

- メニュー制御項目のデータ名：マップ名・MENUUnnnn・A (標準値)  
データ名は、画面属性ダイアログの「メニューバー定義」から表示する、メニューバー定義ダイアログの「制御項目データ名」で変更できます。
- 長さ：2

## 5. 論理マップ生成規則とマッピング規則

長さは、ドローセットアップの「運用管理者用の設定」の[修飾名の設定]ボタンをクリックして表示される、修飾名の設定ダイアログで変更できます。変更するときには、「動的変更の種別」から「候補選択コントロール/コマンドコントロール」を選択し、「修飾名長」で長さを設定します。

### (2) 論理マップ生成規則

《COBOL》

```
{02|03} マップ名-MENUUnnnn-A PIC X(長さ).
```

《C 言語》

```
unsigned char マップ名_MENUUnnnn_A[長さ];
```

## 5.4.3 プッシュボタンボックスの論理マップ生成規則とマッピング規則

プッシュボタンボックスの論理マップ生成規則とマッピング規則について説明します。

### (1) プッシュボタンの定義

プッシュボタンの属性

プッシュボタンダイアログの「動的変更 (AP から表示属性を変更する)」を選ぶと生成されます。

- ボタン制御項目のデータ名：マップ名-BUTTONnnnn-A

データ名は、プッシュボタンダイアログの AP が渡す項目の「データ名」に従います。

- 長さ：2

長さは、ドローセットアップの「運用管理者用の設定」の[修飾名の設定]ボタンをクリックして表示される、修飾名の設定ダイアログで変更できます。変更するときには、「動的変更の種別」から「候補選択コントロール/コマンドコントロール」を選択し、「修飾名長」で長さを設定します。

プッシュボタンのラベルテキスト

プッシュボタンダイアログの「AP からテキストを変更する」を選ぶと生成されます。

- ラベルテキスト項目のデータ名：マップ名-BUTTONnnnn-O

データ名は、プッシュボタンダイアログの AP が渡す項目の「データ名」で変更できます。

- 長さ：ラベルテキストのデータ長

### (2) プッシュボタン属性の論理マップ生成規則とマッピング規則

プッシュボタン属性の論理マップ生成例とマッピング規則について説明します。



## (a) 論理マップ生成規則

《COBOL》

{02 03} マップ名-BUTTONnnnn-A PIC X(長さ).
--------------------------------------

《C 言語》

unsigned char マップ名_BUTTONnnnn_A[長さ];
--------------------------------------

## (b) マッピング規則

AP が制御項目に代入した内容と表示結果

「動的変更 (AP から表示属性を変更する)」を指定していない場合は、標準の属性を使って表示します。

AP が制御項目に代入した内容	マッピングオプション	表示結果
セットアップで指定してある修飾名と同じ	マージ	プッシュボタンのボタン属性の修飾名に対応する属性を使って表示する。
先頭にデータ有無コード、または設定していない修飾名		標準の属性を使って表示する。
セットアップで指定してある修飾名と同じ	論理マップ	プッシュボタンのボタン属性の修飾名に対応する属性を使って表示する。
先頭にデータ有無コード、または設定していない修飾名		(表示形態が「全面書換」のとき) 標準の属性を使って表示する。 (表示形態が「一部上書」で直前に表示したマップと同じ場合) 前回の属性のまま表示する。 (表示形態が「一部上書」で直前に表示したマップと異なる場合) 標準の属性を使って表示する。
-	物理マップ	標準の属性を使って表示する。

(凡例)

- : 該当しない。

注

修飾名と修飾名に対応したボタン属性情報は、ドロースセットアップの「表示属性の動的変更」の「コマンドコントロール」タブで変更できます。

## (3) プッシュボタンのラベルテキストの論理マップ生成規則とマッピング規則

プッシュボタンのラベルテキストの論理マップ生成規則とマッピング規則について説明します。

## 5. 論理マップ生成規則とマッピング規則

### (a) 論理マップ生成規則

《COBOL》

```
{02|03} マップ名-BUTTONnnnn-0 PIC X(長さ).
```

《C 言語》

```
unsigned char マップ名_BUTTONnnnn_0[長さ];
```

### (b) マッピング規則

AP が論理項目に代入した内容と表示結果

「AP からテキストを変更する」を指定していない場合は、テキストに指定したデータを表示します。テキストに文字列を指定していない場合は、空白を表示します。

AP が論理項目に代入した内容	マッピング オプション	表示結果
すべてデータ	マージ,	代入されたデータを表示する。
先頭 1 文字がデータ有無コード	論理マップ	データ有無コード (1F) <sub>16</sub> を設定した場合、例として X'1F414234' または X'1F1F1F' のようなデータが該当する。この場合、テキストに指定したデータを表示する。テキストに文字列を指定していない場合、空白を表示する。
データの後半にデータ有無コード		データ有無コード (1F) <sub>16</sub> を仮定した場合、例として X'4142431F1F' のようなデータが該当する。この場合、データ有無コード以前のデータを表示する。
画面の表示長より長いデータ		代入されたデータを左寄せで桁寄せし、余りを切り捨てて表示する。
-	物理マップ	テキストに指定したデータを表示する。テキストに文字列を指定していない場合、空白を表示する。

(凡例)

- : 該当しない。

注

ターゲットが文字コード EUC (AIX, Solaris, Linux) 用、メインフレーム - PC 分散用の場合が該当します。

### (4) プッシュボタンの入力論理マップ生成規則とマッピング規則

プッシュボタンの入力論理マップ生成規則とマッピング規則について説明します。

## (a) 入力論理マップ生成規則

《COBOL》

{02 03} マップ名-INCI PIC X(長さ).
------------------------------

《C 言語》

unsigned char マップ名_INCI [長さ];
-------------------------------

## (b) マッピング規則

入力操作と結果（制御項目の内容）

入力操作	結果（制御項目の内容）
ボタンボックスの中からボタンを選ぶ	選んだボタンに対応するイベント通知コードを制御項目に代入する。
ボタンボックスの中からボタンを選ばない (イベント通知用のキーを押した場合)	イベント通知コードを指定したキーによって画面送信をしたときに、対応するイベント通知コードを制御項目に代入する。

#### 5.4.4 ラジオボタンボックスの論理マップ生成規則とマッピング規則

ラジオボタンボックスの論理マップ生成規則とマッピング規則について説明します。

## (1) ラジオボタンボックスの定義

ラジオボタンの属性

固定 / 可変ラジオボタンダイアログで「動的変更（AP から表示属性を変更する）」を選ぶことで生成されます。

- 固定ラジオボタンの属性
  - 固定ボタン制御項目のデータ名：マップ名-RADIOnnnn-A  
データ名は、固定ラジオボタンダイアログの「制御項目データ名」で変更できます。
  - 長さ：2  
長さは、ドローセットアップの「運用管理者用の設定」の [ 修飾名の設定 ] ボタンをクリックして表示される、修飾名の設定ダイアログで変更できます。変更するときには、「動的変更の種別」から「候補選択コントロール / コマンドコントロール」を選択し、「修飾名長」で長さを設定します。
- 可変ラジオボタンの属性
  - 可変ボタン制御項目のデータ名：マップ名-FIELDnnnn-A  
データ名は、可変ラジオボタンボックスダイアログの AP が受け取る項目の「データ名」に従います。

## 5. 論理マップ生成規則とマッピング規則

- 長さ：2

長さは、ドローセットアップの「運用管理者用の設定」の [ 修飾名の設定 ] ボタンをクリックして表示される、修飾名の設定ダイアログで変更できます。変更するときには、「動的変更の種別」から「候補選択コントロール/コマンドコントロール」を選択し、「修飾名長」で長さを設定します。

### 可変ラジオボタン用論理テーブル

- 可変用論理テーブルのデータ名：マップ名-RADIOnnnn-O
- 回数：ボタンの数の指定値
- 可変ボタンラベル項目のデータ名：マップ名-RADIO-LABELnnnn-O  
データ名は、可変ラジオボタンボックスダイアログのラベルの「データ名」で変更できます。
- データ型：ラベルのデータ型で指定
- 長さ：ラベルのデータ長で指定
- 可変ボタンコード項目のデータ名：マップ名-RADIO-CODEnnnn-O  
データ名は、可変ラジオボタンボックスダイアログの通知コードの「データ名」で変更できます。
- 長さ：2  
長さは、可変ラジオボタンボックスダイアログの AP が受け取る項目の「データ長」に従います。

### ラジオボタンボックス

この項目は可変/固定ラジオボタンボックス共通です。

- ボックス単位通知項目のデータ名：マップ名-FIELDnnnn-I  
データ名は、ラジオボタンボックスダイアログの AP が受け取る項目の「データ名」で変更できます。
- 長さ：AP が受け取る項目のデータ長で指定

## (2) ラジオボタンボックス属性の論理マップ生成規則とマッピング規則

ラジオボタンボックス属性の論理マップ生成規則とマッピング規則について説明します。この項目は、ダイアログで「動的変更 ( AP から表示属性を変更する )」を指定したときだけ論理マップが生成されます。

### (a) 論理マップ生成規則

《COBOL》

```
{02|03} マップ名-RADIOnnnn-A PIC X(長さ) [OCCURS 回数] .
```

《C 言語》

```
unsigned char マップ名_RADIOnnnn_A [[回数]] [長さ];
```

## (b) マッピング規則

AP が制御項目に代入した内容と表示結果

《「動的変更 (AP から表示属性を変更する)」を指定した場合》

AP が制御項目に代入した内容	マッピングオプション	表示結果
セットアップで指定してある修飾名と同じ	マージ	ラジオボタンの修飾名に対応する属性を使って表示する。
先頭にデータ有無コード、または設定していない修飾名		標準の属性を使って表示する。
セットアップで指定してある修飾名と同じ	論理マップ	ラジオボタンの修飾名に対応する属性を使って表示する。
先頭にデータ有無コード、または設定していない修飾名		(表示形態が「全面書換」のとき) 標準の属性を使って表示する。 (表示形態が「一部上書」で直前に表示したマップと同じ場合) 前回の属性のまま表示する。 (表示形態が「一部上書」で直前に表示したマップと異なる場合) 標準の属性を使って表示する。
-	物理マップ	標準の属性を使って表示する。

(凡例)

- : 該当しない。

注

修飾名と修飾名に対応したボタン属性情報は、ドローセットアップの「表示属性の動的変更」の「候補選択コントロール」タブで変更できます。

《「動的変更 (AP から表示属性を変更する)」を指定しない場合》

標準の属性を使って表示します。

## (3) ラジオボタンボックス可変用論理テーブルの論理マップ生成規則とマッピング規則

ラジオボタンボックス可変用論理テーブルの論理マップ生成規則とマッピング規則について説明します。

## (a) 論理マップ生成規則

《COBOL》

```
{02|03} マップ名-RADIOnnnn-0 [OCCURS 回数].
{03|04} マップ名-RADIO-LABELnnnn-0 PIC {9(長さ)|X(長さ)|N(長さ÷2)|編集文字}.
{03|04} マップ名-RADIO-CODEnnnn-0 PIC {9(長さ)|X(長さ)|N(長さ÷2)|編集文字}.
```

## 5. 論理マップ生成規則とマッピング規則

### 《C 言語》

```
struct {
    unsigned char マップ名_RADIO_LABELnnnn_0[長さ];
    unsigned char マップ名_RADIO_CODEnnnn_0 [長さ];
} マップ名-RADIOnnnn_0 [[回数]] ;
```

### (b) マッピング規則

AP が論理項目に代入した内容と表示結果

《ラベル項目を指定した場合》

AP が論理項目に代入した内容	マッピング オプション	表示結果
すべてデータ	マージ,	代入されたデータを表示する。
すべて (00) <sub>16</sub>	論理マップ	ヌルクリアして表示する。
すべて (20) <sub>16</sub>		空白で表示する。
先頭 1 文字がデータ有無コード		データ有無コード (1F) <sub>16</sub> を設定した場合、例として X'1F414234' または X'1F1F1F' のようなデータが該当する。 (コード項目のデータがある) コード項目のデータを表示する。 (コード項目のデータがない) ボタン図形だけでラベルは表示しない。
データの後半にデータ有無コード		データ有無コード (1F) <sub>16</sub> を仮定した場合、例として X'4142431F1F' のようなデータが該当する。 ラベル項目の桁寄せ向きに従って桁寄せし、埋字指定の内容で埋字をして表示する。
画面の表示長より長いデータ		ラベル項目の桁寄せ向きに従って桁寄せし、余りを切り捨てて表示する。
-	物理マップ	何も表示しない。

### (凡例)

- : 該当しない。

### 注

ラベル項目のデータ型とコード項目のデータ型が同じ場合だけ、ラベル項目にデータ有無コードを設定できます。

## (4) ラジオボタンの入力論理マップ生成規則とマッピング規則

ラジオボタンの入力論理マップ生成例とマッピング規則について説明します。

## (a) 入力論理マップ生成規則

## 《COBOL》

- 論理マップ可変部の集団項目化を指定した場合

```
03 マップ名-FIELDnnnn-H.
04 マップ名-FIELDnnnn-I PIC {X(長さ)|N(長さ÷2)|9(長さ)}.
```

- 論理マップ可変部の集団項目化を指定しない場合

```
02 マップ名-FIELDnnnn-I PIC {X(長さ)|N(長さ÷2)|9(長さ)}.
```

## 《C 言語》

```
unsigned char マップ名_FIELDnnnn_I[長さ];
```

## (b) マッピング規則

入力操作と結果（論理項目の内容）

入力操作	結果（論理項目の内容）
ボタンボックス中からボタンを選ぶ	選んだボタンに対応する定数を論理項目に設定する。
ボタンボックス中からボタンを選ばない	<p>（固定ラジオボタンの場合）</p> <p>ボタン属性情報で「選択済みにする」を指定したボタンに対応する定数を論理項目に設定する。複数のボタン属性情報に「選択済みにする」を指定した場合、最初に指定したラジオボタンに対応する定数を論理項目に設定する。</p> <p>「選択済みにする」を指定しなかった場合、初期クリア文字に従って論理項目をクリアする。</p> <p>（可変ラジオボタンの場合）</p> <p>初期クリア文字に従って論理項目をクリアする。</p>

### 5.4.5 チェックボタンボックスの論理マップ生成規則とマッピング規則

チェックボタンボックスのマッピング規則は、基本的にラジオボタンボックスと同じです。マッピング規則の詳細については、「5.4.4 ラジオボタンボックスの論理マップ生成規則とマッピング規則」を参照してください。

ここでは、チェックボタンの定義、論理マップ生成規則、およびラジオボタンと異なるマッピング規則について説明します。

#### (1) チェックボタンボックスの定義

チェックボタンの属性

固定 / 可変チェックボタンダイアログで「動的変更（AP から表示属性を変更する）」を選ぶことで生成されます。

## 5. 論理マップ生成規則とマッピング規則

- ボタン制御項目のデータ名：マップ名-FIELDnnnn-A  
データ名は、チェックボタンダイアログの AP が受け取る項目の「データ名」に従います。
- 長さ：2  
長さは、ドローセットアップの「運用管理者用の設定」の [ 修飾名の設定 ] ボタンをクリックして表示される、修飾名の設定ダイアログで変更できます。変更するときには、「動的変更の種別」から「候補選択コントロール/コマンドコントロール」を選択し、「修飾名長」で長さを設定します。

### 可変チェックボタン用論理テーブル

- 可変用論理テーブルのデータ名：マップ名-CHECKnnnn-O
- 回数：ボタンの数の指定値
- 可変ボタンラベル項目のデータ名：マップ名-CHECK-LABELnnnn-O  
データ名は、可変チェックボタンボックスダイアログのラベルの「データ名」で変更できます。
- データ型：ラベルのデータ型で指定
- 長さ：ラベルのデータ長で指定
- 可変ボタンコード項目のデータ名：マップ名-CHECK-CODEnnnn-O  
データ名は、可変チェックボタンボックスダイアログの通知コードの「データ名」で変更できます。
- 長さ：2  
長さは、可変チェックボタンボックスダイアログの AP が受け取る項目の「データ長」に従います。

### チェックボタンボックス

この項目は可変 / 固定チェックボタンボックス共通です。

- ボックス単位通知項目のデータ名：マップ名-FIELDnnnn-I  
データ名は、チェックボタンダイアログの AP が受け取る項目の「データ名」で変更できます。
- 長さ：AP が受け取る項目のデータ長で指定

## (2) チェックボタンボックス属性の論理マップ生成規則とマッピング規則

チェックボタンボックス属性の論理マップ生成規則とマッピング規則について説明します。この項目は、ダイアログで「動的変更 ( AP から表示属性を変更する )」を指定したときだけ論理マップが生成されます。

### (a) 論理マップ生成規則

《COBOL》

{02 03} マップ名-FIELDnnnn-A PIC X(長さ) [OCCURS 回数] .
--



## 《C 言語》

```
unsigned char マップ名_FIELDnnnn_A [[回数]] [長さ];
```

## (b) マッピング規則

AP が制御項目に代入した内容と表示結果

ラジオボタンボックスと同様です。詳細については、「5.4.4(2)(b) マッピング規則」を参照してください。

## (3) チェックボタンボックス可変用論理テーブルの論理マップ生成規則とマッピング規則

チェックボタンボックス可変用論理テーブルの論理マップ生成規則とマッピング規則について説明します。

## (a) 論理マップ生成規則

## 《COBOL》

```
{02|03} マップ名-CHECKnnnn-0 [OCCURS 回数].
{03|04} マップ名-CHECK-LABELnnnn-0 PIC {9(長さ)|X(長さ)|N(長さ÷2)|編集文字}.
{03|04} マップ名-CHECK-CODEnnnn-0 PIC {9(長さ)|X(長さ)|N(長さ÷2)|編集文字}.
```

## 《C 言語》

```
struct {
    unsigned char マップ名_CHECK_LABELnnnn_0[長さ];
    unsigned char マップ名_CHECK_CODEnnnn_0[長さ];
} マップ名-CHECKnnnn_0 [[回数]];
```

## (b) マッピング規則

AP が論理項目に代入した内容と表示結果

ラジオボタンボックスと同様です。詳細については、「5.4.4(3)(b) マッピング規則」を参照してください。

## (4) チェックボタンの入力論理マップ生成規則とマッピング規則

チェックボタンの入力論理マップ生成例とマッピング規則について説明します。

## 5. 論理マップ生成規則とマッピング規則

### (a) 入力論理マップ生成規則

#### 《COBOL》

- 論理マップ可変部の集団項目化を指定した場合

```
03 マップ名-FIELDnnnn-H [OCCURS 回数] .
04 マップ名-FIELDnnnn-I PIC {X(長さ)|N(長さ÷2)|9(長さ)}.
```

- 論理マップ可変部の集団項目化を指定しない場合

```
02 マップ名-FIELDnnnn-I PIC {X(長さ)|N(長さ÷2)|9(長さ)} [OCCURS 回数] .
```

#### 《C 言語》

```
unsigned char マップ名_FIELDnnnn_I [[回数]] [長さ];
```

### (b) マッピング規則

#### 入力操作と結果（論理項目の内容）

入力操作	結果（論理項目の内容）
ボタンボックス中からボタンを選ぶ	選んだボタンに対応する定数を論理項目に代入する。複数のボタンを選んだ場合、選んだすべてのボタンに対応する定数を論理項目に代入する。
ボタンボックス中からボタンを選ばない、またはボタンボックスにフォーカスを位置づけない	<p>（固定チェックボタンの場合）</p> <p>ボタン属性情報で「フォーカス設定（フォーカスを本ボタンに位置づける）」を指定したボタンに対応する定数を論理項目に代入する。複数のボタン属性情報に初期選択を指定した場合、選んだすべてのボタンに対応する定数を論理項目に代入する。ボタン属性情報を指定しなかった場合、初期値で指定した定数を論理項目に代入する。初期値を指定しなかった場合、初期クリア文字に従って論理項目をクリアする。</p> <p>（可変ラジオボタンの場合）</p> <p>初期クリア文字に従って論理項目をクリアする。</p>

## 5.4.6 リストボックスの論理マップ生成規則とマッピング規則

リストボックスの論理マップ生成規則とマッピング規則について説明します。

### (1) リストボックスの定義

リストボックスの属性

リストボックスダイアログで「動的変更（AP から表示属性を変更する）」を選ぶことで生成されます。

- リスト制御項目のデータ名：マップ名 -FIELDnnnn-A  
データ名は、リストボックスダイアログの AP が受け取る項目の「データ名」に従

います。

- 長さ：2

長さは、ドローセットアップの「運用管理者用の設定」の [ 修飾名の設定 ] ボタンをクリックして表示される、修飾名の設定ダイアログで変更できます。変更するときには、「動的変更の種別」から「候補選択コントロール/コマンドコントロール」を選択し、「修飾名長」で長さを設定します。

#### 論理テーブル

- 可変用論理テーブルのデータ名：マップ名-LISTnnnn-O
- 回数：指定したリスト項目数
- 可変リストラベル項目のデータ名：マップ名-LIST-LABELnnnn-O  
データ名は、リストボックスダイアログのラベルの「データ名」で変更できます。
- データ型：ラベルのデータ型で指定
- 長さ：ラベルのデータ長で指定
- 可変リストコード項目のデータ名：マップ名-LIST-CODEnnnn-O  
データ名は、リストボックスダイアログの通知コードの「データ名」で変更できます。
- 長さ：2  
長さは、リストボックスダイアログの AP が受け取る項目の「データ長」に従います。

#### リストボックス

- ボックス単位通知項目のデータ名：マップ名-FIELDnnnn-I  
データ名は、リストボックスダイアログの AP が受け取る項目の「データ名」で変更できます。
- 長さ：AP が受け取る項目のデータ長で指定

## (2) リストボックスのボックス属性の論理マップ生成規則とマッピング規則

リストボックスのボックス属性の論理マップ生成規則とマッピング規則について説明します。この項目は、ダイアログで「動的変更 ( AP から表示属性を変更する )」を指定したときだけ論理マップが生成されます。

### (a) 論理マップ生成規則

《COBOL》

```
{02|03} マップ名-LISTnnnn-A PIC X(長さ) [OCCURS 回数] .
```

《C 言語》

```
unsigned char マップ名_LISTnnnn_A [[回数]] [長さ];
```

## 5. 論理マップ生成規則とマッピング規則

### (b) マッピング規則

AP が制御項目に代入した内容と表示結果

《「動的変更 (AP から表示属性を変更する)」を指定した場合》

AP が制御項目に代入した内容	マッピングオプション	表示結果
セットアップで指定してある修飾名と同じ	マージ	リストボックスの修飾名に対応する属性を使って表示する。
先頭にデータ有無コード, または設定していない修飾名		標準の属性を使って表示する。
セットアップで指定してある修飾名と同じ	論理マップ	リストボックスの修飾名に対応する属性を使って表示する。
先頭にデータ有無コード, または設定していない修飾名		(表示形態が「全面書換」のとき) 標準の属性を使って表示する。
		(表示形態が「一部上書」で直前に表示したマップと同じ場合) 前回の属性のまま表示する。
-	物理マップ	標準の属性を使って表示する。

(凡例)

- : 該当しない。

注

修飾名と修飾名に対応したリスト属性情報は, ドローセットアップの「表示属性の動的変更」の「候補選択コントロール」タブで変更できます。

《「動的変更 (AP から表示属性を変更する)」を指定しない場合》

標準の属性を使って表示します。

### (3) リストボックスの論理テーブルの論理マップ生成規則とマッピング規則

リストボックスの論理テーブルの論理マップ生成規則とマッピング規則について説明します。

#### (a) 論理マップ生成規則

《COBOL》

```
{02|03} マップ名-LISTnnnn-0 [OCCURS 回数].
{03|04} マップ名-LIST-LABELnnnn-0 PIC {9(長さ)|X(長さ)|N(長さ÷2)|編集文字}.
{03|04} マップ名-LIST-CODEnnnn-0 PIC {9(長さ)|X(長さ)|N(長さ÷2)|編集文字}.
```

## 《C 言語》

```

struct {
    unsigned char マップ名_LIST_LABELnnnn_0[長さ];
    unsigned char マップ名_LIST_CODEnnnn_0[長さ];
} マップ名-LISTnnnn_0 [[回数]];

```

## (b) マッピング規則

AP が論理項目に代入した内容と表示結果

《ラベル項目を指定した場合》

AP が論理項目に代入した内容	マッピング オプション	表示結果
すべてデータ	マージ, 論理マップ	代入されたデータを表示する。
すべて (00) <sub>16</sub> <sup>1</sup>		ヌルクリアして表示する。
すべて (20) <sub>16</sub> <sup>1</sup>		空白で表示する。
先頭 1 文字がデータ有無コード <sup>2</sup>		データ有無コード (1F) <sub>16</sub> を設定した場合、例として X'1F414234' または X'1F1F1F' のようなデータが該当する。 (コード項目の指定がある) コード項目のデータを表示する。 (コード項目の指定がない) 何も表示しない。
データの後半にデータ有無コード		データ有無コード (1F) <sub>16</sub> を仮定した場合、例として X'4142431F1F' のようなデータが該当する。テキスト項目の桁寄せ向きに従って桁寄せし、埋字指定の内容で埋字をして表示する。
画面の表示長より長いデータ		テキスト項目の桁寄せ向きに従って桁寄せし、余りを切り捨てて表示する。
-	物理マップ	何も表示しない。

## ( 凡例 )

- : 該当しない。

## 注 1

リストの後半に該当する項目がある場合、項目は圧縮されます。

## 注 2

ラベル項目のデータ型とコード項目のデータ型が同じ場合だけ、ラベル項目にデータ有無コードを設定できます。

## (4) リストボックスの入力論理マップ生成規則とマッピング規則

リストボックスの入力論理マップ生成例とマッピング規則について説明します。

## 5. 論理マップ生成規則とマッピング規則

### (a) 入力論理マップ生成規則

#### 《COBOL》

- 論理マップ可変部の集団項目化を指定した場合

```
03 マップ名-FIELDnnnn-H [OCCURS 回数] .
04 マップ名-FIELDnnnn-I PIC {X(長さ)|N(長さ÷2)|9(長さ)}.
```

- 論理マップ可変部の集団項目化を指定しない場合

```
02 マップ名-FIELDnnnn-I PIC {X(長さ)|N(長さ÷2)|9(長さ)} [OCCURS 回数] .
```

#### 《C 言語》

```
unsigned char マップ名_FIELDnnnn_I [[回数]] [長さ];
```

### (b) マッピング規則

#### 入力操作と結果（論理項目の内容）

入力操作	結果（論理項目の内容）
リストボックス中からリストを選ぶ	選んだリストに対応する定数を論理項目に代入する。単一選択リストボックスは最後に選択したリストを、複数選択リストボックスで複数のリストを選んだ場合、選択したすべてのリストに対応する定数を論理項目に代入する。
リストボックスの中からリストを選ばない、またはリストボックスにフォーカスを位置づけない	リスト属性情報で初期選択を指定したリストに対応する定数を論理項目に代入する。ただし、単一選択リストボックスは最初に選択したリストを、複数選択リストボックスで複数のリスト属性情報に初期選択を指定した場合は指定したすべてのリストに対応する定数を論理項目に代入する。リスト属性を指定しなかった場合、論理項目の初期値で指定した定数を論理項目に代入する。 論理項目の初期値を指定しなかった場合、ドロースettingsアップの論理マップ属性で指定した初期クリア文字に従って論理項目を初期化する。

## 5.4.7 コンボボックスの論理マップ生成規則とマッピング規則

コンボボックスの論理マップ生成規則とマッピング規則について説明します。

### (1) コンボボックスの定義

コンボボックスは、選択されたデータや入力されたデータを表示するボックスと、入力するデータの候補を表示するメニューから成ります。

コンボボックスの属性

コンボボックスダイアログで「動的変更（AP から表示属性を変更する）」を選び、属性項目データ名を指定することで生成されます。

- 制御項目のデータ名：マップ名-FIELDnnnn-A  
データ名は、コンボボックスダイアログの AP が受け取る通知コードの項目の「データ名」に従います。
- 長さ：2  
長さは、ドローセットアップの「運用管理者用の設定」の [ 修飾名の設定 ] ボタンをクリックして表示される、修飾名の設定ダイアログで変更できます。変更するときには、「動的変更の種別」から「キャラクタコントロール ( キーエントリ / 選択エントリ )」を選択し、「修飾名長」で長さを設定します。

#### ボックスの定義

固定コンボボックス、および可変コンボボックスに対して生成されます。

- ボックス項目のデータ名：マップ名-FIELDnnnn-I/O  
データ名は、コンボボックスダイアログの AP が受け取る通知コードの項目の「データ名」で変更できます。
- 長さ：AP が受け取る項目のデータ長で指定

#### 可変コンボボックス用論理テーブルの定義

可変メニューダイアログで指定した内容で生成されます。

- 可変用論理テーブルのデータ名：マップ名-POPUPnnnn-O
- 回数：メニューの項目数の指定値
- 可変メニューラベル項目のデータ名：マップ名-POPUP-LABELnnnn-O  
データ名は、可変コンボメニューダイアログのラベルの「データ名」で変更できません。
- データ型：ラベルのデータ型で指定
- 長さ：コンボボックスの長さの指定値-2
- 可変メニューコード項目のデータ名：マップ名-POPUP-CODEnnnn-O  
データ名は、可変コンボメニューダイアログの AP に返す通知コードの「データ名」で変更できます。
- 長さ：2  
長さは、コンボボックスダイアログの AP が受け取る通知コードの項目の「データ長」に従います。

## (2) コンボボックスの論理マップ生成規則とマッピング規則

### (a) 項目属性の論理マップ生成規則とマッピング規則

コンボボックスの項目属性に関する論理マップ生成規則とマッピング規則について説明します。この論理マップは、コンボボックスダイアログで「動的変更 ( AP から表示属性を変更する )」を選び、属性項目データ名を指定することで生成されます。

#### 論理マップ生成規則

《COBOL》

{02 03} マップ名-FIELDnnnn-A PIC X(長さ).
-------------------------------------

## 5. 論理マップ生成規則とマッピング規則

### 《C 言語》

```
unsigned char マップ名_FIELDnnnn_A[長さ];
```

#### マッピング規則

- AP が制御項目に代入した内容と表示結果

《「動的変更（AP から表示属性を変更する）」を指定した場合》

AP が制御項目に代入した内容	マッピングオプション	表示結果
修飾名と同じ	マージ	コンボボックスの修飾名に対応する属性を使って表示する。
先頭にデータ有無コード，または修飾名以外（上記以外）		標準の属性を使って表示する。
修飾名と同じ	論理マップ	コンボボックスの修飾名に対応する属性を使って表示する。
先頭にデータ有無コード，または修飾名以外（上記以外）		（表示形態が「全面書換」の場合） 標準の属性を使って表示する。 （表示形態が「一部上書」で直前に表示したマップと同じ場合） 前回の属性をそのまま表示する。 （表示形態が「一部上書」で直前に表示したマップと異なる場合） 標準の属性を使って表示する。
-	物理マップ	標準の属性を使って表示する。

#### （凡例）

- : 該当しない。

#### 注

修飾名と修飾名に対応したテキスト・フィールド変更属性情報は，ドロースेटアップの「表示属性の動的変更」の「キャラクタコントロール」タブで変更できます。

《「動的変更（AP から表示属性を変更する）」を指定しない場合》

標準の属性を使って表示します。

#### （b）出力論理マップ生成規則とマッピング規則

コンボボックスの出力論理マップ生成規則とマッピング規則について説明します。

#### 出力論理マップ生成規則

### 《COBOL》

```
{02|03} マップ名-FIELDnnnn-0 PIC X(長さ).
```



## 《C 言語》

unsigned char マップ名_FIELDnnnn_0[長さ];
-------------------------------------

## マッピング規則

- AP が論理項目に代入した内容と表示結果

《入力形式がキー入力可能で、初期値を指定しない場合》

AP が論理項目に 代入した内容	マッピング オプション	表示結果
すべてデータ	マージ, 論理マップ	(コード項目に同一のデータが指定されている場合) 対応するラベル項目のデータを表示する。 (コード項目に同一のデータが指定されていない場 合) 代入されたデータを表示する。
すべて (00) <sub>16</sub> のデータ		ヌルクリアして表示する。
すべて (20) <sub>16</sub> のデータ		空白で表示する。
先頭 1 文字がデータ有無コード		データ有無コード (1F) <sub>16</sub> を設定した場合、例として '1F414234' または X'1F1F1F' のようなデータが該当 する。 (表示形態が「全面書換」の場合) 何も表示しない。 (表示形態が「一部上書」で前回と同じマップの場 合) 直前のデータのまま表示する。 (表示形態が「一部上書」で前回と異なるマップの場 合) 何も表示しない。
データの後半にデータ有無コー ド		データ有無コード (1F) <sub>16</sub> を仮定した場合、例として X'4142431F1F' のようなデータが該当する。 (データ有無コードを切り捨てたデータとメニューの コード項目に同一のデータが指定されている場合) 対応するラベル項目のデータを表示する。 (データ有無コードを切り捨てたデータとメニューの コード項目に同一のデータが指定されていない場合) 論理項目の桁寄せ向きに従って桁寄せする。ま た、埋字指定の内容に従って埋字をする。
画面の表示長より長いデータ		(コード項目に同一のデータが指定されている場合) 対応するラベル項目のデータを表示する。 (コード項目に同一のデータが指定されていない場 合) 論理項目の桁寄せ向きに従って桁寄せし、余り を切り捨てて表示する。
-	物理マップ	何も表示しない。

(凡例)

- : 該当しない。

5. 論理マップ生成規則とマッピング規則

《入力形式がキー入力可能で、初期値を指定する場合》

AP が論理項目に 代入した内容	マッピング オプション	表示結果
先頭 1 文字がデータ有無コード	マージ, 論理マップ	《先頭 1 文字がデータ有無コードのときの結果》を参照のこと。
すべてデータ		代入されたデータを表示する。
データの後半にデータ有無コード		データ有無コード (1F) <sub>16</sub> を仮定した場合、例として X'4142431F1F' のようなデータが該当する。論理項目の向きと、指定した埋字の内容に従って表示する。
-	物理マップ	初期値を指定しない場合と同じになる。

(凡例)

- : 該当しない。

《先頭 1 文字がデータ有無コードのときの結果》

初期値の内容	結果
すべて空白 ((20) <sub>16</sub> )	項目全体を空白で表示する。
すべて 0 ((30) <sub>16</sub> )	項目全体を 0 で表示する。
すべてヌル ((00) <sub>16</sub> )	項目全体をヌルクリアして表示する。
繰り返し文字指定	項目全体を繰り返し文字で表示する。
初期値の長さ画面の表示長が等しい	初期値を項目に表示する。
初期値の長さが画面の表示長より短い	定義で指定した「桁寄せ」、および埋字に従って表示する。埋字なしを指定した場合は、左寄せで初期値を項目に表示する。
初期値の長さが画面の表示長より長い	「桁寄せ」に従って桁寄せし、余りを切り捨てて項目を表示する。

## 《入力形式がメニュー選択だけで、初期値を指定しない場合》

AP が論理項目に 代入した内容	マッピング オプション	表示結果
すべてデータ	マージ, 論理マップ	(コード項目に同一のデータが指定されている場合) 対応するラベル項目のデータを表示する。 (コード項目に同一のデータが指定されていない場合) 先頭のラベル項目のデータを表示する。
先頭 1 文字がデータ有無コード		データ有無コード (1F) <sub>16</sub> を設定した場合、例として X'1F414234' または X'1F1F1F' のようなデータが該当する。 (表示形態が「全面書換」の場合) 先頭のラベル項目のデータを表示する。 (表示形態が「一部上書」で前回と同じマップの場合) 直前のデータのまま表示する。 (表示形態が「一部上書」で前回と異なるマップの場合) 先頭のラベル項目のデータを表示する。
データの後半にデータ有無コード		データ有無コード (1F) <sub>16</sub> を仮定した場合、例として X'4142431F1F' のようなデータが該当する。 (データ有無コードを切り捨てたデータとメニューのコード項目に同一のデータが指定されている場合) 対応するラベル項目のデータを表示する。 (データ有無コードを切り捨てたデータとメニューのコード項目に同一のデータが指定されていない場合) 先頭のラベル項目のデータを表示する。
画面の表示長より長いデータ		(コード項目に同一のデータが指定されている場合) 対応するラベル項目のデータを表示する。 (コード項目に同一のデータが指定されていない場合) 先頭のラベル項目のデータを表示する。
-		物理マップ

(凡例)

- : 該当しない。

## 《入力形式がメニュー選択だけで、初期値を指定する場合》

AP が論理項目に 代入した内容	マッピング オプション	表示結果
先頭 1 文字がデータ有無コード	マージ, 論理マップ	《先頭 1 文字がデータ有無コードのときの結果》を参照のこと。
すべてデータ		(コード項目に同一のデータが指定されている場合) 対応するラベル項目のデータを表示する。 (コード項目に同一のデータが指定されていない場合) 先頭のラベル項目のデータを表示する。

5. 論理マップ生成規則とマッピング規則

AP が論理項目に 代入した内容	マッピング オプション	表示結果
データの後半にデータ有無コード		データ有無コード (1F) <sub>16</sub> を仮定した場合、例として X'4142431F1F' のようなデータが該当する。 (データ有無コードを切り捨てたデータとメニューのコード項目に同一のデータが指定されている場合) 対応するラベル項目のデータを表示する。 (データ有無コードを切り捨てたデータとメニューのコード項目に同一のデータが指定されていない場合) 先頭のラベル項目のデータを表示する。
-	物理マップ	《先頭 1 文字がデータ有無コードのときの結果》を参照のこと。

(凡例)

- : 該当しない。

《先頭 1 文字がデータ有無コードのときの結果》

初期値の内容	結果
メニューのコード項目に指定されているデータと同じ	対応するラベル項目のデータを表示する。
メニューのコード項目に指定されているデータと異なる	先頭のラベル項目のデータを表示する。

(c) 入力論理マップ生成規則とマッピング規則

コンボボックスの入力論理マップ生成規則とマッピング規則について説明します。

論理マップ生成規則

《COBOL》

- 論理マップ可変部の集団項目化を指定した場合

03 マップ名-FIELDnnnn-H. 04 マップ名-FIELDnnnn-I PIC {X(長さ) N(長さ÷2) 9(長さ)}.
--

- 論理マップ可変部の集団項目化を指定しない場合

02 マップ名-FIELDnnnn-I PIC {X(長さ) N(長さ÷2) 9(長さ)}.
--

《C 言語》

unsigned char マップ名_FIELDnnnn_I[長さ];
-------------------------------------

## マッピング規則

- 入力操作と結果（論理項目の内容）

## 《入力形式がキー入力可能の場合》

入力操作	結果（論理項目の内容）
コンボボックスにカーソルを位置づけ、表示された候補の中からメニューを選択 <sup>1</sup> する	選んだメニュー（フォーカスを位置づけたラベル）に対応したコードを論理項目に代入する。
コンボボックスにカーソルを位置づけ表示された候補からメニューを選択しない、またはコンボボックスにカーソルを位置づけない	論理項目の初期値で指定したコードを論理項目に代入する。論理項目の初期値が指定されていない場合、初期クリア文字の指定に従って論理項目をクリアする。
半角のデータを入力 <sup>2</sup>	データを論理項目に代入する。
全角、または半角/全角混在のデータを入力 <sup>2</sup>	
フィールドキーを押す（すべて(00) <sub>16</sub> ）データ入力 <sup>2</sup>	データ消去通知文字に従って論理項目をクリアする。

## 注 1

選択とは、カーソルでメニューを選んで、復改キーを押す操作です。この操作で、コードが論理項目に代入されます。

## 注 2

コンボボックスにカーソルを位置つけてデータを入力する場合は該当します。

## 《入力形式がメニュー選択だけの場合》

入力操作	結果（論理項目の内容）
コンボボックスにカーソルを位置づけ、表示された候補の中からメニューを選択する	選んだメニュー（フォーカスを位置づけたラベル）に対応したコードを論理項目に代入する。
コンボボックスにカーソルを位置づけ表示された候補からメニューを選択しない、またはコンボボックスにカーソルを位置づけない	論理項目の初期値で指定したコードを論理項目に代入する。論理項目の初期値が指定されていない場合、初期クリア文字の指定に従って論理項目をクリアする。

## 注

選択とは、カーソルでメニューを選んで、スペースキーまたは復改キーを押す操作です。この操作で、コードが論理項目に代入されます。

## (3) コンボボックスのメニューの論理マップ生成規則とマッピング規則

## (a) メニュー属性のマッピング規則

メニュー属性のマッピング規則は、標準の属性を使ってメニューを表示します。

## 5. 論理マップ生成規則とマッピング規則

### (b) ラベルの論理マップ生成規則とマッピング規則

可変用論理テーブル名を指定した場合のコンボボックスの論理マップ生成規則とマッピング規則について説明します。

論理マップ生成例（入力形式がメニュー選択だけの場合）

《COBOL》

```
03 マップ名-POPUPnnnn-0 OCCURS 回数.
04 マップ名-POPUP-LABELnnnn-0 PIC {9(長さ)|X(長さ)|N(長さ÷2)|編集文字}.
04 マップ名-POPUP-CODEEnnnn-0 PIC X(長さ).
```

《C 言語》

```
struct {
    unsigned char マップ名_POPUP_LABELnnnn_0[長さ];
    unsigned char マップ名_POPUP_CODEEnnnn_0[長さ];
} マップ名_POPUPnnnn_0[回数];
```

論理マップ生成例（入力形式がキー入力可能な場合）

《COBOL》

```
03 マップ名-POPUPnnnn-0 OCCURS 回数.
04 マップ名-POPUP-CODEEnnnn-0 PIC {X(長さ)|N(長さ÷2)|9(長さ)}.
```

《C 言語》

```
struct {
    unsigned char マップ名_POPUP_CODEEnnnn_0[長さ];
} マップ名_POPUPnnnn_0[回数];
```

マッピング規則

- AP が論理項目に代入した内容と表示結果

《入力形式がメニュー選択だけの場合》

AP が論理項目に代入した内容	マッピング オプション	表示結果
すべてデータ	マージ, 論理マップ	代入されたデータを表示する。
すべて (00) <sub>16</sub> <sup>1</sup>		ヌルクリアして表示する。
すべて (20) <sub>16</sub> <sup>1</sup>		空白で表示する。

AP が論理項目に代入した内容	マッピングオプション	表示結果
先頭 1 文字がデータ有無コード <sup>2</sup>		データ有無コード (1F) <sub>16</sub> を設定した場合、例として X'1F414234' または X'1F1F1F' のようなデータが該当する。 (コード項目の指定がある場合) コード項目のデータを表示する。 (コード項目の指定がない場合) 何も表示しない。
データの後半にデータ有無コード		データ有無コード (1F) <sub>16</sub> を仮定した場合、例として X'4142431F1F' のようなデータが該当する。 指定したラベル項目の向き、および埋字の内容に従って表示する。
画面の表示長より長いデータ		ラベル項目の桁寄せ向きに従って桁寄せし、余りを切り捨てて表示する。
-	物理マップ	何も表示しない。

(凡例)

- : 該当しない。

注 1

リストの後半に該当する項目がある場合、項目は圧縮されます。

注 2

ラベル項目のデータ型とコード項目のデータ型が同じ場合だけ、ラベル項目にデータ有無コードを設定できます。

《入力形式がキー入力可能の場合》

マッピングオプション	表示結果
マージ, 論理マップ	コード項目のデータを表示する。
物理マップ	何も表示しない。

### (c) コンボボックスのメニューの表示

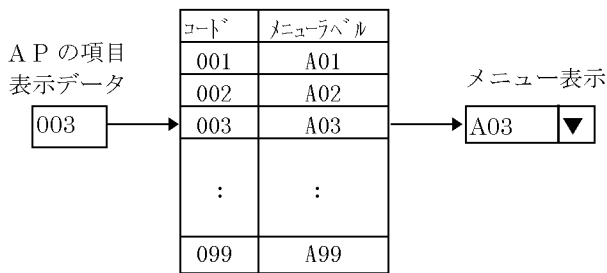
コンボボックスのメニューの表示に関するマッピング規則について説明します。項目の表示データに対応するコンボボックスのメニューコードの値によって表示結果が異なります。

入力形式がメニュー選択だけの場合

《メニューコード内に項目表示データの値があり、項目長とメニューコード長が等しい》

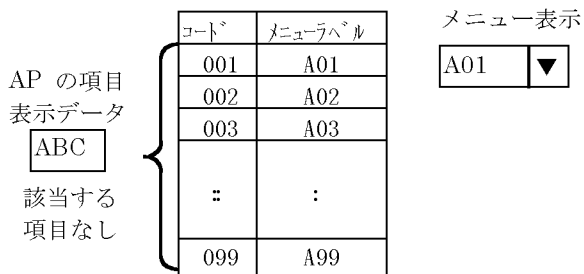
先頭のメニューから表示し、項目表示データと同じメニューコードに対応するメニューに、カーソルを位置づけます。

5. 論理マップ生成規則とマッピング規則



《メニューコード内に項目表示データの値がないか、項目長とメニューコード長が等しくない》

先頭メニューを表示し、カーソルを位置づけます。



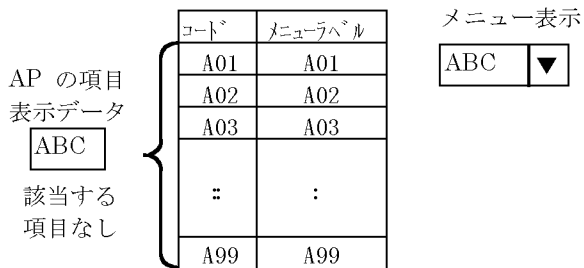
入力形式がキー入力可能な場合

《メニューコード内にキー入力したデータの値があり、項目長とメニューコード長が等しい》

入力形式がメニュー選択だけの場合と同じです。キー入力の場合、「AP の項目表示データ」は「キー入力したデータ」となります。

《メニューコード内にキー入力したデータの値がないか、項目長とメニューコード長が等しくない》

キー入力したデータを表示します。





## 5.4.8 フォーカス・カーソル位置の論理マップ生成規則とマッピング規則

フォーカス・カーソル位置の論理マップ生成規則について説明します。

### (1) フォーカス・カーソル項目の定義

#### (a) フォーカス項目の定義

出力フォーカス項目の定義

- 出力フォーカス項目：マップ名 -OUTFOCUS-O
- 長さ：2

データ名は、ドローセットアップの「カーソルとフォーカス」ダイアログで変更できません。

入力フォーカス項目の定義

- 入力フォーカス項目：マップ名 -INFOCUS-I
- 長さ：2

データ名は、ドローセットアップの「カーソルとフォーカス」ダイアログで変更できません。

#### (b) カーソル項目の定義

フィールドボックスを複数定義した場合、それぞれに割り当てるために、カーソル項目を複数定義できます。ここでは、2 個目以降の定義について示します。

1 個目の定義については、「5.3.2(1) 出力カーソル項目の定義」、「5.3.3(1) 入力カーソル項目の定義」を参照してください。

行列 (2 進) カーソルの標準値

- 出力カーソル項目データ名：(行) マップ名 -OUTCURSnL, (列) マップ名 -OUTCURSnC
- 入力カーソル項目データ名：(行) マップ名 -INCURSnN, (列) マップ名 -INCURSnM
- 長さ：4

データ名は、ドローセットアップの「カーソルとフォーカス」ダイアログで変更できません。

論理カーソルの標準値

- 出力カーソル項目データ名：マップ名 -OUTCURSnO
- 入力カーソル項目データ名：マップ名 -INCURSnI
- 長さ：2

データ名は、ドローセットアップの「カーソルとフォーカス」ダイアログで変更できません。

5. 論理マップ生成規則とマッピング規則

す。

(2) 出力フォーカス・カーソル位置の論理マップ生成規則とマッピング規則

(a) フォーカスとカーソルを同時に制御する場合

論理マップにはカーソル制御項目だけが生成されます。フォーカス制御項目は生成されません。

論理マップ生成例

「5.3.2(2) 論理マップ生成規則」を参照してください。

マッピング規則

- AP がカーソル項目データ名の領域に代入した内容と結果

《フィールドボックスの定義がある場合》

AP がカーソル項目データ名の領域に代入した内容	結果
(00) <sub>16</sub> クリア, またはマップ生成機能が生成するフォーカス・カーソル定数以外の値	<p>《初期フォーカス, 初期カーソルの指定がある》</p> <ul style="list-style-type: none"> <li>• フォーカスは, 初期フォーカスを指定したオブジェクトに位置づく。</li> <li>• カーソルは, 初期カーソルを指定したフィールドの先頭に位置づく。</li> </ul> <p>《初期フォーカスの指定があり, 初期カーソルの指定がない》</p> <ul style="list-style-type: none"> <li>• フォーカスは, 初期フォーカスを指定したオブジェクトに位置づく。</li> <li>• カーソルは, フィールドボックス内の先頭の行列位置にある入力できるフィールドに位置づく。カーソルはフィールドの先頭に位置づく。</li> </ul> <p>《初期フォーカスの指定がなく, 初期カーソルの指定がある》</p> <ul style="list-style-type: none"> <li>• フォーカスは, ウィンドウ上の先頭の行列位置にあるオブジェクトに位置づく。</li> <li>• カーソルは, 初期カーソルを指定したフィールドの先頭に位置づく。</li> </ul> <p>《初期フォーカス, 初期カーソルの指定がない》</p> <ul style="list-style-type: none"> <li>• フォーカスは, ウィンドウ上の先頭の行列位置にあるオブジェクトに位置づく。</li> <li>• カーソルは, フィールドボックス内の先頭の行列位置にある入力できるフィールドの先頭に位置づく。</li> </ul>

AP がカーソル項目データ名の領域に代入した内容	結果
マップ生成機能が生成するフィールドボックス内のフィールドのフォーカス・カーソル定数	フォーカスはフィールドボックスに位置づく。カーソルはフォーカス・カーソル定数に対応したフィールドの先頭に位置づく。フィールドボックスを複数定義している場合、フォーカスの位置づかないフィールドボックスのカーソルは、初期カーソルを指定したフィールドの先頭に位置づく。初期カーソルの指定がない場合、フィールドボックス内の先頭の行列位置にある入力できるフィールドの先頭に位置づく。
マップ生成機能が生成するフィールドボックス外のオブジェクトのフォーカス・カーソル定数	フォーカスはフォーカス・カーソル定数に対応したオブジェクトに位置づく。カーソルは初期カーソルの指定に従う。フィールドボックスを複数定義している場合、フォーカスの位置づかないフィールドボックスのカーソルは、初期カーソルの指定に従う。 《初期カーソルの指定がある》 初期カーソルを指定したフィールドの先頭にカーソルが位置づく。 《初期カーソルの指定がない》 フィールドボックス内の先頭の行列位置にある入力できるフィールドの先頭にカーソルが位置づく。

## 《フィールドボックスの定義がない場合》

AP がカーソル項目データ名の領域に代入した内容	結果
(00) <sub>16</sub> クリア, または マップ生成機能が生成するフォーカス・カーソル定数以外の値	《初期フォーカスの指定がある》 初期フォーカスを指定したオブジェクトにフォーカスが位置づく。 《初期フォーカスの指定がない》 ウィンドウ上の先頭の行列位置にあるオブジェクトにフォーカスが位置づく。
マップ生成機能が生成するフォーカス・カーソル定数	フォーカス・カーソル定数に対応したオブジェクトにフォーカスが位置づく。

## (b) フォーカスとカーソルを別々に制御する場合

論理マップには、フォーカス制御項目とカーソル制御項目が生成されます。フォーカス制御の論理マップ生成例とマッピング規則について説明します。カーソル制御のマッピング規則については「5.3.2(3) マッピング規則」を参照してください。フォーカス定数については「5.5.3 フォーカス定数の論理マップ生成規則」を参照してください。

## 論理マップ生成規則

## 《COBOL》

- 行列(2進)カーソルを指定した場合

Windows リトルエンディアン用, Linux 用

{02 03} マップ名-OUTCURSL	PIC S9(4) COMP-5.	
{02 03} マップ名-OUTCURSG	PIC S9(4) COMP-5.	
{02 03} マップ名-OUTCURSnL	PIC S9(4) COMP-5.	.....※
{02 03} マップ名-OUTCURSnC	PIC S9(4) COMP-5.	.....※
{02 03} マップ名-OUTFOCUS-0	PIC X(2).	

5. 論理マップ生成規則とマッピング規則

Windows ビッグエンディアン用, AIX / HP-UX / Solaris 用

{02 03} マップ名-OUTCURSL	PIC S9(4) COMP.	
{02 03} マップ名-OUTCURSC	PIC S9(4) COMP.	
{02 03} マップ名-OUTCURSnL	PIC S9(4) COMP.	.....※
{02 03} マップ名-OUTCURSnC	PIC S9(4) COMP.	.....※
{02 03} マップ名-OUTFOCUS-0	PIC X(2).	

注

フィールドボックスを複数定義している場合に展開されます。

- 論理カーソルを指定した場合

{02 03} マップ名-OUTCURS-LOCO	PIC X(2).	
{02 03} マップ名-OUTCURSn0	PIC X(2).	.....※
{02 03} マップ名-OUTFOCUS-0	PIC X(2).	

注

フィールドボックスを複数定義している場合に展開されます。

《C 言語》

- 行列 (2 進) カーソルを指定した場合

unsigned char マップ名_OUTCURSL[2];	
unsigned char マップ名_OUTCURSC[2];	
unsigned char マップ名_OUTCURSnL[2];	.....※
unsigned char マップ名_OUTCURSnC[2];	.....※
unsigned char マップ名_OUTFOCUS_0[2];	

注

フィールドボックスを複数定義している場合に展開されます。

- 論理カーソルを指定した場合

unsigned char マップ名_OUTCURS_LOCO[2];	
unsigned char マップ名_OUTCURSn0[2];	.....※
unsigned char マップ名_OUTFOCUS_0[2];	

注

フィールドボックスを複数定義している場合に展開されます。

## マッピング規則

- AP がフォーカス項目データ名の領域に代入した内容と結果

AP がフォーカス項目データ名の領域に代入した内容	結果
(00) <sub>16</sub> クリア, またはマップ生成機能が生成するフォーカス定数以外の値	《初期フォーカスの指定がある》 初期フォーカスを指定したオブジェクトにフォーカスが位置づく。 《初期フォーカスの指定がない》 ウィンドウ上の先頭の行列位置にあるオブジェクトにフォーカスが位置づく。
マップ生成機能が生成するフォーカス定数	フォーカス定数に対応したオブジェクトにフォーカスが位置づく。

## (3) 入力フォーカス・カーソル位置の論理マップ生成規則とマッピング規則

## (a) フォーカスとカーソルを同時に制御する場合

論理マップにはカーソル制御項目だけが生成されます。フォーカス制御項目は生成されません。フィールドボックスを定義した画面で、複数のプッシュボタンボックスのフォーカス位置を識別して制御したい場合には、フォーカスとカーソルを別々に制御してください。

## 論理マップ生成規則

「5.3.3(2)(b) 論理カーソルを指定した場合」を参照してください。

## マッピング規則

- AP がカーソル項目データ名の領域に代入される内容

《フィールドボックスの定義がある場合》

入力操作	結果
画面送信時のフォーカス位置がプッシュボタンに位置づいている (ほかのオブジェクトにフォーカスが位置づいている場合で、マウスによるプッシュボタン選択は該当しない)	通知コードデータ名 T に対応するフォーカス・カーソル定数を代入する。
画面送信時のフォーカス位置がフィールドボックスに位置づいていて、カーソル位置が入力・入出力フィールド内に収まっていない	(00) <sub>16</sub> を代入する。
画面送信時のフォーカス位置がフィールドボックスに位置づいていて、カーソル位置が入力・入出力フィールド内に収まっている	該当するフィールドに対応するカーソル定数を代入する。
画面送信時のフォーカス位置がプッシュボタン、フィールドボックス以外のオブジェクトに位置づいている	該当するオブジェクトに対応するフォーカス・カーソル定数を代入する。

## 5. 論理マップ生成規則とマッピング規則

### 注

GUI 画面で入力単位が「フィールド単位」または「イベント単位」の場合、キー操作をしたオブジェクトのフォーカス定数、またはカーソル定数を代入します。

《フィールドボックスの定義がない場合》

入力操作	結果
画面送信時のフォーカス位置がプッシュボタンに位置づいている (ほかのオブジェクトにフォーカスが位置づいている場合で、マウスによるプッシュボタン選択は該当しない)	《プッシュボタンにボックス名称の指定がある》 ボックス名・T に対応するフォーカス・カーソル定数を代入する。 《プッシュボタンにボックス名称の指定がない》 通知コードデータ名 T に対応するフォーカス・カーソル定数を代入する。
画面送信時のフォーカス位置がプッシュボタン以外のオブジェクトに位置づいている	該当するオブジェクトに対応するフォーカス・カーソル定数を代入する。

### 注

入力単位が「フィールド単位」または「イベント単位」の場合、キー操作をしたオブジェクトの定数を代入します。

### (b) フォーカスとカーソルを別々に制御する場合

論理マップにはフォーカス制御項目とカーソル制御項目が生成されます。フォーカス制御の論理マップ生成例とマッピング規則について説明します。カーソル制御のマッピング規則については「5.3.3(3) マッピング規則」を参照してください。フォーカス定数については「5.5.3 フォーカス定数の論理マップ生成規則」を参照してください。

#### 論理マップ生成規則

《COBOL》

#### • 行列 (2 進) カーソルを指定した場合

Windows リトルエンディアン用, Linux 用

{02 03} マップ名-INCURSN	PIC S9(4) COMP-5.	
{02 03} マップ名-INCURSM	PIC S9(4) COMP-5.	
{02 03} マップ名-INCURSnN	PIC S9(4) COMP-5.	.....※
{02 03} マップ名-INCURSnM	PIC S9(4) COMP-5.	.....※
{02 03} マップ名-INFOCUS-I	PIC X(2).	

Windows ビッグエンディアン用, AIX / HP-UX / Solaris 用

{02 03} マップ名-INCURSN	PIC S9(4) COMP.	
{02 03} マップ名-INCURSM	PIC S9(4) COMP.	
{02 03} マップ名-INCURSnN	PIC S9(4) COMP.	.....※
{02 03} マップ名-INCURSnM	PIC S9(4) COMP.	.....※
{02 03} マップ名-INFOCUS-I	PIC X(2).	

### 注

フィールドボックスを複数定義している場合に展開されます。

- 論理カーソルを指定した場合

```
{02|03} マップ名-INCURS-LOGI    PIC X(2).
{02|03} マップ名-INCURSnI      PIC X(2).      .....※
{02|03} マップ名-INFOCUS-I    PIC X(2).
```

注

フィールドボックスを複数定義している場合に展開されます。

《C 言語》

- 行列（2進）カーソルを指定した場合

```
unsigned char マップ名_INCURSN[2];
unsigned char マップ名_INCURSM[2];
unsigned char マップ名_INCURSnN[2];      .....※
unsigned char マップ名_INCURSnM[2];      .....※
unsigned char マップ名_INFOCUS_I[2];
```

注

フィールドボックスを複数定義している場合に展開されます。

- 論理カーソルを指定した場合

```
unsigned char マップ名_INCURS_LOGI[2];
unsigned char マップ名_INCURSnI[2];      .....※
unsigned char マップ名_INFOCUS_I[2];
```

注

フィールドボックスを複数定義している場合に展開されます。

マッピング規則

- AP がフォーカス項目データ名の領域に代入した内容と結果

入力操作	結果
画面送信時のフォーカス位置がプッシュボタンに位置づいている (ほかのオブジェクトにフォーカスが位置づいている場合で、マウスによるプッシュボタン選択は該当しない)	《プッシュボタンにボックス名称の指定がある》 ボックス名-T に対応するフォーカス定数を代入する。 《プッシュボタンにボックス名称の指定がない》 通知コードデータ名 T に対応するフォーカス定数を代入する。
画面送信時のフォーカス位置がプッシュボタン以外のオブジェクトに位置づいている	該当するオブジェクトに対応するフォーカス定数を代入する。
画面送信時のフォーカス位置がフィールドボックスに位置づいている	フィールドボックス内の先頭の行列位置にあるフィールドに対応するフォーカス定数を代入する。

注

入力単位が「フィールド単位」または「イベント単位」の場合、キー操作をしたオブジェクト

## 5. 論理マップ生成規則とマッピング規則

の定数を代入します。

### 5.4.9 二次ウィンドウの表示に関するマッピング規則

二次ウィンドウのマッピング規則について説明します。

#### (1) 初期表示（ウィンドウが何も表示されていない）の場合

出力するマップ	表示形態	画面表示結果	フォーカス位置
一次ウィンドウ定義のマップ	一部上書， 全面書換， 自動	一次ウィンドウ定義時のウィンドウの表示位置と大きさを一次ウィンドウを表示する	出力した一次ウィンドウ
二次ウィンドウ定義のマップ	-	表示されない（XMAP3 でエラー）	-

（凡例）

- : 該当しない。

注

ウィンドウが表示できない（実画面からウィンドウがはみ出る）場合は、ウィンドウが表示できるように XMAP3 が位置を補正します。

#### (2) 一次ウィンドウだけが表示されている場合

出力するマップ	表示形態	画面表示結果 <sup>1</sup>	フォーカス位置	
一次ウィンドウ定義のマップ	表示されているものと同じマップ名	一部上書	出力した一次ウィンドウ	
	表示されているものと異なるマップ名	全面書換		表示・印刷セットアップの一次
		一部上書， 全面書換， 自動		ウィンドウの位置を「ウィンドウ操作を優先する」にした場合、直前に一次ウィンドウを表示していた位置に今回表示する大きさを表示する。「画面定義を優先する」を指定した場合、定義した位置に表示する。



出力するマップ		表示形態	画面表示結果 <sup>1</sup>	フォーカス位置
二次ウィンドウ定義のマップ	一次ウィンドウ定義中に、同じ ID (ウィンドウ種別番号) の定義がある。	一部上書、 全面書換、 自動	(一次ウィンドウ定義で二次ウィンドウ位置を指定した場合) 一次ウィンドウ定義中の対応する ID (二次ウィンドウ用 ID) の表示位置 <sup>2</sup> に表示する。 (二次ウィンドウ定義でウィンドウ位置を指定した場合) 二次ウィンドウ定義時の表示位置に二次ウィンドウを表示する。	出力した二次ウィンドウ
	一次ウィンドウ定義中に、二次ウィンドウの定義なし、または同じ ID (ウィンドウ識別番号) の二次ウィンドウの定義なし。		(一次ウィンドウ定義で二次ウィンドウ位置を指定した場合) 一次ウィンドウの位置に二次ウィンドウを表示する。 (二次ウィンドウ定義でウィンドウ位置を指定した場合) 二次ウィンドウ定義時の表示位置に二次ウィンドウを表示する。	

## 注 1

ウィンドウが表示できない (実画面からウィンドウがはみ出る) 場合は、ウィンドウが表示できるように XMAP3 が位置を補正します。

## 注 2

端末オペレータによって、一次ウィンドウが移動されている場合、二次ウィンドウも同様に移動された位置に出力されます。

## (3) 一次ウィンドウと二次ウィンドウが表示されている場合

出力するマップ		表示形態	画面表示結果 <sup>1</sup>	フォーカス位置
一次ウィンドウ定義のマップ	表示されているものと同じマップ名	全面書換	二次ウィンドウは消去される。	出力した一次ウィンドウ
		一部上書、 自動	二次ウィンドウはそのまま表示される。	画面を表示した時点で対話権があった二次ウィンドウ
	表示されているものと異なるマップ名	一部上書、 全面書換、 自動	一次ウィンドウを表示する。二次ウィンドウは消去される。	出力した一次ウィンドウ

5. 論理マップ生成規則とマッピング規則

出力するマップ		表示形態	画面表示結果 <sup>1</sup>	フォーカス位置
二次ウィンドウ定義のマップ	表示されているものと同じマップ	全面書換	<p>(一次ウィンドウ定義で二次ウィンドウ位置を指定した場合) 一次ウィンドウ定義中の対応する ID (二次ウィンドウ用 ID) の表示位置 <sup>2</sup> に、直前の二次ウィンドウの大きさで表示する。</p> <p>(一次ウィンドウ定義で二次ウィンドウ位置を指定しない場合) 一次ウィンドウの位置に直前の二次ウィンドウの大きさで表示する。</p> <p>(二次ウィンドウ定義でウィンドウ位置を指定した場合) 二次ウィンドウ定義時の表示位置に直前の二次ウィンドウの大きさで表示する。</p>	出力した二次ウィンドウ
		一部上書	直前の二次ウィンドウを表示していた位置 <sup>3</sup> にそのときと同じ大きさで二次ウィンドウを表示する。	

出力するマップ	表示形態	画面表示結果 <sup>1</sup>	フォーカス位置
表示されているものと異なるマップで、二次ウィンドウの ID (ウィンドウ種別番号) は現在表示されているものと同じ	一部上書、 全面書換、 自動	<p>(一次ウィンドウ定義で二次ウィンドウ位置を指定した場合)</p> <p>一次ウィンドウ定義中の対応する ID (二次ウィンドウ用 ID) の表示位置 <sup>2</sup> に表示する。このとき、直前の一次ウィンドウはそのまま表示され、二次ウィンドウは出力したウィンドウの内容と置き換えられる。</p> <p>(一次ウィンドウ定義で二次ウィンドウ位置を指定しない場合)</p> <p>一次ウィンドウの表示位置に二次ウィンドウを表示する。このとき、直前の一次ウィンドウはそのまま表示され、二次ウィンドウは出力したウィンドウの内容と置き換えられる。</p> <p>(二次ウィンドウ定義でウィンドウ位置を指定した場合)</p> <p>二次ウィンドウ定義時の表示位置に二次ウィンドウを表示する。このとき、直前の一次ウィンドウはそのまま表示され、二次ウィンドウは出力したウィンドウの内容と置き換えられる。</p>	出力した二次ウィンドウ
表示されているものと異なるマップで、二次ウィンドウの ID (ウィンドウ種別番号) も異なる	一部上書、 全面書換、 自動	<p>(一次ウィンドウ定義で二次ウィンドウ位置を指定した場合)</p> <p>一次ウィンドウ定義中の対応する ID (二次ウィンドウ用 ID) の表示位置 <sup>2</sup> に表示する。このとき、直前の二次ウィンドウと一次ウィンドウは表示したままになる。</p> <p>(一次ウィンドウ定義で二次ウィンドウ位置を指定しない場合)</p> <p>一次ウィンドウの表示位置に二次ウィンドウを表示する。このとき、直前の二次ウィンドウと一次ウィンドウは表示したままになる。</p> <p>(二次ウィンドウ定義でウィンドウ位置を指定した場合)</p> <p>二次ウィンドウ定義時の表示位置に二次ウィンドウを表示する。このとき、直前の二次ウィンドウと一次ウィンドウは、表示したままになる。</p>	

注 1

ウィンドウが表示できない(実画面からウィンドウがはみ出る)場合は、ウィンドウが表示で

## 5. 論理マップ生成規則とマッピング規則

きるように XMAP3 が位置を補正します。

### 注 2

端末オペレータによって、一次ウィンドウが移動されている場合、二次ウィンドウも同様に移動された位置に出力されます。

### 注 3

端末オペレータによってウィンドウが移動されている場合は、その位置に表示されます。

## 5.4.10 出力グラフィックの論理マップ生成規則とマッピング規則

### (1) 出力グラフィックの定義

出力グラフィックをドローで定義することで生成されます。

- 出力グラフィック項目のデータ名：マップ名-GRAPHnnnn-O  
データ名は、出力グラフィックダイアログの「データ名」で変更できます。
- 長さ  
ファイル名指定：12  
フルバス 64：64  
フルバス 128：128  
フルバス 259：259

### (2) 出力グラフィックの論理マップ生成規則とマッピング規則

出力グラフィックの論理マップ生成規則とマッピング規則について説明します。

#### (a) 可変項目の出力論理マップの生成規則

《COBOL》

```
[02 | 03] マップ名-GRAPHnnnn-O PIC X(長さ).
```

《C 言語》

```
unsigned char マップ名_GRAPHnnnn_0[長さ];
```

## (b) 可変項目のマッピング規則

## AP が出力項目データ名の領域に代入した内容と結果

AP が出力項目データ名の領域に代入した内容	結果
すべてデータの場合	(指定されたファイルが存在した場合) 指定されたファイルを読み込み、グラフィックを表示する。 (指定されたファイルが存在しない場合) 表示されない。
先頭 1 文字がデータ有無コード	データ有無コード (1F) <sub>16</sub> を設定した場合、例として X'1F414234' または X'1F1F1F' のようなデータが該当する。このとき、表示形態が「全面書換」か「一部上書」によって次のように異なる。 全面書換の場合：何も表示しない。 一部上書の場合：直前に表示したデータのまま表示する。
データの後半にデータ有無コード、ヌルまたは空白	データ有無コード、ヌル ((00) <sub>16</sub> )、または空白 ((20) <sub>16</sub> ) を (xx) <sub>16</sub> と仮定した場合、例として X'414243xxxx' のようなデータが該当する。この場合、後半の X'xxxx' を切り捨てたデータの X'414243' が指定されたファイル名となる。 (指定されたファイルが存在した場合) 指定されたファイルを読み込み、グラフィックを表示する。 (指定されたファイルが存在しない場合) 表示されない。

## 注

データが「\*CLIP」の場合、クリップボード中のグラフィックを表示します。

## 5.4.11 日付・時刻項目の論理マップ生成規則とマッピング規則

## (1) 日付・時刻項目の定義

GUI 画面と GUI 画面内のフィールドボックスおよび CUI 画面では、各項目の名称が次の表のように異なります。

表 5-9 可変項目の名称の違い

日付・時刻項目	GUI 画面	フィールドボックスおよび CUI 画面
出力項目	出力日付・時刻テキストボックス	出力日付・時刻フィールド
入出力項目	入出力日付・時刻テキストボックス	入出力日付・時刻フィールド

各項目の名称は、それぞれの画面での名称に置き換えてください。

## 出力項目

出力日付・時刻テキストボックス、または出力日付・時刻フィールドに対して生成されます。

- 出力項目：マップ名 -FIELDnnnn-O

## 5. 論理マップ生成規則とマッピング規則

- 繰り返し回数：縦，または横の反復回数
- 長さ：論理項目の長さ。指定がないときはフィールドのデータ長
- データ型：テキストボックス，フィールドのデータ型で設定

表 5-10 日付・時刻の出力項目のデータ型と使用できる項目

データ型	使用できる項目
99999	数字編集項目 (COBOLで桁数に満たない値を代入すると， COBOLの仕様によって「0」が設定されます)
文字(XX)	文字

### 注

桁数に満たない値を代入し，その残りにデータ有無コードを指定している場合，「桁寄せ：左」，「埋字：スペース」で埋字されます。その他のマッピング規則については，「5.4.11(2) 日付・時刻項目の出力論理マップ生成規則とマッピング規則」を参照してください。

### 入力項目

入出力日付・時刻テキストボックス，または入出力日付・時刻フィールドに対して生成されます。次に示す入力項目以外の定義内容は出力項目と同じになります。

- 入力項目：マップ名-FIELDnnnn-I

日付，時刻入力項目では，入力する桁数をキー入力時にチェックするため，桁数に満たないデータの入力ができません。しかし，「入力済み」属性が設定されている場合，画面表示上のデータがそのまま入力データとなり，日付・時刻の出力項目のデータ型と同じデータ型指定に合ったデータで処理されます。日付・時刻の出力項目のデータ型については，「表 5-10 日付・時刻の出力項目のデータ型と使用できる項目」を参照してください。

### 入出力項目

入出力日付・時刻テキストボックス，または入出力日付・時刻フィールドに対して生成されます。論理マップは，入力論理マップと出力論理マップに分けて出力されます。定義内容は出力項目と同じになります。

## (2) 日付・時刻項目の出力論理マップ生成規則とマッピング規則

日付・時刻項目の出力論理マップ生成規則と論理マップ生成規則について説明します。

### (a) 可変項目の出力論理マップの生成規則

《COBOL》

- 文字項目の場合

```
{02|03} マップ名-FIELDnnnn-0 PIC X(長さ) [OCCURS 回数] .
```

- 数字編集項目の場合

```
{02|03} マップ名-FIELDnnnn-0 PIC 編集文字 [OCCURS 回数] .
```

#### 《C 言語》

C 言語使用時は、数字編集項目は指定できません。

```
unsigned char マップ名_FIELDnnnn_0 [[回数]] [長さ];
```

#### (b) 可変項目のマッピング規則

AP が出力項目データ名の領域に代入した内容と結果

AP が出力項目データ名の領域に代入した内容	マッピングオプション	結果
すべてデータの場合	マージ,	代入されたデータを表示する。
先頭 1 文字がデータ有無コード	論理マップ	データ有無コード (1F) <sub>16</sub> を設定した場合、例として X'1F414234' または X'1F1F1F' のようなデータが該当する。このとき、表示形態が「全面書換」か「一部上書」によって次のように異なる。 全面書換の場合：何も表示しない。 一部上書の場合：直前に表示したデータのまま表示する。
データの後半にデータ有無コード		データ有無コード (1F) <sub>16</sub> を仮定した場合、例として X'4142431F1F' のようなデータが該当する。この場合、代入されたデータを表示する（不足分は表示されない）。
-	物理マップ	データを表示しない。

#### (凡例)

- : 該当しない。

### (3) 日付・時刻項目の入力論理マップ生成規則とマッピング規則

日付・時刻項目の入力論理マップ生成規則と論理マップ生成規則について説明します。

#### (a) 入出力日付・時刻フィールドおよび入出力日付・時刻テキストを指定した場合

入力論理マップの生成規則

#### 《COBOL》

論理マップ可変部の集団項目化を指定した場合

- 数字項目の場合

```
03 マップ名-FIELDnnnn-H [OCCURS 回数] .
04 マップ名-FIELDnnnn-I PIC 編集文字.
```

## 5. 論理マップ生成規則とマッピング規則

- 文字項目の場合

```
03 マップ名-FIELDnnnn-H [OCCURS 回数].
04 マップ名-FIELDnnnn-I PIC X(長さ).
```

論理マップ可変部の集団項目化を指定していない場合

- 文字項目の場合

```
02 マップ名-FIELDnnnn-I PIC X(長さ) [OCCURS 回数].
```

《C 言語》

```
unsigned char マップ名_FIELDnnnn_I [[回数]] [長さ];
```

マッピング規則

- 入力操作と入力項目データ名の領域の内容

《使用目的が日本語以外の場合》

入力操作	結果（入力項目データ名の領域の内容）
データを入力	データを入力項目データ名の領域に代入する。
フィールドキーを押す (すべて(00) <sub>16</sub> のデータ 入力)	データ消去通知文字に従って入力項目データ名の領域をクリアする。 フィールドキーを押したときのマッピング規則については、「5.3.10 フィールドキーを押したときのマッピング規則」を参照のこと。
入力操作なし	初期値クリア文字に従って入力データ名の領域をクリアする。入力操作が なかったときのマッピング規則の詳細については、「5.3.12 入力操作がな かったときのマッピング規則」を参照のこと。

### 5.4.12 トグルフィールドの論理マップ生成規則とマッピング規則

トグルフィールドの論理マップ生成規則とマッピング規則について説明します。

#### (1) トグルフィールドの定義

トグルフィールドの属性

トグルフィールドダイアログで「動的変更（AP から表示属性を変更する）」を選ぶことで生成されます。

- 制御項目のデータ名：マップ名-FIELDnnnn-A

データ名は、トグルフィールドダイアログの「データ名」に従います。

- 長さ：2

長さは、ドローセットアップの「運用管理者用の設定」の[修飾名の設定]ボタンをクリックして表示される、修飾名の設定ダイアログで変更できます。変更するときには、「動的変更の種別」から「候補選択コントロール/コマンドコントロール」



を選択し、「修飾名長」で長さを設定します。

トグルフィールドの定義

- ボックス項目のデータ名：マップ名-FIELDnnnn-I  
データ名は、トグルフィールドダイアログの「データ名」で変更できます。
- 長さ：AP が受け取る項目のデータ長で指定

## (2) トグルフィールド属性の論理マップ生成規則とマッピング規則

トグルフィールド属性の論理マップ生成規則とマッピング規則について説明します。この項目は、ダイアログで「動的変更（AP から表示属性を変更する）」を指定したときだけ論理マップが生成されます。

### (a) 論理マップ生成規則

《COBOL》

```
[02|03] マップ名-FIELDnnnn-A PIC X(長さ) [OCCURS 回数] .
```

《C 言語》

```
unsigned char マップ名_FIELDnnnn_A [[回数]] [長さ];
```

### (b) マッピング規則

AP が制御項目に代入した内容と表示結果

《「動的変更（AP から表示属性を変更する）」を指定した場合》

AP が制御項目に代入した内容	マッピングオプション	表示結果
セットアップで指定してある修飾名と同じ	マージ	トグルフィールドの修飾名に対応する属性を使って表示する。
先頭にデータ有無コード、または設定していない修飾名		標準の属性を使って表示する。
セットアップで指定してある修飾名と同じ	論理マップ	トグルフィールドの修飾名に対応する属性を使って表示する。
先頭にデータ有無コード、または設定していない修飾名		(表示形態が「全面書換」のとき) 標準の属性を使って表示する。 (表示形態が「一部上書」で直前に表示したマップと同じ場合) 前回の属性のまま表示する。 (表示形態が「一部上書」で直前に表示したマップと異なる場合) 標準の属性を使って表示する。
-	物理マップ	標準の属性を使って表示する。

## 5. 論理マップ生成規則とマッピング規則

(凡例)

- :該当しない。

注

修飾名と修飾名に対応したフィールド属性情報は、ドロースेटアップの「表示属性の動的変更」の「候補選択コントロール」タブで変更できます。

《「動的変更 (AP から表示属性を変更する)」を指定しない場合》  
標準の属性を使って表示します。

### (3) トグルフィールドの入力論理マップ生成規則とマッピング規則

トグルフィールドの入力論理マップ生成例とマッピング規則について説明します。

#### (a) 論理マップ生成規則

《COBOL》

- 論理マップ可変部の集団項目化を指定した場合

```
03 マップ名-FIELDnnnn-H.  
04 マップ名-FIELDnnnn-I PIC {X(長さ)|N(長さ÷2)|9(長さ)}.
```

- 論理マップ可変部の集団項目化を指定しない場合

```
02 マップ名-FIELDnnnn-I PIC {X(長さ)|N(長さ÷2)|9(長さ)}.
```

《C 言語》

```
unsigned char マップ名_FIELDnnnn_I[長さ];
```

#### (b) マッピング規則

入力操作と結果 (論理項目の内容)

入力操作	結果 (論理項目の内容)
フィールド中の文字を変更する	選んだ文字に対応する定数を論理項目に設定する。
フィールド中の文字を変更しない、またはフィールドにカーソルを位置づけない	初期クリア文字に従って論理項目をクリアする。

## 5.4.13 スピンボックスの論理マップ生成規則とマッピング規則

スピンボックスの論理マップ生成規則とマッピング規則について説明します。

### (1) スピンボックスの定義

スピンボックスの属性

スピンボックスダイアログで「動的変更 (AP から表示属性を変更する)」を選ぶと生

成されます。

- 制御項目のデータ名：マップ名・FIELDnnnn-A  
データ名は、スピボックスダイアログの AP が渡す項目の「データ名」に従います。
- 長さ：2  
長さは、ドローセットアップの「運用管理者用の設定」の [ 修飾名の設定 ] ボタンをクリックして表示される、修飾名の設定ダイアログで変更できます。変更するときには、「動的変更の種別」から「キャラクタコントロール (キーエントリ/選択エントリ)」を選択し、「修飾名長」で長さを設定します。

スピボックスの定義

- ボックス項目のデータ名：マップ名・FIELDnnnn-I/O  
データ名は、スピボックスダイアログの AP が受け取る項目の「データ名」で変更できます。
- 長さ：AP が受け取る項目のデータ長で指定

## (2) スピボックスの論理マップ生成規則とマッピング規則

### (a) 項目属性の論理マップ生成規則とマッピング規則

スピボックスの項目属性に関する論理マップ生成規則とマッピング規則について説明します。この論理マップは、スピボックスダイアログで「動的変更 (AP から表示属性を変更する)」を選び、属性項目データ名を指定することで生成されます。

論理マップ生成規則

《COBOL》

```
{02|03} マップ名-FIELDnnnn-A PIC X(長さ).
```

《C 言語》

```
unsigned char マップ名_FIELDnnnn_A[長さ];
```

マッピング規則

- AP が制御項目に代入した内容と表示結果

《「動的変更 (AP から表示属性を変更する)」を指定した場合》

AP が制御項目に代入した内容	マッピングオプション	表示結果
修飾名と同じ	マージ	スピボックスの修飾名に対応する属性を使って表示する。
先頭にデータ有無コード、または修飾名以外 (上記以外)		標準の属性を使って表示する。

## 5. 論理マップ生成規則とマッピング規則

AP が制御項目に代入した内容	マッピングオプション	表示結果
修飾名と同じ	論理マップ	スピンボックスの修飾名に対応する属性を使って表示する。
先頭にデータ有無コード、または修飾名以外（上記以外）		(表示形態が「全面書換」の場合) 標準の属性を使って表示する。 (表示形態が「一部上書」で直前に表示したマップと同じ場合) 前回の属性をそのまま表示する。 (表示形態が「一部上書」で直前に表示したマップと異なる場合) 標準の属性を使って表示する。
-	物理マップ	標準の属性を使って表示する。

(凡例)

- : 該当しない。

注

修飾名と修飾名に対応したテキスト・フィールド変更属性情報は、ドローセットアップの「表示属性の動的変更」の「キャラクタコントロール」タブで変更できます。

《「動的変更 (AP から表示属性を変更する)」を指定しない場合》

標準の属性を使って表示します。

(b) 可変項目の出力論理マップ生成規則とマッピング規則

可変項目の出力論理マップ生成規則と論理マップ生成規則について説明します。

出力論理マップの生成規則

《COBOL》

COBOL の可変項目の生成規則は、使用目的によってデータ型が変わります。

- 数字項目の場合

```
[02|03] マップ名-FIELDnnnn-0 PIC 9(長さ) [OCCURS 回数] .
```

- 文字項目の場合

```
[02|03] マップ名-FIELDnnnn-0 PIC X(長さ) [OCCURS 回数] .
```

- 数字編集項目の場合

```
[02|03] マップ名-FIELDnnnn-0 PIC 編集文字 [OCCURS 回数] .
```

《C 言語》

C 言語使用時は、数字編集項目は指定できません。

```
unsigned char マップ名_FIELDnnnn_0 [[回数]] [長さ];
```

## マッピング規則

- AP が出力項目データ名の領域に代入した内容と結果

AP が出力項目データ名の領域に代入した内容	マッピングオプション	結果
すべてデータの場合	マージ,	代入されたデータを表示する。
先頭 1 文字がデータ有無コード	論理マップ	データ有無コード (1F) <sub>16</sub> を設定した場合、例として X'1F414234' または X'1F1F1F' のようなデータが該当する。このとき、表示形態が「全面書換」か「一部上書」によって次のように異なる。 全面書換の場合：最小値を表示する。 一部上書の場合：直前に表示したデータのまま表示する。
データの後半にデータ有無コード		データ有無コード (1F) <sub>16</sub> を仮定した場合、例として X'4142431F1F' のようなデータが該当する。
-	物理マップ	最小値を表示する。

(凡例)

- : 該当しない。

注

代入されたデータを表示します。ただし、値が範囲外の場合には、最初の操作によって最大値、最小値のどちらか近い値を表示します。

## (c) スピンボックスの入力論理マップ生成規則とマッピング規則

スピンボックスの入力論理マップ生成規則と論理マップ生成規則について説明します。

## 入力論理マップの生成規則

## 《COBOL》

COBOL の可変項目の生成規則は、使用目的によってデータ型が変わります。

- 論理マップ可変部の集団項目化を指定した場合
  - 数字項目の場合

03 マップ名-FIELDnnnn-H [OCCURS 回数]. 04 マップ名-FIELDnnnn-I PIC 編集文字.
---

- 文字項目の場合

03 マップ名-FIELDnnnn-H [OCCURS 回数]. 04 マップ名-FIELDnnnn-I PIC X(長さ).
--

- 論理マップ可変部の集団項目化を指定していない場合
  - 数字項目の場合

02 マップ名-FIELDnnnn-I PIC 9(長さ) [OCCURS 回数].
--

## 5. 論理マップ生成規則とマッピング規則

・文字項目の場合

```
02 マップ名-FIELDnnnn-I PIC X(長さ) [OCCURS 回数].
```

《C 言語》

```
unsigned char マップ名_FIELDnnnn_I [[回数]] [長さ];
```

マッピング規則

・入力操作と入力項目データ名の領域の内容

《使用目的が日本語以外の場合》

入力操作	結果（入力項目データ名の領域の内容）
スピンのボックスにフォーカスを位置づけ、ボタン操作で表示データを変更する。	選択したデータを入力項目データ名の領域に代入する。
入力操作なし	初期値クリア文字に従って入力データ名の領域をクリアする。入力操作がなかったときのマッピング規則の詳細については、「5.3.12 入力操作がなかったときのマッピング規則」を参照のこと。

### 5.4.14 数値・金額項目の論理マップ生成規則とマッピング規則

数値・金額項目の論理マップ生成規則とマッピング規則について説明します。

#### (1) 数値・金額項目の定義

- ・入力項目 : マップ名-FIELDnnnn-I  
数字編集文字列：データ型で指定できる数字編集文字列は次のとおり。「9」、「V」、  
「S」
- ・繰り返し回数 : 縦、または横の反復回数（入出力フィールドだけ）

#### (2) 論理マップ生成規則

《COBOL》

```
{02|03} マップ名-FIELDnnnn-H [OCCURS 回数].  
{03|04} マップ名-FIELDnnnn-I PIC 編集文字.
```

注

使用目的が金額の場合、編集文字に小数点を示す「V」を使用できます。

### (3) マッピング規則

#### 入力操作と結果

入力操作	結果
フィールドキーを押した場合	データ消去通知文字に従って入力項目データ名の領域をクリアする。データ消去通知文字が埋字として指定してあるときは、 $0((30)_{16})$ でクリアする。
入力操作がない、または END キーを押した場合。	(初期値が指定してある場合) 初期値で指定した定数を入力項目データ名の領域に代入する。 (初期値が指定していない場合) 初期クリア文字が埋字として指定してあるときは、 $0((30)_{16})$ でクリアする。
全桁に空白を入力した場合	空白 ( $(20)_{16}$ ) で入力項目データ名の領域をクリアする。また、NULL と空白を混在している場合も、すべて空白となる。
数字項目として正しいデータを入力した場合	数字編集項目の指定に従って入力項目データ名の領域に値を代入する。
数字項目として不正なデータを入力した場合	入力チェックを指定した場合、入力時にエラーとなる。入力チェックを指定していない場合、コードエラー通知文字に従って入力項目データ名の領域をクリアする。コードエラー通知文字の標準は HIGH ('X'FF') である。

#### 注

「入力済み」属性を指定して、整数桁や小数桁が合わないデータを表示して画面確定している場合、不正なデータとなります。

## 5.4.15 ウィンドウ表示位置の論理マップ生成規則とマッピング規則

### (1) ウィンドウ表示位置の制御情報

#### 制御情報の標準値

- ウィンドウ位置制御データ名：マップ名・WINDOWW  
データ名は、ドローセットアップの「ドローの設定」から表示する、表示属性の動的変更ダイアログの「データ名」タブで変更できます。
- 長さ：2  
長さは、ドローセットアップの「運用管理者用の設定」の [ 修飾名の設定 ... ] ボタンをクリックして表示される、修飾名の設定ダイアログで変更できます。変更するときには、「動的変更の種別」から「位置属性」を選択し、「修飾名長」で長さを設定します。

### (2) 論理マップ生成規則

ウィンドウ表示位置の制御情報の論理マップ生成例について説明します。

## 5. 論理マップ生成規則とマッピング規則

《COBOL》

```
[02|03] マップ名-WINDOWW PIC X(長さ).
```

《C 言語》

```
unsigned char マップ名_WINDOWW[長さ];
```

### (3) マッピング規則

ウィンドウ表示位置の制御情報のマッピング規則について説明します。

AP がウィンドウ表示位置の制御情報のデータ名の領域に代入した内容とその結果

AP がウィンドウ表示位置の制御情報のデータ名の領域に代入した内容	結果
修飾名と同じ	ウィンドウ表示位置の制御情報の修飾名に対応するウィンドウ制御情報を使って画面が表示される。
データの先頭またはすべてがデータ有無コード、または修飾名以外（上記以外）	標準のウィンドウ表示制御情報を使って画面が表示される。画面属性ダイアログの XYZ 位置に従う。

注

修飾名と修飾名に対応したウィンドウ表示位置の制御情報は、ドローセットアップの「表示属性の動的変更」の「ウィンドウ属性」タブで変更できます。

注

1 回目の画面表示、または「全面書換」で画面が表示される場合にだけ、指定が有効になります。



## 5.5 定数部の論理マップ生成規則とマッピング規則

定数部の論理マップ生成規則について説明します。

### 5.5.1 定数部の論理マップ生成規則

#### (1) 定義部の定義

出力論理マップ長の出力

ドロースेटアップの論理マップ属性ダイアログ中の展開方式の「定数部への論理マップ長出力」を指定したときに生成されます。

カーソル定数の出力

カーソル定数は無条件に生成されます。

#### (2) 論理マップ生成規則

##### (a) 出力論理マップ長の出力の指定

出力論理マップの論理マップ生成例を示します。また、ここで説明する出力論理マップ長とは、定数部だけの論理マップ長です。

##### 《COBOL》

- Windows リトルエンディアン用, Linux 用

```
01 マップ名T PIC S9(4) COMP-5 VALUE +長さ.
```

- Windows ビッグエンディアン用, AIX / HP-UX / Solaris 用  
<論理マップ長が 10,000 バイト未満の場合>

```
01 マップ名T PIC S9(4) COMP VALUE +長さ.
```

<論理マップ長が 10,000 バイト以上の場合>

```
01 マップ名U PIC X(2) VALUE X'長さ'.
01 マップ名T REDEFINES マップ名U PIC S9(4) COMP.
```

##### 《C 言語》

- Windows リトルエンディアン用, AIX / HP-UX / Solaris / Linux 用

```
short マップ名T=長さ;
```

## 5. 論理マップ生成規則とマッピング規則

- Windows ビッグエンディアン用

```
unsigned char マップ名T[2]={0x長さ};
```

### (b) カーソル定数の出力の指定

カーソル定数を指定したときの論理マップ生成例を示します。

繰り返し項目がある場合

《COBOL》

- Windows 用, Linux 用

```
01 マップ名D.  
02 項目データ名D.  
03 FILLER PIC X(2) VALUE X'nnnn'.  
:  
02項目データ名T REDEFINES 項目データ名D OCCURS 回数.
```

- AIX / HP-UX / Solaris 用

<カーソル定数の値が 10,000 未満の場合>

```
01 マップ名D.  
02 項目データ名D.  
03 FILLER PIC S9(4) COMP VALUE +nnnn.  
:  
02項目データ名T REDEFINES 項目データ名D OCCURS 回数.
```

<カーソル定数の値が 10,000 以上の場合>

```
01 マップ名D.  
02 項目データ名D.  
:  
03 FILLER PIC X(2) VALUE X'nnnn'.  
:  
02項目データ名T REDEFINES 項目データ名D OCCURS 回数.
```

《C 言語》

```
struct {  
    unsigned char 項目データ名T [回数][2];  
    :  
} マップ名D={0xnn,0xnn [,...]};
```

繰り返し項目がない場合

《COBOL》

- Windows 用 , Linux 用

```
01 マップ名D.
02 項目データ名T.
03 FILLER PIC X(2) VALUE X'nnnn'.
:
```

- AIX / HP-UX / Solaris 用

<カーソル定数の値が 10,000 未満の場合>

```
01 マップ名D.
02 項目データ名T.
03 FILLER PIC S9(4) COMP VALUE +nnnn.
:
```

<カーソル定数の値が 10,000 以上の場合>

```
01 マップ名D.
02 項目データ名T.
:
03 FILLER PIC X(2) VALUE X'nnnn'.
:
```

《C 言語》

```
struct {
    unsigned char 項目データ名T [2];
    :
} マップ名D={0xnn, 0xnn};
```

## 5.5.2 イベント定数の論理マップ生成規則

イベント定数の論理マップ生成規則について説明します。

### (1) イベント定数の定義

イベント定数は、画面属性ダイアログの「入力単位」で「イベント」を指定すると生成されます。ドロースेटアップの「イベント通知コード」ダイアログの、「通知コードの長さ」が「4」のときだけ指定できます。

## (2) 論理マップ生成規則

《COBOL》

```
01 マップ名D.  
02 項目データ名V.  
03 FILLER PIC X(4) VALUE X'nnnnnnnn'.  
:
```

《C 言語》

```
struct {  
    unsigned char 項目データ名V [4];  
    :  
} マップ名D={0xnn, 0xnn, 0xnn, 0xnn [, ...] };
```

### 5.5.3 フォーカス定数の論理マップ生成規則

フォーカス定数の論理マップ生成規則について説明します。

#### (1) フォーカス定数の定義

フォーカス定数は、無条件に生成されます。

#### (2) フォーカス定数の論理マップ生成例

《COBOL》

繰り返し項目がある場合

- Windows 用, Linux 用

```
01 マップ名D.  
02 項目データ名D.  
03 FILLER PIC X(2) VALUE X'nnnn'.  
:  
02項目データ名T REDEFINES 項目データ名D OCCURS 回数.
```

- AIX / HP-UX / Solaris 用

<フォーカス定数の値が 10,000 未満の場合>

```
01 マップ名D.  
02 項目データ名D.  
03 FILLER PIC S9(4) COMP VALUE +nnnn.  
:  
02項目データ名T REDEFINES 項目データ名D OCCURS 回数.
```

<フォーカス定数の値が 10,000 以上の場合 >

```
01 マップ名D.
02 項目データ名D.
03 FILLER PIC X(2) VALUE X' nnnn' .
:
02項目データ名T REDEFINES 項目データ名D OCCURS 回数.
```

繰り返し項目がない場合

- Windows 用 , Linux 用

```
01 マップ名D.
02 項目データ名T.
03 FILLER PIC X(2) VALUE X' nnnn' .
:
```

- AIX / HP-UX / Solaris 用

<フォーカス定数の値が 10,000 未満の場合 >

```
01 マップ名D.
02 項目データ名T.
03 FILLER PIC S9(4) COMP VALUE +nnnn.
:
```

<フォーカス定数の値が 10,000 以上の場合 >

```
01 マップ名D.
02 項目データ名T.
03 FILLER PIC X(2) VALUE X' nnnn' .
:
```

プッシュボタンボックスの場合

```
01 マップ名D.
02 ボックス名T. ....※
03 FILLER PIC X(2) VALUE X' nnnn' .
:
```

注

プッシュボタンボックスダイアログで定義したボックス名称が展開されます。

## 5. 論理マップ生成規則とマッピング規則

### 《C 言語》

繰り返し項目がある場合

```
struct {  
    unsigned char 項目データ名T  [[回数]] [2];  
    :  
} マップ名D={0xnn, 0xnn [, …] };
```

繰り返し項目がない場合

```
struct {  
    unsigned char 項目データ名T [2];  
    :  
} マップ名D={0xnn, 0xnn};
```

プッシュボタンボックスの場合

```
struct {  
    unsigned char ボックス名T [2]. ……※  
    :  
} マップ名D={0xnn, 0xnn};
```

注

プッシュボタンボックスダイアログで定義したボックス名称が展開されます。

### 5.5.4 動的変更テーブル

実行時に色などの属性を AP から変更するときには、XMAP3 が用意する属性変更用の定数テーブルの修飾名を表示属性の動的変更制御項目に代入します。この属性変更用の定数テーブルを動的変更テーブルといいます。動的変更テーブルは、COBOL 用、C 用の二つを用意しています

## 5.6 予約テキスト・フィールドのマッピング規則

予約テキスト・フィールドのマッピング規則を次に示します。なお、論理マップに予約テキスト・フィールドの情報は生成されません。

### (1) ドローで定義した予約項目名と結果

ドローで定義した項目名	データの表示結果
OpenTP1 がサポートしている予約項目名を指定	<ul style="list-style-type: none"> <li>・「指定した項目長」 = 「OpenTP1 で決められた項目長」 OpenTP1 が設定するデータが表示される。</li> <li>・「指定した項目長」 &gt; 「OpenTP1 で決められた項目長」 データが左寄せで表示される。また、残りは空白が表示される。</li> <li>・「指定した項目長」 &lt; 「OpenTP1 で決められた項目長」 データが左寄せで表示される。また、表示しきれないデータは切り捨てられる。</li> </ul>
OpenTP1 がサポートしていない予約項目名を指定	項目には何も表示されない。

OpenTP1 がサポートしていない予約項目を指定した場合はエラーとなります。このとき詳細コードには  $1036(040C)_{16}$  が返されます。詳しくは「付録 E XMAP3 のリターンコード・詳細コード」を参照してください。





# 6

## 画面定義とコーディングの 関連

この章では、画面定義と AP のコーディング、および画面の各オブジェクトと AP の関連について説明します。

---

6.1 画面属性と AP

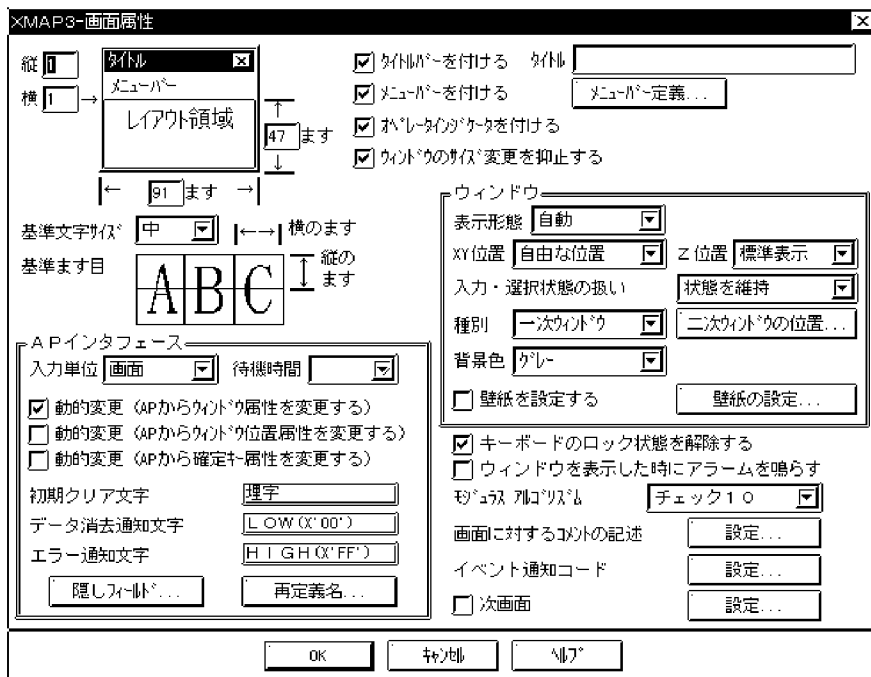
---

6.2 GUI 画面の各オブジェクトと AP

---

## 6.1 画面属性とAP

画面属性は画面属性ダイアログで設定します。AP インタフェースの各項目を指定すると、AP から動的に画面属性を変更できます。画面属性ダイアログを次に示します。



### 6.1.1 ウィンドウの表示形態を変更する

画面属性ダイアログで「動的変更（AP からウィンドウ属性を変更する）」を選びます。

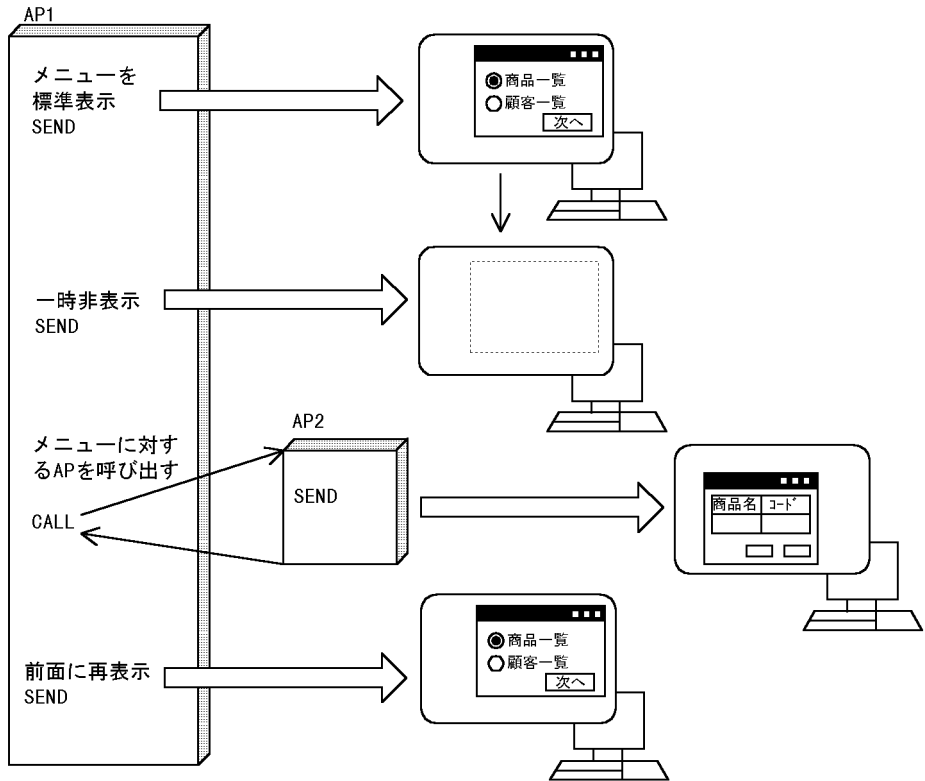
一時的にウィンドウを非表示にし、そのあと、無条件に手前に再表示したりする設定ができます。XMAP3 に対して、画面のオープン/クローズをしないで表示するウィンドウを切り替えられるので、処理の高速化が図れます。

#### (1) 定義画面

「6.1 画面属性とAP」の画面属性ダイアログで「動的変更（AP からウィンドウ属性を変更する）」を選びます。

なお、ウィンドウの表示形態は「表示形態」でも指定できます。どちらとも指定されている場合は、AP からの指定が優先されます。「動的変更（AP からウィンドウ属性を変更する）」が選ばれていない場合は、「表示形態」の標準値が適用されます。

## (2) AP との関係



## 6.1.2 ウィンドウ位置を変更する

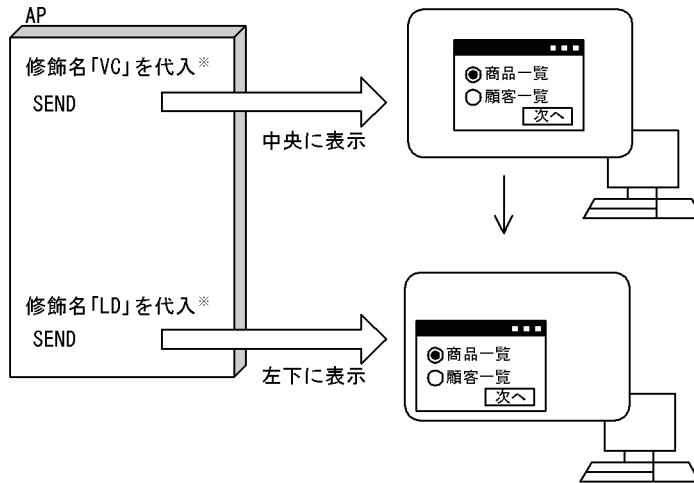
画面属性ダイアログで「動的変更 (AP からウィンドウ位置属性を変更する)」を選びます。AP からウィンドウを画面に表示する位置を変更できます。

## (1) 定義画面

「6.1 画面属性と AP」の画面属性ダイアログで「動的変更 (AP からウィンドウ位置属性を変更する)」を選びます。

なお、ウィンドウ位置は、「XY 位置」でも指定できます。どちらとも指定されている場合は、AP からの指定が優先されます。「動的変更 (AP からウィンドウ位置属性を変更する)」が選ばれていない場合は、「XY 位置」の標準値が適用されます。

## (2) AP との関係



注※ 修飾名「VC」は、ウィンドウを中央に表示するように指定されている標準修飾名です。  
修飾名「LD」は、ウィンドウを左下に表示するように指定されている標準修飾名です。  
これら以外にも、右上、右下、左上に表示させるよう指定されている修飾名があります。

### 6.1.3 PF キーなど確定キーの扱いを変更する

画面属性ダイアログで「動的変更 (AP から確定キー属性を変更する)」を選びます。AP から、ウィンドウ中で使用している確定キーの有効範囲などを変更できます。確定キーの有効範囲は、画面属性ダイアログの「イベント通知コード」で設定できます。画面定義時の仮定値は、標準ではすべての PF キー、PA キー、送信キー、割込キー、スクリーン消去キー、およびボタン / メニューが有効です。

「イベント通知コード」で確定キーを無効としている場合、動的変更で確定キーを有効にすることはできません。そのため、「イベント通知コード」で確定キーを有効に設定し、動的変更で無効とするように AP を作成します。

#### (1) 定義画面

通常、確定キーの設定は、「6.1 画面属性と AP」の画面属性ダイアログで「イベント通知コード」を選択して表示されるイベント通知コードダイアログで設定します。イベント通知コードダイアログを次に示します。

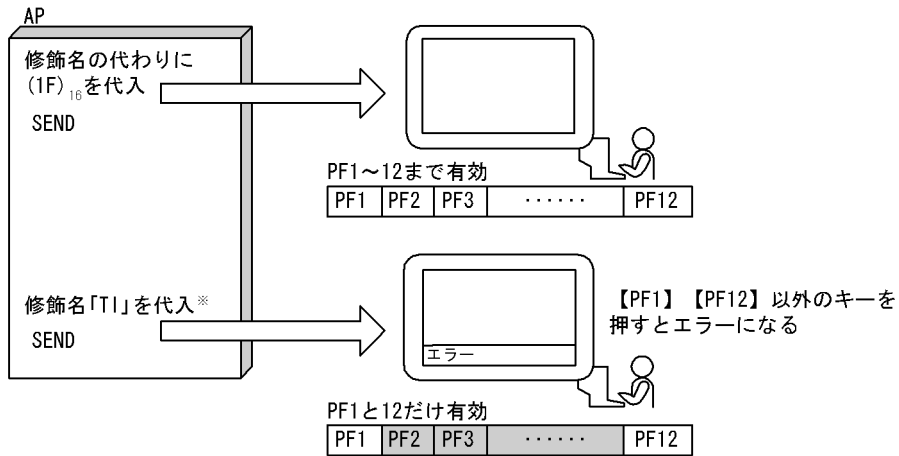


動的変更で確定キー属性を設定する場合、「6.1 画面属性とAP」の画面属性ダイアログで「動的変更（AP から確定キー属性を変更する）」をチェックします。「動的変更」をチェックすると、マップ生成後に属性変更の情報を受け取る領域（制御項目）が自動的に生成されます。APでは、制御項目に修飾名をセットすると、確定キー属性を変更できません。

修飾名は、ドロースेटアップの「修飾名の設定」ダイアログで「確定キー属性」を選択して設定します。動的変更の属性は、ドロースेटアップの表示属性の動的変更ダイアログの「確定キー属性」タブで設定します。ドロースेटアップでの設定については、マニュアル「XMAP3 開発・実行ガイド」を参照してください。

「イベント通知コード」での設定、および「動的変更（AP から確定キー属性を変更する）」のチェックがされている場合は、動的変更の指定が優先されます。「動的変更（AP から確定キー属性を変更する）」が選ばれていない場合は、「イベント通知コード」の標準値が適用されます。

(2) AP との関係



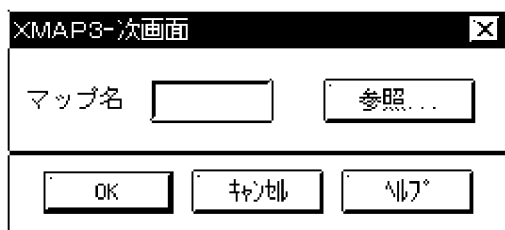
注※ 「TI」はユーザがドローセットアップで設定した修飾名です。この例題では、修飾名「TI」には「PF1」および「PF12」以外のキーを無効にするように指定されていると仮定しています。

6.1.4 次画面を設定する

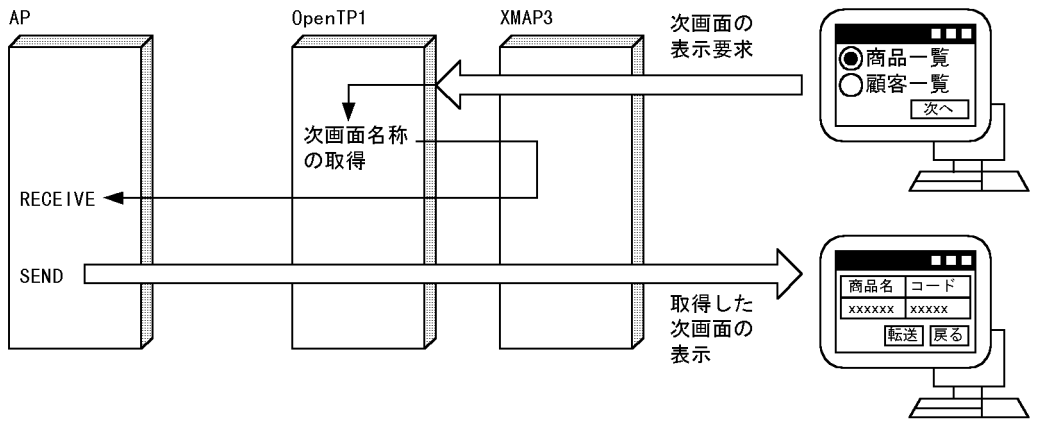
画面属性ダイアログで「次画面」を選びます。次画面を設定しておくことで AP で画面名称を定義していなくても実行時に画面名称を取得して表示できます。なお、次画面は OLTP サーバ構成の環境で設定できます。

(1) 定義画面

「6.1 画面属性と AP」の画面属性ダイアログで「次画面」を選び、次画面ダイアログを表示します。次画面ダイアログを次に示します。



## (2) AP との関係



## 6.2 GUI画面の各オブジェクトとAP

GUI画面の各オブジェクトとAPの関連について説明します。なお、ここで示す例は、ドローセットアップが標準値で設定されていることを前提とし、AP開発言語をCOBOL、マップ名を6文字、実行環境は対話型を想定しています。ドローセットアップについては、マニュアル「XMAP3 開発・実行ガイド」を参照してください。

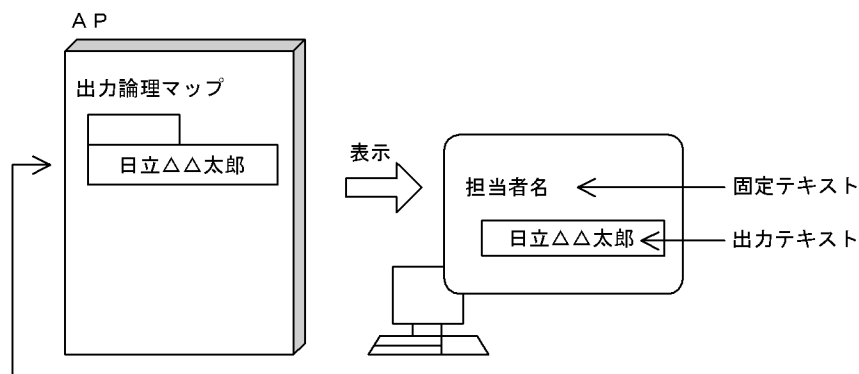
### 6.2.1 出力テキストの表示

#### (1) 仕様

ドローで定義した出力テキストボックスの論理マップに、APからテキスト「日立 太郎」を代入し、送信要求をするとXMAP3が「日立 太郎」を出力テキストボックスに表示します。

AP実行時の出力テキストの表示を次の図に示します。

図 6-1 出力テキストの表示



文字列「日立△△太郎」を出力論理マップに代入する

(凡例)

△：半角の空白

#### (2) 画面定義の概要

画面定義の概要を次に示します。

1. GUI画面定義ウィンドウで、ツールボックスの[出力テキスト(日本語)]を使用して出力テキストボックスを配置する。
2. 出力テキストボックスダイアログで次の表の項目を定義する。

注

「動的変更(APから表示属性を変更する)」をチェックすると、APから表示属性を



変更できます。詳細は「付録 A 属性情報一覧」を参照してください。

定義項目	定義内容
データ名	担当者名 -
データ型	文字 (XX)
桁	20
データ長	20 (この定義内容は固定)

### (3) 生成される論理マップ

マップ名が MAP115 の場合の例

```
01 MAP1150. (出力論理マップ)
:
03 MAP115-担当者名-0 PIC X(20). ... 出力テキストデータ格納エリア
```

#### 注

出力テキストボックスのデータ名を省略した場合、FIELDnnnn- (nnnn : 0001 ~ 9999) が仮定されます。

## 6. 画面定義とコーディングの関連

### (4) COBOL での AP

マップ名称が MAP115 の場合の例

```
PROCEDURE DIVISION.  
:  
MOVE ' 日立 太郎' TO MAP115-担当者名-0. ...出力論理マップ' 出力テキストを代入  
MOVE ' MAP115ND' TO 画面マップ名. ... マップ' 名称格納領域に物理マップ' 名を代入  
TRANSCIEVE DSP FROM MAP1150 INTO MAP1151. ... 画面の送受信
```

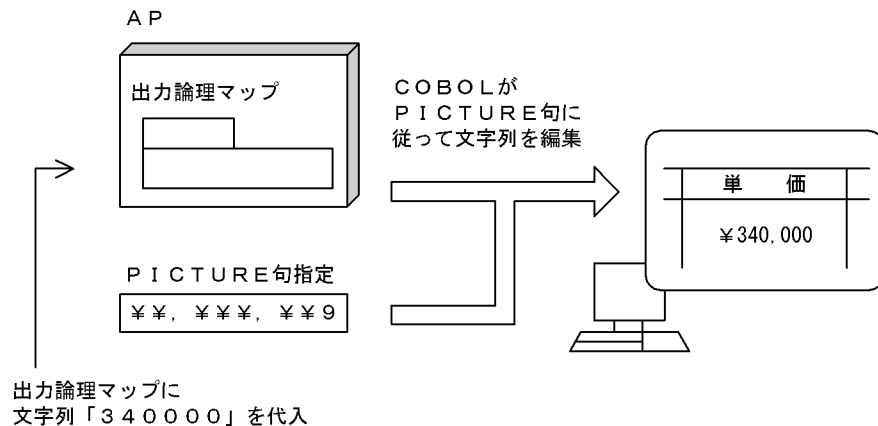
## 6.2.2 出力数字編集項目の利用 (COBOL だけ有効)

### (1) 仕様

単価の出力テキストボックスに「340000」を代入して送信要求すると、「¥340,000」と編集されて画面に表示されます。

AP 実行時の表示、および論理マップに格納される値を次の図に示します。

図 6-2 数字編集項目の利用



### (2) 画面定義の概要

画面定義の概要を次に示します。

1. GUI 画面定義ウィンドウで、ツールボックスの [出力テキストボックス (数字)] を使用して出力テキストボックスを配置する。
2. 出力テキストボックスの属性を出力テキストボックスダイアログで次の表の項目を定義する。

定義項目	定義内容
データ名	単価 1 -
データ型	¥, ¥¥9
桁	10
データ長	10 (この定義内容は固定)

### (3) 生成される論理マップ

マップ名が MAP024 の場合の例

01 MAP0240. (出力論理マップ)  
:  
03 MAP024-単価1-0 PIC ¥¥, ¥¥¥, ¥¥9 ... 「単価」の出力エリア

## (4) COBOL での AP

マップ名が MAP024 の場合の例

```

WORKING-STORAGE SECTION.
  COPY MAP024.      ... 出力論理マップの取り込み

PROCEDURE DIVISION
:
  MOVE 340000 TO MAP024-単価 1-0.    ... 出力テキストボックスにデータを代入
  MOVE ' MAP024ND' TO 画面マップ名.  ... 物理マップ名を代入
  TRANSCIEVE DSP FROM MAP024 INTO MAP024I.  ... 画面の送受信
    
```

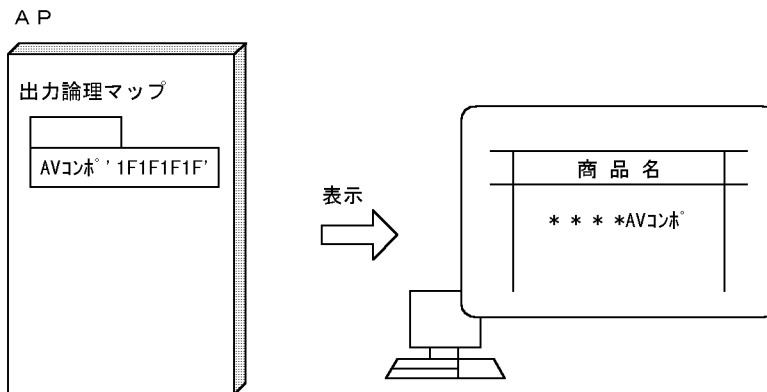
### 6.2.3 出力テキストでの埋字と桁寄せ向きの利用

#### (1) 仕様

出力テキストボックスの「商品名」に「AV コンボ'1F1F1F1F'」を代入すると右に桁寄せし、余った部分は「\*」で置き換えます。

AP 実行時の表示、および論理マップに格納される値を次の図に示します。

図 6-3 埋字と桁寄せ向きの利用



注

この図の「1F」はデータ有無コードを表します。データ有無コードを利用した AP 作成については「6.2.7 特定の入出力テキストだけを書き換える（データ有無コードの利用）」を参照してください。

#### (2) 画面定義の概要

画面定義の概要を次に示します。

1. GUI 画面定義ウィンドウで、ツールボックスの [出力テキストボックス (英数)] を

使用して「商品名」の出力テキストボックスを配置する。

2. 出力テキストボックスダイアログで次の表の項目を定義する。

定義項目	定義内容
データ名	商品名 1 -
データ型	文字 (XX)(この定義内容は固定)
桁	10
データ長	10 (この定義内容は固定)
桁寄せ	右
埋字	自由な埋字 (＊)

### (3) 生成される論理マップ

マップ名が MAP025 の場合の例

```
01 MAP0250. (出力論理マップ)
:
03 MAP025-商品名1-0 PIC X(10). ... 「商品名」の出力エリア
```

## 6. 画面定義とコーディングの関連

### (4) COBOL での AP

マップ名が MAP025 の場合の例

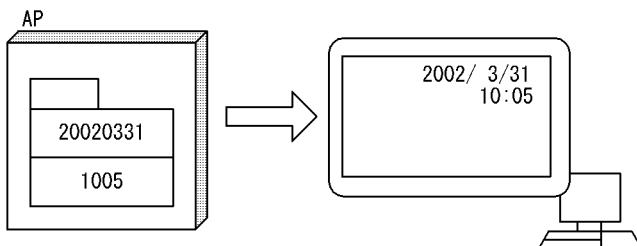
```
WORKING-STORAGE SECTION.  
COPY MAP025.          ... 出力論理マップの取り込み  
  
PROCEDURE DIVISION  
:  
MOVE ALL X'1F' TO MAP025G.          ... 出力論理マップを(1F)16でクリア  
  
MOVE 'AVコンホ' TO MAP025-商品名1-0.  ... 'AVコンホ'を代入  
TRANSCIEVE DSP FROM MAP0250 INTO MAP0251. ... 画面の送受信
```

## 6.2.4 下位項目を利用した日付テキスト・フィールドと時刻 テキスト・フィールド（出力）

### (1) 仕様

AP に「20020331」および「1005」を代入すると、それぞれ出力日付テキストダイアログまたは出力時刻テキストダイアログで指定した表示書式に合わせて「2002/ 3/31」および「10:05」と表示されます。

図 6-4 日付・時刻の出力



### (2) 画面定義の概要

画面定義の概要を次に示します。

#### (a) 出力日付の場合

1. GUI 画面定義ウィンドウで、ツールボックスの [ 出力日付テキスト ] を使用して、出力用の日付テキスト・フィールドを配置する。
2. 出力日付テキストダイアログで次の表の項目を定義する。

#### (b) 出力時刻の場合

1. GUI 画面定義ウィンドウで、ツールボックスの [ 出力時刻テキスト ] を使用して、出力用の時刻テキスト・フィールドを配置する。

2. 出力時刻テキストダイアログで次の表の項目を定義する。

ここでは出力日付のダイアログを例にします。

定義項目		定義内容
日付	表示書式	1997/ 8/ 7
	データ名	日付 -
	データ型	文字 (XX)
	下位項目	指定する
時刻	表示書式	08:07
	データ名	時刻 -
	データ型	文字 (XX)
	下位項目	指定する

The screenshot shows the 'XMAP3-出力日付テキスト' dialog box. It has several sections:

- Text Properties:** Font color (黒), font expansion (標準), font size (標準), font style (標準), font emphasis (標準), background color (透明), and line height (2.5ます).
- Options:** Checkboxes for '枠付き (テキストに枠を付ける)' and '反転表示する', and a '対称反復...' button.
- A Pが渡す項目 (Table):**

データ名	データ長	データ型	格納形式
日付-	8	文字 (XX)	YYYYMMDD
- Buttons:** OK, キャンセル, and F1?.

#### 下位項目の設定

1. 出力日付 (または時刻) テキストダイアログの下位項目の [ 設定 ] ボタンをクリックする。
2. 年月日を「年」「月」「日」、または時分を「時」「分」という下位項目として設定するために、次の表の項目を定義する。

定義項目		定義内容		
		年	月	日
日付	データ名	日付 - 年 -	日付 - 月 -	日付 - 日 -
	桁	4	2	2

## 6. 画面定義とコーディングの関連

定義項目	定義内容		
	年	月	日
データ長 (この定義内容は固定)	4	2	2
データ型	99999	99999	99999

時刻	定義項目	定義内容	
		時	分
	データ名	時刻 - 時 -	時刻 - 分 -
	桁	2	2
	データ長 (この定義内容は固定)	2	2
	データ型	99999	99999

### (3) 生成される論理マップ

マップ名が MAP001 の場合の例

```

01 MAP0010.
   :
02 MAP001G.
   :
03 MAP001-日付-0.
   04 MAP001-日付-年-0 PIC 9999.
   04 MAP001-日付-月-0 PIC 99.
   04 MAP001-日付-日-0 PIC 99.
03 MAP001-時刻-0.
   04 MAP001-時刻-時-0 PIC 99.
   04 MAP001-時刻-分-0 PIC 99.
    
```



#### (4) COBOL での AP

マップ名が MAP001 の場合の例

```

:
WORKING-STORAGE SECTION.
COPY MAP0010. ....出力論理マップの取り込み
:
PROCEDURE DIVISION.
:
MOVE ALL 'X' 1F' TO MAP001G. ....出力論理マップを(1F)16でクリア
MOVE 1997 TO MAP001-日付-年-0.   }
MOVE 09  TO MAP001-日付-月-0.   } .....日付(19970930)を代入
MOVE 30  TO MAP001-日付-日-0.   }
MOVE 10  TO MAP001-時刻-時-0.   } .....時刻(1005)を代入
MOVE 05  TO MAP001-時刻-分-0.   }
:
MOVE 'MAP001ND' TO 画面マップ名. ....物理マップ名を代入
TRANSCIVE DSP FROM MAP0010 INTO MAP0011.
                                     .....画面の送受信

```

### 6.2.5 入出力テキストを使用したデータの送受信

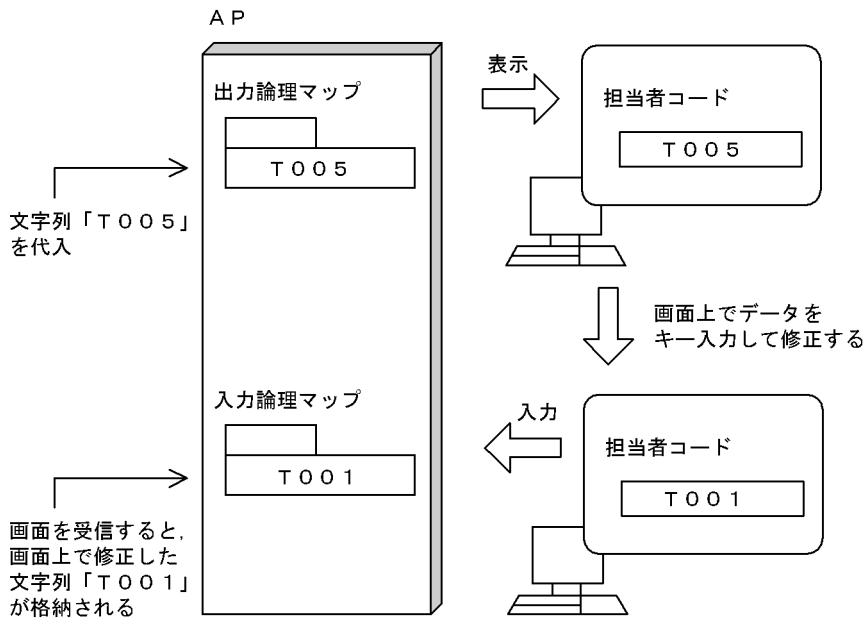
#### (1) 仕様

ドローで定義した入出力テキストボックスの出力論理マップに、AP からテキスト「T005」を代入して、送信要求をすると XMAP3 が画面上に「T005」を表示します。画面上で「T005」を「T001」に修正すると、XMAP3 が入力論理マップに「T001」を格納します。

AP 実行時での入出力テキストの表示例を次の図に示します。

## 6. 画面定義とコーディングの関連

図 6-5 入出力テキストの表示



### (2) 画面定義の概要

画面定義の概要を次に示します。

1. GUI 画面定義ウィンドウで、ツールボックスの [ 入出力テキスト ( 英数 ) ] を使用して入出力テキストボックスを配置する。
2. 入出力テキストボックスダイアログで次の表の項目を定義する。

定義項目	定義内容
使用目的	英数
データ名	担当者コード -
データ型	文字 ( XX ) ( この定義内容は固定 )
桁	4
データ長	4 ( この定義内容は固定 )

The screenshot shows a configuration window for 'MAP114-入出力テキストボックス'. It has several sections:

- 基本設定:** Includes fields for 'テキスト' (Text), '背景色' (Background Color), '枠高' (Frame Height), '文字色' (Text Color), '文字サイズ' (Text Size), '文字の書体' (Text Font), and '文字の強調' (Text Emphasis).
- 使用目的:** Includes '英数' (Alphanumeric) and '半角' (Half-width).
- 空白入力:** Includes '一部&全角 (半角)' (Partially/Full-width (Half-width)).
- 表示方法:** Includes '入力可能-標準表示' (Input possible - Standard display).
- 表示条件:** Includes '自動スクロール' (Auto scroll).
- その他の記述:** A text field for additional notes.
- A Pが受け取る項目:** A table with columns for 'データ名' (Data name), '桁' (Digits), '型' (Type), '初期値 (入力)' (Initial value (Input)). It shows '担当者コード' (Staff code) with '4' digits, '文字 (NN)' type, and '左' (Left) alignment.
- A Pが送る項目:** A table with columns for 'データ名' (Data name), '桁' (Digits), '型' (Type), '初期値 (出力)' (Initial value (Output)). It shows '担当者コード' (Staff code) with '4' digits, '文字 (NN)' type, and '左' (Left) alignment.

## 注 1

入力専用時は、「動的変更 (AP からデータを表示する)」はチェックしません。

## 注 2

「初期値」は指定しません。

## (3) 生成される論理マップ

マップ名が MAP114 の場合の例

01	MAP1140.	(出力論理マップ)	
:	03	MAP114-担当者コード-0 PIC X(4).	… 出力テキストを格納するエリア
01	MAP1141.	(入力論理マップ)	
:	03	MAP114-担当者コード-H.	
	04	MAP114-担当者コード-I PIC X(4).	… 入力テキストを格納するエリア

## 注

入出力テキストボックスのデータ名を省略した場合、FIELDnnnn (nnnn : 0001 ~ 9999) が仮定されます。

## 6. 画面定義とコーディングの関連

### (4) COBOL での AP

マップ名が MAP114 の場合の例

```
WORKING-STORAGE SECTION.  
COPY MAP1140.      ... 出力論理マップの取り込み  
COPY MAP1141.      ... 入力論理マップの取り込み  
COPY X3MODTBL.     ... 動的変更テーブルの取り込み  
77 FLG PIC X(1).   ... 入力完了確認フラグ  
COMMUNICATION SECTION.  
*ディスプレイに対する通信記述項  
CD   DSP FOR I-0 WS      ... 通信記述名:DSP  
    MAP NAME IS 画面マップ名      ... 物理マップ名称格納エリア  
:  
MOVE 'T005' TO MAP114-担当者コード-0.  ... 出力論理マップに担当者コード  
を格納  
MOVE 'MAP114ND' TO 画面マップ名.      ... マップ名称格納エリアに物理マップ名を格納  
TRANSCIVE DSP FROM MAP1140 INTO MAP1141.  ... 画面の送受信  
MOVE '0' TO FLG.  
PERFORM UNTIL FLG = '1'  
*修正したデータの正誤を判定する  
  IF MAP114-担当者コード-I = 'T001'  
  THEN      ... データが正しい場合  
    MOVE 'MAP114ND' TO 画面マップ名      ... マップ名称格納エリアに物理マップ名を格納  
    TRANSCIVE DSP FROM MAP1140 INTO MAP1141  ... 画面の送受信  
    MOVE '1' TO FLG      ... データ入力完了の確認  
  ELSE      ... データが誤りの場合  
    PERFORM エラ-時の処理  
  END-IF  
END-PERFORM.
```

## 6.2.6 入出力テキストでの埋字と桁寄せ向きの利用

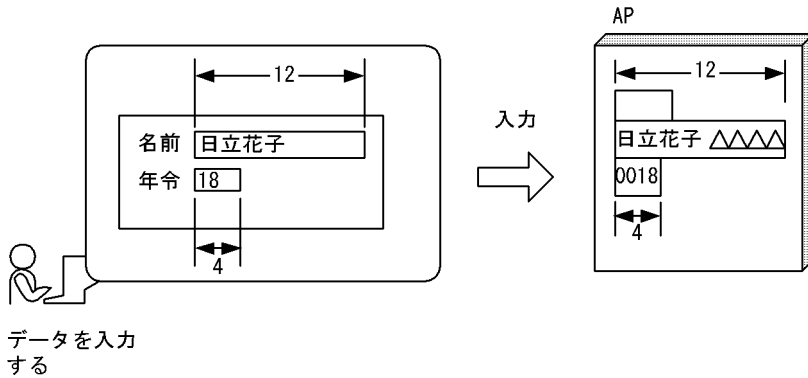
入出力テキストにデータを入力するときの仕様について説明します。

### (1) 仕様

入出力論理マップのデータ項目の長さに満たない長さのデータをキーボードなどから入力したとき、桁寄せ向きに従ってデータを寄せて、余った分は埋字に置き換えます。

名前に「日立花子」、「年令」に「18」と入力したとき、入力論理マップには「日立花子」、「0018」と代入されます。

図 6-6 入出力テキストにデータ入力するときの仕様



## (2) 画面定義の概要

画面定義の概要を次に示します。

1. GUI 画面定義ウィンドウで、ツールボックスの [ 入出力テキスト (日本語) ] および [ 入出力テキスト (数字) ] を使用して入出力テキストボックスを配置する。
2. 入出力テキストボックスで次の表の項目を定義する。

定義項目	定義内容	
使用目的	日本語	数字
データ名	名前 -	年齢 -
データ型	漢字 (NN)	99999
桁寄せ (この定義内容は固定)	左	右
埋字 (この定義内容は固定)	(スペース)	0 (ゼロ)
桁	12	4
データ長 (この定義内容は固定)	12	4

## 6. 画面定義とコーディングの関連

### 使用目的が日本語の場合の定義例

XMAP03-入出力テキストボックス							
種別	テキスト						
使用目的	日本語						
詳細目的	漢字						
空白入力	一部と全角(全・半角)						
表示方法	入力可能-標準表示						
表示書式							
遷移条件	自動スキップ						
背景色	白						
文字色	黒						
文字の書体	標準						
字高	2.5文字						
文字サイズ	標準						
文字の強調	標準						
反転表示する	<input type="checkbox"/>						
カーソル設定 (初期フォーカスを本ボックスに設定する)	<input type="checkbox"/>						
入力済み (入力しなくても表示データをAPIに送す)	<input type="checkbox"/>						
自動送信 (入力直後にAPへ制御を渡す)	<input type="checkbox"/>						
入力必須 (入力しない時はエラーにする)	<input type="checkbox"/>						
カクカク (遷移状態にする)	<input type="checkbox"/>						
この記述							
A Pが受け取る項目							
データ名	名前	データ型	漢字(NN)	桁寄せ	埋字	スペース	初期値(入力)
<input checked="" type="checkbox"/> A Pからデータを表示する							
A Pが送す項目							
データ名	名前	データ型	漢字(NN)	桁寄せ	埋字	スペース	初期値(出力)
<input type="checkbox"/> 動的変更 (A Pから表示属性を変更する)							

### 使用目的が数字の場合の定義例

XMAP03-入出力テキストボックス							
種別	数字						
使用目的	数字						
詳細目的	数字-数字記号						
空白入力	全角(半角)						
表示方法	入力可能-標準表示						
表示書式							
遷移条件	自動スキップ						
背景色	白						
文字色	黒						
文字の書体	標準						
字高	2.5文字						
文字サイズ	標準						
文字の強調	標準						
反転表示する	<input type="checkbox"/>						
カーソル設定 (初期フォーカスを本ボックスに設定する)	<input type="checkbox"/>						
入力済み (入力しなくても表示データをAPIに送す)	<input type="checkbox"/>						
自動送信 (入力直後にAPへ制御を渡す)	<input type="checkbox"/>						
入力必須 (入力しない時はエラーにする)	<input type="checkbox"/>						
カクカク (遷移状態にする)	<input type="checkbox"/>						
この記述							
A Pが受け取る項目							
データ名	年令	データ型	9999	桁寄せ	埋字	ゼロ	初期値(入力)
<input checked="" type="checkbox"/> A Pからデータを表示する							
A Pが送す項目							
データ名	年令	データ型	9999	桁寄せ	埋字	ゼロ	初期値(出力)
<input type="checkbox"/> 動的変更 (A Pから表示属性を変更する)							

## (3) 生成される論理マップ

### マップ名が MAP001 の場合の例

```

01 MAP0010.
  03 MAP001-名前-0 PIC N(6).
  03 MAP001-年令-0 PIC 9999.
01 MAP0011.
  03 MAP001-名前-1.
  04 MAP001-名前-1 PIC N(6).
  03 MAP001-年令-1.
  04 MAP001-年令-1 PIC 9999.
  
```

## (4) COBOL での AP

```

COPY MAP0010.
COPY MAP0011.
PROCEDURE DIVISION
:
:
TRANSCEIVE DSP FROM MAP0010 INTO MAP0011. ...データを入力
:
:
IF MAP001-年齢-1 = 20      ... 埋字/けた寄せされたデータ
  THEN                    をそのまま判定
  PERFORM HATACHI-処理.

```

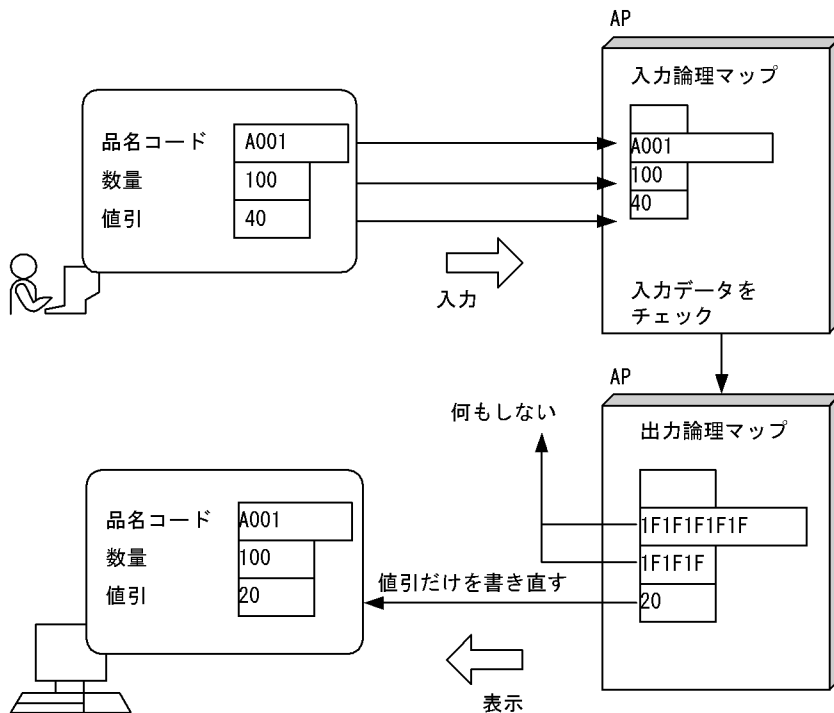
## 6.2.7 特定の入出力テキストだけを書き換える（データ有無コードの利用）

特定のデータだけを書き換える場合について説明します。

## (1) 仕様

品名コード、数量および値引を入力し、AP でチェックします。AP でチェックして、値引だけを書き直します。

図 6-7 データ有無コードの利用



## 6. 画面定義とコーディングの関連

### (2) 画面定義の概要

入出力テキストを定義します。初期値は指定しなくてもかまいません。

### (3) 生成される論理マップ

マップ名が MAP001 の場合の例

```
01 MAP001.  
01 MAP0010.  
  :  
  :  
02 MAP001G.  
  03 MAP001-品名コード-0 PIC X(5).  
  03 MAP001-数量-0 PIC 999.  
  03 MAP001-値引-0 PIC 99.
```

### (4) COBOL での AP

```
WORKING-STORAGE SECTION.  
  COPY MAP0010.  
  :  
  :  
*****  
*   通信記述           *  
*****  
COMMUNICATION SECTION.  
*   ディスプレイに対する通信記述項  
  CD DSP FOR I-O WS  
  :  
  MAPPING MODE IS マッピングモード  
  :  
PROCEDURE DIVISION.  
  :  
  :  


|                     |                       |
|---------------------|-----------------------|
| MOVE 2 TO マッピングモード. | ……論理マップだけのマッピングモードを設定 |
|---------------------|-----------------------|


|                            |                                      |
|----------------------------|--------------------------------------|
| MOVE ALL X'1F' TO MAP001G. | ……データ有無コードの(1F) <sub>16</sub> でクリアする |
|----------------------------|--------------------------------------|


|                         |           |
|-------------------------|-----------|
| MOVE 20 TO MAP001-値引-0. | ……値引だけを代入 |
|-------------------------|-----------|

  
TRANSCIEVE
```

## 6.2.8 初期値の利用

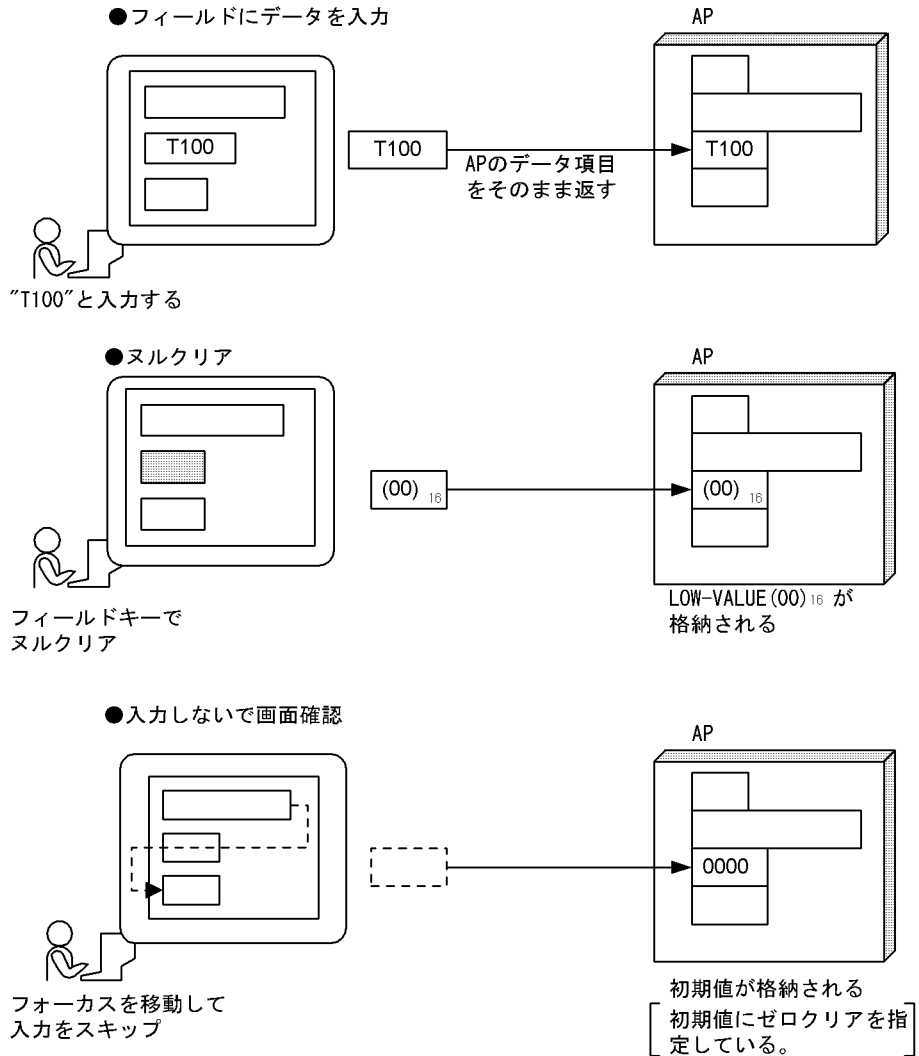
初期値の利用について説明します。



## (1) 仕様

初期値は、入出力テキストボックスで設定しておきます。設定する初期値は「ゼロクリア」です。設定した初期値が反映される場合と、反映されない場合を次の図に示します。

図 6-8 初期値の利用



## (2) 画面定義の概要

データ名、長さおよび初期値（入力）「ゼロクリア」を指定します。ヌルクリア時のデータ消去通知文字は、(00)<sub>16</sub> です。

## 注

ゼロクリアを指定できるのは、使用目的が「英数」「パスワード」および「MCR」

## 6. 画面定義とコーディングの関連

のときだけです。

### (3) 論理マップ

マップ名が MAP001 の場合の例

```
01 MAP0010.  
  :  
01 MAP0011.  
  :  
03 MAP001-コード-H.  
  04 MAP001-コード-I PIC X(4).
```

### (4) COBOL での AP

```
:  
TRANSCEIVE DSP FORM MAP0010 TO MAP0011.  
:  
:  
EVALUATE MAP001-コード-H  
  WHEN LOW-VALUE  
    PERFORM クリア処理.  
  WHEN '0000'  
    PERFORM 未入力処理  
END-EVALUATE.  
IF MAP001-コード-I = 'T100'  
  THEN  
:  
:
```

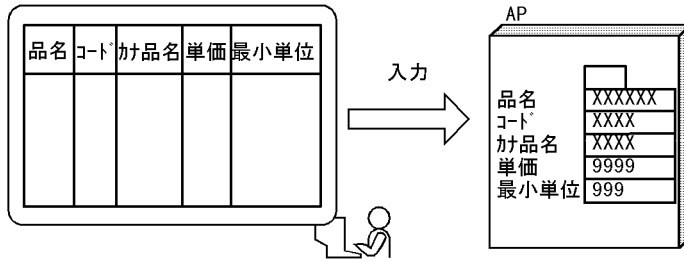
## 6.2.9 入力できるデータ型の設定（詳細目的の利用）

入出力テキストの「詳細目的」を設定し、入出力テキストに入力できるデータ型を設定します。

### (1) 仕様

「品名」「コード」「カナ品名」「単価」「最小単位」データを持つ商品マスタの登録をします。それぞれのデータの詳細目的を次の図に示します。

図 6-9 入出力テキストでの使用目的，詳細目的



入出力項目	使用目的	詳細目的
品名	日本語	混合(全角・半角)
コード	英数	英大・数
カナ品名	カナ	カナ・数・長音
単価	金額	数字
最小単位	数値	数字

## (2) 画面定義の概要

画面定義の概要を示します。なお，ここでは各データ項目の使用目的，および詳細目的の定義で必要な部分だけを載せています。入出力テキストの全体の画面定義は，「6.2.5 入出力テキストを使用したデータの送受信」を参照してください。

目的	定義名
使用目的 <input type="text" value="日本語"/> 詳細目的 <input type="text" value="混在(全角・半角)"/>	「品名」の定義
使用目的 <input type="text" value="英数"/> 詳細目的 <input type="text" value="英大・数"/>	「コード」の定義
使用目的 <input type="text" value="カナ"/> 詳細目的 <input type="text" value="カ・数・長音"/>	「カナ品名」の定義
使用目的 <input type="text" value="金額"/> 詳細目的 <input type="text" value="数字"/>	「単価」の定義
使用目的 <input type="text" value="数値"/> 詳細目的 <input type="text" value="数字"/>	「最小単位」の定義

### (3) 詳細目的とデータ型

設定した詳細目的とデータ型については「付録 A.9 表示属性の解説」の表 A-9 を参照してください。

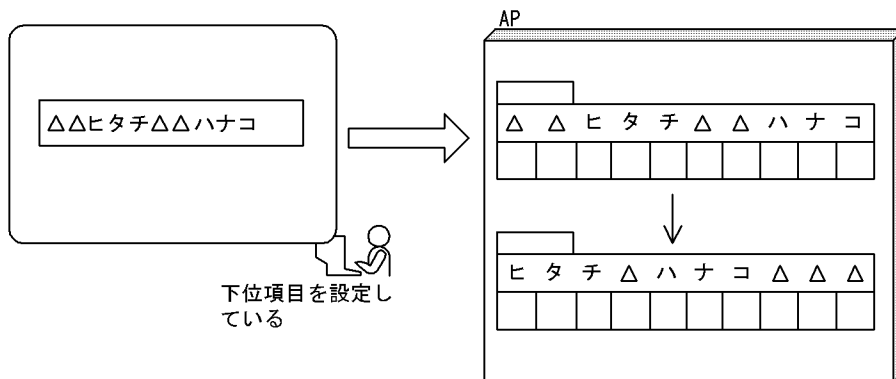
## 6.2.10 下位項目を利用した入出力テキスト

入力したデータは AP で一文字ずつチェックされます。入出力テキストボックスダイアログで「下位項目」の [ 設定 ] ボタンを選ぶと、論理マップ中に、データを一文字の繰り返しとしするチェック用の下位項目が生成されます。AP でのチェックに利用できます。

### (1) 仕様

入出力テキストに「 ヒタチ ハナコ」と入力すると、AP にはそのままデータが渡るので、AP 側でデータ先頭の空白をすべて削除し、データ中央の空白を一つにし、残りのデータ長に空白を埋める処理を行います。

図 6-10 下位項目を利用したときの入力データの扱い



### (2) 画面定義の概要

1. 入出力テキストボックスダイアログで「下位項目」の [ 設定 ] ボタンを選ぶ。
2. 下位項目ダイアログが表示されます。
3. 下位項目ダイアログで「一文字の反復」を指定する。

上位項目 A Pが受け取る項目

データ名 氏名 桁 10 データ長 10

下位項目

データ名	桁	データ長	データ型
氏名-チェック	10	10	1文字の反復

氏名-チェック 10 10 1文字の反復

OK キャンセル ヘルプ

### (3) 生成される論理マップ

マップ名が MAP001 の場合の例

```

:
04 MAP001-氏名-1.
05 MAP001-氏名-チェック-1 PIC X(1) OCCURS(10).

```

## (4) COBOL での AP

```

WORKING-STORAGE SECTION.
  01  氏名エリア
    02  氏名          PIC X OCCURS 10.  ... 「氏名」
  01  W-CNT          PIC S9(4).        ... カウンタ
  01  W-CNT1         PIC S9(4).        ... カウンタ1
      :
COPY  MAP0010.      ... 出力論理マップの取り込み
COPY  MAP0011.      ... 入力論理マップの取り込み
      :
PROCEDURE DIVISION.
      :
MOVE  'MAP001ND' TO 画面マップ名... 物理マップ名の代入
TRANSCEIVE DSP FROM MAP0010 INTO MAP0011... 画面の送受信
      :
MOVE ALL SPACE TO 氏名.          ... 「氏名」をスペースクリア
MOVE 1 TO W-CNT.                ... カウンタの初期化
MOVE 1 TO W-CNT1.              ... カウンタ1の初期化
PERFORM UNTIL ( W-CNT > MAP001-氏名-L )
      ... 入力したバイト数分ループする
  EVALUATE MAP001-氏名-チェック( W-CNT )... 入力文字列のチェック
    WHEN SPACE                    ... スペース入力の場合
      COMPUTE W-CNT = W-CNT + 1 ... 次の入力文字列へ
    WHEN OTHER                    ... 文字入力の場合
      MOVE MAP001-氏名-チェック( W-CNT ) TO 氏名( W-CNT1 )
      ... 文字を「氏名」に代入
      COMPUTE W-CNT1 = W-CNT1 + 1
      ... 「氏名」用カウンタのカウンタアップ
    IF MAP001-氏名-チェック( W-CNT + 1 )
      = SPACE ... 次のスペースか
      THEN
        MOVE SPACE TO 氏名( W-CNT1 )... スペースを「氏名」に代入
        COMPUTE W-CNT1 = W-CNT1 + 1
        ... 「氏名」用カウンタのカウンタアップ
      END-IF
    COMPUTE W-CNT = W-CNT + 1 ... 次の入力文字列へ
  END-EVALUATE
END-PERFORM.

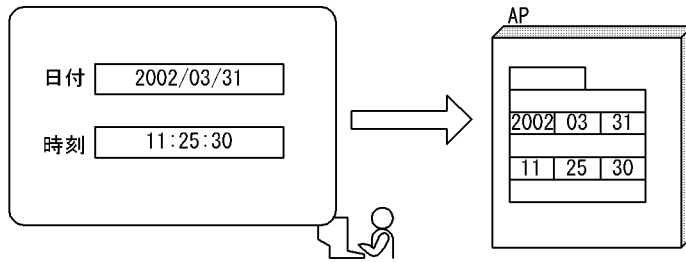
```

## 6.2.11 下位項目を利用した日付テキスト・フィールドと時刻テキスト・フィールド（入出力）

## (1) 仕様

入出力日付テキストダイアログまたは入出力時刻テキストダイアログで表示書式を指定すると、入出力画面に表示されます。ここで入力したデータが AP に渡されるときは、デリミタは削除されます。AP から画面にデータが渡されるときは、指定した表示書式に合わせて表示されます。

図 6-11 日付・時刻の入出力



## (2) 画面定義の概要

画面定義の概要を次に示します。

### (a) 入出力日付

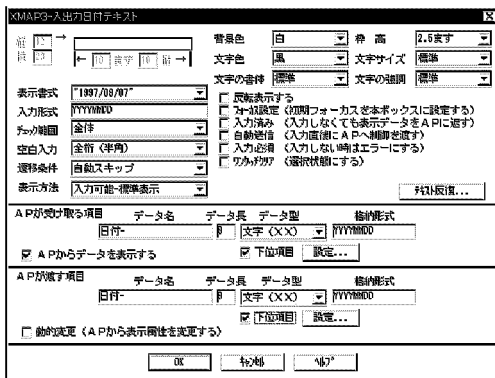
1. GUI 画面定義ウィンドウで、ツールボックスの [ 入出力日付テキスト ] を使用して、入出力用の日付テキスト・フィールドを配置する。
2. 入出力日付テキストダイアログで次の表の項目を定義する。

### (b) 入出力時刻

1. GUI 画面定義ウィンドウで、ツールボックスの [ 入出力時刻テキスト ] を使用して、入出力用の時刻テキスト・フィールドを配置する。
2. 入出力時刻テキストダイアログで次の表の項目を定義する。

定義項目		定義内容
日付	表示書式	1997/08/07
	データ名	日付 -
	データ型	文字 (XX)
	下位項目	指定する
時刻	表示書式	21:08:07
	データ名	時刻 -
	データ型	文字 (XX)
	下位項目	指定する

6. 画面定義とコーディングの関連



下位項目の設定

1. 入出力日付（時刻）テキストダイアログの「AP が受け取る項目」および「AP が渡す項目」それぞれのデータ名に対して下位項目を設定する。
2. 年月日を「年」「月」「日」、または時分秒を「時」「分」「秒」という下位項目として設定するために、次の表の項目を定義する。

定義項目		定義内容		
		年	月	日
日付	データ名	日付 - 年 -	日付 - 月 -	日付 - 日 -
	桁	4	2	2
	データ長（この定義内容は固定）	4	2	2
	データ型	99999	99999	99999

定義項目		定義内容		
		時	分	秒
時刻	データ名	時刻 - 時 -	時刻 - 分 -	時刻 - 秒 -
	桁	2	2	2
	データ長（この定義内容は固定）	2	2	2
	データ型	99999	99999	99999



MAP00-下位項目

上位項目 [A Pが受け取る項目]

データ名 桁 データ長

日付- 2 2

---

下位項目

データ名	桁	データ長	データ型
日付-年-	4	4	9999
日付-月-	2	2	9999
日付-日-	2	2	9999

↑ 上へ挿入

↓ 下へ挿入

移動

削除

日付-日- 2 9999

OK キャンセル ヘルプ

### (3) 生成される論理マップ

```

01 MAP0010. (出力論理マップ)
:
02 MAP001G.
:
03 MAP001-日付-0.
04 MAP001-日付-年-0 PIC 9999.
04 MAP001-日付-月-0 PIC 99.
04 MAP001-日付-日-0 PIC 99.
03 MAP001-時刻-0.
04 MAP001-時刻-時-0 PIC 99.
04 MAP001-時刻-分-0 PIC 99.
04 MAP001-時刻-秒-0 PIC 99.

01 MAP0011. (入力論理マップ)
:
02 MAP001K.
:
03 MAP001-日付-H.
04 MAP001-日付-I.
05 MAP001-日付-年-I PIC 9(4).
05 MAP001-日付-月-I PIC 9(2).
05 MAP001-日付-日-I PIC 9(2).
03 MAP001-時刻-H.
04 MAP001-時刻-I.
05 MAP001-時刻-時-I PIC 9(2).
05 MAP001-時刻-分-I PIC 9(2).
05 MAP001-時刻-秒-I PIC 9(2).

```

## 6. 画面定義とコーディングの関連

### (4) COBOL での AP

マップ名が MAP001 の場合の例

```

:
WORKING-STORAGE SECTION.
01 日付. ....日付用ワークエリア
   02 日付-年 PIC 9(4).
   02 日付-月 PIC 9(2).
   02 日付-日 PIC 9(2).
:
01 時刻. ....時刻用ワークエリア
   02 時刻-時 PIC 9(2).
   02 時刻-分 PIC 9(2).
   02 時刻-秒 PIC 9(2).
:
COPY MAP0010. ....出力論理マップの取り込み
COPY MAP0011. ....入力論理マップの取り込み
:
PROCEDURE DIVISION.
:
MOVE 'MAP001ND' TO 画面マップ名. ....物理マップ名を代入
TRANSCIVE DSP FROM MAP0010 INTO MAP0011. ....画面の送受信
MOVE MAP001-日付-I TO 日付. ....画面から入力した日付をワークエリアに代入
MOVE MAP001-時刻-I TO 時刻. ....画面から入力した時刻をワークエリアに代入

```

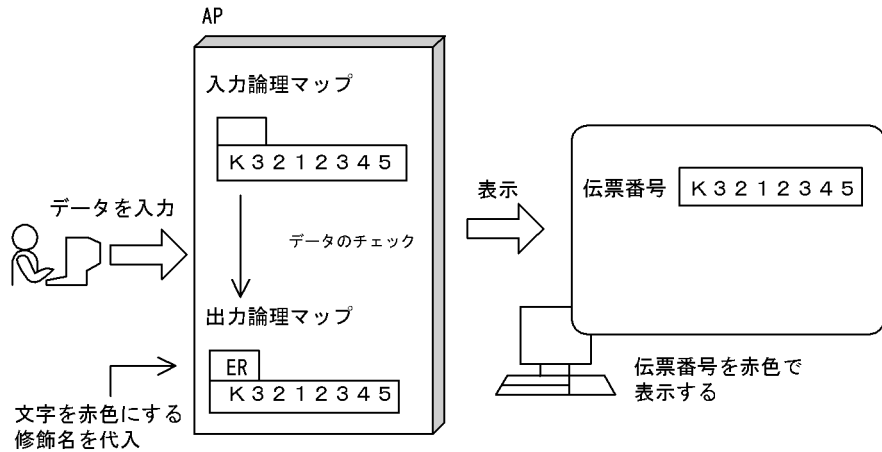
## 6.2.12 テキストの文字色の動的変更

入出力テキストおよび出力テキストの文字色を AP で変更する仕組みを説明します。

### (1) 仕様

「伝票番号」の入出力テキストボックスに入力されたデータが誤りの場合、注目してもらうために、文字を赤色で表示します。AP でテキスト色を変更する仕組みを次の図に示します。

図 6-12 入出力テキストの文字色を AP で変更する



## 注

出力テキストでも同じように、AP から表示属性を変更できます。表示属性の動的変更は文字色だけでなく、オブジェクトの表示 / 非表示の切り替えもできます。これらの動的変更はドロースेटアップで設定します。

## (2) 画面定義の概要

画面定義の概要を次に示します。

1. GUI 画面定義ウィンドウでツールボックスの [ 入出力テキスト ( 英数 ) ] を使用して入出力テキストボックスを配置する。
2. 入出力テキストボックスダイアログで次の表の項目を定義する。

定義項目	定義内容
データ名	伝票番号 -
桁	8
データ長	8 ( この定義内容は固定 )
動的変更 ( AP から表示属性を変更する )	指定する

## 6. 画面定義とコーディングの関連

The screenshot shows the 'XMAP3-入力テキストボックス' dialog box. It is divided into several sections:

- Text Input Settings:** Includes fields for '項目' (Text), '使用目的' (Use Purpose), '詳細目的' (Detail Purpose), '空白入力' (Blank Input), '表示方法' (Display Method), '換字書式' (Conversion Format), and '遷移条件' (Transition Condition).
- Appearance and Behavior:** Includes '背景色' (Background Color), '文字色' (Text Color), '文字の書体' (Text Font), '文字の強調' (Text Emphasis), and several checkboxes for '反転表示する' (Invert Display), 'フォーカス指定' (Focus Designation), '入力済み' (Input Completed), '自動送信' (Auto Send), '入力必須' (Input Required), and 'ウインドウ? (選択状態にする)' (Window? (Select State)).
- Additional Description:** A text field for 'その他の記述' (Other Description) and a '詳細戻理...' (Detail Return...) button.
- A Pが受け取る項目 (A P Receives Item):** A table with columns for 'データ名' (Data Name), '桁' (Digits), 'データ長' (Data Length), 'データ型' (Data Type), '桁寄せ' (Justify), '埋字' (Fill Character), and '初期値(入力)' (Initial Value (Input)). The example shows '伝票番号' (Invoice Number) with 8 digits, type '数字(\\N)', left justify, space fill, and '指定しない' (None).
- A Pが渡す項目 (A P Passes Item):** A similar table for output. The example shows '伝票番号' (Invoice Number) with 8 digits, type '数字(\\N)', left justify, space fill, and '指定しない' (None).
- Dynamic Change:** A checkbox for '動的変更 (A Pから表示属性を変更する)' (Dynamic Change (Change Display Attributes from A P)) and a '動的変更...' (Dynamic Change...) button.
- Buttons:** 'OK', 'キャンセル' (Cancel), and 'ヘルプ?' (Help?) buttons at the bottom.

### (3) 生成される論理マップ

マップ名が MAP439 の場合の例

```

01 MAP4390.
:
03 MAP439-伝票番号-A PIC X(2). ... 動的变化情報格納エリア
03 MAP439-伝票番号-O PIC X(8). ... 出力データ格納エリア
01 MAP4391.
:
03 MAP439-伝票番号-H.
04 MAP439-伝票番号-I PIC X(8). ... 入力データ格納エリア
    
```

#### 注

AP 実行時に入出力テキストボックスの表示属性を変更する場合、マップ生成後に生成される動的变化情報格納エリア（制御項目）に修飾名を代入します。修飾名は、XMAP3 が動的变化テーブルとして用意しています。AP では、この定数テーブルの修飾名を制御項目に代入してください。

## (4) COBOL での AP

マップ名が MAP539 の場合の例

### 1. 通常を表示をする

表示属性は画面定義の標準属性を適用する。

```
MOVE XMAP-NODATA TO MAP439-伝票番号-A. ……標準の表示属性を
MOVE 表示データ TO MAP439-伝票番号-0. ……使用することを指示
MOVE 0          TO マッピングモード.
TRANSCEIVE
```

### 2. 文字色を赤に変更して表示する

修飾名「ER」は標準の変更属性を使用します。この赤色をほかの色にするためには、あらかじめドロースettingsアップで設定が必要です。

あらかじめ設定されている修飾名と表示属性については、「付録 A 属性情報一覧」を参照してください。

```
WORKING-STORAGE SECTION.
COPY X3MODTBL. ……動的変更テーブルの取り込み
:
MOVE XMAP-IN-ATTR1 TO MAP439-伝票番号-A. ……修飾名を代入
MOVE XMAP-NODATA TO MAP439-伝票番号-0. ……表示データを変更しない指示
MOVE 2 TO マッピングモード. ……「論理マップだけ」の
TRANSCEIVE ……マッピングモードを指定
```

### 注

マッピングモードが「論理マップだけ」、表示形態が「一部上書」の条件で画面表示する場合、修飾名としてデータ有無コードや定義されていない修飾名を代入しても、ドロー定義時の属性には戻りません。ドロー定義時の属性に戻したい場合は、ドロースettingsアップですべての属性を「変更なし」に設定した修飾名を代入してください。

## 6.2.13 パスワードの利用

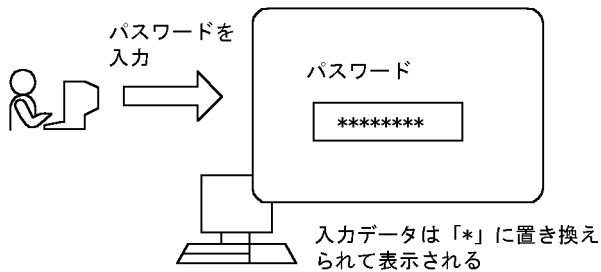
### (1) 仕様

パスワードの入出力テキストボックスにデータを入力します。表示属性の指定で、表示方法を「入力可能 - パスワード表示」にすると、入力したパスワードを「\*」に置き換えて表示できます。

AP 実行時での表示を次の図に示します。

## 6. 画面定義とコーディングの関連

図 6-13 パスワードの利用

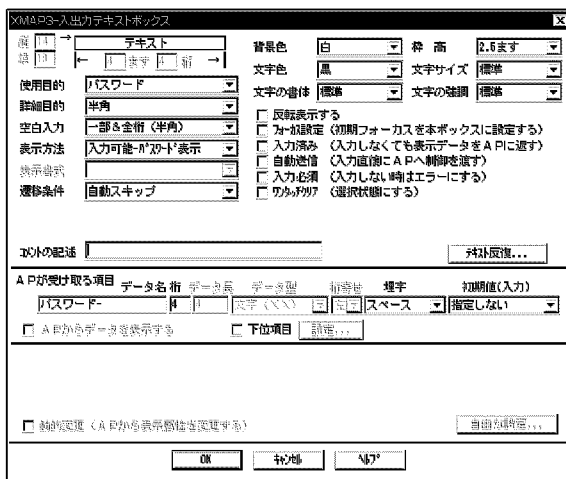


### (2) 画面定義の概要

画面定義の概要を次に示します。

1. GUI 画面定義ウィンドウで、ツールボックスの [ 入出力テキスト (英数) ] を使用して入出力テキストボックスを配置する。
2. 入出力テキストボックスダイアログで次の表の項目を定義する。

定義項目	定義内容
使用目的	パスワード
データ名	パスワード -
桁	4
データ長	4 (この定義内容は固定)
データ型	文字 (XX) (この定義内容は固定)



注

「初期値」は指定しません。

### (3) 生成される論理マップ

マップ名が MAP436 の場合の例

```
01 MAP4360. (出力論理マップ)
:
01 MAP4361. (入力論理マップ)
:
03 MAP436-パスワード-H.
04 MAP436-パスワード-I PIC X(4).
```

### (4) COBOL での AP

マップ名が MAP436 の場合の例

```
WORKING-STORAGE SECTION.
  COPY MAP4360.          ... 出力論理マップの取り込み
  COPY MAP4361.          ... 入力論理マップの取り込み
:
PROCEDURE DIVISION.

TRANSCEIVE DSP FROM MAP4360 INTO MAP4361.    ... 画面の送受信
* 入力されたパスワードの正誤判定処理      (表示と入力)
IF MAP436-パスワード-I = 'XMAP'
  THEN                                         ... 入力データが正しい場合
    MOVE MAP436-パスワード-I TO パスワードエリア
  ELSE                                         ... 入力データが誤りの場合
    PERFORM エラー時の処理
END-IF.
```

## 6.2.14 MCR 入力専用のテキストの利用

### (1) 仕様

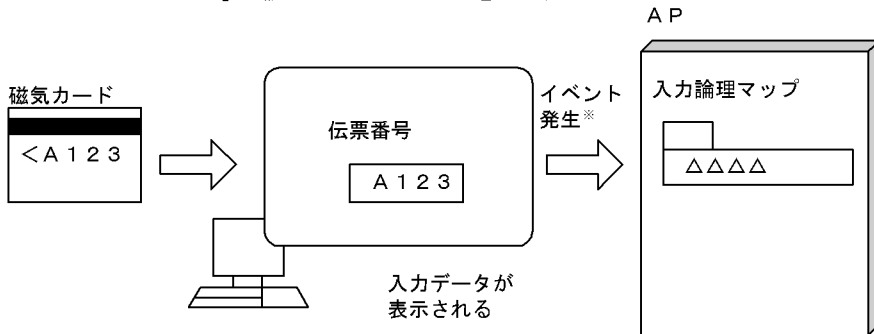
入出力テキストボックスに MCR からデータを入力します。ここでは正しい ID コードが「>」であることが仮定されています。

入力されるデータと AP 実行時の表示を次の図に示します。

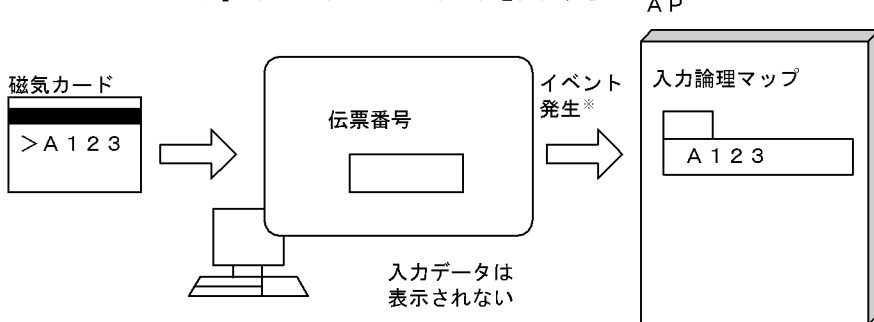
## 6. 画面定義とコーディングの関連

図 6-14 MCR 入力専用のテキストの利用

1. IDコードが「<」（誤り）のMCRデータを入力する



2. IDコードが「>」（正しい）のMCRデータを入力する



注※ イベント通知項目にはクリア文字が設定されます。  
イベント通知項目とは、イベント通知コードを格納する領域です。

### （2）画面定義の概要

画面定義の概要を次に示します。

1. GUI画面定義ウィンドウで、ツールボックスの[入出力テキスト(英数)]を使用して入力テキストボックスを配置する。
2. 入出力テキストの属性を入出力テキストボックスダイアログで次の表の項目を定義する。

定義項目	定義内容
使用目的	MCR
データ名	MCR -
桁	10
データ長	10(この定義内容は固定)



### (3) 生成される論理マップ

```

01 MAP4370. (出力論理マップ)
:
01 MAP4371. (入力論理マップ)
:
03 MAP437-MCR-H.
04 MAP437-MCR-I PIC X(10). ... MCRデータの入力エリア

```

### (4) COBOL での AP

マップ名が MAP437 の場合の例

```

WORKING-STORAGE SECTION.
COPY MAP4370. ... 出力論理マップの取り込み
COPY MAP4371. ... 入力論理マップの取り込み
COPY X3MODTBL. ... 動的変更テーブルの取り込み
COMMUNICATION SECTION.
CD DSP FOR I-O WS ... 通信記述名:DSP
MAP NAME IS 画面マップ名 ... 物理マップ名称格納エリア
:
PROCEDURE DIVISION.
MOVE 'MAP437ND' TO 画面マップ名. ... 物理マップ名を代入
TRANSCIEVE DSP FROM MAP4370 INTO MAP4371. ... 画面の送受信
IF MAP437-MCR-I = 'A123' ... 入力データの参照
THEN
:

```

## 6.2.15 フィールドボックスの入出力

フィールドボックスの入出力は、フィールドの集まりとして定義されます。

AP とのインタフェース上は、各フィールドの入出力となります。各機能の使用方法については、次に示す参照先をご覧ください。

フィールドボックスの入出力		参照先
出力フィールド	文字系	出力テキストボックス
	数字系	6.2.1 ~ 3
	日付・時刻	6.2.4
入出力フィールド	文字系	入出力テキストボックス
	数字系	
	パスワード	
	MCR	
	日付・時刻	6.2.11
ポップアップ フィールド	自動・固定	ポップアップテキスト
	自動・可変	6.2.17
	手動・固定	6.2.19
	手動・可変	
トグルフィールド	-	6.2.20

(凡例)

- : 該当しない。

## 6.2.16 複数行フィールドからのデータの抽出

### (1) 仕様

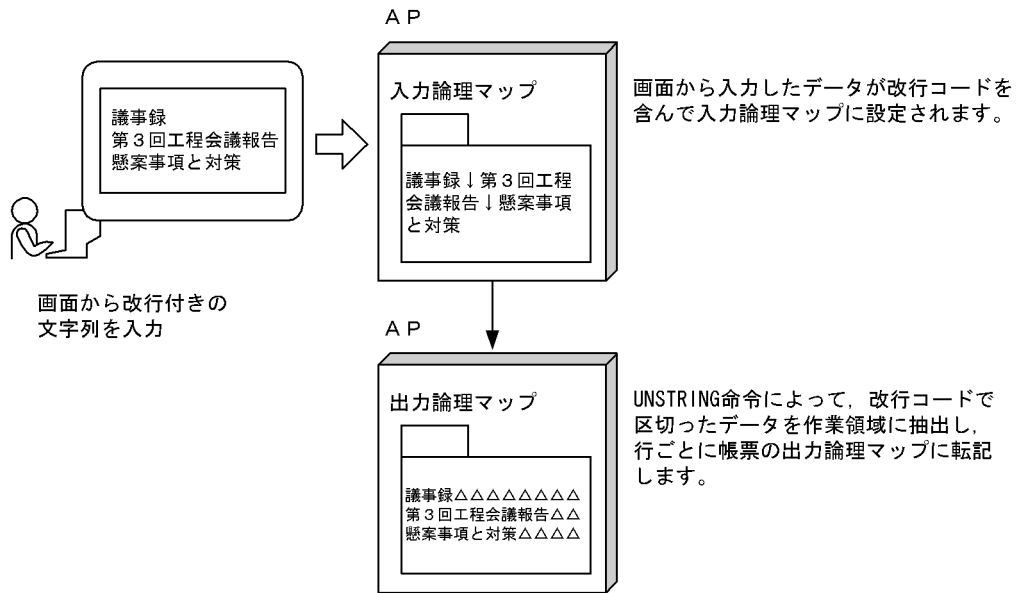
画面から入力した改行コードを含んだ文字列からデータを抽出して帳票に印刷します。

画面のフィールドボックスの入出力フィールドに改行を含んだ文字列を入力すると、入力論理マップには、改行コードを含んだ文字列「議事録 第3回工程会議報告 懸案事項と対策」( は改行コードを示します) が格納されます。改行コードは、LF(0A)<sub>16</sub> 固定です。

AP は入力論理マップの入出力フィールドから、改行コードで区切ったデータを抽出し、帳票の出力論理マップに転記します。

GUI 上の操作では、クリップボードを経由して改行を含むデータをやり取りできます。論理マップに格納される値と AP の処理を次の図に示します。

図 6-15 複数行フィールドからのデータの抽出



## (2) 画面定義の概要

画面定義の概要を次に示します。

1. GUI 画面定義ウィンドウで、ツールボックスの [フィールドボックス] を使用してフィールドボックスを縦サイズ 22，横サイズ 42 で配置する。
2. GUI 画面定義ウィンドウで、ツールボックスの [入出力フィールド (日本語)] を使用して入出力フィールドを 1 行 1 列目から 10 行 40 列目まで配置する。
3. 入出力フィールドダイアログで次の表の項目を定義する。

定義項目	定義内容
使用目的	日本語
詳細目的	混在 (全角・半角)
空白入力	一部 & 全桁 (全・半角)
表示方法	入力可能 - 標準表示
遷移条件	自動スキップ
改行入力 (改行コードを認識する)	指定する
データ名	FIELD0001-
桁	400
埋字	LOW (X'00')
初期値 (入力)	指定しない

## 6. 画面定義とコーディングの関連

### (3) 帳票定義の概要

帳票定義については、マニュアル「XMAP3 プログラミングガイド 帳票編」を参照してください。

1. 帳票定義ウィンドウで、ツールボックスの [ 出力フィールド (日本語) ] を使用して出力フィールドを配置する。
2. 出力フィールドダイアログで次の表の項目を定義する。

定義項目	定義内容
使用目的	日本語
データ名	FIELD0001-
桁	40
フィールド反復	方向 縦, 回数 10
埋字	スペース
初期値	指定しない

XMAP3-出力フィールド

縦 1 ます → フィールド  
横 3 ます ← 40 ます 40 桁 →

使用目的 日本語 文字サイズ 標準 文字の書体 標準  
 文字の拡大 標準 文字の間隔 標準 文字の強調 標準  
 網掛け 標準網掛け 文字色 黒  
 コメントの記述 フォントの回復...

A Pが渡す項目

データ名	桁	データ長	データ型	桁寄せ	埋字	初期値
FIELD0001-	40	40	文字 (XX)	左	スペース	指定しない

下位項目 設定... 自由な設定...

動的変更 (A Pから表示属性を変更する)

OK キャンセル ヘルプ

#### (4) 生成される論理マップ

画面定義のマップ名が MAP001 の場合の例

```
01 MAP001I. (入力論理マップ)
:
:   04 MAP001-FIELD0001-I PIC X(400).
:
01 MAP001O. (出力論理マップ)
:
:   03 MAP001-FIELD0001-O PIC X(400).
:
```

帳票定義のマップ名が MAP002 の場合の例

```
01 MAP002O. (出力論理マップ)
:
:   03 MAP002-FIELD0001-O PIC X(40) OCCURS 10.
:
```

#### (5) COBOL での AP

行データを取り出す AP の例を次に示します。

## 6. 画面定義とコーディングの関連

画面のマップ名が MAP001 , 帳票のマップ名が MAP002 の場合の例

```
IDENTIFICATION DIVISION.  
:  
DATA DIVISION.  
COPY MAP0010.  ...画面の出力論理マップの取り込み  
COPY MAP0011.  
COPY MAP0020.  ...帳票の出力論理マップの取り込み  
:  
77 NEW-LINE-CD PIC X(1) VALUE '0A'.  
01 WORK-AREA.  
03 LINE-DATA PIC X(400) OCCURS 10.  ...行データを抽出するエリア  
:  
PROCEDURE DIVISION.  
:  
TRANSCIBE DSP FROM MAP0010 INTO MAP0011.  
:  
UNSTRING MAP001-FIELD0001-1 DELIMITED BY NEW-LINE-CD  
INTO LINE-DATA(1) LINE-DATA(2) LINE-DATA(3) LINE-DATA(4)  
LINE-DATA(5) LINE-DATA(6) LINE-DATA(7) LINE-DATA(8)  
LINE-DATA(9) LINE-DATA(10).  
:  
MOVE LINE-DATA(1) TO MAP002-FIELD0001-0(1).  
:  
MOVE LINE-DATA(10) TO MAP002-FIELD0001-0(10).  
:  
:
```

### 注

画面で入力するデータが、帳票出力項目の1行の長さを超える場合は、UNSTRING文を使わずに、入力論理マップ中の改行コードをチェックしながら帳票の出力論理マップへ転記する必要があります。

UNSTRING文については、マニュアル「COBOL2002 言語 拡張仕様編」またはマニュアル「COBOL85 言語」を参照してください。

## 6.2.17 可変ポップアップテキストを使用したデータの入力

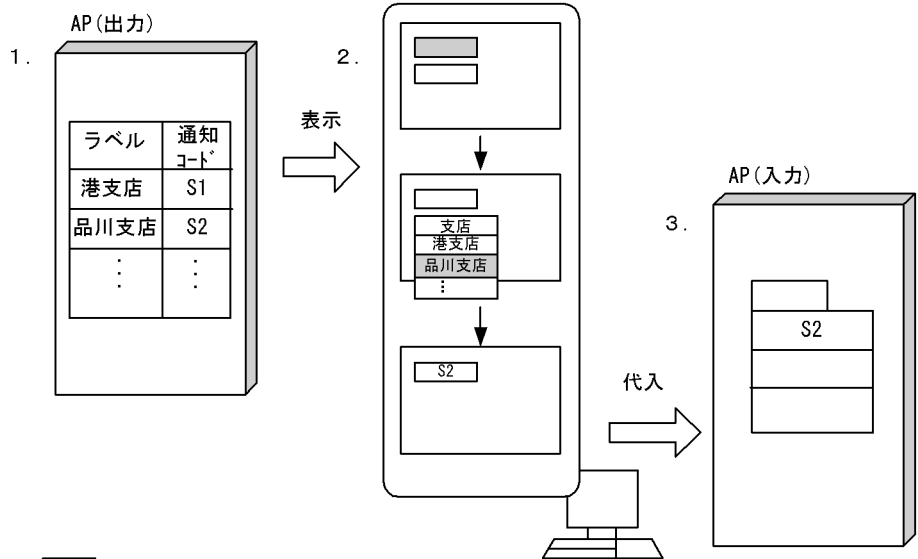
### (1) 自動的にポップアップメニューを表示する

#### (a) 仕様

「支店コード」の入出力テキストにフォーカスが位置づけられると、ポップアップメニューが自動的に表示されます。「支店」のポップアップメニューからデータを選択すると、対応する通知コードが入力論理マップに代入されます。「支店コード」の入出力テキストには、選択されたメニューに対応する通知コード「S2」が表示されます。

可変ポップアップメニューを使用した、AP 実行時のポップアップメニューの自動表示を次の図に示します。

図 6-16 ポップアップメニューを表示する（自動）



(凡例)  : 選択されている状態

1. 可変ポップアップの場合、ラベルおよび通知コードは、AP中の論理マップに定義されています。
2. 自動的にポップアップを表示する場合、「支店コード」の入出力テキストにフォーカスが位置づくとき、ポップアップメニューが自動的に表示されます。
3. ポップアップメニューで選択したラベルに対応する通知コードが、APに返されます。

#### (b) 画面定義の概要

画面定義の概要を次に示します。

1. ポップアップに表示するメニューを AP から動的に変更できるようにするには、GUI 画面定義ウィンドウで、ツールボックスの [可変ポップアップ] を使用して配置する。
2. 可変ポップアップメニューダイアログで次の表の項目を定義する。

定義項目		定義内容
使用目的		POP (自動 - 日本語)
詳細目的		混在 (全角・半角)
データ名		支店コード -
データ型		文字 (XX) (この定義内容は固定)
桁 (AP が受け取る項目)		2
データ長		2 (この定義内容は固定)
ラベル	データ名	ポップラベル -
	データ型	文字 (XX)
	桁	10

## 6. 画面定義とコーディングの関連

定義項目	定義内容
	データ長 10 (この定義内容は固定)
通知コード	データ名 ポップコード -
	データ型 文字 (XX) (この定義内容は固定)
	桁 2 (この定義内容は固定)
	データ長 2 (この定義内容は固定)
メニューの項目数	6
タイトルを付ける	指定する (支店)

### 注

通知コードとして半角文字専用にする場合は、「POP (自動 - 英数)」を選んでください。

### (c) 生成される論理マップ

マップ名が MAP123 の場合の例

```

01 MAP1230. (出力論理マップ)
:
03 MAP123-支店コード-0 PIC X(2).    ... 出力テキストを代入するエリア
03 MAP123-POPUP0001-0 OCCURS 6.    ... 論理テーブル名
04 MAP123-ポップコード-0 PIC X(2).  ... 通知コード項目
04 MAP123-ポップラベル-0 PIC X(10). ... ラベル項目
01 MAP1231. (入力論理マップ)
:
03 MAP123-支店コード-H.
04 MAP123-支店コード-I PIC X(2).    ... 入力テキストを代入するエリア
    
```

### 注

入出力テキストボックスのデータ名を省略した場合、FIELDnnnn- が仮定されます。



ポップアップメニューのデータ名を省略した場合、メニューラベル名 POPUP-LABELnnnn、メニューコード名 POPUP-CODEnnnn が仮定されます (nnnn : 0001 ~ 9999)。

また、可変ポップアップメニューに対して一つの論理テーブル名が付けられます。

(d) COBOL での AP

マップ名が MAP123 の場合の例

WORKING-STORAGE SECTION.	
COPY MAP1230.	… 出力論理マップの取り込み
COPY MAP1231.	… 入力論理マップの取り込み
COPY X3MODTBL.	… 動的変更テーブルの取り込み
77 MINATO PIC N(5) VALUE N' 港支店'.	
77 SINAGAWA PIC N(5) VALUE N' 品川支店'.	
77 NERIMA PIC N(5) VALUE N' 練馬支店'.	
:	
COMMUNICATION SECTION.	
CD DSP FOR I-O WS	… 通信記述名 : DSP
MAP NAME IS 画面マップ名	… 物理マップ名称格納エリア
:	
MOVE 'S1' TO MAP123-ポップコード-0(1).	} … 出力論理マップへの メニュー項目の通知コード とラベルの設定
MOVE MINATO TO MAP123-ポップラベル-0(1).	
MOVE 'S2' TO MAP123-ポップコード-0(2).	
MOVE SINAGAWA TO MAP123-ポップラベル-0(2).	
MOVE 'S3' TO MAP123-ポップコード-0(3).	
MOVE NERIMA TO MAP123-ポップラベル-0(3).	
:	
MOVE 'MAP123ND' TO 画面マップ名.	… 物理マップ名称を代入
TRANSCIEVE DSP FROM MAP1230 INTO MAP1231.	… 画面の送受信
:	
IF MAP123-支店コード-1 = 'S1'	… 入力した通知コードの参照
THEN	
:	

## (2) 手動でポップアップメニューを表示する

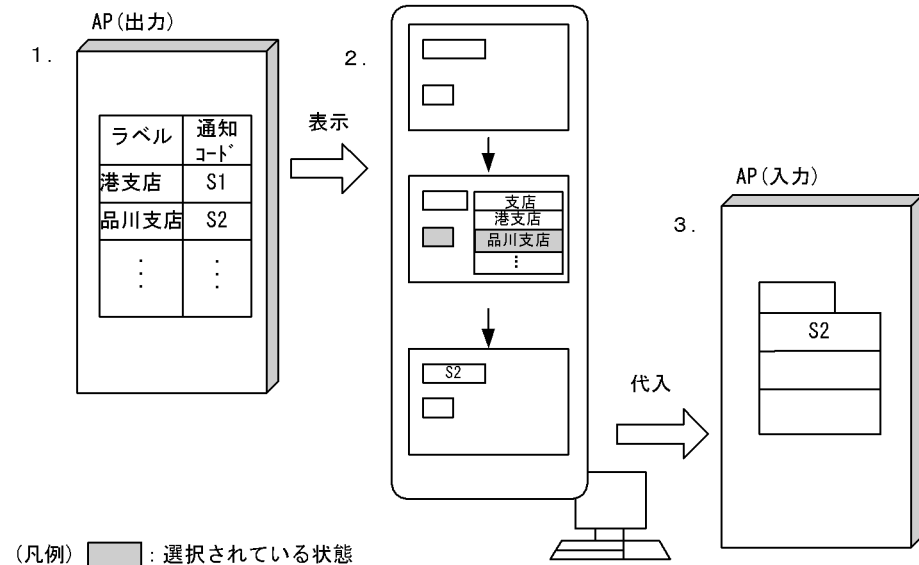
(a) 仕様

[支店] のプッシュボタンか PF キーを押すと、「支店」のポップアップメニューが表示されます。ポップアップメニューからデータを選択すると、対応する通知コードが入力論理マップに格納されます。「支店コード」の入出力テキストには、選択したデータに対応する通知コード「S2」が表示されます。

可変ポップアップメニューを使用した、AP 実行時でのポップアップメニューの手動表示を次の図に示します。

6. 画面定義とコーディングの関連

図 6-17 ポップアップメニューの表示



- (凡例)  : 選択されている状態
1. 可変ポップアップの場合、ラベルおよび通知コードは、AP中の論理マップに定義されています。
  2. 手でポップアップを表示する場合、ポップアップメニューを表示するボタンを押します。  
ここでは、「支店コード」を表示する「ボタン」を押します。
  3. ポップアップメニューで選択したラベルに対応する通知コードが、APIに返されます。

注

なお、手でポップアップメニューを表示する場合、ボタン以外に PF キーも使用できます。

(b) 画面定義の概要

画面定義の概要を次に示します。

ポップアップメニュー、またはキーボードで検索結果を入力する入出力テキストを定義します。

1. ポップアップに表示するメニューを AP から動的に変更できるようにするには、GUI 画面定義ウィンドウで、ツールボックスの [ 可変ポップアップ ] を使用して配置する。
2. 可変ポップアップメニューダイアログで次の表の項目を定義する。

定義項目	定義内容
使用目的	POP ( 手動 - 日本語 )
詳細目的	混在 ( 全角・半角 )
最初にメニューを開く	指定しない
データ名	支店コード -

定義項目		定義内容
データ型		文字 (XX) (この定義内容は固定)
桁 (AP が受け取る項目)		2
データ長		2 (この定義内容は固定)
ラベル	データ名	ポップラベル -
	データ型	文字 (XX)
	桁	10
	データ長	10 (この定義内容は固定)
通知コード	データ名	ポップコード -
	データ型	文字 (XX) (この定義内容は固定)
	桁	2 (この定義内容は固定)
	データ長	2 (この定義内容は固定)
AP から初期値を変更する		指定する

## 注

通知コードとして半角文字専用にする場合は、「POP (手動 - 英数)」を選んでください。

## 《ポップアップメニューの定義》

ポップアップメニューの定義は、自動的にポップアップメニューを表示するときと同じです。「6.2.17(1)(b) 画面定義の概要」を参照してください。

## 《手動ポップアップの設定》

ポップアップメニューを PF1 キーで表示させるための定義をします。

1. GUI 画面定義ウィンドウの、画面属性ダイアログで、「イベント通知コード」の [設定] ボタンを選択する。
2. イベント通知コードのダイアログで次の項目を定義する。

## 6. 画面定義とコーディングの関連

定義項目	定義内容
ポップアップの表示タイミング	「確定キーを使用する」を選択します。
動作	ポップアップ表示 (「確定キーを使用する」を選んだ場合だけ指定できます。PF キーにポップアップ表示の動作を割り当てます)。

### 注

プッシュボタンやメニューバーで手動ポップアップを表示する場合は、手動ポップアップの定義をしたあとに、プッシュボタン定義やメニューバー定義のダイアログを開き、「選択時の動作」で「ポップアップ表示」の定義をしてください。

### 《プッシュボタンの定義》

ポップアップメニューを表示させるためのプッシュボタンを定義します。

1. ポップアップ表示のプッシュボタンは GUI 画面定義ウィンドウで、[ プッシュボタン ( 単独 / 追加 ) ] を使用して配置する。
2. プッシュボタンダイアログで次の項目を定義する。

定義項目	定義内容
ラベル	支店
選択時の動作	ポップアップ表示

### (c) 生成される論理マップ

生成される論理マップは、自動的にポップアップメニューを表示するときと同じです。「6.2.17(1)(c) 生成される論理マップ」を参照してください。

### (d) COBOL での AP

COBOL での AP は、自動的にポップアップメニューを表示するときと同じです。「6.2.17(1)(d) COBOL での AP」を参照してください。

## (3) アクセスキーを使用したポップアップのメニュー選択

アクセスキーは自動ポップアップでも手動ポップアップでも利用できます。ここでは、自動ポップアップを仮定します。

### (a) 仕様

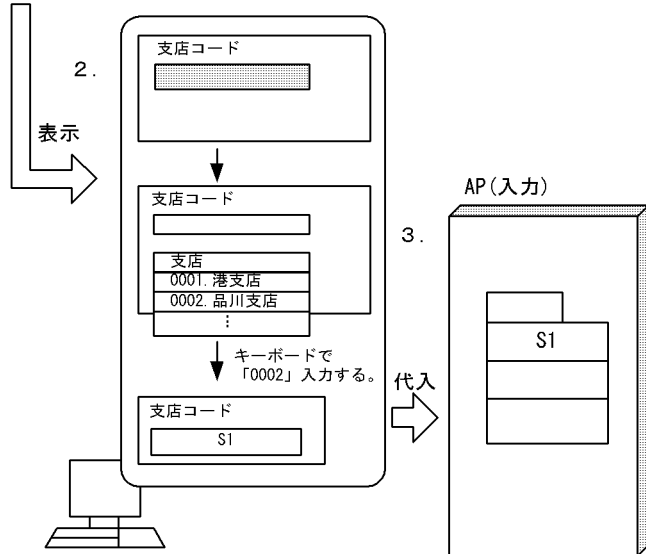
「支店コード」の入出力テキストにフォーカスが位置づけられると、ポップアップメニューが自動的に表示されます。「支店」のポップアップメニューからアクセスキーを使用してデータを選択すると、対応する通知コードが入力論理マップに代入されます。「支店コード」の入出力テキストには、選択されたメニューに対応する通知コード「S2」が表示されます。


可変ポップアップメニューを使用した、AP 実行時のポップアップメニューの自動表示を次の図に示します。

図 6-18 アクセスキーでポップアップメニューを選択する（自動）

## 1. AP(出力)

アクセスキー	ラベル	通知コード
0001	0001. 港支店	S1
0002	0002. 品川支店	S2
:	:	:



(凡例)  : 選択されている状態

1. 可変ポップアップで、「メニューデータをファイルで指定する」をチェックしていない場合は、ラベル、通知コード、選択ラベルおよびアクセスキーは、AP中の論理マップに定義されています。
2. 自動ポップアップの場合、「支店コード」の入出力テキストにフォーカスが位置づくとき、ポップアップメニューが自動的に表示されます。  
手動ポップアップの場合は、キー操作などでポップアップメニューを表示します。\*
3. アクセスキーを入力すると、対応する通知コードが、APに返されます。  
「支店コード」の入出力テキストにも、通知コードと同じ値が表示されます。

注※ アクセスキーは、ポップアップメニューが表示されているときだけ有効です。

## (b) 画面定義の概要

画面定義の概要を次に示します。

1. ポップアップに表示するメニューを AP から動的に変更できるようにするには、GUI 画面定義ウィンドウで、ツールボックスの [ 可変ポップアップ ] を使用して配置する。
2. 可変ポップアップメニューダイアログで次の表の項目を定義する。

## 6. 画面定義とコーディングの関連

定義項目	定義内容	
使用目的	POP (自動 - 日本語)	
詳細目的	混在 (全角・半角)	
アクセスキーを使用する	指定する	
長さ (アクセスキー)	4	
選択ラベルと通知コードを同じ値にする	指定する	
タイトルを付ける	指定する (支店)	
データ名	支店コード -	
データ型	文字 (XX) (この定義内容は固定)	
桁 (AP が受け取る項目)	2	
データ長	2 (この定義内容は固定)	
メニューデータをファイルで指定する	指定しない	
ラベル	データ名	ポップラベル -
	データ型	文字 (XX)
	桁	10
	データ長	10 (この定義内容は固定)
通知コード	データ名	ポップコード -
	データ型	文字 (XX) (この定義内容は固定)
	桁	2 (この定義内容は固定)
	データ長	2 (この定義内容は固定)
アクセスキー	データ名	ポップキー -
	データ型	文字 (XX) (この定義内容は固定)
	桁	4 (この定義内容は固定)
メニューの項目数	6	

### 注

通知コードとして半角文字専用にする場合は、「POP (自動 - 英数)」または「POP (手動 - 英数)」を選んでください。

## (c) 生成される論理マップ

マップ名が MAP123 の場合の例

```

01 MAP1230. (出力論理マップ)
:
03 MAP123-支店コード-0 PIC X(2).    ... 出力テキストを代入するエリア
03 MAP123-POPUP0001-0 OCCURS 6.    ... 論理テーブル名
04 MAP123-ポップコード-0 PIC X(2).  ... 通知コード項目
04 MAP123-ポップラベル-0 PIC X(10). ... ラベル項目
04 MAP123-ポップキー-0 PIC X(4).   ... アクセスキー項目
01 MAP1231. (入力論理マップ)
:
03 MAP123-支店コード-H.
04 MAP123-支店コード-I PIC X(2).    ... 入力テキストを代入するエリア

```

## 注

入出力テキストボックスのデータ名を省略した場合、FIELDnnnn- が仮定されます。

ポップアップメニューのデータ名を省略した場合、メニューラベル名

POPUP-LABELnnnn-、メニューコード名 POPUP-CODEnnnn- が仮定されます

(nnnn: 0001 ~ 9999)

また、可変ポップアップメニューに対して一つの論理テーブル名が付けられます。

## 6. 画面定義とコーディングの関連

### (d) COBOL での AP

マップ名が MAP123 の場合の例

WORKING-STORAGE SECTION.	
COPY MAP1230.	… 出力論理マップの取り込み
COPY MAP1231.	… 入力論理マップの取り込み
COPY X3MODTBL.	… 動的変更テーブルの取り込み
77 MINATO PIC N(5) VALUE N' 港支店'.	
77 SINAGAWA PIC N(5) VALUE N' 品川支店'.	
77 NERIMA PIC N(5) VALUE N' 練馬支店'.	
:	
COMMUNICATION SECTION.	
CD DSP FOR I-O WS	… 通信記述名: DSP
MAP NAME IS 画面マップ名	… 物理マップ名称格納エリア
:	
MOVE 'S1' TO MAP123-ポップコード-0(1).	} … 出力論理マップへの メニュー項目の通知コード とラベルの設定
MOVE MINATO TO MAP123-ポップラベル-0(1).	
MOVE '0001' TO MAP123-ポップキー-0(1).	
MOVE 'S2' TO MAP123-ポップコード-0(2).	
MOVE SINAGAWA TO MAP123-ポップラベル-0(2).	
MOVE '0002' TO MAP123-ポップキー-0(2).	
MOVE 'S3' TO MAP123-ポップコード-0(3).	
MOVE NERIMA TO MAP123-ポップラベル-0(3).	
MOVE '0003' TO MAP123-ポップキー-0(3).	
:	
MOVE 'MAP123ND' TO 画面マップ名.	… 物理マップ名称を代入
TRANSCIEVE DSP FROM MAP1230 INTO MAP1231.	… 画面の送受信
:	
IF MAP123-支店コード-I = 'S1'	… 入力した通知コードの参照
THEN	
:	

### (4) ポップアップが選択されたときの選択ラベルと通知コードを別々にする

ポップアップの選択ラベルと通知コードを同じ値にするかどうかは、自動ポップアップでも手動ポップアップでも指定できます。ここでは、自動ポップアップを仮定します。

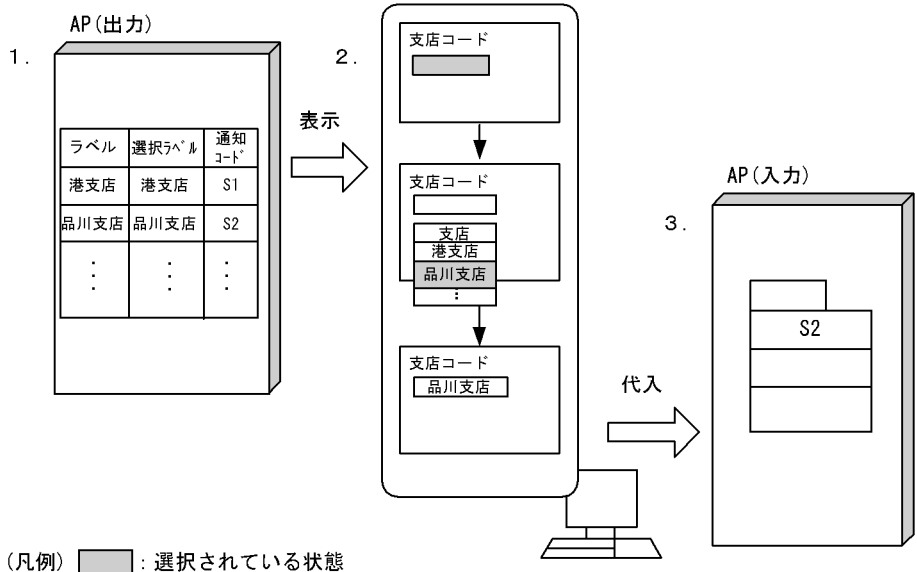
#### (a) 仕様

「支店コード」の入出力テキストにフォーカスが位置づけられると、ポップアップメニューが自動的に表示されます。「支店」のポップアップメニューからデータを選択すると、対応する通知コードが入力論理マップに代入されます。「支店コード」の入出力テキストには、選択されたメニューに対応する選択ラベル「品川支店」が表示されます。APには、選択ラベルに対応した通知コード「S2」が返されます。

可変ポップアップメニューを使用した、AP 実行時のポップアップメニューの自動表示を次の図に示します。



図 6-19 ポップアップが選択されたときの選択ラベルと通知コードを別々にする（自動）



1. 可変ポップアップで「メニューデータをファイルで指定する」をチェックしていない場合は、ラベル、選択ラベルおよび通知コードは、AP中の論理マップに定義されています。
2. 自動ポップアップの場合は、「支店コード」の入出力テキストにフォーカスが位置すると、ポップアップメニューが自動的に表示されます。手動ポップアップの場合は、キー操作などでメニューを表示します。
3. ポップアップメニューで選択したラベルに対応する選択ラベルが表示され、選択ラベルに対応した通知コードがAPに返されます。

### (b) 画面定義の概要

画面定義の概要を次に示します。

1. ポップアップに表示するメニューを AP から動的に変更できるようにするには、GUI 画面定義ウィンドウで、ツールボックスの [ 可変ポップアップ ] を使用して配置する。
2. 可変ポップアップメニューダイアログで次の表の項目を定義する。

定義項目	定義内容
使用目的	POP (自動 - 日本語)
詳細目的	混在 (全角・半角)
アクセスキーを使用する	指定しない
選択ラベルと通知コードを同じ値にする	指定しない
タイトルを付ける	指定する (支店)
データ名	支店コード -
データ型	文字 (XX) (この定義内容は固定)
桁 (AP が受け取る項目)	2

6. 画面定義とコーディングの関連

定義項目	定義内容	
データ長	2 (この定義内容は固定)	
メニューデータをファイルで指定する	指定しない	
ラベル	データ名	ポップラベル -
	データ型	文字 (XX)
	桁	10
	データ長	10 (この定義内容は固定)
通知コード	データ名	ポップコード -
	データ型	文字 (XX) (この定義内容は固定)
	桁	2 (この定義内容は固定)
	データ長	2 (この定義内容は固定)
選択ラベル	データ名	ポップ表示 -
	データ型	文字 (XX)
	桁	10
	データ長	10 (この定義内容は固定)
メニューの項目数	6	

注

通知コードとして半角文字専用にする場合は、「POP (自動 - 英数)」または「POP (手動 - 英数)」を選んでください。

## (c) 生成される論理マップ

マップ名が MAP123 の場合の例

```

01 MAP1230. (出力論理マップ)
:
03 MAP123-支店コード-0 PIC X(2).    ... 出力テキストを代入するエリア
03 MAP123-POPUP0001-0 OCCURS 6.    ... 論理テーブル名
04 MAP123-ポップコード-0 PIC X(2).  ... 通知コード項目
04 MAP123-ポップラベル-0 PIC X(10). ... ラベル項目
04 MAP123-ポップ表示-0 PIC X(10).  ... 選択ラベル
01 MAP1231. (入力論理マップ)
:
03 MAP123-支店コード-H.
04 MAP123-支店コード-I PIC X(2).    ... 入力テキストを代入するエリア

```

## 注

入出力テキストボックスのデータ名を省略した場合、FIELDnnnn- が仮定されます。  
 ポップアップメニューのデータ名を省略した場合、メニューラベル名  
 POPUP-LABELnnnn-、メニューコード名 POPUP-CODEnnnn- が仮定されます  
 (nnnn : 0001 ~ 9999)。  
 また、可変ポップアップメニューに対して一つの論理テーブル名が付けられます。

## 6. 画面定義とコーディングの関連

### (d) COBOL での AP

マップ名が MAP123 の場合の例

WORKING-STORAGE SECTION.	
COPY MAP1230.	… 出力論理マップの取り込み
COPY MAP1231.	… 入力論理マップの取り込み
COPY X3MODTBL.	… 動的変更テーブルの取り込み
77 MINATO PIC N(5) VALUE N' 港支店'.	
77 SINAGAWA PIC N(5) VALUE N' 品川支店'.	
77 NERIMA PIC N(5) VALUE N' 練馬支店'.	
:	
COMMUNICATION SECTION.	
CD DSP FOR I-O WS	… 通信記述名: DSP
MAP NAME IS 画面マップ名	… 物理マップ名称格納エリア
:	
MOVE 'S1' TO MAP123-ポップコード-0(1).	} … 出力論理マップへの メニュー項目の通知コード とラベルの設定
MOVE MINATO TO MAP123-ポップラベル-0(1).	
MOVE MINATO TO MAP123-ポップ表示-0(1).	
MOVE 'S2' TO MAP123-ポップコード-0(2).	
MOVE SINAGAWA TO MAP123-ポップラベル-0(2).	
MOVE SINAGAWA TO MAP123-ポップ表示-0(2).	
MOVE 'S3' TO MAP123-ポップコード-0(3).	
MOVE NERIMA TO MAP123-ポップラベル-0(3).	
MOVE NERIMA TO MAP123-ポップ表示-0(3).	
:	
MOVE 'MAP123ND' TO 画面マップ名.	… 物理マップ名称を代入
TRANSCIEVE DSP FROM MAP1230 INTO MAP1231.	… 画面の送受信
:	
IF MAP123-支店コード-I = 'S1'	… 入力した通知コードの参照
THEN	
:	

## 6.2.18 ポップアップメニューファイルを使用したデータの 入力

### (1) ポップアップメニューの表示 (メニュー形式 標準の場合)

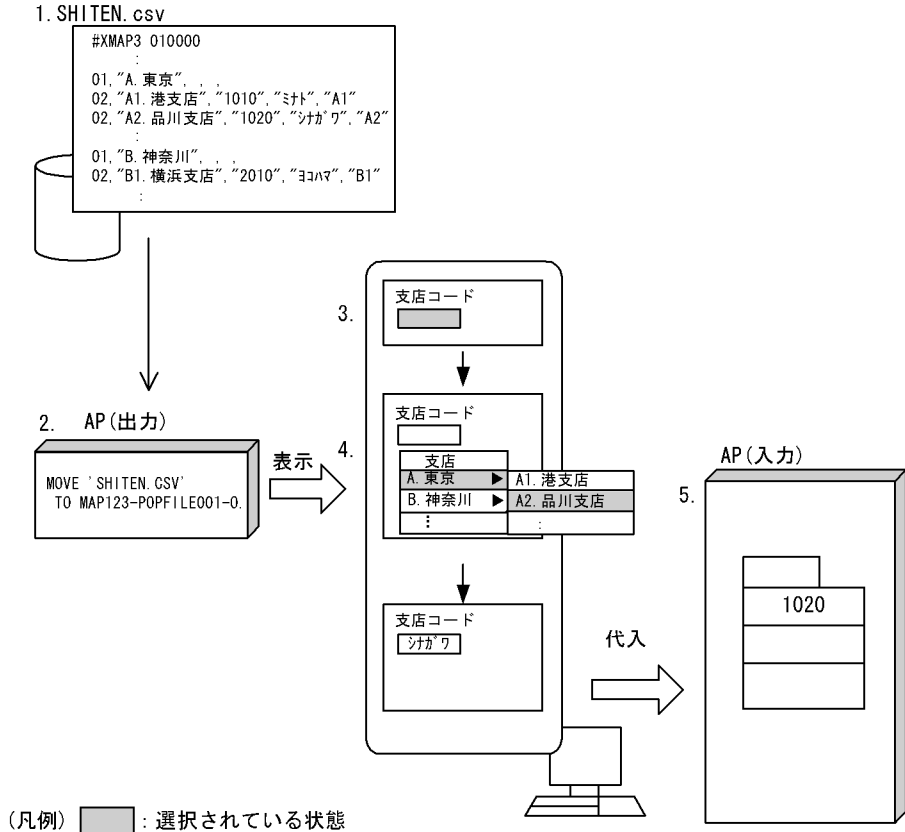
メニュー形式で「標準 (メニュー)」を指定すると、ポップアップメニューファイルに格納されているメニュー項目が階層構造の場合は、下位のメニュー項目がカスケードメニューで表示されます。メニュー形式の指定は、自動ポップアップでも手動ポップアップでも利用できます。ここでは、手動ポップアップを仮定します。

#### (a) 仕様

「支店コード」の入出力テキストにフォーカスが位置づけられると、ポップアップメニューがポップアップメニューファイル (SHITEN.csv) から読み込まれ、下位のメニュー項目はカスケード表示されます。「支店」のポップアップメニューからデータを選択すると、対応する通知コードが入力論理マップに代入されます。「支店コード」の入出力テキストには、選択されたメニューに対応する選択ラベルが表示されます。

ポップアップメニューファイルを使用した、AP 実行時のポップアップメニューのカスケード表示を次の図に示します。

図 6-20 ポップアップメニューを表示する（メニュー形式 標準（メニュー））



- ポップアップメニューファイルは、あらかじめポップアップメニューエディタで作成しておきます。ポップアップメニューファイル作成の注意事項については、「ポップアップメニューファイル作成時の注意事項」を参照してください。
- APで、読み込むファイル名を設定します。ドローで設定したファイル名は初期値となります。
- ドローの定義で「最初にメニューを開く」を指定しておく、「支店コード」の入出力テキストにフォーカスが位置したとき、ポップアップメニューが自動的に表示されます。
- ドローの定義でメニュー形式に「標準（メニュー）」を指定しておく、ポップアップメニューファイルに格納されているメニュー項目が階層構造の場合は、下位のメニュー項目がカスケードメニューで表示されます。
- ポップアップメニューで選択したラベルに対応する通知コードが、APに返されます。

#### ポップアップメニューファイル作成時の注意事項

ポップアップメニューファイルは、ユーティリティのポップアップメニューエディタで作成してください。ポップアップメニューエディタの詳細はマニュアル「XMAP3 開発・実行ガイド」を参照してください。ポップアップメニューエディタ以外のエディタで作成する場合は次に示す規則に従ってください。

## 6. 画面定義とコーディングの関連

### 1. 保守情報の記述

ファイルの先頭には保守情報を記述できます。ポップアップメニューエディタでファイルを作成すると、この情報は必ず付加されます。それ以外のエディタで作成する場合、この情報は省略できます。ただし、作成日時などは記述することをお勧めします。

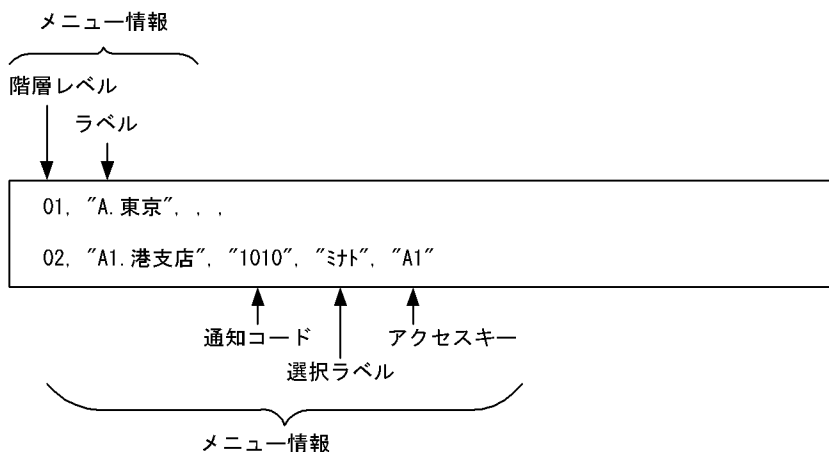
格納するデータはシフト JIS 固定です。

保守情報を次に示します。

#XMAP3 vvrzz	…ファイルのバージョン情報	010000固定
#Date yyyy/mm/dd	…作成/変更日付	
#Time hh:mm:ss	…作成/変更日付	

### 2. メニュー情報の記述

保守情報に続く行はメニュー情報です。次にメニュー情報を示します。



- ラベル、通知コードおよび選択ラベルは、それぞれ 64 桁以内で指定します（全角 1 文字は、2 桁となります）。アクセスキーは 4 桁以内で指定します。
- 階層レベルとラベルは省略できません。
- 項目は「,」で区切ります。通知コード以降の項目を省略する場合も「,」は必要です。

(例)

選択ラベルを省略する場合

02, "A1. 港支店", "1010", , "A1"

- 階層レベル以外の各項目は引用符「"」で囲みます。項目中に「"」がある場合は「"」を 2 回連続で指定します。
- 選択ラベルは省略できます。なお、選択ラベルを指定する場合は、最下位の階層のメニュー項目にだけ指定してください。
- アクセスキーは省略できます。
- アクセスキーには、英大文字と数字だけ指定します。
- アクセスキーのあとには、空白を含むデータは指定できません。

- アクセスキーは、各階層で必ず桁数をそろえて指定します。また、各階層内でユニークになるように指定します。
- 階層レベルは 8 階層（01 ~ 08）まで指定できます。同階層のメニュー項目は 256 項目以下、全階層のメニュー項目の和は 32,000 項目以下になるようにしてください。
- 階層レベルは 01 から昇順に指定します。階層レベルを飛ばさないように（01 レベルの次に 03 レベルを指定したりしないように）注意してください。

階層レベルが昇順に並んだ正しい指定	{	01, "関東", . . .
		02, "東京", . . .
		03, "港支店", "0A10", "ミト", "A1"
		03, "品川支店", "0A20", "シナガワ", "A2"
階層レベル02が飛ばされた誤った指定	{	01, "関東", . . .
		03, "港支店", "0A10", "ミト", "A1"
		03, "品川支店", "0A20", "シナガワ", "A2"

- ポップアップメニューファイルでは、空白の行または改行だけの行は指定できません。
- あるレベルのメニュー項目に対し下位レベルのメニュー項目がある場合は、該当するメニュー項目の行の次に、下位レベルの行を指定してください。

01, "関東", . . .	} 01レベルの「関東」の下位レベルの行とみなされます
02, "東京", . . .	
03, "港支店", "0A10", "ミト", "A1"	
03, "品川支店", "0A20", "シナガワ", "A2"	
01, "関西", . . .	} 01レベルの「関東」の次に指定されていないため、「関東」の下位レベルの行とみなされません。直前の01レベルの行「関西」の下位レベルの行とみなされます。
02, "神奈川", . . .	
02, "大阪", . . .	
:	

- 通知コードは、最下位の階層のメニュー項目にだけ指定してください。
- 1 行には、一つのメニュー情報だけ指定します。また、1 行の終わりには、必ず改行を指定してください。
- ポップアップメニューファイルの最終行にも、改行を指定してください。

#### (b) 画面定義の概要

画面定義の概要を次に示します。

1. ポップアップに表示するメニューを AP から動的に変更できるようにするには、GUI 画面定義ウィンドウで、ツールボックスの [ 可変ポップアップ ] を使用して配置する。
2. 可変ポップアップメニューダイアログで次の表の項目を定義する。

## 6. 画面定義とコーディングの関連

定義項目	定義内容
使用目的	POP (手動 - カナ)
データ名	支店コード -
データ型	文字 (XX) (この定義内容は固定)
桁 (AP が受け取る項目)	4
データ長	4 (この定義内容は固定)
メニューデータをファイルで指定する	指定する
メニュー形式	標準 (メニュー)
最初にメニューを開く	指定する
タイトルを付ける	指定する (支店)
データ名 (ファイル)	POPFILE0001-
データの渡し方	ファイル名指定
初期値 (ファイル)	SHITEN.csv

### 注

通知コードとして半角文字専用にする場合は、「POP (自動 - 英数)」または「POP (手動 - 英数)」を選んでください。



## (c) 生成される論理マップ

マップ名が MAP123 の場合の例

```

01 MAP1230. (出力論理マップ)
:
03 MAP123-支店コード-0 PIC X(4).    ...出力テキストを代入するエリア
03 MAP123-POPF0001-0 PIC X(12).    ...ポップアップメニューファイルのファイル名指定エリア

01 MAP1231. (入力論理マップ)
:
03 MAP123-支店コード-H.
04 MAP123-支店コード-I PIC X(4).    ...入力テキストを代入するエリア

```

## 注

入出力テキストボックスのデータ名を省略した場合、FIELDnnnn- が仮定されます。

## (d) COBOL での AP

マップ名が MAP123 の場合の例

```

WORKING-STORAGE SECTION.
COPY MAP1230.                ... 出力論理マップの取り込み
COPY MAP1231.                ... 入力論理マップの取り込み
COPY X3MODTBL.               ... 動的変更テーブルの取り込み
:
COMMUNICATION SECTION.
CD DSP FOR I-0 WS            ... 通信記述名: DSP
MAP NAME IS 画面マップ名    ... 物理マップ名称格納エリア
:
MOVE 'MAP123ND' TO 画面マップ名. ... 物理マップ名称を代入
MOVE ALL X'1F' TO MAP123-POPF0001-0. ... 必ずX'1F'でクリアする。
MOVE 'SHITEN.GSV' TO MAP123-POPF0001-0(1:10). ... ポップアップメニューファイル名の設定
TRANSCIEVE DSP FROM MAP1230 INTO MAP1231. ... 画面の送受信
:
IF MAP123-支店コード-I = '1020' ... 入力した通知コードの参照
THEN
:

```

## (2) ポップアップメニューの表示 (メニュー形式 リストの場合)

メニュー形式で「リスト - ポップアップ型」または「リスト - ダイアログ型」を指定すると、メニューデータファイルに格納されているメニュー項目が階層構造の場合、下位のメニュー項目は下位レベルのリストとして表示されます。メニュー形式の指定は、自動ポップアップでも手動ポップアップでも利用できます。ここでは、手動ポップアップを仮定します。また、メニュー形式は「リスト - ポップアップ型」を仮定します。

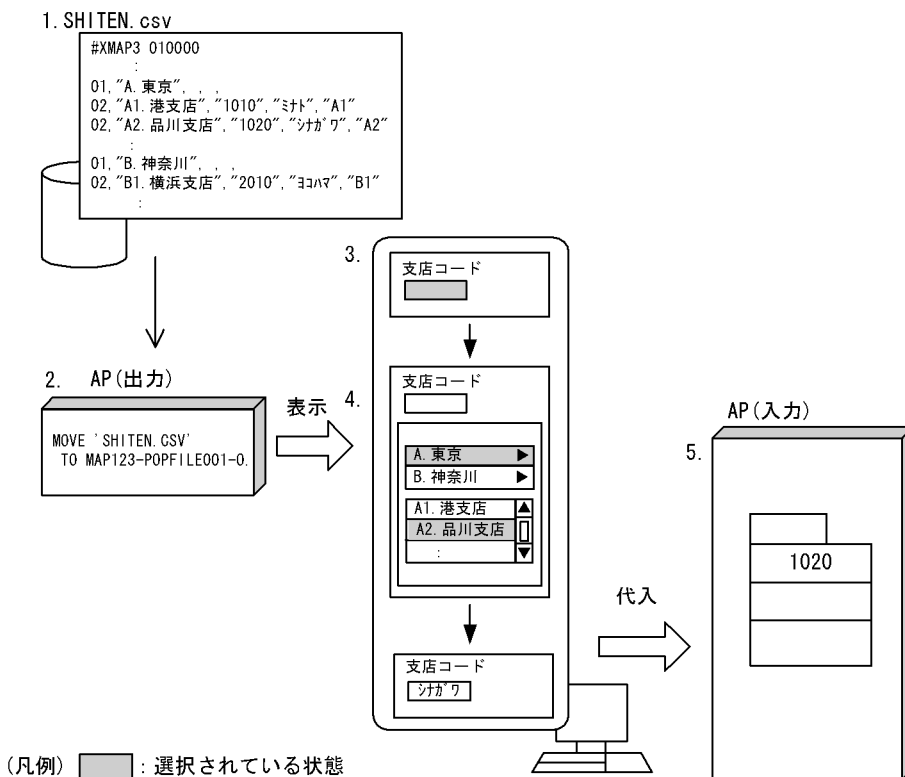
## 6. 画面定義とコーディングの関連

### (a) 仕様

「支店コード」の入出力テキストにフォーカスが位置づけられると、ポップアップメニューがポップアップメニューファイル (SHITEN.csv) から読み込まれ、下位のメニュー項目がリスト表示されます。ポップアップメニューからデータを選択すると、対応する通知コードが入力論理マップに代入されます。「支店コード」の入出力テキストには、選択されたメニューに対応する選択ラベルが表示されます。

ポップアップメニューファイルを使用した、AP 実行時のポップアップメニューのリスト表示を次の図に示します。

図 6-21 ポップアップメニューを表示する (メニュー形式 リスト-ポップアップ型)



1. ポップアップメニューファイルは、あらかじめポップアップメニューエディタで作成しておきます。
2. APで、読み込むファイル名を設定します。ドローで設定したファイル名は初期値となります。
3. ドローの定義で「最初にメニューを開く」を指定しておく、「支店コード」の入出力テキストにフォーカスが位置づいたとき、ポップアップメニューが自動的に表示されます。
4. ドローでの定義で、メニュー形式に「リスト-ポップアップ型」または「リスト-ダイアログ型」を指定すると、ポップアップメニューファイルに格納されているメニュー項目が階層構造の場合、下位のメニュー項目は下位レベルのリストとして表示されます。ここでは、「リスト-ポップアップ型」が指定されていると仮定します。
5. ポップアップメニューで選択したラベルに対応する通知コードが、APに返されます。

## (b) 画面定義の概要

画面定義の概要を次に示します。

1. ポップアップに表示するメニューを AP から動的に変更できるようにするには、GUI 画面定義ウィンドウで、ツールボックスの [ 可変ポップアップ ] を使用して配置する。
2. 可変ポップアップメニューダイアログで次の表の項目を定義する。

定義項目	定義内容
使用目的 <sup>1</sup>	POP ( 手動 - カナ )
データ名	支店コード -
データ型	文字 ( XX ) ( この定義内容は固定 )
桁 ( AP が受け取る項目 )	4
データ長	4 ( この定義内容は固定 )
メニューデータをファイルで指定する	指定する
メニュー形式	リスト - ポップアップ型 <sup>2</sup>
最初にメニューを開く	指定する
データ名 ( ファイル )	POPFIL0001-
データの渡し方	ファイル名指定
初期値 ( ファイル )	SHITEN.csv

## 注 1

通知コードとして半角文字専用にする場合は、「POP ( 自動 - 英数 )」または「POP ( 手動 - 英数 )」を選んでください。

## 注 2

リスト表示をする場合は、「リスト - ダイアログ型」も選べます。

## 6. 画面定義とコーディングの関連

### (c) 生成される論理マップ

マップ名が MAP123 の場合の例

```
01 MAP1230. (出力論理マップ)
:
03 MAP123-支店コード-0 PIC X(4).    ...出力テキストを代入するエリア
03 MAP123-POPFIL0001-0 PIC X(12).   ...ポップアップメニューファイルのファイル名指定エリア

01 MAP1231. (入力論理マップ)
:
03 MAP123-支店コード-H.
04 MAP123-支店コード-I PIC X(4).    ...入力テキストを代入するエリア
```

#### 注

入出力テキストボックスのデータ名を省略した場合、FIELDnnnn- が仮定されます。

### (d) COBOL での AP

マップ名が MAP123 の場合の例

```
WORKING-STORAGE SECTION.
COPY MAP1230.                ... 出力論理マップの取り込み
COPY MAP1231.                ... 入力論理マップの取り込み
COPY X3MODTBL.              ... 動的変更テーブルの取り込み
:
COMMUNICATION SECTION.
CD DSP FOR I-O WS           ... 通信記述名:DSP
MAP NAME IS 画面マップ名   ... 物理マップ 名称格納エリア
:
MOVE 'MAP123ND' TO 画面マップ名. ... 物理マップ 名称を代入
MOVE ALL X'1F' TO MAP123-POPFIL0001-0. ... 必ずX'1F' でクリアする。
MOVE 'SHITEN.CSV' TO MAP123-POPFIL0001-0(1:10). ... ポップアップメニューファイル名の設定
TRANSCIEVE DSP FROM MAP1230 INTO MAP1231. ... 画面の送受信
:
IF MAP123-支店コード-I = '1020' ... 入力した通知コードの参照
THEN
:
```

## 6.2.19 固定ポップアップテキストを使用したデータの入力

### (1) 自動的にポップアップメニューを表示する

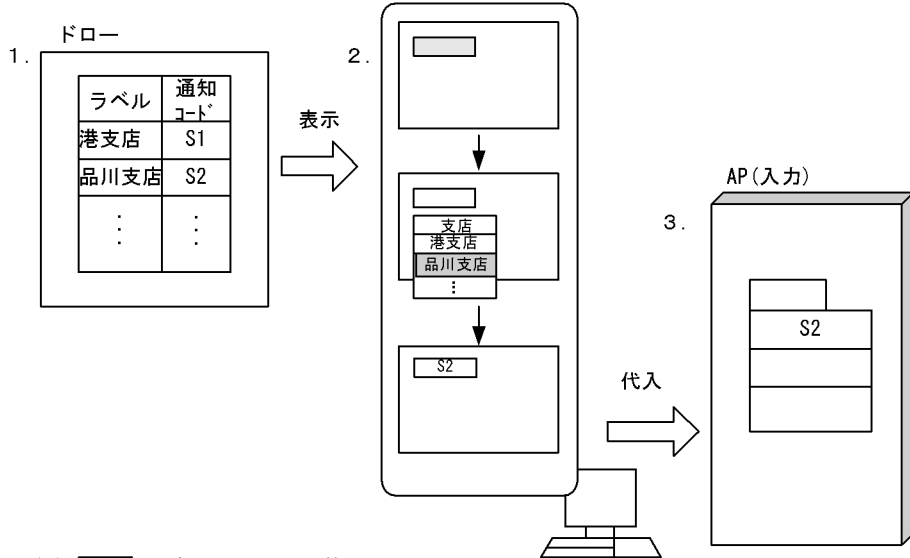
#### (a) 仕様

「支店コード」の入出力テキストにフォーカスが位置づけられると、ポップアップメニューが自動的に表示されます。「支店」のポップアップメニューからデータを選択すると、対応する通知コードが入力論理マップに格納されます。「支店コード」の入出力テキ

ストには、選択されたメニューに対応する通知コード「S2」が表示されます。

固定ポップアップメニューを使用した、AP 実行時のポップアップメニューの自動表示を次の図に示します。

図 6-22 ポップアップメニューを表示する（自動）



(凡例)  : 選択されている状態

1. 固定ポップアップの場合、ラベルおよび通知コードは、ドロワーで定義されています。
2. 自動的にポップアップを表示する場合、「支店コード」の入出力テキストにフォーカスが位置づくとき、ポップアップメニューが自動的に表示されます。
3. ポップアップメニューで選択したラベルに対応する通知コードが、APIに返されます。

#### (b) 画面定義の概要

画面定義の概要を次に示します。

1. ポップアップに表示するメニューを、画面定義以降に変更する必要がない場合には、GUI 画面定義ウィンドウで、ツールボックスの [ 固定ポップアップ ] を使用して配置する。
2. 固定ポップアップメニューダイアログで次の表の項目を定義する。

定義項目	定義内容
使用目的	POP (自動 - 日本語)
詳細目的	混在 (全角・半角)
選択ラベルと通知コードを同じ値にする	指定する
タイトルを付ける	指定する (支店)
データ名	支店コード -
データ型	文字 (XX) (この定義内容は固定)

## 6. 画面定義とコーディングの関連

定義項目	定義内容
桁	2
データ長	2 (この定義内容は固定)
動の変更 (AP から表示属性を変更する)	指定しない
AP から初期値を変更する	指定する

### 注

通知コードとして半角文字専用にする場合は、「POP (自動 - 英数)」を選んでください。

また、次に示す定義内容を「メニューの値」のラベルと通知コードに追加していきます。

ラベル	通知コード
港支店	S1
品川支店	S2
練馬支店	S3
足立支店	S4
豊島支店	S5

The screenshot shows the 'XMAPQ-固定ポップアップメニュー' configuration window. It includes fields for '使用目的' (Usage Purpose) set to 'POP(自動-日本語)', '詳細目的' (Detailed Purpose) set to '現在(全角・半角)', and checkboxes for 'アクセスキーを使用する' (Use Access Key), '選択ラベルと通知コードを同じ値にする' (Use same value for selected label and notification code), '外枠を付ける' (Attach frame) set to '別枠 両面', and '最初に表示メニューを強制' (Force menu display at start). Below these are fields for 'APが取り取る項目' (Item taken by AP) and 'APが値する項目' (Item valued by AP), with '動的変更 (APから表示属性を変更する)' (Dynamic change) unchecked and 'APから初期値を変更する' (Change initial value from AP) checked. The 'メニューの値' (Menu Values) section contains a table:

メニューの値	メニューの項目数	ラベル	通知コード
1	5	港支店	S1
2	5	品川支店	S2
3	5	練馬支店	S3
4	5	足立支店	S4
5	5	豊島支店	S5

## (c) 生成される論理マップ

マップ名が MAP124 の場合の例

```

01 MAP1240. (出力論理マップ)
   03 MAP124-支店コード-0 PIC X(2). ... 出力テキストを代入するエリア

01 MAP1241. (入力論理マップ)
   03 MAP124-支店コード-H.
   04 MAP124-支店コード-I PIC X(2). ... 入力テキストを代入するエリア

```

## 注

入出力テキストボックスのデータ名を省略した場合、FIELDnnnnn- (nnnn : 0001 ~ 9999) が仮定されます。

## (d) COBOL での AP

マップ名が MAP124 の場合の例

```

WORKING-STORAGE SECTION.
  COPY MAP1240. ... 出力論理マップの取り込み
  COPY MAP1241. ... 入力論理マップの取り込み
  COPY X3MODTBL. ... 動的変更テーブルの取り込み
COMMUNICATION SECTION.
  CD DSP FOR I-0 WS ... 通信記述名 : DSP
  MAP NAME IS 画面マップ名 ... 物理マップ名称格納エリア
  :
  MOVE 'MAP124ND' TO 画面マップ名. ... 物理マップ名称を代入
  TRANSCEIVE DSP FROM MAP1240 INTO MAP1241. ... 画面の送受信
  :
  IF MAP124-支店コード-I = 'S1' ... 【支店】ボタンを押してポップアップメニューを表示し、【品川支店】を選択
  THEN ... 選択された通知コードの判定
  :

```

## (2) 手動でポップアップメニューを表示する

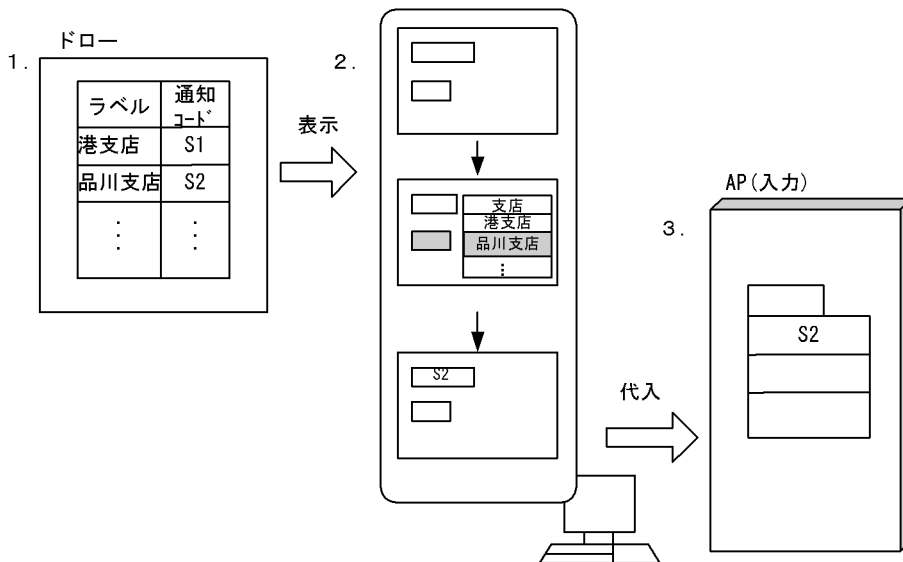
## (a) 仕様

[支店] のプッシュボタンか PF1 キーを押すと、「支店」のポップアップメニューが表示されます。ポップアップメニューからデータを選択すると、対応する通知コードが入力論理マップに格納されます。「支店コード」の入出力テキストには、選択したデータに対応する通知コード「S2」が表示されます。

固定ポップアップメニューを使用した、AP 実行時でのポップアップメニューの手動表示を次の図に示します。

6. 画面定義とコーディングの関連

図 6-23 ポップアップメニューの表示



(凡例)  : 選択されている状態

1. 固定ポップアップの場合、ラベルおよび通知コードは、ドロウで定義されています。
2. 手でポップアップを表示する場合、ポップアップメニューを表示するボタンを押します。ここでは、「支店コード」を表示する「ボタン」を押します。
3. ポップアップメニューで選択したラベルに対応する通知コードが、APIに返されます。

注

なお、手でポップアップメニューを表示する場合、ボタン以外に PF キーも使用できます。

(b) 画面定義の概要

画面定義の概要を次に示します。

ポップアップメニュー、またはキーボードで検索結果を入力する入出力テキストを定義します。

1. ポップアップに表示するメニューを、画面定義以降に変更する必要がある場合には、GUI 画面定義ウィンドウで、ツールボックスの [ 固定ポップアップ ] を使用して配置する。
2. 固定ポップアップダイアログで次の表の項目を定義する。

定義項目	定義内容
使用目的	POP ( 手動 - 日本語 )
詳細目的	混在 ( 全角・半角 )
最初にメニューを開く	指定しない
選択ラベルと通知コードを同じ値にする	指定する ( この定義内容は固定 )
タイトルを付ける	指定する ( 支店 )



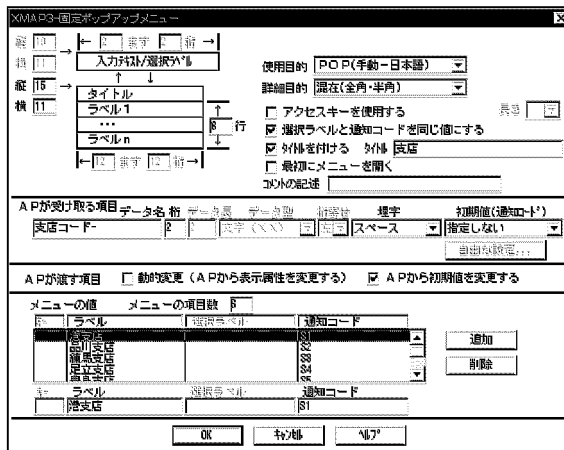
定義項目	定義内容
データ名	支店コード -
データ型	文字 (XX)(この定義内容は固定)
桁	2
データ長	2(この定義内容は固定)
動的変更 (AP から表示属性を変更する)	指定しない
AP から初期値を変更する	指定する

## 注

通知コードとして半角文字専用にする場合は、「POP(手動-英数)」を選んでください。

また、次に示す定義内容をラベルと通知コードのリストボックスに追加していきます。

ラベル	通知コード
港支店	S1
品川支店	S2
練馬支店	S3
足立支店	S4
豊島支店	S5



## 《手動ポップアップの設定》

ポップアップメニューを PF1 キーで表示させるための定義をします。

1. GUI 画面定義ウィンドウの、画面属性ダイアログで、「イベント通知コード」の [設定] ボタンを選択する。
2. イベント通知コードのダイアログで次の項目を定義する。

## 6. 画面定義とコーディングの関連

定義項目	定義内容
ポップアップの表示タイミング	「確定キーを使用する」を選択します。
アクション	ポップアップ表示 (「確定キーを使用する」を選んだ場合だけ指定できます。PF キーにポップアップ表示の動作を割り当てます)。

### 注

プッシュボタンやメニューバーで手動ポップアップを表示する場合は、手動ポップアップの定義をしたあとに、プッシュボタンやメニューバーでのポップアップ表示の定義をしてください。

### 《プッシュボタンの定義》

ポップアップメニューを表示させるためのプッシュボタンを定義します。

1. ポップアップ付きのプッシュボタンは GUI 画面定義ウィンドウで、[ プッシュボタン ( 単独 / 追加 ) ] を使用して配置する。
2. プッシュボタンダイアログで次の項目を定義する。

定義項目	定義内容
ラベル	支店
選択時の動作	ポップアップ表示

### (c) 生成される論理マップ

生成される論理マップは、自動的にポップアップメニューを表示するときと同じです。

「6.2.19(1)(c) 生成される論理マップ」を参照してください。

### (d) COBOL での AP

COBOL での AP は自動的にポップアップメニューを表示するときと同じです。

「6.2.19(1)(d) COBOL での AP」を参照してください。

## (3) アクセスキーを使用した固定ポップアップのメニュー選択

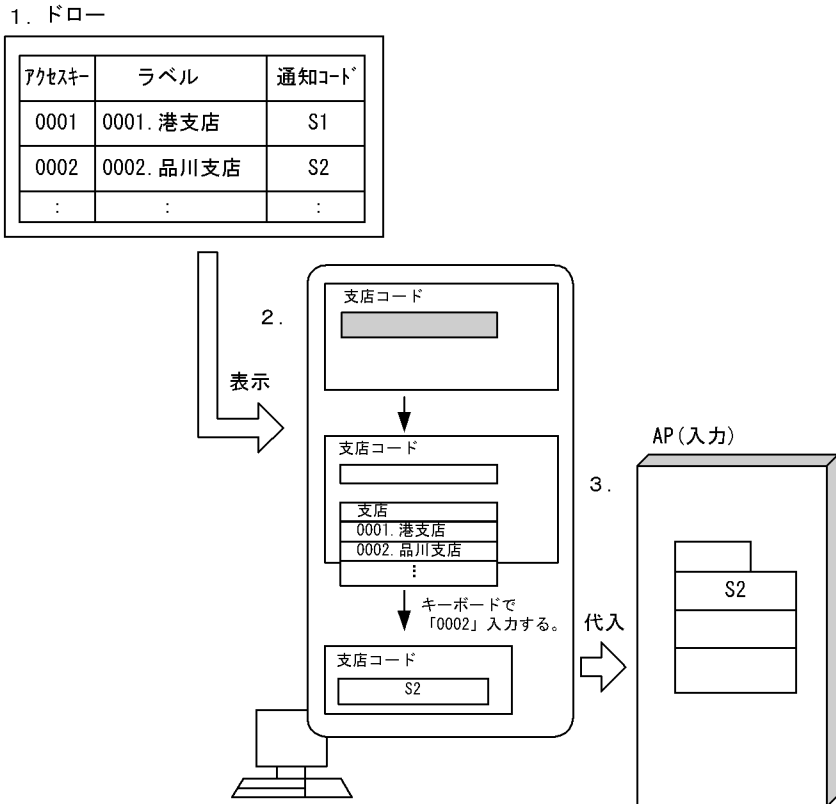
アクセスキーは自動ポップアップでも手動ポップアップでも利用できます。ここでは、自動ポップアップを仮定します。

### (a) 仕様

「支店コード」の入出力テキストにフォーカスが位置づけられると、ポップアップメニューが自動的に表示されます。「支店」のポップアップメニューからアクセスキーを使用してデータを選択すると、対応する通知コードが入力論理マップに代入されます。「支店コード」の入出力テキストには、選択されたメニューに対応する通知コード「S2」が表示されます。

固定ポップアップメニューを使用した、AP 実行時のポップアップメニューの自動表示を次の図に示します。

図 6-24 アクセスキーでポップアップメニューを選択する（自動）



1. 固定ポップアップの場合、ラベル、通知コード、選択ラベルおよびアクセスキーは、ドローに定義されています。
2. 自動ポップアップの場合、「支店コード」の入出力テキストにフォーカスが位置づくとき、ポップアップメニューが自動的に表示されます。  
手動ポップアップの場合は、キー操作などでポップアップメニューを表示します。\*
3. アクセスキーを入力すると、対応する通知コードが、APに返されます。  
「支店コード」の入出力テキストにも、通知コードと同じ値が表示されます。

注※ アクセスキーは、ポップアップメニューが表示されているときだけ有効です。

### (b) 画面定義の概要

画面定義の概要を次に示します。

1. ポップアップに表示するメニューを AP から動的に変更できるようにするには、GUI 画面定義ウィンドウで、ツールボックスの [ 固定ポップアップ ] を使用して配置する。
2. 固定ポップアップメニューダイアログで次の表の項目を定義する。

## 6. 画面定義とコーディングの関連

定義項目	定義内容
使用目的	POP (自動 - 日本語)
詳細目的	混在 (全角・半角)
長さ (アクセスキー)	4
アクセスキーを使用する	指定する
選択ラベルと通知コードを同じ値にする	指定する
タイトルを付ける	指定する (支店)
データ名	支店コード -
データ型	文字 (XX) (この定義内容は固定)
桁	2
データ長	2 (この定義内容は固定)
メニューの項目数	6

### 注

通知コードとして半角文字専用にする場合は、「POP (自動 - 英数)」または「POP (手動 - 英数)」を選んでください。

また、次に示す定義内容を「メニューの値」のラベルと通知コードに追加していきます。

キー	ラベル	通知コード
0001	0001. 港支店	S1
0002	0002. 品川支店	S2
0003	0003. 練馬支店	S3
0004	0004. 足立支店	S4
0005	0005. 豊島支店	S5

メニューの値	メニューの項目数	通知コード
0001 品川支店	1	S1
0002 品川支店	1	S2
0003 品川支店	1	S3
0004 品川支店	1	S4
0005 品川支店	1	S5
0001 品川支店	1	S1

### (c) 生成される論理マップ

生成される論理マップは、自動的にポップアップメニューを表示するときと同じです。  
「6.2.19(1)(c) 生成される論理マップ」を参照してください。

### (d) COBOL での AP

COBOL での AP は自動的にポップアップメニューを表示するときと同じです。  
「6.2.19(1)(d) COBOL での AP」を参照してください。

## (4) ポップアップが選択されたときの選択ラベルと通知コードを別々にする

ポップアップの選択ラベルと通知コードを同じ値にするかどうかは、自動ポップアップでも手動ポップアップでも指定できます。ここでは、自動ポップアップを仮定します。

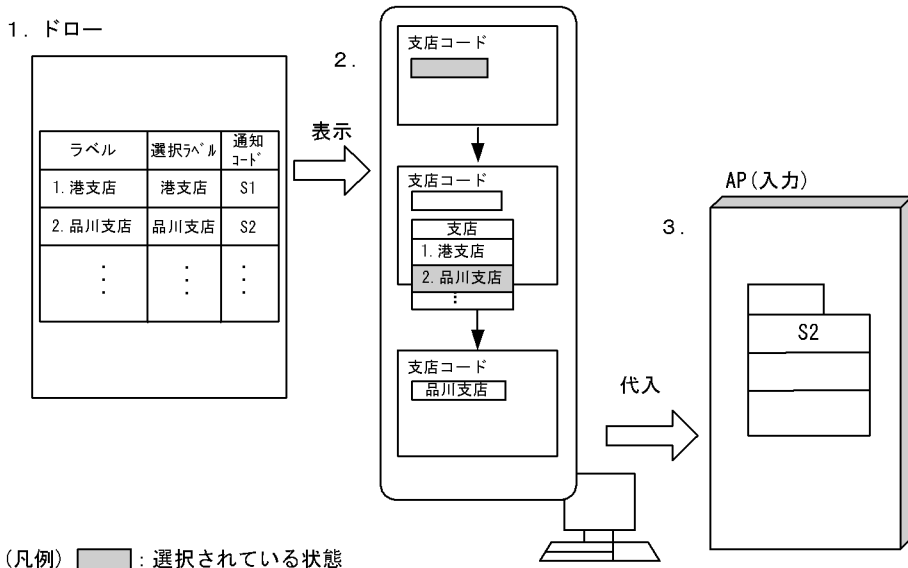
### (a) 仕様

「支店コード」の入出力テキストにフォーカスが位置づけられると、ポップアップメニューが自動的に表示されます。「支店」のポップアップメニューからデータを選択すると、対応する通知コードが入力論理マップに代入されます。「支店コード」の入出力テキストには、選択されたメニューに対応する選択ラベル「品川支店」が表示されます。AP には、選択ラベルに対応した通知コード「S2」が返されます。

固定ポップアップメニューを使用した、AP 実行時のポップアップメニューの自動表示を次の図に示します。

6. 画面定義とコーディングの関連

図 6-25 ポップアップが選択されたときの選択ラベルと通知コードを別々にする（自動）



1. 固定ポップアップの場合、ラベル、選択ラベルおよび通知コードは、ドローに定義されています。
2. 自動ポップアップの場合は、「支店コード」の入出力テキストにフォーカスが位置すると、ポップアップメニューが自動的に表示されます。手動ポップアップの場合は、キー操作などでメニューを表示します。
3. ポップアップメニューで選択したラベルに対応する選択ラベルが表示され、選択ラベルに対応した通知コードがAPに返されます。

(b) 画面定義の概要

画面定義の概要を次に示します。

1. ポップアップに表示するメニューを AP から動的に変更できるようにするには、GUI 画面定義ウィンドウで、ツールボックスの [ 固定ポップアップ ] を使用して配置する。
2. 固定ポップアップメニューダイアログで次の表の項目を定義する。

定義項目	定義内容
使用目的	POP (自動 - 日本語)
詳細目的	混在 (全角・半角)
アクセスキーを使用する	指定する
選択ラベルと通知コードを同じ値にする	指定しない
タイトルを付ける	指定する (支店)
データ名	支店コード -
データ型	文字 (XX) (この定義内容は固定)
桁	2

定義項目	定義内容
データ長	2 (この定義内容は固定)
メニューの項目数	6

## 注

通知コードとして半角文字専用にする場合は、「POP (自動 - 英数)」または「POP (手動 - 英数)」を選んでください。

また、次に示す定義内容を「メニューの値」のラベルと通知コードに追加していきます。

キー	ラベル	選択ラベル	通知コード
1	1. 港支店	港支店	S1
2	2. 品川支店	品川支店	S2
3	3. 練馬支店	練馬支店	S3
4	4. 足立支店	足立支店	S4
5	5. 豊島支店	豊島支店	S5

## (c) 生成される論理マップ

生成される論理マップは、自動的にポップアップメニューを表示するときと同じです。  
「6.2.19(1)(c) 生成される論理マップ」を参照してください。

## (d) COBOL での AP

COBOL での AP は自動的にポップアップメニューを表示するときと同じです。  
「6.2.19(1)(d) COBOL での AP」を参照してください。

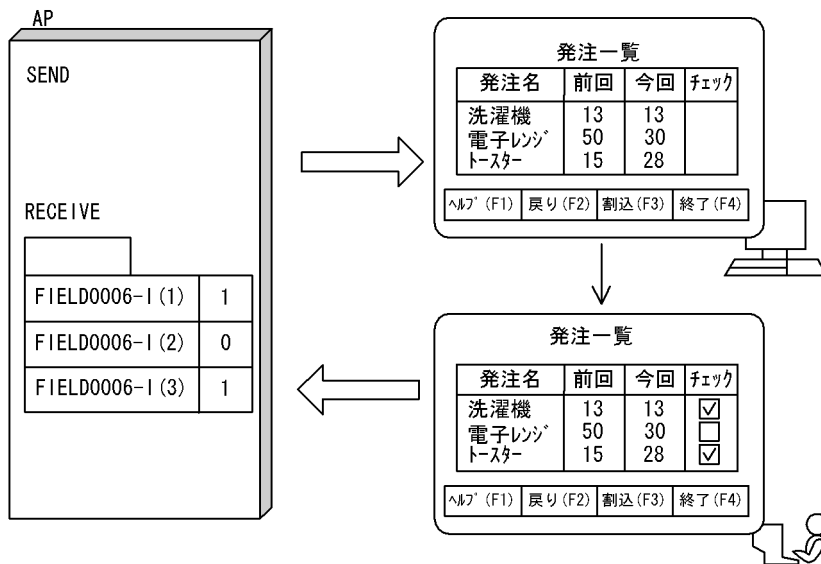
## 6.2.20 トグルフィールド

### (1) 仕様

「発注一覧」の表の各行をチェックし、内容を確認後、発注処理を許可するものに「」印を付けます。画面入力後、APでは「」が付いているかどうかを判定し、付いているフィールドに対する処理を行います。

トグルフィールドでの選択を次の図に示します。

図 6-26 トグルフィールド



### (2) 画面定義の概要

画面定義の概要を次に示します。

1. GUI画面のフィールドボックス中にフレームを定義し、その中へツールボックスの [トグルフィールド] を使用してトグルフィールドを配置する。
2. トグルフィールドダイアログで次の表の項目を定義する。

定義項目	定義内容
使用目的	数字
トグル種別	チェック ( <input checked="" type="checkbox"/> )
通知コード	1
データ名	FIELD0006- (任意)



定義項目	定義内容
桁	1
データ長	1 (この定義内容は固定)

### (3) 生成される論理マップ

マップ名が MAP001 の場合の例

```

01 MAP001I. ....入力論理マップ
02 MAP001S
02 MAP001O
02 MAP001K.
:
03 MAP001-FRAME0001-I OCCURS 8.
04 MAP001-FIELD0006-H.
05 MAP001-FIELD0006-I PIC 9. ....トグルフィールド

```

## (4) COBOL での AP

```

WORKING-STORAGE SECTION.
* 論理マップのコピー展開
  COPY MAP0010.
  COPY MAP0011.
* 動的変更テーブルのコピー展開
  COPY X3MODTBL.
  :
  :
COMMUNICATION SECTION.
  CD DSP FOR I-O WS
  MAP NAME IS 画面マップ名…………… 物理マップ名称格納エリア
  SYMBOLIC TERMINAL IS 画面端末名.
                        …………… 画面入出力する端末名
  :
PROCEDURE DIVISION.
  :
  MOVE 'DSP001' TO 画面端末名. …………… 端末名の設定
  MOVE 'MAP001ND' TO 画面マップ名. ……………物理マップ名の設定
  MOVE ALL X'1F' TO MAP001G.
                        ……………出力エリアのクリア処理
  :
  TRANSCEIVE DSP FROM MAP0010 INTO MAP0011.
                        ……………画面の表示&入力
  :
  :
  IF MAP0001-FIELD0006-H(01) = '1'
                        ……………トグルフィールドが選択された時の処理を行う
  THEN
    MOVE '発注' TO MAP001-FIELD0005-0
  ELSE
    MOVE ' ' TO MAP001-FIELD0005-0
  END-IF.
  :
  :

```

## 6.2.21 入出力・出力フィールドの一括反復（フレームの使用）

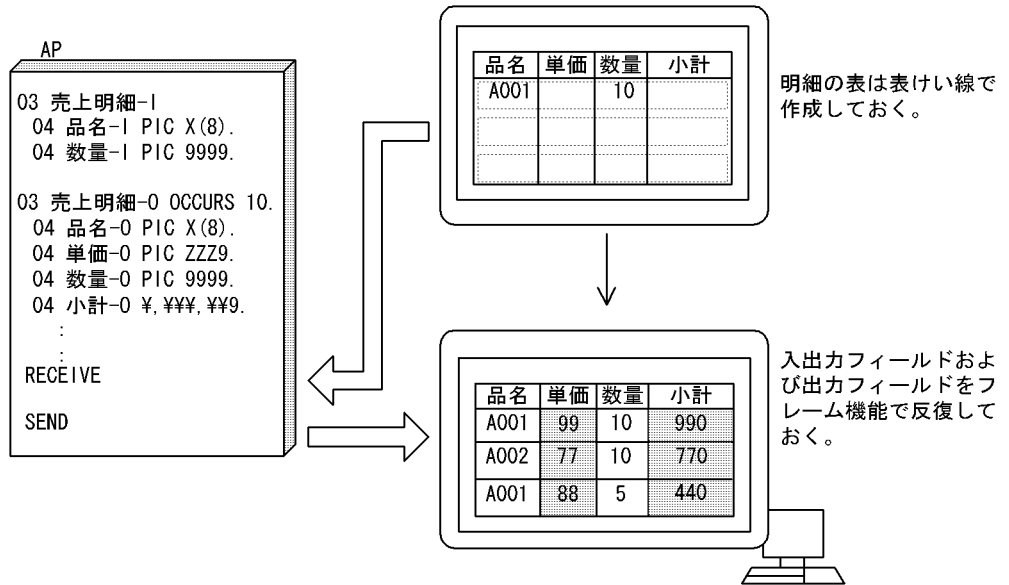
入出力フィールドと出力フィールドを1行とみなして、縦方向または横方向に反復表示します。APでは、この1列および1行を集団項目として扱います。なお、GUI画面ではフィールドボックス内でだけ指定できます。

## (1) 仕様

1行の中に入出力フィールドとして「品名」「数量」、出力フィールドとして「単価」「小計」がある明細があります。これらのフィールドを、ツールボックスの[フレーム(縦繰り返し)]でまとめると、異なるフィールドから成る1行とみなされ、行単位での縦反復ができます。フレームを使用してフィールドを反復させる場合は、まず、フレームで反復する領域を指定します。そして、指定した領域の先頭行に反復させたいフィールドを設定します。反復させたいフィールドが複数ある場合は、一つずつ設定します。

フレームでの反復を次の図に示します。

図 6-27 フレームでの縦反復



## (2) 画面定義の概要

画面定義の概要を次に示します。

1. GUI 画面のフィールドボックス中にフレームを配置する。
2. フレームダイアログで次の表の項目を定義する。

定義項目	定義内容
データ名	売上明細

## 6. 画面定義とコーディングの関連

MAP3-フレーム

縦 1  
横 5

3 実行  
11 実行

A Pが渡す項目      フレームの反復  
データ名      方向  縦    横  
売上明細      回数   3  
                 間隔   0

OK      キャンセル      ヘルプ

### (3) 生成される論理マップ

マップ名が MAP002 の場合の例

```

<出力論理マップ>
01 MAP0020.
02 MAP002L PIC S9(4) COMP-5 VALUE +258.
02 MAP002Z PIC S9(4) COMP VALUE +0.
02 MAP002G.
03 MAP002-CNTRLO PIC X(2).
03 MAP002-OUTCURS-LOC0 PIC X(2).
03 MAP002-売上明細-0 OCCURS 10.
04 MAP002-品名-0 PIC X(8).
04 MAP002-単価-0 PIC ZZZ9.
04 MAP002-数量-0 PIC 9999.
04 MAP002-小計-0 PIC ¥,¥¥¥,¥¥9.

<入力論理マップ>
01 MAP002I.
02 MAP002S PIC S9(4) COMP-5 VALUE +130.
02 MAP0020 PIC S9(4) COMP VALUE +0.
02 MAP002K.
03 MAP002-INC1 PIC X(4).
03 MAP002-INCURS-LOC1 PIC X(2).
03 MAP002-売上明細-I OCCURS 10.
04 MAP002-品名-H.
05 MAP002-品名-I PIC X(8).
04 MAP002-数量-H.
05 MAP002-数量-I PIC 9999.
    
```

## (4) COBOL での AP

```

WORKING-STORAGE SECTION.
* 論理マップのCOPY展開
  COPY MAP0020.
  COPY MAP0021.
* 動的変更テーブルのCOPY展開
  COPY X3MODTBL.
  :
  :
COMMUNICATION SECTION.
  CD DSP FOR I-0 WS
  MAP NAME IS 画面マップ名 ..... 物理マップ名称格納エリア
  SYMBOLIC TERMINAL IS 画面端末名. .... 画面入出力する端末名

  :
  :
PROCEDURE DIVISION.
  :
  MOVE 'DSP001' TO 画面端末名. .... 端末名の設定
  MOVE 'MAP002ND' TO 画面マップ名. .... 物理マップ名の設定
  MOVE ALL X'1F' TO MAP002G.
  ..... 出力エリアのクリア処理

  TRANSCIEVE DSP FROM MAP0020 INTO MAP0021.
  : ..... 画面の表示&入力(品名と数量を入力する)
  :
  MOVE xxxxxx TO T-WORK. .... 品名に対応する単価をDBより取得する
  :
  MOVE T-WORK TO MAP002-単価-0(1).
  ..... フレーム中の「単価」への出力データ設定
  COMPUTE S-WORK = T-WORK * MAP002-数量-1(1). ... 単価と数量から小計を求める
  MOVE S-WORK TO MAP002-小計-0(1).
  ..... フレーム中の「小計」への出力データ設定
  : .....以降, 入力のあった明細行分の処理を行う
  :
  TRANSCIEVE DSP FROM MAP0020 INTO MAP0021. ... 画面の再表示&入力
  :

```

## 6.2.22 メニュー選択コンボボックス(可変)を使用した候補の選択

## (1) 仕様

「地区名」のメニュー選択コンボボックスに「札幌」を初期表示しています。矢印ボタンを押すと、メニュー項目(ラベル)の一覧が表示されます。メニュー項目(ラベル)を選択すると、選択したラベルがボックス内に表示されます。画面の確定後には、選択したラベルに対応する通知コードが入力論理マップに代入されます。

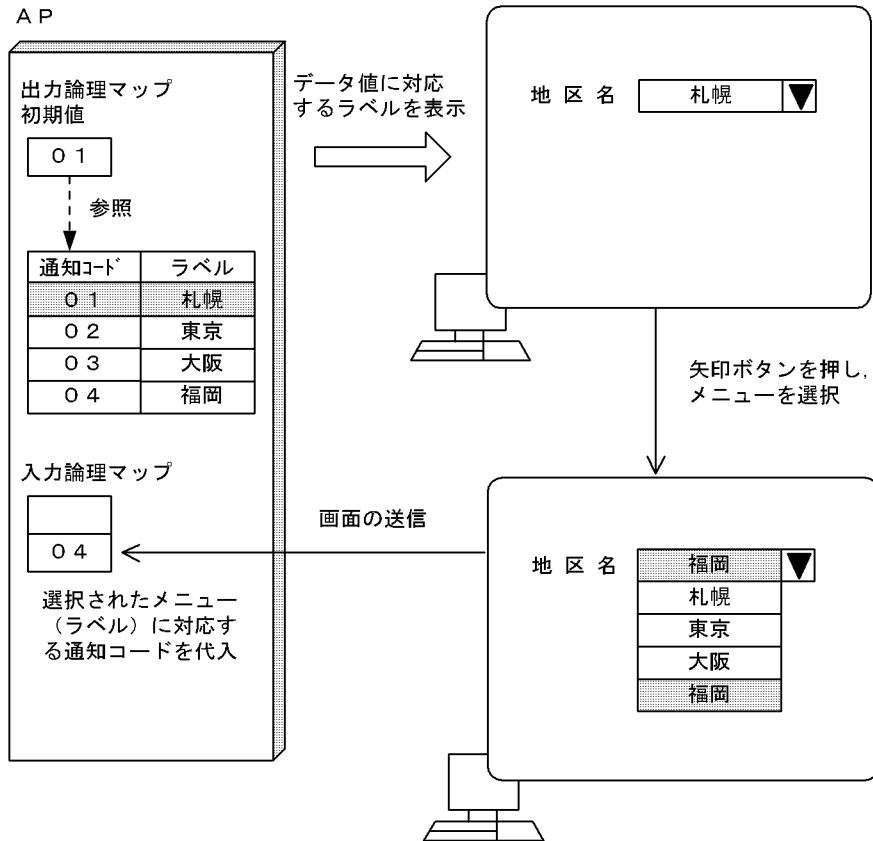
なお、画面表示時に2番目以降のメニューラベルを表示させたい場合は、メニューラベルと通知コードを必ず設定してください。メニューラベルおよび通知コードにデータ有無コードを設定して、前回表示時のメニューを継続した場合は、先頭のメニューラベル

## 6. 画面定義とコーディングの関連

が表示されます。

メニュー選択コンボボックス（可変）を使用した AP 実行時のコンボボックスの表示を次の図に示します。

図 6-28 メニュー選択コンボボックス（可変）を使用したデータの入力



### （2）画面定義の概要

画面定義の概要を次に示します。

1. GUI 画面定義ウィンドウで、ツールボックスの [ 可変コンボボックス ] を使用してコンボボックスを配置する。
2. コンボボックスダイアログで次の表の項目を定義する。

定義項目		定義内容
使用目的		日本語
桁		10
AP が受け取る通知コード	データ名	地区名 -

定義項目		定義内容
の項目	桁	2
	データ長	2 (この定義内容は固定)

## 注

通知コードとして半角文字専用にする場合は、「英数」を選んでください。

## 3. 可変コンボメニューダイアログで、次の表の項目を定義する。

定義項目		定義内容
メニューの項目数		4
ラベル	データ名	地区名 - LABEL -
	データ型	文字 (XX)
AP に返す通知コード	データ名	地区名 - CODE -

### (3) 生成される論理マップ

マップ名が MAP413 の場合の例

01	MAP4130. (出力論理マップ)		
:			
02	MAP413G.		… 一括クリ7用名称
03	MAP413-CNTRL0	PIC X(2).	
03	MAP413-地区名-0	PIC X(2).	… 出力時の初期選択エリア
03	MAP413-POPUP0001-0	OCCURS 4.	… 論理テーブル名称
04	MAP413-地区名-CODE-0	PIC X(2).	… 通知コード項目
04	MAP413-地区名-LABEL-0	PIC X(10).	… ラベル項目
01	MAP4131. (入力論理マップ)		
:			
02	MAP413K.		… 一括クリ7用名称
03	MAP413-INC1	PIC X(4).	
03	MAP413-地区名-H.		
04	MAP413-地区名-I	PIC X(2).	… 通知コードの入力エリア

#### 注

「AP が受け取る通知コードの項目」のデータ名を省略した場合、FIELDnnnn- が仮定されます。「メニュー」のデータ名を省略した場合、ラベル名 POPUP-LABELnnnn-、AP に返すメニューコード名 POPUP-CODEnnnn- が仮定されます (nnnn : 0001 ~ 9999)。

また、可変コンボボックスに対して一つの論理テーブル名が付けられます。



## (4) COBOL での AP

マップ名が MAP413 の場合の例

```

WORKING-STORAGE SECTION.
* 論理マップのCOPY展開
COPY    MAP4130.          .....   出力論理マップ
COPY    MAP4131.          .....   入力論理マップ
* 動的変更テーブルのCOPY展開
COPY    X3MODTBL.
COMMUNICATION SECTION.          .....   通信記述
CD DSP FOR I-O WS              .....   通信記述名: DSP
MAP NAME IS 画面マップ名      .....   物理マップ 名格納エリア
SYMBOLIC TERMINAL IS 画面端末名 .....   端末名: 画面端末名
:
PROCEDURE DIVISION.
* 画面表示処理
MOVE 'DSP001' TO 画面端末名.
:
MOVE ALL X'1F' TO MAP413G.      .....出力論理マップのクリア

* 出力論理マップへのメニューの通知コードとラベルの設定
MOVE '01' TO MAP413-地区名-CODE-0(1).
MOVE '札幌' TO MAP413-地区名-LABEL-0(1).

MOVE '01' TO MAP413-地区名-0.    ...   初期状態の設定
MOVE 'MAP413ND' TO 画面マップ名. ...   物理マップ 名称の設定
TRANSCEIVE DSP FROM MAP4130 INTO MAP4131. ....画面の送受信
EVALUATE MAP413-地区名-I         .....選択されたメニューに
                                対応する通知コードの処理

                                WHEN '01'
                                  PERFORM 北海道地区の処理
                                WHEN '02'
                                  PERFORM 東日本地区の処理
                                WHEN '03'
                                  PERFORM 西日本地区の処理
                                WHEN '04'
                                  PERFORM 九州地区の処理

END-EVALUATE.

```

### 6.2.23 メニュー選択コンボボックス（固定）を使用した候補の選択

#### (1) 仕様

「6.2.22 メニュー選択コンボボックス（可変）を使用した候補の選択」と同じ仕様で、メニュー選択コンボボックス（固定）を利用する場合の説明をします。

#### (2) 画面定義の概要

画面定義の概要を次に示します。

1. GUI 画面定義ウィンドウで、ツールボックスの [ 固定コンボボックス ] を使用してコ

## 6. 画面定義とコーディングの関連

ンボボックスを配置する。

2. コンボボックスダイアログで次の表の項目を定義する。

定義項目	定義内容
使用目的	日本語
桁	10
AP が受け取る 通知コードの項目	データ名
	桁
	データ長
	2 (この定義内容は固定)

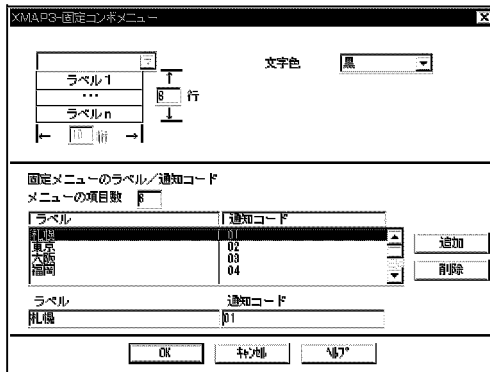
注

通知コードとして半角文字専用にする場合は、「英数」を選んでください。

### メニューの定義

1. 固定コンボメニューダイアログで、次に示す内容をラベルと通知コードのリストに追加する。
2. 「ラベル」は、メニュー項目として表示されます。「通知コード」は、AP に返すコードデータの値です。

ラベル	通知コード
札幌	01
東京	02
大阪	03
福岡	04



### (3) 生成される論理マップ

マップ名が MAP413 の場合の例

01	MAP4130. (出力論理マップ)	
:		
02	MAP413G.	… 一括クリ用名称
03	MAP413-CNTRL0	PIC X(2).
03	MAP413-地区名-0	PIC X(2). … 出力時の初期選択エリア
01	MAP413I. (入力論理マップ)	
:		
02	MAP413K.	… 一括クリ用名称
03	MAP413-INCL	PIC X(4).
03	MAP413-地区名-H.	
04	MAP413-地区名-I	PIC X(2). … 通知コードの入力エリア

#### 注

「AP が受け取る通知コードの項目」のデータ名を省略した場合、FIELDnnnn- (nnnn : 0001 ~ 9999) が仮定されます。

## 6. 画面定義とコーディングの関連

### (4) COBOL での AP

マップ名が MAP413 の場合の例

```
WORKING-STORAGE SECTION.
* 論理マップのCOPY展開
COPY    MAP4130.  .....   出力論理マップ
COPY    MAP4131.  .....   入力論理マップ
* 動的変更テーブルのCOPY展開
COPY    X3MODTBL.
COMMUNICATION SECTION.      .....   通信記述
CD DSP FOR I-0 WS  .....   通信記述名:DSP
MAP NAME IS 画面マップ名  .....   物理マップ名格納エリア
:
PROCEDURE DIVISION.
* 画面表示処理
MOVE ALL X'1F' TO MAP413G.  ...   出力論理マップのクリア
MOVE '01' TO MAP413-地区名-0. ... 初期状態の設定
MOVE 'MAP413ND' TO 画面マップ名. ... 物理マップ名称の設定
TRANSC EIVE DSP FROM MAP4130 INTO MA04131.
.....
.....   画面の送受信
EVALUATE MAP413-地区名-1  .....   選択されたメニューに
.....   対応する通知コードの
.....   処理

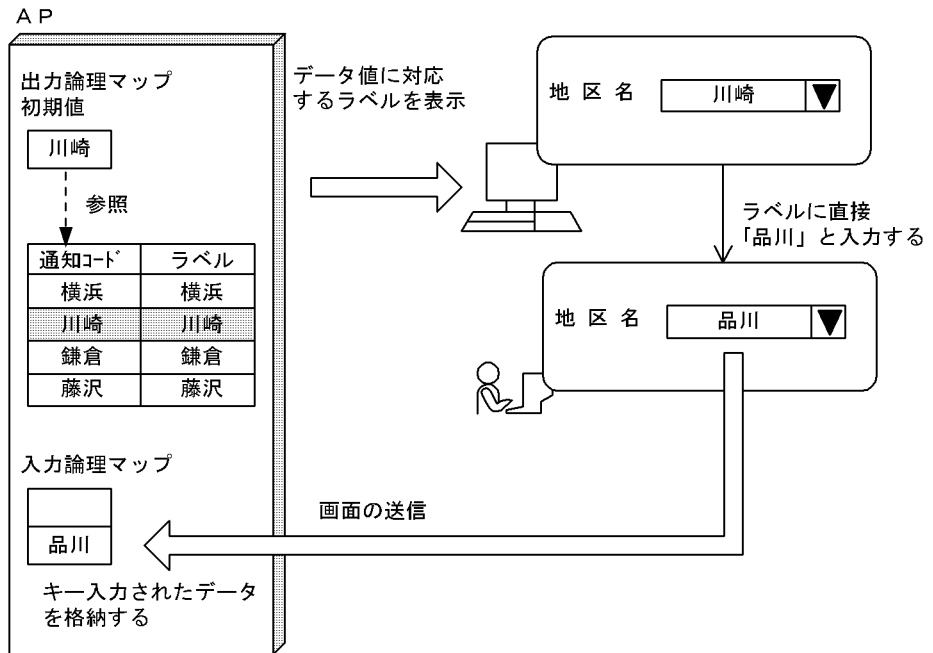
      WHEN '01'
        PERFORM 北海道地区の処理
      WHEN '02'
        PERFORM 東日本地区の処理
      WHEN '03'
        PERFORM 西日本地区の処理
      WHEN '04'
        PERFORM 九州地区の処理
END-EVALUATE.
```

## 6.2.24 キー入力コンボボックス（可変）を使用したデータの 入力

### (1) 仕様

「地区名」のキー入力コンボボックスに「川崎」を初期表示しています。コンボボックスにフォーカスを位置づけ、ラベルの内容をキー入力で書き換えます。画面の確定後には、ラベルに入力したデータに対応するコードが入力論理マップに代入されます。キー入力コンボボックス（可変）を使用した AP 実行時のコンボボックスの表示を次の図に示します。

図 6-29 キー入力コンボボックス（可変）を使用したデータの入力



## (2) 画面の概要

画面定義の概要を次に示します。

1. GUI 画面定義ウィンドウで、ツールボックスの [ キー入力可変コンボボックス ] を使用してコンボボックスを配置する。
2. キー入力コンボボックスダイアログで次の表の項目を定義する。

定義項目	定義内容	
使用目的	日本語	
桁	4	
AP が受け取る通知コードの項目	データ名	地区名
	桁	4
	データ長	4 (この定義内容は固定)
	データ型	文字 (XX) (この定義内容は固定)

注

通知コードとして半角文字専用にする場合は、「英数」を選んでください。

## 6. 画面定義とコーディングの関連

このダイアログボックスには、以下のような設定項目があります。

- 文字色: 黒
- 文字サイズ: 標準
- 文字の書体: 標準
- 文字の強調: 標準
- 背景色: 白
- 字 高: 2.5ます
- 使用目的: 日本語
- 詳細目的: 現在(全角・半角)
- 表示方法: 入力可能-標準表示
- 遷移条件: 自動スキップ
- 反転表示する
- フォーカス設定 (初期フォーカスを本ボックスに設定する)
- 入力済み (入力しなくても通知コードをAPに返す)
- 自動送信 (入力直後にAPへ制御を返す)
- 入力必須 (入力しない時はエラーにする)
- のみが有効 (選択状態にする)

また、「動的変更 (APから表示属性を変更する)」と「APから初期値を変更する」のチェックボックスが有効になっています。

3. キー入力可変コンボメニューダイアログで、次の表の項目を定義する。

定義項目	定義内容	
メニューの項目数	4	
ラベル&通知コード	データ名	地区名 - CODE -
	データ型	キー入力コンボボックスダイアログの「APが受け取る通知コードの項目」の値と同じ

このダイアログボックスには、以下のような設定項目があります。

- 文字色: 黒
- メニューの項目数: 4
- ラベル&通知コードのデータ名: 地区名-CODE-
- データ型: 文字 (XX)
- データ長: 2
- データ型: 左
- データ型: スペース

## (3) 生成される論理マップ

マップ名が MAP001 の場合の例

01	MAP0010. (出力論理マップ)			
	:			… 一括クリア用名称
02	MAP001G.			
	03	MAP001-CNTRL0	PIC X(2).	
	03	MAP001-地区名-0	PIC X(4).	… 出力時の初期選択エリア
	03	MAP001-POPUP0001-0	OCCURS 4.	… 論理テーブル名称
	04	MAP001-地区名-CODE-0	PIC X(4).	… 通知コード項目
01	MAP0011. (入力論理マップ)			
	:			… 一括クリア用名称
	02	MAP001		
	03	MAP001-INCI	PIC X(4)	
	03	MAP001-地区名-H.		
	04	MAP001-地区名-I	PIC X(4).	… 通知コードの入力エリア

(4) COBOL での AP

```

WORKING-STORAGE SECTION.
*論理マップのCOPY展開
COPY MAP0010.                ..... 出力論理マップ
COPY MAP0011.                ..... 入力論理マップ
COMMUNICATION SECTION.
CD DSP FOR I-O WS           ..... 通信記述
MAP NAME IS画面マップ名    ..... 物理マップ名称格納エリア
SYMBOLIC TERMINAL IS画面端末名 ..... 端末名：画面端末名
:
PROCEDURE DIVISION.
*画面表示処理
MOVE 'DSP001' TO 画面端末名.
:
MOVE ALL X'1F' TO MAP001G.    ..... 出力論理マップのクリア

*出力論理マップへのメニューの通知コードとラベルの設定
MOVE '横浜' TO MAP001-地区名-CODE-0(1).

MOVE '川崎' TO MAP001-地区名-0.    ..... 初期状態を川崎に設定
MOVE 'MAP001ND' TO 画面マップ名.  ..... 物理マップ名称の設定
TRANSCEIVE DSP FROM MAP001 INTO MAP0011.  ..... 画面の送受信
EVALUATE MAP001-地区名-1          ..... 入力された通知コードに対応する処理
    WHEN '横浜'
        PERFORM 横浜地区の処理
    WHEN '川崎'
        PERFORM 川崎地区の処理
    WHEN '鎌倉'
        PERFORM 鎌倉地区の処理
    WHEN '藤沢'
        PERFORM 藤沢地区の処理
    WHEN OTHER
        PERFORM その他の地区の処理

END-EVALUATE.
    
```

6.2.25 キー入力コンボボックス (固定) を使用したデータの  
 の入力

(1) 仕様

「6.2.24 キー入力コンボボックス (可変) を使用したデータの入力」と同じ仕様で、  
 キー入力コンボボックス (固定) を利用する場合を説明します。

(2) 画面定義の概要

画面定義の概要を次に示します。

1. GUI 画面定義ウィンドウで、ツールボックスの [ キー入力固定コンボボックス ] を使  
 用してコンボボックスを配置する。
2. キー入力コンボボックスダイアログで次の表の項目を定義する。



定義項目		定義内容
使用目的		日本語
桁		4
AP が受け取る	データ名	地区名 -
通知コードの項目	桁	4
	データ長	4 (この定義内容は固定)

## 注

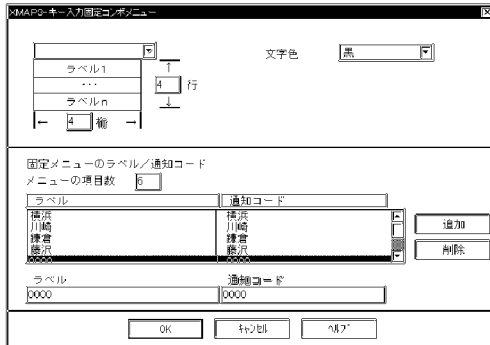
通知コードとして半角文字専用にする場合は、「英数」を選んでください。

## メニューの定義

1. キー入力固定コンボメニューダイアログで、次に示す内容をラベルのリストに追加する。
2. 「ラベル」はメニュー項目として表示されます。「通知コード」はAP に返すコードデータの値です。

ラベル	通知コード
横浜	ラベルの指定が反映される
川崎	
鎌倉	
藤沢	

## 6. 画面定義とコーディングの関連



### (3) 生成される論理マップ

マップ名が MAP001 の場合の例

01	MAP0010. (出力論理マップ)	
:		
02	MAP001G.	…一括クリア用名称
03	MAP001-CNTRLO PIC X(2).	
03	MAP001-OUTCURS-LOC0 PIC X(2).	
03	MAP001-地区名-0 PIC X(6).	…出力時の初期選択エリア
01	MAP0011. (入力論理マップ)	
:		
02	MAP001K.	…一括クリア用名称
03	MAP001-INCI PIC X(4).	
03	MAP001-INCURS-LOC1 PIC X(2).	
03	MAP001-地区名-H.	
04	MAP001-地区名-I PIC X(6).	…通知コードの入力エリア

## (4) COBOL での AP

マップ名が MAP001 の場合の例

```

WORKING-STORAGE SECTION.
*論理マップのCOPY展開
COPY MAP0010.          ..... 出力論理マップ
COPY MAP0011.          ..... 入力論理マップ
COMMUNICATION SECTION.
..... 通信記述
CD DSP FOR I-O WS      ..... 通信記述名 : DSP
MAP NAME IS画面マップ名 ..... 物理マップ名称格納エリア
SYMBOLIC TERMINAL IS画面端末名 ..... 端末名 : 画面端末名
:
PROCEDURE DIVISION.
*画面表示処理
MOVE 'DSP001' TO 画面端末名.
:
MOVE ALL X'1F' TO MAP001G.          ..... 出力論理マップのクリア

MOVE '川崎' TO MAP001-地区名-0.      ..... 初期状態を川崎に設定
MOVE 'MAP001ND' TO 画面マップ名.    ..... 物理マップ名称の設定
TRANSCEIVE DSP FROM MAP001 INTO MAP0011. ..... 画面の送受信
EVALUATE MAP001-地区名-1
..... 入力された通知コードに
..... 対応する処理

      WHEN '横浜'
        PERFORM 横浜地区の処理
      WHEN '川崎'
        PERFORM 川崎地区の処理
      WHEN '鎌倉'
        PERFORM 鎌倉地区の処理
      WHEN '藤沢'
        PERFORM 藤沢地区の処理
      WHEN OTHER
        PERFORM その他の地区の処理

END-EVALUATE.

```

## 6.2.26 可変ラジオボタンを使用した候補の選択

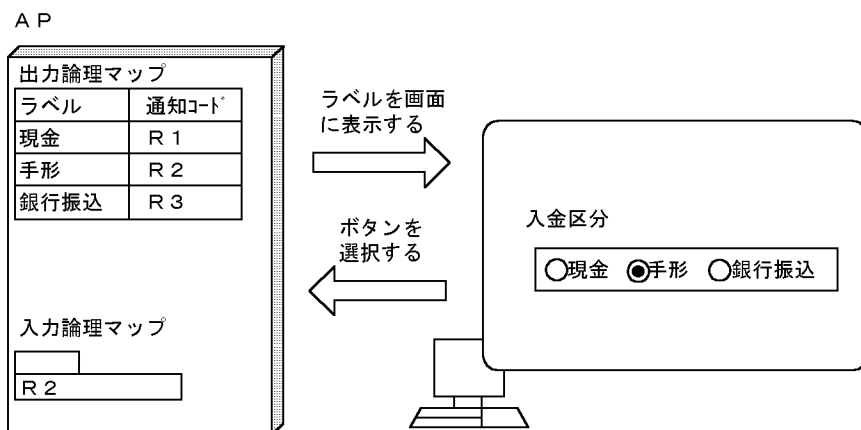
## (1) 仕様

入金区分のラジオボタンボックスで [現金], [手形], または [銀行振込] のラジオボタンから一つだけボタンを選択するとオン状態で表示し、それぞれのボタンに対応した通知コードを入力論理マップに代入します。

可変ラジオボタンを使用する場合、表示するボタンのラベル、およびボタンに対応する通知コードを AP で出力論理マップに代入します。

AP 実行時でのラジオボタンの表示を次の図に示します。

図 6-30 可変ラジオボタンを使用したデータの選択



## (2) 画面定義の概要

画面定義の概要を次に示します。

1. GUI 画面定義ウィンドウで、ツールボックスの [ 可変ラジオボタン ] を使用してラジオボタンを 3 個配置する。
2. 可変ラジオボタンボックスダイアログで次の表の項目を定義する。

定義項目		定義内容	
使用目的		日本語	
桁		8	
AP が受け取る項目	データ名	入金区分 -	
	データ型	文字 (XX)	
	桁	2	
	データ長	2 (この定義内容は固定)	
AP が渡す項目	ラベル	データ名	入金ラベル -
		データ型	文字 (XX)
		桁	8
		データ長	8 (この定義内容は固定)
	通知コード	データ名	入金コード -
		データ型	文字 (XX) (この定義内容は固定)
		桁	2 (この定義内容は固定)
		データ長	2 (この定義内容は固定)

注

通知コードとして半角文字専用にする場合は、「英数」を選んでください。

MAP116-可変ラジオボタンボックス

ボタン

ボタンの数: 1

使用目的: 日本語

表示方法: 標準表示

ラベル

背景色: 透明

ラベル

文字色: 黒

文字サイズ: 標準

文字の書体: 標準

文字の強調: 標準

A Pが受け取る項目

データ名	桁	データ型	桁属性	埋字
入金区分-	2	文字 (XX)		スペース

A Pが渡す項目

データ名	桁	データ型	桁属性	埋字
入金ラベル-	8	文字 (XX)		スペース
入金コード-	2	文字 (XX)		

動的変更 (A Pから表示属性を変更する)

OK キャンセル ヘルプ

### (3) 生成される論理マップ

マップ名が MAP116 の場合の例

```

01 MAP1160. (出力論理マップ)
:
03 MAP116-RADIO0001-0 OCCURS 3. ...ラジオボタンの通知コードとラベルを設定する論理テーブル
04 MAP116-入金コード-0 PIC X(2). ... 通知コード

04 MAP116-入金ラベル-0 PIC X(8). ... ラベル項目
01 MAP1161. (入力論理マップ)
:
03 MAP116-入金区分-H.
04 MAP116-入金区分-I PIC X(2). ... ラジオボタン入力のエリア

```

#### 注

AP が受け取る項目のデータ名を省略した場合は FIELDnnnn- が仮定されます。

ラジオボタンの通知コードのデータ名を省略した場合は、RADIO-CODEnnnn- が仮定されます。

ラベルのデータ名を省略した場合、RADIO-LABELnnnn- が仮定されます。

それぞれの場合、nnnn は「0001 ~ 9999」を示します。また、同一の属性を定義したボタンには、一つの論理テーブル名称が付けられます。

## 6. 画面定義とコーディングの関連

### (4) COBOL での AP

マップ名が MAP116 の場合の例

```
WORKING-STORAGE SECTION.
COPY MAP1160.      ... 出力論理マップの取り込み
COPY MAP1161.      ... 入力論理マップの取り込み
COPY X3MODTBL.     ... 動的変更テーブルの取り込み
COMMUNICATION SECTION.
CD DSP FOR I-O WS      ... 通信記述名:DSP
MAP NAME IS 画面マップ名      ... 物理マップ名称格納エリア
:
MOVE 'R1' TO MAP116-入金コード0(1).
MOVE N'現金' TO MAP116-入金ラベル0(1).
MOVE 'R2' TO MAP116-入金コード-0(2).
MOVE N'手形' TO MAP116-入金ラベル0(2).
MOVE 'R3' TO MAP116-入金コード-0(3).
MOVE N'銀行振込' TO MAP116-入金ラベル0(3).
MOVE 'MAP116ND' TO 画面マップ名.      ... 画面マップ名称を代入
TRANSCEIVE DSP FROM MAP1160 INTO MAP1161. ... 画面の送受信
:
EVALUATE MAP116-入金区分-I
WHEN 'R1'
PERFORM 現金処理
WHEN 'R2'
PERFORM 手形処理
WHEN 'R3'
PERFORM 銀行振込処理
END-EVALUATE.
```

... 出力論理マップへのボタンの通知コードとラベルの設定

... 画面マップ名称を代入

... 画面の送受信

... 選択されたボタンの通知コード判定

## 6.2.27 固定ラジオボタンを使用した候補の選択

### (1) 仕様

「6.2.26 可変ラジオボタンを使用した候補の選択」と同じ仕様で、固定ラジオボタンを利用する場合の説明をします。

固定ラジオボタンを使用する場合、ラジオボタンダイアログで指定したボタンのラベルを画面に表示し、選択されたラベルに対応する通知コードを入力論理マップに代入します。

### (2) 画面定義の概要

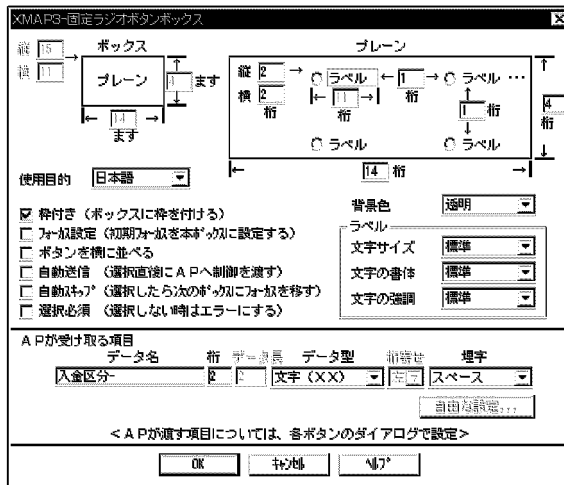
画面定義の概要を次に示します。

1. GUI 画面定義ウィンドウで、ツールボックスの [ 固定ラジオボタン ] を使用して、ラジオボタンを 3 個配置する。
2. 固定ラジオボタンダイアログで次の表の項目を定義する。

定義項目	定義内容	
使用目的	日本語	
枠付き（ボックスに枠を付ける）	指定する	
AP が受け取る項目	データ名	入金区分
	データ型	文字（XX）
	桁	2
	データ長	2（この定義内容は固定）

## 注

通知コードとして半角文字専用にする場合は、「英数」を選んでください。



## 3. 固定ラジオボタンダイアログで、次の表の項目を定義する。

定義項目	定義内容		
	[ 現金 ] ボタン	[ 手形 ] ボタン	[ 銀行振込 ] ボタン
ラベル	現金	手形	銀行振込
通知コード	R1	R2	R3

## 6. 画面定義とコーディングの関連

XMAPPG-固定ランオプション

○ラベル

表示方法: 標準表示      ラベル色: 黒

強制設定 (初期値を本機に設定する)       選択済みにする

不活性 (選択できない状態にする)

ラベル: 現金

通知コード: R01

動的変更 (A Pから表示属性を変更する)      制御項目が一致: R0100008

< A Pが受け取る項目はボタンボックスで指定済み >

OK      キャンセル      ヘルプ

### (3) 生成される論理マップ

マップ名が MAP117 の場合の例

```
01 MAP1170. (出力論理マップ)
:
01 MAP1171. (入力論理マップ)
:
03 MAP117-入金区分-H.
04 MAP117-入金区分-I PIC X(2). ... ラジ林' ン入力のエア
```

#### 注

AP が受け取る項目のデータ名を省略した場合、FIELDnnnn- (nnnn: 0001 ~ 9999) が仮定されます。



#### (4) COBOL での AP

マップ名が MAP117 の場合の例

```

WORKING-STORAGE SECTION.
  COPY MAP1170.    ... 出力論理マップの取り込み
  COPY MAP1171.    ... 入力論理マップの取り込み
  COPY X3MODTBL.   ... 動的変更テーブルの取り込み
COMMUNICATION SECTION.
  CD DSP FOR I-O WS           ... 通信記述名:DSP
  MAP NAME IS 画面マップ名    ... 物理マップ名称格納エリア
  :
  MOVE 'MAP117ND' TO 画面マップ名.
  TRANSCIEVE DSP FROM MAP1170 INTO MAP1171. ... 画面の送受信
  :
  EVALUATE MAP117-入金区分-I
    WHEN 'R1'
      PERFORM 現金処理
    WHEN 'R2'
      PERFORM 手形処理
    WHEN 'R3'
      PERFORM 銀行振込処理
  END-EVALUATE.

```

… 選択されたボタンの通知コード判定

### 6.2.28 可変チェックボタンを使用した候補の選択

#### (1) 仕様

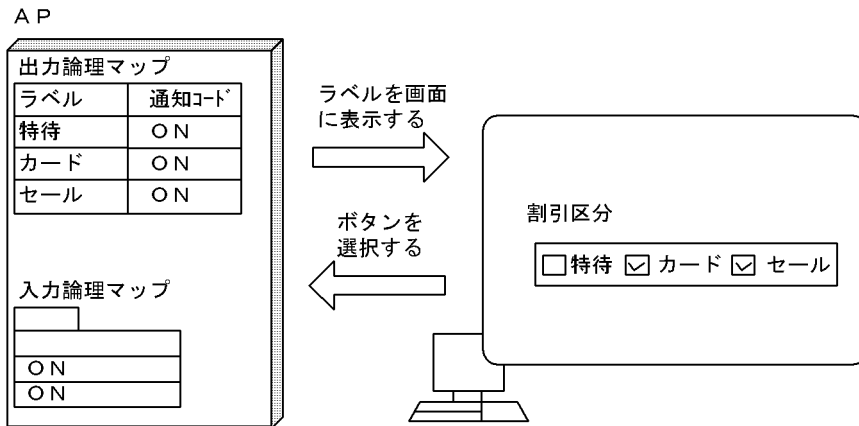
「割引区分」のチェックボタンボックスで [ 特待 ], [ カード ], または [ セール ] のチェックボタンを選択するとオン状態で表示します。選択されたボタンに対応した通知コードは、入力論理マップに代入されます。

可変チェックボタンは、AP でチェックボタンのラベル、およびボタンに対応する通知コードを出力論理マップに格納します。

AP 実行時での可変チェックボタンの表示を次の図に示します。

6. 画面定義とコーディングの関連

図 6-31 可変チェックボタンを使用したデータの選択



(2) 画面定義の概要

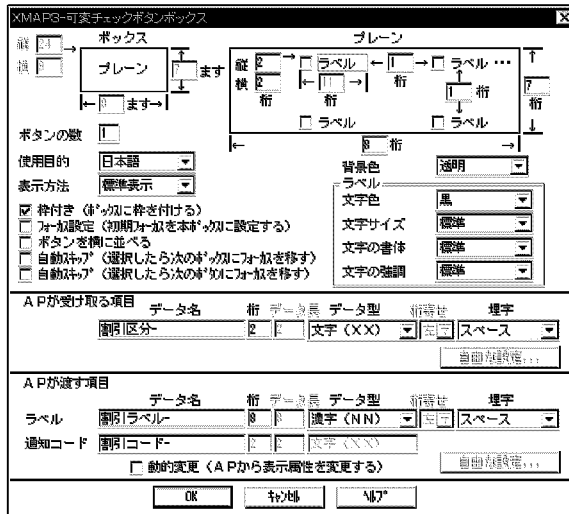
画面定義の概要を次に示します。

1. GUI 画面定義ウィンドウで、ツールボックスの [ 可変チェックボタン (複数) ] を使用して、チェックボタンを 3 個配置する。
2. 可変チェックボタンボックスダイアログで次の表の項目を定義する。

定義項目		定義内容	
使用目的		日本語	
桁		8	
AP が受け取る項目	データ名	割引区分 -	
	データ型	文字 (XX)	
	桁	2	
	データ長	2 (この定義内容は固定)	
AP が渡す項目	ラベル	データ名	割引ラベル -
		データ型	漢字 (NN)
		桁	8
		データ長	8 (この定義内容は固定)
	通知コード	データ名	割引コード -
		データ型	文字 (XX) (この定義内容は固定)
		桁	2 (この定義内容は固定)
		データ長	2 (この定義内容は固定)

注

通知コードとして半角文字専用にする場合は、「英数」を選んでください。



### (3) 生成される論理マップ

マップ名が MAP118 の場合の例

01	MAP1180. (出力論理マップ)		
:			
	03	MAP118-CHECK0001-0 OCCURS 3.	… チェックボタンのコード 定数とラベルを設定する論理テーブル
	04	MAP118-割引コード-0 PIC X(2).	… ボタンのコード 定数項目
	04	MAP118-割引ラベル-0 PIC N(4).	… ボタンのラベル項目
01	MAP1181. (入力論理マップ)		
:			
	03	MAP118-割引区分-H OCCURS 3.	
	04	MAP118-割引区分-I PIC X(2).	… チェックボタン入力エリア

#### 注

「AP が受け取る項目」のデータ名を省略した場合、FIELDnnnn- が仮定されます。  
 「AP が渡す項目」のラベルのデータ名を省略した場合、CHECK-LABELnnnn-、通知コードのデータ名を省略した場合、CHECK-CODEnnnn- が仮定されます  
 (nnnn : 0001 ~ 9999)。

また、同一の属性を定義したボタンには、一つの論理テーブル名が付けられます。

## 6. 画面定義とコーディングの関連

### (4) COBOL での AP

マップ名が MAP118 の場合の例

WORKING-STORAGE SECTION.	
COPY MAP1180.	… 出力論理マップの取り込み
COPY MAP1181.	… 入力論理マップの取り込み
COPY X3MODTBL.	… 動的変更テーブルの取り込み
COMMUNICATION SECTION.	
CD DSP FOR I-O WS	… 通信記述名: DSP
MAP NAME IS 画面マップ名	… 物理マップ名称格納エリア
:	
MOVE SPACE TO MAP118-割引区分-1(1).	} …チェックボタン入力の データエリアをクリア
MOVE SPACE TO MAP118-割引区分-1(2).	
MOVE SPACE TO MAP118-割引区分-1(3).	
MOVE 'ON' TO MAP118-割引コード-0(1).	} …出力論理マップへの ボタンのコード定数と ラベルの設定
MOVE '特待' TO MAP118-割引ラベル-0(1).	
MOVE 'ON' TO MAP118-割引コード-0(2).	
MOVE 'カード' TO MAP118-割引ラベル-0(2).	
MOVE 'ON' TO MAP118-割引コード-0(3).	} …物理マップ名称を代入 …画面の送受信
MOVE 'セール' TO MAP118-割引ラベル-0(3).	
MOVE 'MAP118ND' TO 画面マップ名.	
TRANSCEIVE DSP FROM MAP1180 INTO MAP1181.	
:	
IF MAP118-割引区分-1(1) = 'ON'	} …選択されたボタンの 条件判定とその処理
THEN	
PERFORM 特待処理	
END-IF.	
IF MAP118-割引区分-1(2) = 'ON'	
THEN	
PERFORM カード処理	
END-IF.	
IF MAP118-割引区分-1(3) = 'ON'	
THEN	
PERFORM セール処理	
END-IF.	

## 6.2.29 固定チェックボタンを使用した候補の選択

### (1) 仕様

「6.2.28 可変チェックボタンを使用した候補の選択」と同様の仕様で、固定チェックボタンを使用した場合の例を説明します。

固定チェックボタンは、あらかじめチェックボタンダイアログで選択したボタンのラベルを画面に表示し、選択されたラベルに対応する通知コードを入力論理マップに代入します。

## (2) 画面定義の概要

画面定義の概要を次に示します。

1. GUI 画面定義ウィンドウで、ツールボックスの [ 固定チェックボタン (複数) ] を使用して、チェックボタンを 3 個配置する。
2. 固定チェックボタンの属性を固定チェックボタンダイアログで定義する。次の表の項目を定義する。

定義項目		定義内容		
		[ 特待 ] ボタン	[ カード ] ボタン	[ セール ] ボタン
使用目的		日本語	日本語	日本語
ラベル		特待	カード	セール
通知コード		ON	ON	ON
AP が受け取る項目	データ名	特待 -	カード -	セール -
	データ型	文字 (XX)	文字 (XX)	文字 (XX)
	桁	2	2	2
	データ長 (この定義内容は固定)	2	2	2

注

通知コードとして半角文字専用にする場合は、「英数」を選んでください。

## 6. 画面定義とコーディングの関連

### (3) 生成される論理マップ

マップ名が MAP119 の場合の例

01 MAP1190. (出力論理マップ)	
:	
01 MAP1191. (入力論理マップ)	
:	
03 MAP119-特待-H.	} … チェックボタンの格納領域
04 MAP119-特待-I PIC X(2).	
03 MAP119-カード-H.	
04 MAP119-カード-I PIC X(2).	
03 MAP119-セール-H.	
04 MAP119-セール-I PIC X(2).	

注

AP が受け取る項目のデータ名を省略した場合、FIELDnnnn- (nnnn : 0001 ~ 9999) が仮定されます。

### (4) COBOL での AP

マップ名称が MAP119 の場合

WORKING-STORAGE SECTION.	
COPY MAP1190.     … 出力論理マップの取り込み	
COPY MAP1191.     … 入力論理マップの取り込み	
COPY X3MODTBL.    … 動的変更テーブルの取り込み	
COMMUNICATION SECTION.	
CD DSP FOR I-O WS                   … 通信記述名: DSP	
MAP NAME IS 画面マップ名         … 物理マップ名称格納エリア	
:	
MOVE 'MAP119ND' TO 画面マップ名.     … 物理マップ名を代入	
TRANSCIVE DSP FROM MAP1190 INTO MAP1191. … 画面の送受信	
:	
IF MAP119-特待-I = 'ON'	} … 選択されたボタンの 条件判定とその処理
THEN	
PERFORM 特待処理	
END-IF.	
IF MAP119-カード-I = 'ON'	
THEN	
PERFORM カード処理	
END-IF.	
IF MAP119-セール-I = 'ON'	
THEN	
PERFORM セール処理	
END-IF.	

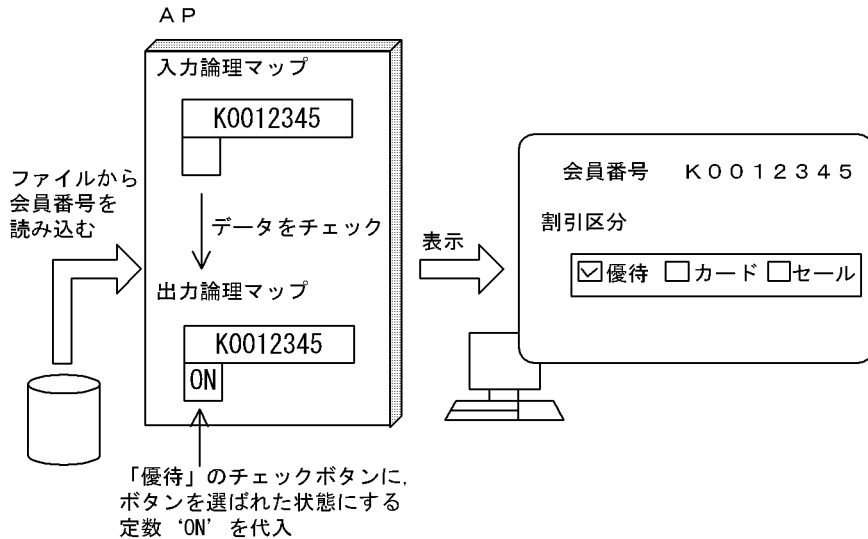
### 6.2.30 チェックボタンの選択状態の動的変更

ここでは固定チェックボタンを使用して説明しますが、可変チェックボタンでも同じように変更できます。

#### (1) 仕様

ファイルから読み込んだ「会員番号」を判断し、優待会員であれば「割引区分」のチェックボタンボックスの [ 優待 ] が選択された状態で画面を表示します。AP 実行時に論理マップに代入される値を次の図に示します。

図 6-32 チェックボタンの選択状態の動的変更



#### (2) 画面定義の概要

画面定義の概要を次に示します。

1. GUI 画面定義ウィンドウでツールボックスの [ 固定チェックボタン (複数) ] を使用して固定チェックボタンを 3 個配置する。
2. 固定チェックボタンダイアログで次の表の項目を定義する。

定義項目	定義内容		
	[ 優待 ] ボタン	[ カード ] ボタン	[ セール ] ボタン
使用目的	日本語	日本語	日本語
ラベル	優待	カード	セール
通知コード	ON	ON	ON
データ名	優待 -	カード -	セール -

## 6. 画面定義とコーディングの関連

定義項目	定義内容		
	[ 優待 ] ボタン	[ カード ] ボタン	[ セール ] ボタン
桁	2	2	2
データ長 (この定義内容は固定)	2	2	2

### 注

このほか、「動的変更 (AP から属性を変更する)」のチェックボタンをオン状態にして指定します。

### 注

通知コードとして半角文字専用にする場合は、「英数」を選んでください。会員番号データを読み込み、優待会員かどうかをチェックするための領域を定義します。

3. GUI 画面定義ウィンドウでツールボックスの [ 入出力テキスト (英数) ] を使用して英数の入出力テキストボックスを配置する。

4. 入出力テキストボックスダイアログで次の表の項目を定義する。

定義項目	定義内容	
AP が受け取る項目	データ名	会員番号 -
	データ型	文字 (XX) (この定義内容は固定)
	桁	8
	データ長	8 (この定義内容は固定)
AP が渡す項目	データ名	会員番号 - (この定義内容は固定)
	データ型	文字 (XX) (この定義内容は固定)
	桁	8 (この定義内容は固定)
	データ長	8 (この定義内容は固定)



### (3) 生成される論理マップ

マップ名が MAP412 の場合の例

01 MAP4120. (出力論理マップ)	
:	
03 MAP412-優待-A PIC X(2).	} ... ボタンの動的変更情報格納エリア
03 MAP412-カード-A PIC X(2).	
03 MAP412-セール-A PIC X(2).	
03 MAP412-会員番号-0 PIC X(8).	
01 MAP4121. (入力論理マップ)	
:	
03 MAP412-優待-H.	} ... 選択したボタンの 通知コードを代入するエリア
04 MAP412-優待-I PIC X(2).	
03 MAP412-カード-H.	
04 MAP412-カード-I PIC X(2).	
03 MAP412-セール-H.	
04 MAP412-セール-I PIC X(2).	
03 MAP412-会員番号-H.	} ... 「会員番号」の入力エリア
04 MAP412-会員番号-I PIC X(8).	

#### 注 1

入出力テキストボックスのデータ名を省略した場合、FIELDnnnnr (nnnn : 0001 ~ 9999) が仮定されます。

#### 注 2

AP 実行時にボタンの属性を変更する場合、マップ生成後に生成される動的変更情報格納エリア (制御項目) に属性情報を代入します。属性情報は、XMAP3 が動的変更テーブルとして用意しています。AP 実行時は、このテーブルの修飾名を動的変更情報格納エリアに代入します。

## 6. 画面定義とコーディングの関連

### (4) COBOL での AP

マップ名が MAP412 の場合の例

```
WORKING-STORAGE SECTION.  
  COPY MAP4120.          ... 出力論理マップの取り込み  
  COPY MAP4121.          ... 入力論理マップの取り込み  
  COPY X3MODTBL.         ... 動的変更テーブルの取り込み  
COMMUNICATION SECTION.  
CD DSP FOR I-O WS      ... 通信記述名: DSP  
MAP NAME IS 画面マップ名.  ... 物理マップ名格納エリア  
PROCEDURE DIVISION.  
* ファイルからデータを読み込む  
IF MAP412-会員番号-I = 'K0012345'  
  THEN                  ... 優待会員の場合  
  MOVE XMAP-BUTTON-SEL2 TO MAP412-優待-A.  ... ボタン属性を選択状態  
  ELSE                  ... にする修飾名を代入  
  CONTINUE  
END-IF.  
MOVE MAP412-会員番号-I TO MAP412-会員番号-O.  ... 出力エリアヘデータを代入  
* 画面の送受信  
MOVE 'MAP412ND' TO 画面マップ名.  
TRANSCIEVE DSP FROM MAP4120 INTO MAP4121.
```

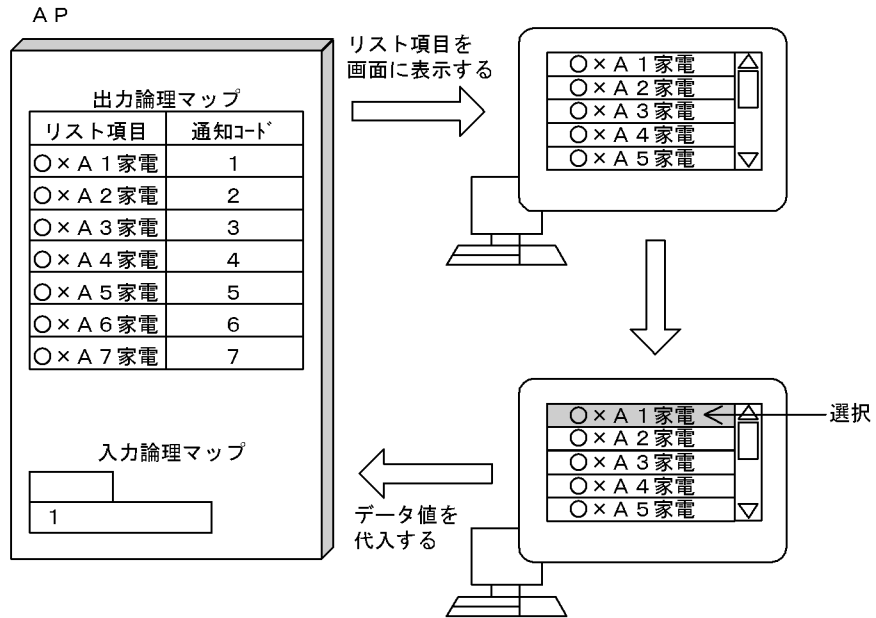
## 6.2.31 リストボックスを使用した項目の選択

### (1) 仕様

店名のリスト項目を画面に表示します。リスト中からデータを一つだけ選択すると反転表示し、データに対応するコード定数が入力論理マップに代入されます。APで出力論理マップにリスト項目と、それに対応する通知コードを代入します。送信要求をすると、リスト項目がリストボックス上に表示されます。リスト項目を選択したあとで送信（実行）キーを押すと、XMAP3が入力論理マップにリスト項目に対応した通知コードを格納します。

単一選択リストボックスを使用した、AP実行時でのリストボックスの表示を次の図に示します。

図 6-33 単一選択リストボックスの表示



## (2) 画面定義の概要

画面定義の概要を次に示します。

1. GUI 画面定義ウィンドウで、ツールボックスの [ リストボックス ( 単一選択 ) ] を使用してリストボックスを配置する。
2. 単一選択リストボックスダイアログで次の表の項目を定義する。

定義項目		定義内容
使用目的		日本語
AP が受け取る項目	データ名	選択リスト -
	データ型	文字 ( XX )
	桁	2
	データ長	2 ( この定義内容は固定 )
ラベル	データ名	リストラベル -
	データ型	漢字 ( NN )
	桁	12
	データ長	12 ( この定義内容は固定 )
通知コード	データ名	リストコード -
	データ型	文字 ( XX ) ( この定義内容は固定 )
	桁	2 ( この定義内容は固定 )

## 6. 画面定義とコーディングの関連

定義項目	定義内容
データ長	2 (この定義内容は固定)
リストの項目数 (ブレーンの行)	7

### 注

通知コードとして半角文字専用にする場合は、「英数」を選んでください。

### (3) 生成される論理マップ

マップ名が MAP121 の場合の例

```

01 MAP1210. (出力論理マップ)
:
03 MAP121-LIST0001-0 OCCURS 7. ... 論理テーブル名
04 MAP121-リストラベル-0 PIC N(6). ... リスト項目のラベル
04 MAP121-リストコード-0 PIC X(2). ... リスト項目の通知コード
01 MAP1211. (入力論理マップ):
03 MAP121-選択リスト-H.
04 MAP121-選択リスト-I PIC X(2). ... リスト項目入力のデータエリア
    
```

### 注

「AP が受け取る項目」のデータ名を省略した場合、FIELDnnnn- が仮定されます。「ラベル」のデータ名を省略した場合、LIST-LABELnnnn-、通知コードのデータ名を省略した場合、LIST-CODEnnnn- が仮定されます (nnnn: 0001 ~ 9999)。また、単一選択リストボックスダイアログで「LISTnnnn-」の論理項目が付けられません。

## (4) COBOL での AP

マップ名称が MAP121 の場合

```

WORKING-STORAGE SECTION.
COPY MAP1210.    ... 出力論理マップの取り込み
COPY MAP1211.    ... 入力論理マップの取り込み
COPY X3MODTBL.   ... 動的変更テーブルの取り込み
77 A1 PIC N(6) VALUE N'○×A1家電'.
77 A2 PIC N(6) VALUE N'○×A2家電'.
:
77 A7 PIC N(6) VALUE N'○×A7家電'.
COMMUNICATION SECTION.
CD DSP FOR I-O WS           ... 通信記述名:DSP
MAP NAME IS 画面マップ名   ... 物理マップ名称格納エリア
* データ値'1'~'7'をリストコード(1)~(7)に設定する。ラベルA1~A7をリストラベル(1)~(7)に設定する
MOVE '1' TO MAP121-リストコード-0(1).
MOVE A1 TO MAP121-リストラベル-0(1).
MOVE '2' TO MAP121-リストコード-0(2).
MOVE A2 TO MAP121-リストラベル-0(2).
:
MOVE '7' TO MAP121-リストコード-0(7).
MOVE A7 TO MAP121-リストラベル-0(7).
TRANSCIEVE DSP FROM MAP1210 INTO MAP1211.    ... 画面の送受信
* 選択されたリストに対応するデータ値を参照して各処理に分岐する
EVALUATE MAP121-選択リスト-1
  WHEN '1'
    PERFORM A1家電の処理
  WHEN '2'
    PERFORM A2家電の処理
:
  WHEN '7'
    PERFORM A7家電の処理
END-EVALUATE.

```

… 出力論理マップへのリスト項目通知コードとラベルの設定

… 選択されたリスト項目の判定とその処理

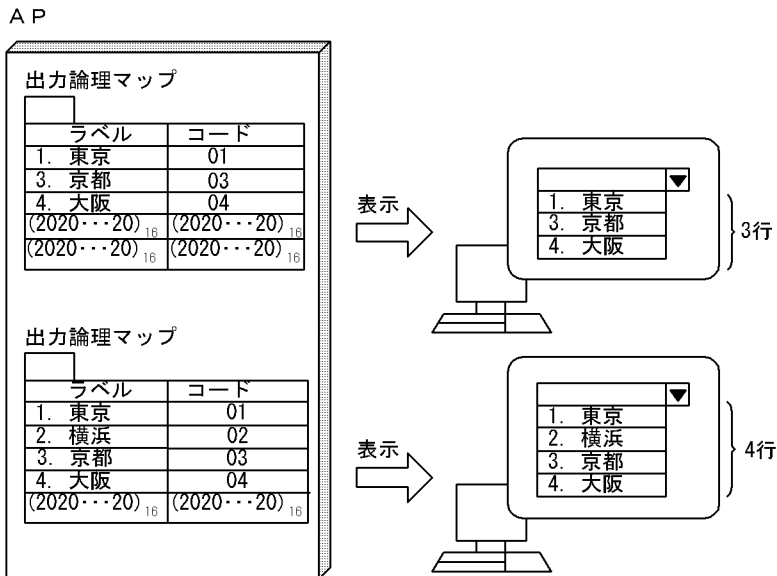
### 6.2.32 コンボボックスの項目数の動的変更

#### (1) 仕様

可変コンボボックスで表示するメニュー項目数を、AP から変更します。メニュー項目を上から順に表示し、後半のデータのないメニュー項目には空白（スペースデータ）を代入しておきます。空白を代入している後半のメニュー項目は表示されません。なお、途中のメニュー項目に代入された空白はデータとみなされるので、これらのメニュー項目はデータが何も入っていない状態で表示されます。コンボボックスのメニュー項目数の動的変更を次の図に示します。

6. 画面定義とコーディングの関連

図 6-34 コンボボックスの項目数の動的変更



(2) 画面定義の概要

画面定義の概要を次に示します。

1. GUI 画面定義ウィンドウでツールボックスの [ 可変コンボボックス ] を使用して、コンボボックスを配置する。
2. 可変コンボボックスのコンボボックスダイアログで、次の表の項目を定義する。

定義項目		定義内容
使用目的		日本語
桁		22
AP が受け取る通知コードの項目	データ名	地区
	桁	4
	データ長	4 (この定義内容は固定)

注

通知コードとして半角文字専用にする場合は、「英数」を選んでください。

XMAP3-コンボボックス

縦 18  
横 15

← 2.6 ます 2.2 桁 →

背景色 白  
文字色 黒  
文字サイズ 標準  
文字の書体 標準  
文字の強調 標準

使用目的 日本語  
表示方法 選択可能-標準表示  
遷移条件 自動スキップ  
枠高 2.6 ます

反転表示する  
 フォーカス設定 (初期フォーカスを本ボックスに設定する)  
 選択済み (選択しなくても通知コードを A P に返す)  
 自動送信 (選択直後に A P へ制御を渡す)

コマの記述

A P が受け取る通知コードの項目

データ名	桁	データ長	データ型	桁寄せ	埋字
地区-	4	4	文字 (XX)	左	スペース

動的変更 (A P から表示属性を変更する)  
 A P から初期値を変更する

初期値(通知コード\*) 指定しない

3. 可変コンボボックスの可変コンボメニューダイアログで、次の表の項目を定義する。

定義項目		定義内容
ボックスサイズ (ラベルの行数)		10 行
AP が渡す項目	メニューの項目数	20
ラベル	データ名	地区ラベル -
	データ型	文字 (XX)
AP に返す通知コード	データ名	地区コード -
	桁	4 (この定義内容は固定)
	データ長	4 (この定義内容は固定)

XMAP3-可変コンボメニュー

ラベル1  
...  
ラベルn

↑  
10 行  
↓

← 2.2 桁 →

文字色 黒

A P が渡す項目  
メニューの項目数 20

ラベル	データ名	桁	データ長	データ型	桁寄せ	埋字
地区ラベル-	地区-	22	22	文字 (XX)	左	スペース

AP に返す通知コード\* 地区コード- 4 4 文字 (XX)

## 6. 画面定義とコーディングの関連

### (3) 生成される論理マップ

マップ名が MAP001 の場合の例

```
01 MAP0010. (出力論理マップ)
:
02 MAP001G.
:
03 MAP001-POPUP0001-0 OCCURS 20. . . . 論理テーブル名称
04 MAP001-地区コード-0 . . . 通知コード項目
PIC X(4).
04 MAP001-地区ラベル-0 . . . ラベル項目
PIC X(22).
03 MAP001-地区-0 PIC X(4). . . . 出力時の初期選択エリア
*
01 MAP0011. (入力論理マップ)
:
02 MAP001K.
:
03 MAP001-地区-H. . . . 通知コードの入力エリア
04 MAP001-地区-I PIC X(4).
```

#### 注

「AP が受け取る通知コードの項目」のデータ名を省略した場合、FIELDnnnn- が仮定されます。「AP が渡す項目」のデータ名を省略した場合、ラベルでは POPUP-LABELnnnn- が、通知コードでは POPUP-CODEnnnn- がそれぞれ仮定されます (nnnn : 0001 ~ 9999)。

また、可変コンボボックスに対して一つの論理テーブル名が付けられます。



## (4) COBOL での AP

```

WORKING-STORAGE SECTION.
  COPY  MAP0010.      . . . 出力論理マップの取り込み
  COPY  MAP0011.      . . . 入力論理マップの取り込み
*
  COPY  X3MODTBL.     . . . 動的変更テーブルの取り込み
  :
COMMUNICATION SECTION.
  CD    DSP FOR 1-0 WS
  MAP  NAME           IS 画面マップ名 . . . 端末名格納エリア
  SYMBOLIC TERMINAL IS 画面端末名 . . . 物理マップ名格納エリア
  :
*
PROCEDURE DIVISION.
  :
*  端末名の設定
  MOVE 'DSP001' TO 画面端末名.

***** 地区データの設定 *****
  PERFORM 地区データ設定処理.

  :

***** コンボボックスのラベル数（表示サイズ）の動的変更 *****
*** コンボボックスの表示サイズを、動的に変更します ***
* コンボボックスの後半のラベルを表示しない部分に *
* スペースデータを設定すると → 表示サイズを圧縮します *
*****
* 本例では、スペースデータを設定して表示サイズを圧縮させます *
* 定義時のサイズ：20行 → 画面表示時のサイズ：17行 *
  MOVE ALL SPACES TO MAP001-POPUP0001-0(18).
  MOVE ALL SPACES TO MAP001-POPUP0001-0(19).
  MOVE ALL SPACES TO MAP001-POPUP0001-0(20).
*****

* マップ名の設定
  MOVE 'MAP001ND' TO 画面マップ名.

* 画面の表示と入力
  TRANSCEIVE DSP FROM MAP0010 INTO MAP0011.

  :
  :

```

## 6.2.33 プッシュボタンを使用した処理の選択

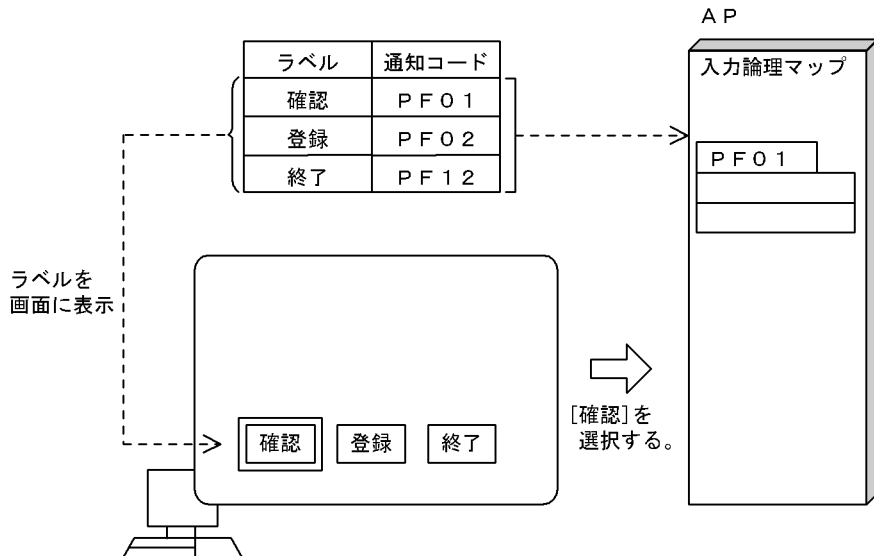
## (1) 仕様 1

[確認],[登録],[終了]のプッシュボタンを選択すると、それぞれのプッシュボタンに対応するイベント通知コードを入力論理マップに代入します。APでは選択されたボタンに対応するイベント通知コードを判定して、各処理へ分岐します。

## 6. 画面定義とコーディングの関連

AP 実行時でのプッシュボタンの表示、および入力論理マップに代入される値を次の図に示します。

図 6-35 プッシュボタンを使用した処理の選択



### (a) 画面定義の概要

画面定義の概要を次に示します。

1. GUI画面定義ウィンドウで、ツールボックスの[プッシュボタン(複数)]を使用してボタンを3個配置する。
2. 各プッシュボタンの属性をプッシュボタンダイアログで定義する。次の表の項目を定義する。

定義項目		定義内容		
		[確認]ボタン	[登録]ボタン	[終了]ボタン
ラベル	テキスト	確認	登録	終了
APへ通知するコード		PF01	PF02	PF12

#### 注

これらは標準の仕様です。APへ通知するコードを変更する場合は、ドロースettingsアップで変更してください。

## (b) 生成される論理マップ

マップ名が MAP125 の場合の例

```

01 MAP1250. (出力論理マップ)
:
01 MAP1251. (入力論理マップ)
:
03 MAP125-INCI PIC X(4). ... イベント通知コードの格納領域(イベント通知項目)

```

## (c) COBOL での AP

マップ名が MAP125 の場合の例

```

WORKING-STORAGE SECTION.
* 論理マップのCOPY展開
COPY MAP1250. .... 出力論理マップ
COPY MAP1251. .... 入力論理マップ
TRANSCEIVE DSP FROM MAP1250 INTO MAP1251.....画面の表示と入力
EVALUATE MAP125-INCI ..... プッシュボタンのチェック
                                と該当処理の呼び出し

                                WHEN 'PF01'
                                    PERFORM 確認処理
                                WHEN 'PF02'
                                    PERFORM 登録処理

```

## (2) 仕様2 (ボタンにフォーカスが遷移しないようにする)

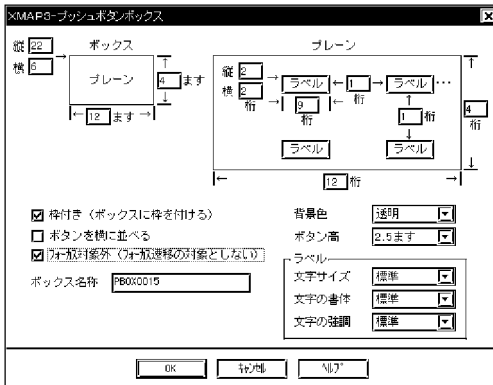
誤入力などを防ぐために、ボタンにフォーカスが遷移しないように設定できます。プッシュボタンボックスダイアログの「フォーカス対象外 (フォーカス遷移の対象としない)」を指定すると、そのボックスへはフォーカスが遷移しなくなります。

## 6. 画面定義とコーディングの関連

### (a) 画面定義の概要

画面定義の概要を次に示します。

1. GUI 画面定義ウィンドウに配置されているボタンボックスのうち、フォーカスを遷移させないボタンボックスを選択する。
2. プッシュボタンボックスダイアログで、「フォーカス対象外（フォーカス遷移の対象としない）」を指定する。



## (b) 生成される論理マップ

マップ名が MAP001 の場合の例

```

《出力論理マップ(MAP0010)》
01 MAP0010.
  02 MAP001L      PIC S9(4)  COMP-5
                    VALUE +24.
  02 MAP001Z      PIC S9(4)  COMP
                    VALUE +0.
  02 MAP001G.
    03 MAP001-CNTRLO      PIC X(2).
    03 MAP001-OUTCURS-LOGO  PIC X(2).
    03 MAP001-BUTTON0001-A  PIC X(2).      ……ボタンの動的変更情報格納エリア
*
01 MAP001T      PIC S9(4)  COMP-5
                    VALUE +24.
01 MAP001D.
  02 MAP001-INCT.
    03 FILLER      PIC X(2)  VALUE X'0004'.
  02 MAP001-PBOX0015T.
    03 FILLER      PIC X(2)  VALUE X'8001'.

《入力論理マップ(MAP0011)》
01 MAP0011.
  02 MAP001S      PIC S9(4)  COMP-5
                    VALUE +22.
  02 MAP0010      PIC S9(4)  COMP
                    VALUE +0.
  02 MAP001K.
    03 MAP001-INCI  PIC X(4).      ……………イベント通知項目
    03 MAP001-INCURS-LOGI  PIC X(2).

```

## 6. 画面定義とコーディングの関連

### (c) COBOL での AP

```
WORKING-STORAGE SECTION.
* 論理マップの展開
COPY MAP0010.
COPY MAP0011.
* 動的変更テーブルの展開
COPY X3MODTBL.
COMMUNICATION SECTION.
CD DSP FOR I-O WS
MAP NAME IS 画面マップ名
      :
PROCEDURE DIVISION.
* 画面表示処理
  MOVE 'DSP001' TO 画面端末名.
  MOVE ALL 'X' 1F' TO MAP001G.
  MOVE 'MAP001ND' TO 画面マップ名.
  TRANSCEIVE DSP FROM MAP0010 INTO MAP0011..... 画面の表示と入力
  EVALUATE MAP001-INC1..... プッシュボタンのチェックと該当処理の呼び出し
    WHEN 'PF01'
      PERFORM 確認処理
    WHEN 'PF02'
      PERFORM 登録処理
    WHEN 'PF12'
      PERFORM 終了処理
    WHEN OTHER
      PERFORM エラー処理
  END-EVALUATE.
```

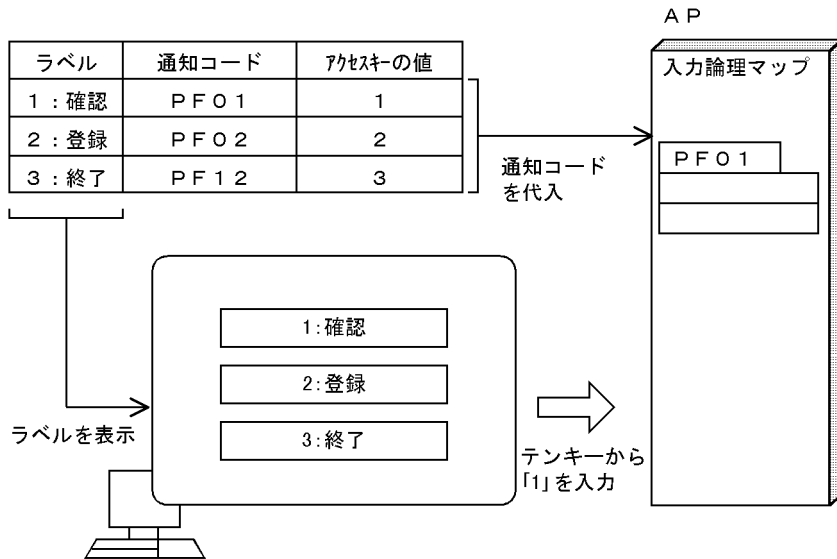
### (3) 仕様 3 (キーでプッシュボタンを選択できるようにする)

プッシュボタンダイアログの「アクセスキーを使用する」を指定し、アクセスキーでプッシュボタンを選択できるようにします。アクセスキーに数字(0~9)を指定すると、テンキー上の数字(0~9)でプッシュボタンを操作できるようになります。

[確認] ボタン, [登録] ボタンおよび [終了] ボタンを選んで、それぞれのプッシュボタンに対応するイベント通知コードを入力論理マップに格納します。APでは、選択されたボタンに対応するイベント通知コードを判定して、各処理に分岐させます。

AP実行時でのプッシュボタンの表示、および入力論理マップに代入される値を次の図に示します。

図 6-36 テンキー選択のプッシュボタンを使用した処理の選択



## (a) 画面定義の概要

画面定義の概要を次に示します。

1. GUI 画面定義ウィンドウに配置されているボタンボックスのうち、テンキーで選択するボタンボックスを選択する。
2. プッシュボタンダイアログで、「アクセスキーを使用する」および「アクセスキー」を指定する。次の表の項目を指定する。

定義項目		定義内容		
		[ 確認 ] ボタン	[ 登録 ] ボタン	[ 終了 ] ボタン
ラベル	テキスト	確認	登録	終了
アクセスキーを使用する		指定する	指定する	指定する
アクセスキー		1	2	3
通知コード		PF01	PF02	PF12

## 注

これらは標準の仕様です。通知コード（イベント通知コード）を変更する場合は、ドローストアップで変更してください。

6. 画面定義とコーディングの関連

(b) 生成される論理マップ

マップ名が MAP001 の場合の例

```

《出力論理マップ(MAP0010)》
01 MAP0010.
  02 MAP001L      PIC S9(4)  COMP-5
                   VALUE +24.
  02 MAP001Z      PIC S9(4)  COMP
                   VALUE +0.
  02 MAP001G.
    03 MAP001-CNTRL0      PIC X(2).
    03 MAP001-OUTCURS-LOC0 PIC X(2).
*
01 MAP001T      PIC S9(4)  COMP-5
                   VALUE +24.
01 MAP001D.
  02 MAP001-INCT.
    03 FILLER      PIC X(2)  VALUE X'0004'.
  02 MAP001-PBOX0001T.
    03 FILLER      PIC X(2)  VALUE X'8001'.

《入力論理マップ(MAP0011)》
01 MAP0011.
  02 MAP001S      PIC S9(4)  COMP-5
                   VALUE +22.
  02 MAP0010      PIC S9(4)  COMP
                   VALUE +0.
  02 MAP001K.
    03 MAP001-INCI PIC X(4).      .....イベント通知項目
    03 MAP001-INCURS-LOC1 PIC X(2).
    
```



## (c) COBOL での AP

```

WORKING-STORAGE SECTION.
* 論理マップの展開
COPY MAP0010.
COPY MAP0011.
* 動的変更テーブルの展開
COPY X3MODTBL.
COMMUNICATION SECTION.
CD DSP FOR I-O WS
MAP NAME IS 画面マップ名
      :
PROCEDURE DIVISION.
* 画面表示処理
MOVE 'DSP001' TO 画面端末名.
MOVE ALL X'1F' TO MAP001G.
MOVE 'MAP001ND' TO 画面マップ名.
TRANSCIEVE DSP FROM MAP0010 INTO MAP0011..... 画面の表示と入力
EVALUATE MAP001-INCI..... プッシュボタンのチェックと該当処理の呼び出し
  WHEN 'PF01'
    PERFORM 確認処理
  WHEN 'PF02'
    PERFORM 登録処理
  WHEN 'PF12'
    PERFORM 終了処理
  WHEN OTHER
    PERFORM エラー処理
END-EVALUATE.

```

## 6.2.34 プッシュボタンの表示の動的変更

## (1) 仕様 1

「顧客コード」の入出力テキストボックスにデータが入力された場合、プッシュボタンボックスにフォーカスが位置づけられると、[ OK ] のプッシュボタンを活性の状態で表示します。

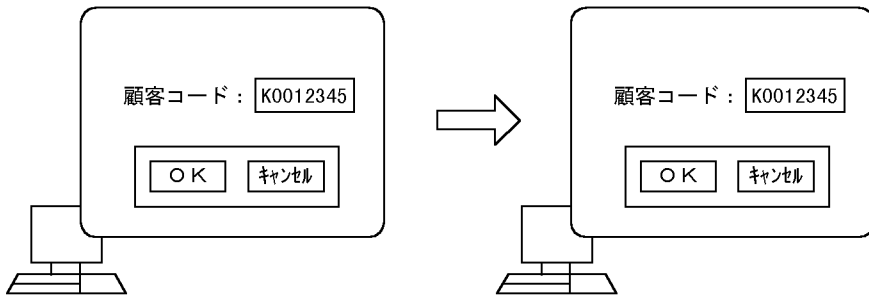
データの入力がない場合、プッシュボタンボックスにフォーカスが位置づくとき、[ OK ] のプッシュボタンを不活性の状態で表示します。

プッシュボタンの表示の動的変更を次の図に示します。

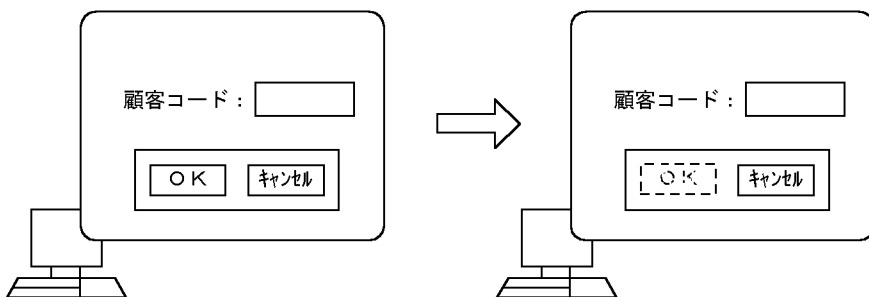
## 6. 画面定義とコーディングの関連

図 6-37 プッシュボタンを不活性状態に表示する

- データの入力がある場合



- データの入力がない場合



### (a) 画面定義の概要

画面定義の概要を次に示します。

#### 入出力テキストの定義

1. GUI 画面定義ウィンドウで、ツールボックスの [ 入出力テキスト ( 英数 ) ] を使用して顧客コードの入出力テキストボックスを配置する。
2. 入出力テキストボックスの属性を入出力テキストダイアログで定義する。次の表の項目を定義する。

定義項目		定義内容
使用目的		英数
自動送信 ( 入力直後に AP へ制御を渡す )		指定する
AP が受け取る項目	データ名	顧客コード -
	データ型	文字 ( XX ) ( この定義内容は固定 )
	桁	8
	データ長	8 ( この定義内容は固定 )
AP が渡す項目	データ名	顧客コード -

定義項目	定義内容
データ型	文字 (XX) (この定義内容は固定)
桁	8 (この定義内容は固定)
データ長	8 (この定義内容は固定)

### プッシュボタンの定義

1. GUI 画面定義ウィンドウで、ツールボックスの [ プッシュボタン (複数) ] を使用してボタン 2 個を配置する。
2. プッシュボタンの属性をプッシュボタンダイアログで定義する。次の表の項目を定義する。

定義項目		定義内容	
		[ OK ] ボタン	[ キャンセル ] ボタン
ラベル	テキスト	OK	キャンセル
AP へ通知するコード		A001	A002
動的変更 (AP から表示属性を変更する)		指定する	指定しない
データ名		BUTTON0001-	指定しない

### 注

AP へ通知するコードとして A001 や A002 を定義した場合、[ Ctrl ] + [ F1 ]、[ Ctrl ] + [ F2 ] などを押しても同じ通知コードが AP に返ります。

プッシュボタン専用の通知コードを AP に返したい場合は、A061 ~ A071 を定義してください。

## 6. 画面定義とコーディングの関連

### (b) 生成される論理マップ

マップ名が MAP127 の場合の例

01	MAP1270.	(出力論理マップ)	
:			
03	MAP127-顧客コード-0	PIC X(8).	… 出力テキストを代入するエリア
03	MAP127-BUTTON001-A	PIC X(2).	… ボタン用動的変更情報格納エリア
01	MAP1271.	(入力論理マップ)	
:			
03	MAP127-顧客コード-H.		
04	MAP127-顧客コード-1	PIC X(8).	… 入力テキストを代入するエリア

## (c) COBOL での AP

マップ名称が MAP127 の場合

```

WORKING-STORAGE SECTION.
  COPY MAP1270.          ... 出力論理マップの取り込み
  COPY MAP1271.          ... 入力論理マップの取り込み
  COPY X3MODTBL.         ... 動的変更テーブル取り込み
* XMAP-BUTTON-SEL1 PIC X(2) VALUE "PT". ... ボタンを不活性にする修飾名
* XMAP-NODATA PIC X(1) VALUE X"1F". ... データ有無コード
COMMUNICATION SECTION.
CD DSP FOR I=0 WS          ... 通信記述名:DSP
  MAP NAME IS 画面マップ名 ... 物理マップ名称格納エリア
:
MOVE 'MAP127ND' TO 画面マップ名. ... 物理マップ名を代入

TRANSCEIVE DSP FROM MAP1270 INTO MAP1271. ... 画面の送受信
IF MAP127-顧客コード=I = SPACE
  THEN
    MOVE XMAP-BUTTON-SEL1 TO MAP127-BUTTON0001-A ...入力データがない場合
    ...ボタンを不活性にする修飾名を格納エリアに代入
  ELSE
    MOVE XMAP-NODATA TO MAP127-BUTTON0001-A ...入力データがある場合
    ...動的変更情報格納エリアのクリア
END-IF.
MOVE 'MAP127ND' TO 画面マップ名. ... 物理マップ名を代入
TRANSCEIVE DSP FROM MAP1270 INTO MAP1271. ... 画面の送受信

```

## (2) 仕様 2

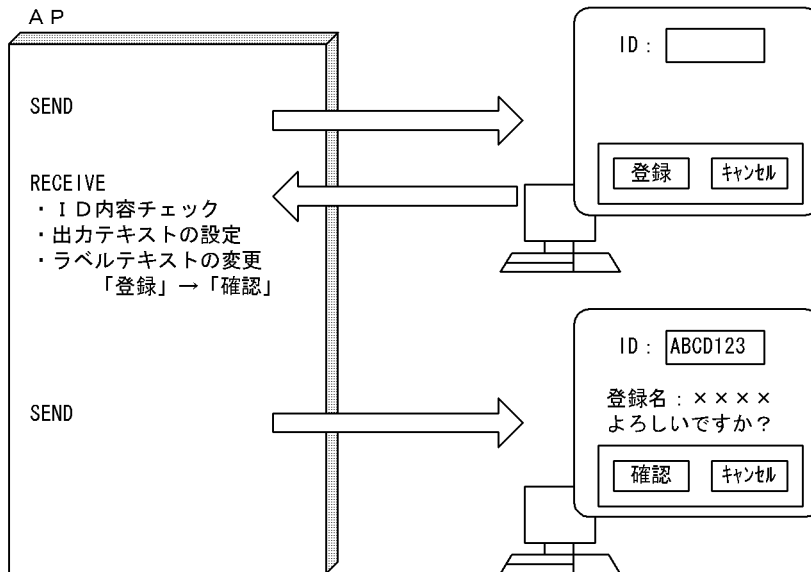
プッシュボタンのテキストラベルを変更します。

1. 「ID」の入出力テキストボックスにデータを入力し,[登録]ボタン選ぶ。
2. APで内容をチェックし,出力テキストに「登録名」と「メッセージ」を設定する。
3. プッシュボタンのラベルテキストを[登録]から[確認]に変更する。

プッシュボタンの表示の動的変更を次の図に示します。

6. 画面定義とコーディングの関連

図 6-38 プッシュボタンのラベルテキストを変更する



(a) 画面定義の概要

画面定義の概要を次に示します。ここでは、プッシュボタンの定義についてだけ説明します。

1. GUI画面定義ウィンドウで、ツールボックスの[プッシュボタン(複数)]を使用してボタン2個を配置する。
2. プッシュボタンの属性をプッシュボタンダイアログで定義する。次の表の項目を定義する。

定義項目		定義内容	
		[登録]ボタン	[キャンセル]ボタン
APへ通知するコード		A061	A062
テキスト	テキスト	登録	キャンセル
	APからテキストを変更する	指定する	指定しない
	データ名	BUTTON0002-	-
	桁	9	-

(凡例)

- : 該当しない。

注

APへ通知するコードとしてA061やA062を定義した場合、プッシュボタン専用となります。[Ctrl]+[F1],[Ctrl]+[F2]などを押した場合と同じ通知コードをAPに返したい場合は、A001～A060を定義してください。

XMAP03-プロパティ

ラベル: [ ] [ ] [ ] [ ]      ラベルの種類: テキスト

初期テキスト: [ ]      文字色: 黒

テキスト: [ ]      テキスト揃え: 中央

ビットマップ

	標準	交代1	交代2	交代3	縦×横
活性用	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> 金体カド付
不活性用	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> 確定...

選択時の動作: [ A Pへ通知する ]      表示方法: [ 標準表示 ]

通知コード: [ A081 ]

有効設定 (初期カドを本格的に設定する)      データ名: [ INC ]

不活性 (選択できない状態にする)

三連インクリメントを参照する       アクセスキーを使用する      アクセスキー: [ ]

A Pが渡す項目      データ名

動的変更 (A Pから表示属性を変更する)      [ BUTTON002- ]

[ A Pからテキストを変更する ]      桁      [ ]      桁      [ ]      左      [ ]

[ OK ]      [ キャンセル ]      [ ヘルプ ]

## (b) 生成される論理マップ

マップ名が MAP001 の場合の例

```

01 MAP0010. (出力論理マップ)
:
03 MAP001-BUTTON0002-0      PIX X(9). ... ラベルテキスト項目名
:

```

## 6. 画面定義とコーディングの関連

### (c) COBOL での AP

マップ名称が MAP001 の場合

```
WORKING-STORAGE SECTION.
COPY MAP0010.           ... 出力論理マップの取り込み
COPY MAP0011.           ... 入力論理マップの取り込み
COPY X3MODTBL.         ... 動的変更テーブルの取り込み
COMMUNICATION SECTION.
CD DSP FOR I-O WS      ... 通信記述名 : DSP
MAP NAME IS 画面マップ名. ... 物理マップ名称格納エリア
:
MOVE 'MAP001ND' TO 画面マップ名. ... 物理マップ名を代入
MOVE ALL X'1F' TO MAP001G.
TRANSCIEVE DSP FROM MAP0010 INTO MAP0011. ... 画面の送受信
EVALUATE MAP001-INCL ... ボタンのチェックと該当処理の呼び出し
WHEN 'A061'
  PERFORM 登録依頼処理
  MOVE '確認' TO MAP001-BUTTON002-0. ... ラベルテキスト変更
  MOVE 'MAP001ND' TO 画面マップ名. ... 物理マップ名を代入
  TRANSCIEVE DSP FROM MAP0010 INTO MAP0011. ... 画面の送受信
  IF MAP001-INCL = 'A061' ... ボタンのチェックと該当処理の呼び出し
  THEN
    PERFORM 登録処理
  ELSE
    PERFORM 終了処理
  END-IF.
WHEN 'A062'
  PERFORM 終了処理
WHEN OTHER
  PERFORM エラー処理
END-EVALUATE.
```

## 6.2.35 メニューバーを使用した処理の選択

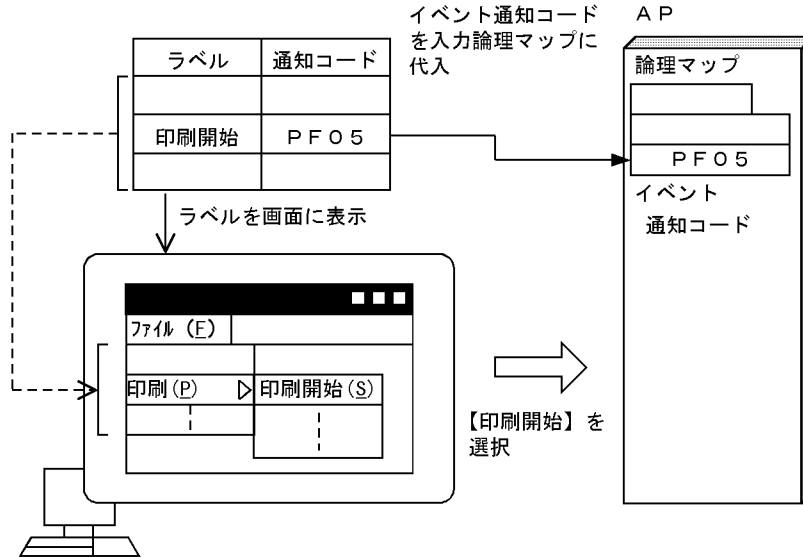
### (1) 仕様

メニューバーの [ ファイル ] を選択すると、プルダウンメニューが表示されます。プルダウンメニューの [ 印刷 ] を選択すると、カスケードメニューが表示されます。カスケードメニューの [ 印刷開始 ] を選択すると、入力論理マップにはイベント通知コード「PF05」が代入されます。

メニューバーを使用した処理の選択を次の図に示します。



図 6-39 メニューバーを使用した処理の選択



## (2) 画面定義の概要

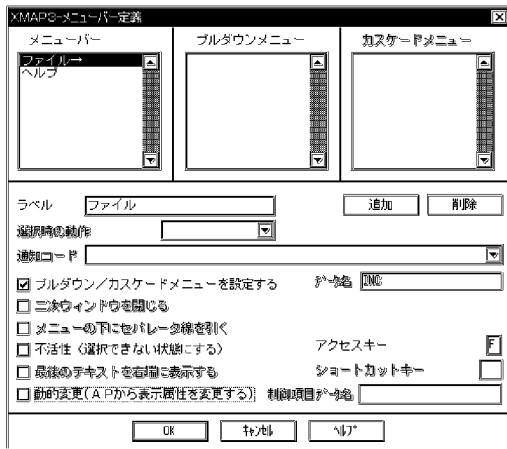
画面定義の概要を次に示します。

### (a) メニューバーの定義

1. GUI 画面定義ウィンドウの画面属性定義ダイアログで、[メニューバーを付ける]のチェックボタンを指定したあと、[メニューバー定義]を選択する。
2. メニュー定義ダイアログでメニューバーを定義する。次の表の項目を定義する。

定義項目	定義内容
ラベル	ファイル
プルダウン・カスケードメニューを設定する	指定する
アクセスキー	F

## 6. 画面定義とコーディングの関連



### (b) プルダウンメニューの定義

1. メニューバーを定義したあとで、プルダウンメニューを表示したいメニューバーを選択する。
2. プルダウンメニューのリストボックスをクリックしたあとで、プルダウンメニューの次の表の項目を定義する。

定義項目	定義内容
ラベル	印刷
プルダウン・カスケードメニューを設定する	指定する
アクセスキー	P

### (c) カスケードメニューの定義

1. プルダウンメニューを定義したあと、カスケードメニューを表示したいメニューバーを選択する。
2. カスケードメニューのリストボックスをクリックしたあとで、カスケードメニューの次の表の項目を定義する。

定義項目	定義内容
ラベル	印刷開始
アクセスキー	S
APへ通知するコード	PF05
選択時の動作	APに通知する

### (3) 生成される論理マップ

マップ名が MAP126 の場合の例

```
01 MAP126I. (入力論理マップ)
:
03 MAP126-INCI PIC X(4). ... 通知コードの格納領域
```

### (4) COBOL での AP

マップ名が MAP126 の場合の例

```
WORKING-STORAGE SECTION.
COPY MAP1260. ... 出力論理マップの取り込み
COPY MAP126I. ... 入力論理マップの取り込み
COPY X3MODTBL. ... 動的変更テーブルの取り込み

TRANSCEIVE DSP FROM MAP1260 INTO MAP126I. ... 画面の送受信
                                         カスケードメニューから【印刷開始】
                                         を選択する
:
EVALUATE MAP126-INCI } ... 選択したメニューの
  WHEN 'PF05'         } 「APへ通知するコード」を判定
  PERFORM 印刷開始処理
END-EVALUATE.
```

## 6.2.36 出力グラフィックの利用

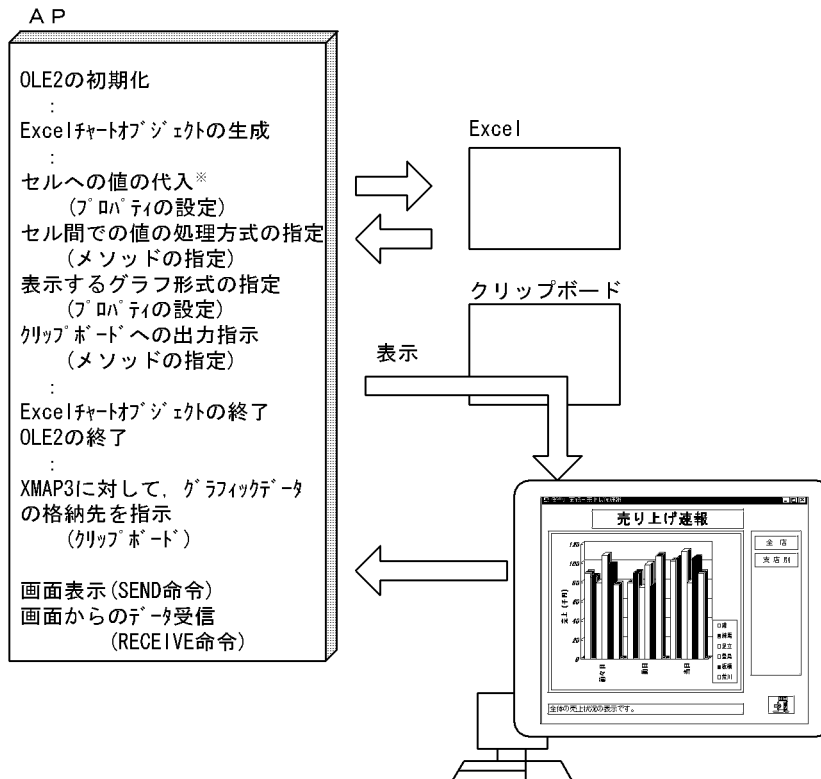
### (1) 仕様

Excel を使用して、グラフをビットマップ形式でクリップボード上に作成します。

作成したビットマップ形式の入力元を出力論理マップに代入します。このとき、入力元はクリップボードなので、「\* CLIP」を指定します。

出力グラフィックを利用したグラフの表示を次の図に示します。

図 6-40 出力グラフィックを利用したグラフの表示



注

この例では Excel のセルに、AP から直接、値を設定しています。そのほかにも、設定する値を CSV 形式 (Comma Separated Values 形式) で別ファイルに出力しておき、それを Excel のシートに読み込むように指示する方法もあります。

《Excel での準備》

「仕様」の図で示すように、OLE2 (Object Linking and Embedding 2) オートメーション機能を使用すると Excel では何も用意しておく必要はありません。しかし、使用するグラフや処理のパターンが一定であれば、それら进行处理する Excel のブック・シートを前もって作成しておき、直接使用すると、AP での OLE2 の操作を軽減できます。

また、図 6-40 の例では、セル間の値の処理方式、および表示するグラフ形式の指定を前もってシートに設定しておく、AP からの指定は不要になります。

- OLE2 オートメーションのサービス機能

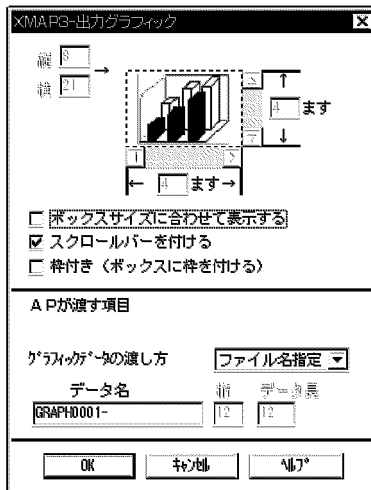
OLE2 オートメーション機能については、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL85 ユーザーズガイド」を参照してください。

## (2) 画面定義の概要

画面定義の概要を次に示します。

1. GUI 画面定義ウィンドウでツールボックスの [出力グラフィック] を使用して、領域を配置する。必要な桁数分、ドラッグして大きさを調整する。このとき、スクロールバーを付けることもできる。
2. 出力グラフィックダイアログで次の表の項目を指定する。

定義項目	定義内容
グラフィックデータの渡し方	ファイル名指定
データ名	GRAPH0001-
桁	12 (この定義内容は固定)
データ長	12 (この定義内容は固定)
スクロールバーを付ける	指定する
枠付き (ボックスに枠を付ける)	指定しない



## (3) 生成される論理マップ

マップ名が GRAPH1 の場合の例

```
01 GRAPH10.
:
02 GRAPH1-GRAPH0001-0 PIC X(12). ...グラフィックデータのファイル名指定エリア
```

(4) COBOL での AP

```

WORKING-STORAGE SECTION.
COPY GRAPH10. ....出力論理マップの取り込み
COPY GRAPH11. ....入力論理マップの取り込み
COPY X3MODTBL. ....動的変更テーブルの取り込み

COMMUNICATION SECTION.
CD DSP FOR I-O WS .....通信記述名：DSP
MAP NAME IS 画面マップ名 .....物理マップ名称格納エリア
:
MOVE ' GRAPHIND ' TO 画面マップ名.....物理マップ名を代入
*****
* グラフを、クリップボードから、貼り付ける *
* OLE2機能を使用する *
*****
グラフ表示 SECTION.
INVOKE 'EXCEL.APPLICATION' 'CREATEOBJ' RETURNING APP-OBJ.
.....OLE2の初期化とEXCELのAPPLICATIONのOLEオブジェクトの生成
SET 'APPLICATION.VISIBLE' WITH APP-OBJ TO 1. .... EXCELを表示する
INVOKE APP-OBJ 'WORKBOOKS.OPEN' USING VALUE
'C:\TEMP\GRAPH01.XLS'. ....作成済みのGRAPH01.XLSをオープンする
INVOKE APP-OBJ
'APPLICATION.WORKBOOKS(1).WORKSHEETS("SHEET1").SELECT'.
.....該当するシート"SHEET1"を選択する
SET 'APPLICATION.WORKBOOKS(1)' &
'.WORKSHEETS("SHEET1").CELLS(2,2).VALUE'
WITH APP-OBJ TO '73'.
SET 'APPLICATION.WORKBOOKS(1)' &
'.WORKSHEETS("SHEET1").CELLS(3,2).VALUE'
WITH APP-OBJ TO '63'.
SET 'APPLICATION.WORKBOOKS(1)' &
'.WORKSHEETS("SHEET1").CELLS(4,2).VALUE'
WITH APP-OBJ TO '55'.
SET 'APPLICATION.WORKBOOKS(1)' &
'.WORKSHEETS("SHEET1").CELLS(5,2).VALUE'
WITH APP-OBJ TO '53'.
SET 'APPLICATION.WORKBOOKS(1)' &
'.WORKSHEETS("SHEET1").CELLS(6,2).VALUE'
WITH APP-OBJ TO '43'.

INVOKE APP-OBJ
'APPLICATION.WORKBOOKS(1)' &
'.WORKSHEETS("SHEET1").CHARTOBJECTS("グラフ 7").CHART'
RETURNING CELL-OBJ. ...."SHEET1"内の"グラフ7"のCHARTオブジェクトを求める

INVOKE CELL-OBJ 'COPYPICTURE'. ...."グラフ 7"をクリップボードにコピーする
* OLE2の終了
SET 'APPLICATION.WORKBOOKS(1).SAVED' WITH APP-OBJ TO 1.
..... SAVEDプロパティをTRUE(保存しない)に設定し
ブックを閉じる

INVOKE APP-OBJ 'APPLICATION.QUIT'. ] ..... EXCELアプリケーションの終了
SET APP-OBJ TO NULL. ] ..... OLEオブジェクトの解放
SET CELL-OBJ TO NULL.
    
```

各セルの値を設定して  
グラフを変更する

OLE2の処理

(続く)

(続き)

```
MOVE ALL '1F' TO GRAPH1-GRAPH0001-0. ....必ず'1F'でクリアする。
MOVE '*CLIP' TO GRAPH1-GRAPH0001-0(1:5).
...クリップボードからグラフィックデータを読み込むことを示す文字列 '*CLIP'を設定する。
TRANSCIEVE DSP FROM GRAPH10 INTO GRAPH11. ....画面の送受信
```

## 6.2.37 GUI画面でのファンクションキーの利用

### (1) 仕様

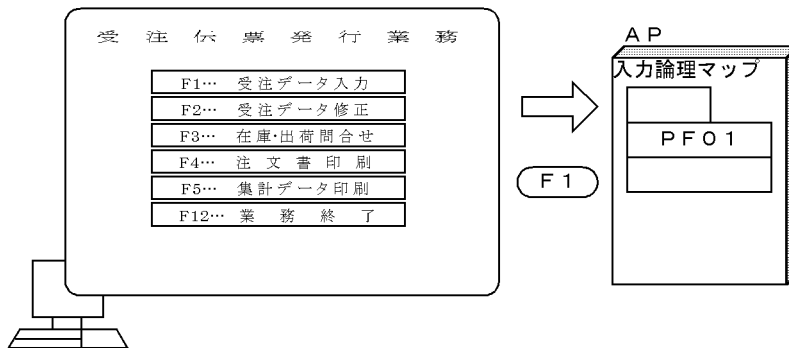
ファンクションキーを押して機能を選択します。

メニュー画面を表示し、APはどのプッシュボタン、またはファンクションキーが押されたかをチェックし、それぞれの処理部へ分岐します。

F1キーを押すとXMAP3が入力論理マップのイベント通知コードデータ名の領域に「PF01」を代入する例を次の図に示します。

なお、この例では、PFキーとプッシュボタンは同じイベント通知をします。

図 6-41 ファンクションキーを使用した処理の選択



### (2) 画面定義の概要

画面定義の概要を次に示します。

1. GUI画面定義ウィンドウで、ツールボックスの[プッシュボタン]を使用してボタンを配置してメニュー画面をレイアウトします。
2. 各ボタンの通知コードは、PFキーの通知コードと同じイベント通知コードにします。

ラベル	APへ通知するコード
F1 ~ F12	PF01 ~ PF12

## 6. 画面定義とコーディングの関連

ボタンとPFキーについて

この例では、ボタンとPFキーのイベント通知を同じにしていますが、ボタンがない場合も、PFキーは有効になります。また、PFキーのイベント通知コードと、ボタンのイベント通知コードを別々にして、区別して使用することもできます。

### (3) 生成される論理マップ

マップ名がMAP107の場合の例

```
01 MAP1070. (出力論理マップ)
:
01 MAP1071. (入力論理マップ)
:
03 MAP107-INCI PIC X(4). ...ファンクションキーに対応する通知コードの格納領域
```

### (4) COBOLでのAP

マップ名称がMAP107の場合

```
:
PROCEDURE          DIVISION.
* 画面表示処理
  MOVE 'DSP001' TO 画面端末名. ....端末名を代入
  MOVE ALL X'1F' TO MAP107G.
                                ....出力エリアのクリア
  MOVE 'MAP107ND' TO 画面マップ名. ...マップ名を代入
  TRANSCEIVE DSP FROM MAP1070 INTO MAP1071.
                                ....画面の表示と入力
  EVALUATE MAP107-INCI          ....ファンクションキーのチェックと該当処理の
                                呼び出し
    WHEN 'PF01'
      DISABLE DSP              ....メニューを消す
      CALL 受注データ確認処理 ... 処理を呼び出す
    WHEN 'PF02'
      DISABLE DSP
      CALL 受注データ修正処理
    WHEN 'PF03'
      DISABLE DSP
      CALL 在庫・出荷問合せ処理
    WHEN OTHER
      CONTINUE
  END-EVALUATE.
```

## 6.2.38 二次ウィンドウの表示

### (1) 仕様

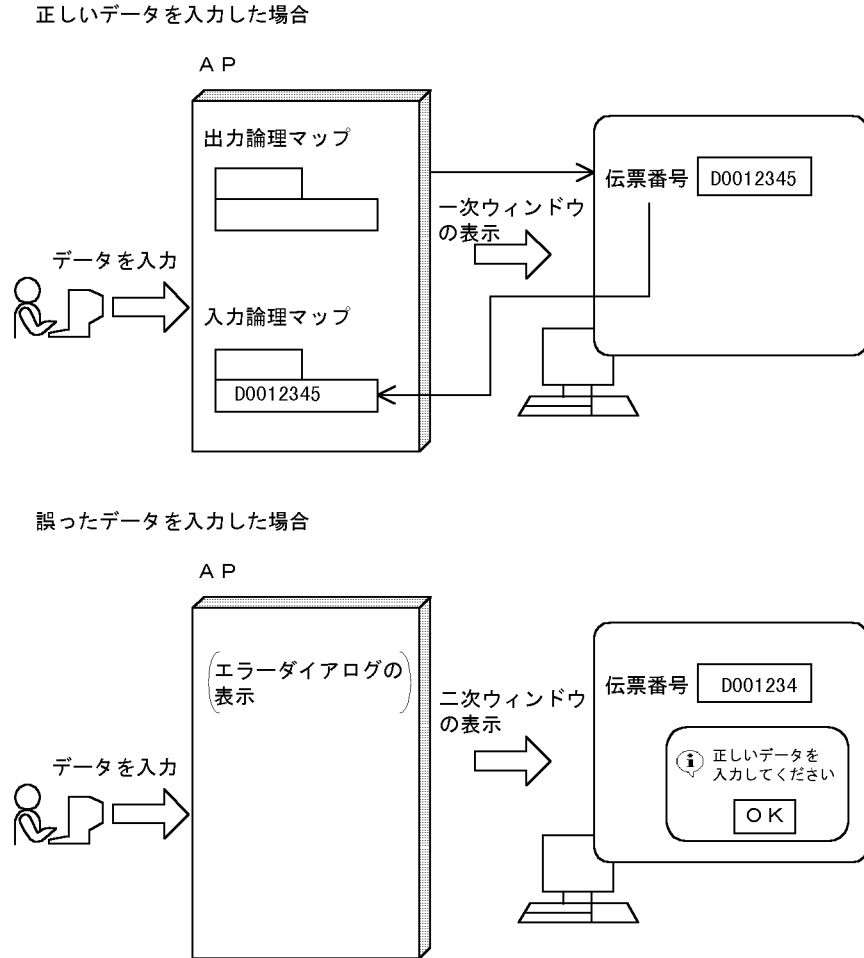
一次ウィンドウの入出力テキスト「伝票番号」欄に間違った伝票番号が入力されると、二次ウィンドウでエラーメッセージを表示します。



プッシュボタンの [OK] をクリックすると二次ウィンドウは閉じられます。

AP 実行時のエラーメッセージの表示を次の図に示します。

図 6-42 二次ウィンドウの表示



二次ウィンドウについて

- 二次ウィンドウのプッシュボタンをクリックしても、二次ウィンドウが閉じないようになります。
- 二次ウィンドウを開いたまま、もう一つの二次ウィンドウを表示できます。
- 表示されている二次ウィンドウの一部を書き換えられます。

## (2) 画面定義の概要

画面定義の概要を次に示します。



## (4) COBOL での AP

一次ウィンドウのマップ名が MAP102, 二次ウィンドウのマップ名が MAP103 の場合の例

```

WORKING-STORAGE      SECTION.
    77      入力データ確認フラグ      PIC X(1) VALUE '0'.
    ....., 入力データの正誤判定フラグ
    77      入力用ワーク              PIC X(8).
    ....., 入力データがデータベースにあるかの確認用ワークエリア

* 論理マップのCOPY展開
    COPY   MAP102. .... 出力論理マップ (一次ウィンドウ)
    COPY   MAP102I. .... 入力論理マップ (一次ウィンドウ)
    COPY   MAP103. .... 出力論理マップ (二次ウィンドウ)
    COPY   MAP103I. .... 入力論理マップ (二次ウィンドウ)

* 動的変更テーブルのCOPY展開
    COPY   X3MODTBL.

COMMUNICATION        SECTION.
    CD DSP FOR I-O WS .....通信記述
    .....データレイに対する通信記述項

    MAP NAME IS 画面マップ名 .....物理マップ名称格納エリア
    SYMBOLIC TERMINAL IS 画面端末名 .....端末名:画面端末名
    :

PROCEDURE DIVISION.
*   画面表示処理
    MOVE 'DSP001' TO 画面端末名. .... 端末名を代入
    MOVE ALL X'1F' TO MAP102G. .... 出力エリアのクリア
    MOVE ALL X'1F' TO MAP103G.
    MOVE 'MAP102ND' TO 画面マップ名. ... 物理マップ名を代入
    TRANSCIVE DSP FROM MAP1020 INTO MAP102I. ....一次ウィンドウの表示

*   入力データ確認
    PERFORM UNTIL 入力データ確認フラグ = '1'
*   --- 入力データがデータベースにあるかの確認処理 --- *
        IF MAP102-伝票番号-1 = 入力用ワーク
            THEN ..... 入力データがデータベースにある場合
                MOVE MAP102-伝票番号-1 TO MAP102-伝票番号-0
                .....出力エリアにデータを代入
                MOVE '1' TO 入力データ確認フラグ
            ELSE
                MOVE '0' TO 入力データ確認フラグ
                MOVE 'MAP103ND' TO 画面マップ名
                TRANSCIVE DSP FROM MAP1030 INTO MAP103I
                .....二次ウィンドウの表示 (エラーメッセージ表示)
        END-IF
    END-PERFORM.

```

## 6.2.39 初期フォーカスの動的変更

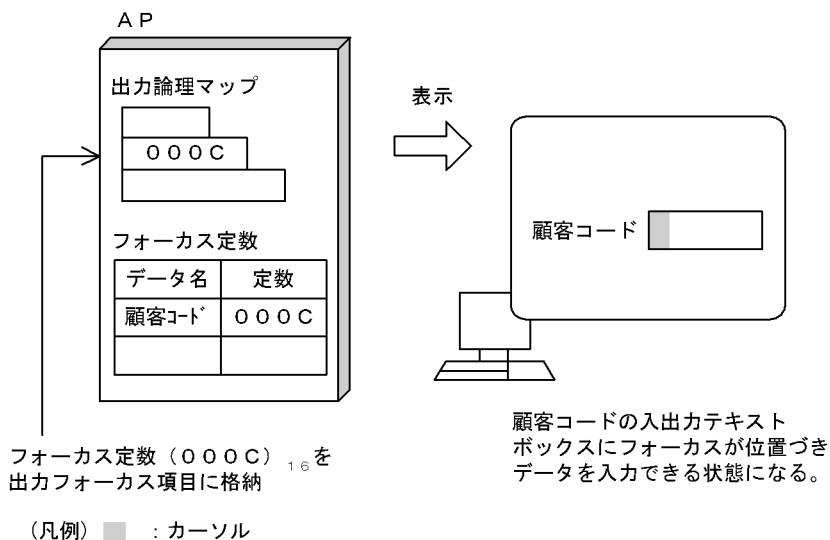
## (1) 仕様

画面を表示させたときに, AP からの指示によって「顧客コード」の入出力テキストボックスにフォーカスを位置づけます。AP からフォーカスを位置づける指定をしていないと, フォーカス位置は, 初期フォーカス位置に指定されます。

## 6. 画面定義とコーディングの関連

AP 実行時のフォーカスの位置づけを次の図に示します。

図 6-43 フォーカスの位置づけ



フォーカス定数は、基本的に対応するオブジェクトの数だけ、XMAP3が自動生成します。APはこれらの定数を使用します。定数は、各オブジェクトのAPが受け取る項目のデータ名に「T」を付けて、定数テーブルとしています。なお、この定数は初期カーソル設定でも使用でき、フィールドボックスへのフォーカスと、ボックス内の入出力フィールドへのカーソルを同時に設定できます。

### (2) 画面定義の概要

画面定義の概要を次に示します。

1. GUI画面定義ウィンドウで、ツールボックスの[入出力テキスト(英数)]を使用して入出力テキストボックスを配置する。
2. 入出力テキストボックスダイアログで次の項目を定義する。

定義項目	定義内容
データ名	顧客コード -
桁	8
データ長	8 (この定義内容は固定)

### (3) 生成される論理マップ

マップ名が MAP411 の場合の例

01	MAP4110. (出力論理マップ)	
:		
03	MAP411-OUTCURS-LOGO PIC X(2).	… 出力フォーカス位置格納エリア
03	MAP411-顧客コード-O PIC X(8).	… 出力テキストを代入するエリア
01	MAP411D. (フォーカス定数テーブル)	
02	MAP411-顧客コード-T.	… 入出力テキストボックス位置の定数
03	FILLER PIC X(2) VALUE X'000C'.	
01	MAP411I. (入力論理マップ)	
:		
03	MAP411-INCURS-LOGO PIC X(2).	… 入力フォーカス位置格納エリア
03	MAP411-顧客コード-H.	
04	MAP411-顧客コード-I PIC X(8).	… 入力テキストを代入するエリア

### (4) COBOL での AP

マップ名が MAP411 の場合の例

WORKING-STORAGE	SECTION.	
* 論理マップのCOPY展開		
COPY	MAP4110.	……出力論理マップ
COPY	MAP411I.	……入力論理マップ
* 動的変更テーブルのCOPY展開		
COPY	X3MODTBL.	
COMMUNICATION	SECTION.	……通信記述
CD DSP FOR 1-0 WS		……通信記述名: DSP
MAP NAME IS	画面マップ名	……物理マップ名称格納エリア
SYMBOLIC TERMINAL IS	画面端末名	…… 端末名: 画面端末名
:		
PROCEDURE	DIVISION.	
画面表示処理	SECTION.	
MOVE 'DSP001'	TO 画面端末名.	…端末名を代入
MOVE ALL X'1F'	TO MAP411G.	…出力エリアのクリア
MOVE MAP411-顧客コード-T	TO MAP411-OUTCURS-LOGO.	…フォーカス定数の設定
MOVE 'MAP411ND'	TO 画面マップ名.	…物理マップ名を代入
TRANSCIVE DSP FROM	MAP4110 INTO MAP411I.	…画面の表示

## 6.2.40 隠しフィールドの設定

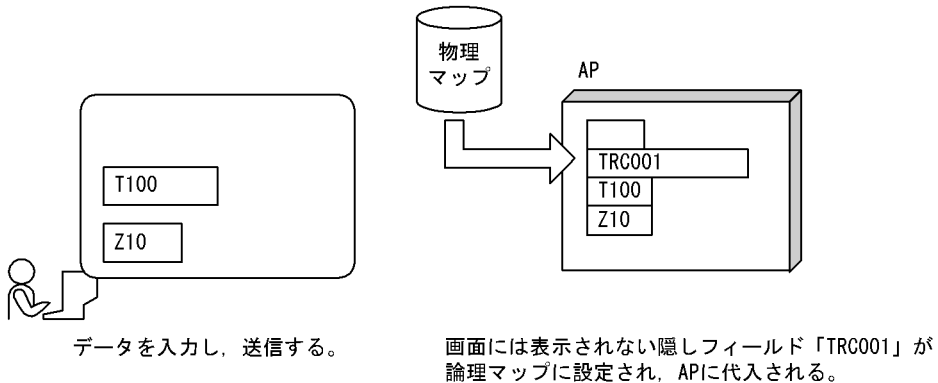
### (1) 仕様

画面から入力したとき、必ず論理マップの先頭に定数「TRC001」を指定するようにします。出力用 AP と入力用 AP が別プログラムになっているとき、AP はこの定数を参照して、どこからの入力か判断します。

隠しフィールドの設定を次の図に示します。

## 6. 画面定義とコーディングの関連

図 6-44 隠しフィールドの設定



### (2) 画面定義の概要

画面定義の概要を次に示します。

1. GUI画面定義ウィンドウの画面属性ダイアログで、「隠しフィールド」を選択する。
2. 隠しフィールドダイアログで次の項目を定義します。

定義項目	定義内容
本フィールドを設定	する
データ名	TRAN001
設定する値	TRC001
データ長	9文字固定

#### 設定する値

最大8文字の半角文字列を指定します。最後の9バイト目はスペースが設定されるので、入力できません。

### (3) 生成される論理マップ

#### (a) 入力論理マップ

マップ名が MAP001 の場合の例

```

01 MAP0011.
  02 MAP001S      PIC S9(4)  COMP-5
                   VALUE  +30.
  02 MAP0010      PIC S9(4)  COMP
                   VALUE  +0.
  02 MAP001K.
    03 MAP001-TRAN001I  PIC X(9).  .....隠しフィールド'の定義項目※
    03 MAP001-INCI      PIC X(4).
    03 MAP001-INCURS-LOGI PIC X(2).

```

#### 注

TRANSCIVE 文などで画面に出力したあと、隠しフィールドの定義項目で、定義時に「設定する値」として指定した 9 バイトの文字列が入ります。

#### (b) 出力論理マップ

隠しフィールドに対応する項目はありません。

```

01 MAP0010.
  02 MAP001L      PIC S9(4)  COMP-5
                   VALUE  +8.
  02 MAP001Z      PIC S9(4)  COMP
                   VALUE  +0.
  02 MAP001G.
    03 MAP001-CNTRLO  PIC X(2).
    03 MAP001-OUTCURS-LOGO
                   PIC X(2).
*
01 MAP001T      PIC S9(4)  COMP-5
                   VALUE  +8.
01 MAP001D.
  02 MAP001-TRAN001T.  .....隠しフィールドのダミーのフォーカス定数※
    03 FILLER      PIC X(2)  VALUE  X'0004'.

```

#### 注

隠しフィールドは画面上に表示されないフィールドであるため、フォーカス定数は使用しません。

### (4) 使用時の注意事項

- 隠しフィールドは入力専用なので、必ず「設定する値」を指定してください。「設定する値」を指定しないと半角スペース 9 文字が仮定されます。

## 6. 画面定義とコーディングの関連

- 隠しフィールドに設定する値を画面に出力したり、画面から入力することはできません。
- 隠しフィールドは1画面に一つだけ設定できます。
- トランザクションコードは隠しフィールドで定義してください。



# 7

## AP のコーディング方法

この章では、XMAP3 実行支援の AP インタフェースについて説明します。

---

7.1 XMAP3 実行支援の AP インタフェースについて

---

7.2 COBOL の画面入出力命令

---

7.3 C 言語の画面入出力命令

---

7.4 汎用 AP インタフェース

---

7.5 OLTP サーバ構成での画面表示

---

## 7.1 XMAP3 実行支援の AP インタフェースについて

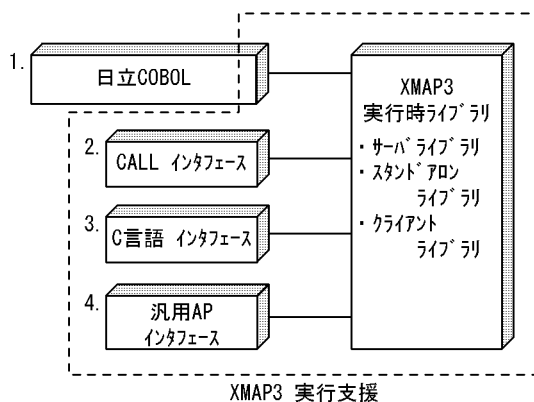
XMAP3 実行支援の AP インタフェースについて説明します。なお、XMAP3 の AP インタフェース (ライブラリ) はマルチスレッドで動作する AP からの実行に対応していません。

### 7.1.1 XMAP3 での画面の入出力の基本

#### (1) AP インタフェースの種類

XMAP3 実行支援には、四つの AP インタフェースがあります。XMAP3 実行支援の詳細についてはマニュアル「XMAP3 開発・実行ガイド」を参照してください。

図 7-1 AP インタフェースの種類



#### 解説

1. COBOL 言語仕様に取り込まれた言語インタフェースです。日立 COBOL で使用できます。すべての OS で共通の文法です。
2. COBOL の CALL インタフェースで呼び出す方法です。日立 COBOL など、標準 CALL インタフェースを持つ COBOL 言語から使用できます。
3. C 言語インタフェースです。XMAP3 共通の C 言語用 AP インタフェースです。また、PC / WS で共通の文法です。
4. PC 専用の汎用 AP インタフェースです。COBOL, C 言語以外の言語用の AP インタフェースです。

#### (2) AP インタフェースの概要

画面用 AP での XMAP3 を使用するときの命令を表 7-1 に示します。また、画面用 AP と XMAP3 の間で渡される情報を図 7-2 に示します。

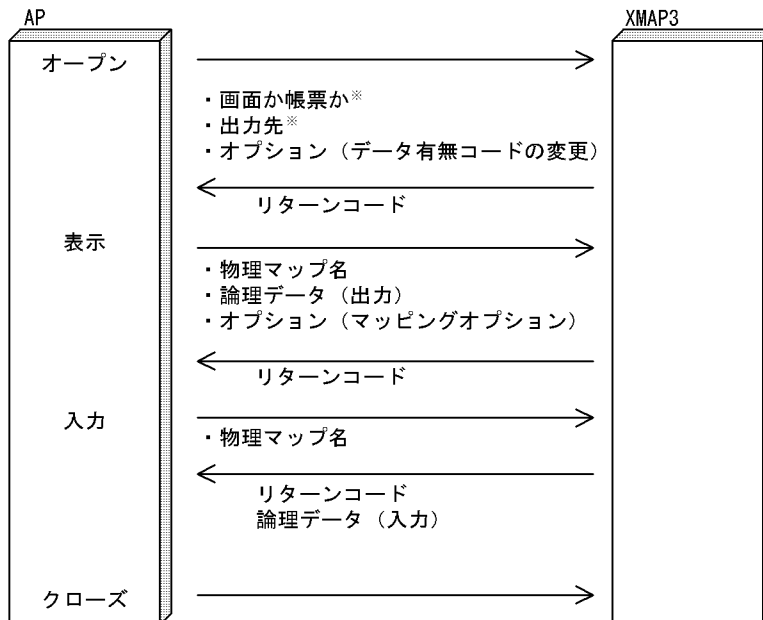
表 7-1 XMAP3 での画面の入出力の基本

要求	日立 COBOL の命令	COBOL の CALL 文	C 言語	汎用関数
オープン	最初の SEND または TRANSCEIVE 文発行時	'OPEN' 'MDO ' 要求	'OPEN' 'MDO ' 要求	XmapDrvCreateOpen
表示	SEND 文または TRANSCEIVE 文	'SEND' 要求	'SEND' 要求	XmapDrvSend
入力	RECEIVE 文または TRANSCEIVE 文	'RECV' 要求	'RECV' 要求	XmapDrvReceive
クローズ	DISABLE 文, GOBACK 文, EXIT PROGRAM 文, CANCEL 文 (AP の) STOP RUN	'CLOS' 要求	'CLOS' 要求	XmapDrvClose

注

一つの AP (プロセス) 内で同時にオープンできる端末数の上限は 15 個です。

図 7-2 XMAP3 と AP 間のデータのやり取り



注※ 必ず指定します。

## 7.1.2 AP 間でオープンを引き継ぐ場合

メッセージを出す共通部分を別 DLL 化したり、プログラムが大きいので分割したりする場合などは、オープンをプログラム間で引き継がせることをお勧めします。オープンを引き継ぐためには、日立 COBOL の場合、プログラムを実行させる AP（ここでは AP1）で環境変数「CBLTERMSHAR=YES」を指定しておきます。なお、「CBLTERMSHAR=YES」は、SEND / RECEIVE / TRANSCEIVE インタフェースのときだけ有効です。CALL インタフェースでは無効になります。

オープンを引き継ぐことによって、AP1 から AP2 を呼び出すとき、AP1 でのクローズ（日立 COBOL のとき DISABLE 文の発行）および AP2 でのオープンは発生しません。そのため画面遷移の時間が短くなります。

使用できる日立 COBOL を OS ごとに次の表に示します。

表 7-2 使用できる日立 COBOL

OS 名	COBOL2002	COBOL85		
		Version 5.0	Version 6	Version 7
Windows 95	x			x
Windows 98		1		
Windows Me		x	2	
Windows NT 4.0				
Windows 2000		x	3	
Windows XP		x	x	
Windows Server 2003	4	x	x	x
Windows Server 2003 x64	5	x	x	x

（凡例）

：使用できる。

x：使用できない。

注 1

使用できるバージョンは 05-01 以降となります。

注 2

使用できるバージョンは 06-03 以降となります。

注 3

使用できるバージョンは OS によって異なります。

- ・ Windows 2000 Server, および Windows 2000 Professional : 06-01 以降

- ・ Windows 2000 Advanced Server : 06-03 以降

- ・ Windows 2000 Datacenter Server : 06-04 以降

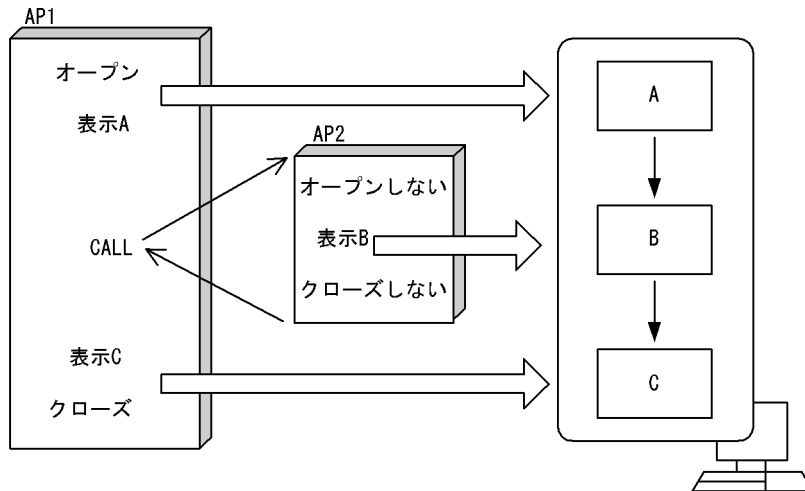
注 4

使用できるバージョンは 01-01 以降となります。

注 5

使用できるバージョンの詳細については、「Readme ファイル」でご確認ください。

図 7-3 AP 間でオープンを引き継ぐ場合



### 7.1.3 AP 分割時の注意

日立 COBOL では、明示的に CALL 文でオープンを要求する場合を除き、一つのコンパイル単位で最初の SEND 文または TRANSCIVE 文が発行されると画面がオープンされます。そのため、1 画面が 1 実行ファイル (.EXE) のような構成にすると、画面ごとにオープンをすることになります。この場合、オープン・クローズ、SEND 文または TRANSCIVE 文を発行する実行ファイル、ビジネス処理をする実行ファイルをそれぞれ分けた方が実行性能が高くなります。ビジネス処理をする実行ファイルは、DLL ファイルでも代用できます。.EXE ファイルにするか .DLL ファイルにするかは、処理の形態によって次のように使い分けてください。

- .EXE：メニュー画面など処理が続く場合（.EXE ファイル間でのデータ受け渡しはない）
- .DLL：処理が単独で閉じる場合（1 ウィンドウで閉じる）

また、複数のコンパイル単位のを合わせて一つの実行ファイルにするときは、各コンパイル単位でオープンを発行しないようにするため、日立 COBOL の実行支援の環境変数で、「CBLTERMSHAR=YES」を指定します。「CBLTERMSHAR=YES」は、SEND / RECEIVE / TRANSCIVE インタフェースのときだけ有効です。CALL インタフェースでは無効になります。

## 7. AP のコーディング方法

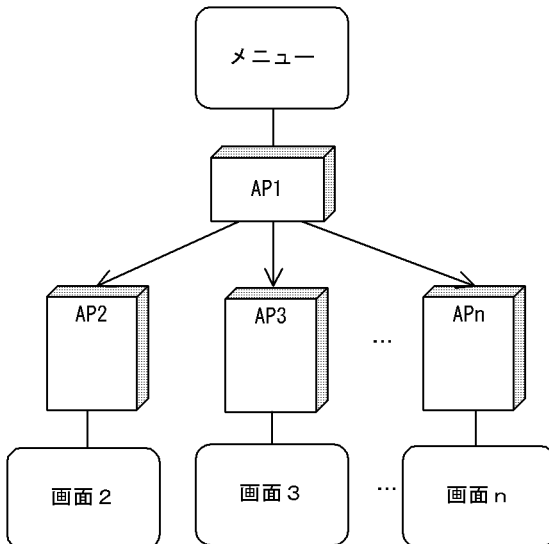
CALL インタフェースの場合、または「CBLTERMSHAR」をサポートしていないバージョンの日立 COBOL を使用している場合で、AP 間のオープン引き継ぎをするときは、ユーザが、XMAP3 インタフェースエリアの情報を引き継ぐようなコーディングをする必要があります。サンプルコーディングとして、AP パターンの GENDSP02 および GENDSP03 が利用できます。

### (1) メインの実行ファイルから別の実行ファイルを呼び出す場合

メインメニュー（AP1）から各サブプログラム（AP2 ~ n）を呼び出すケースを次の図に示します。

- AP1, AP2 ~ n の各 AP で、オープン ~ クローズを行う。
- AP1 から AP2 ~ n を呼び出すときは、表示している画面を一度消してから、新しい画面を表示し直す。

図 7-4 メインメニュー（AP1）から各サブプログラム（AP2 ~ n）を呼び出すケース

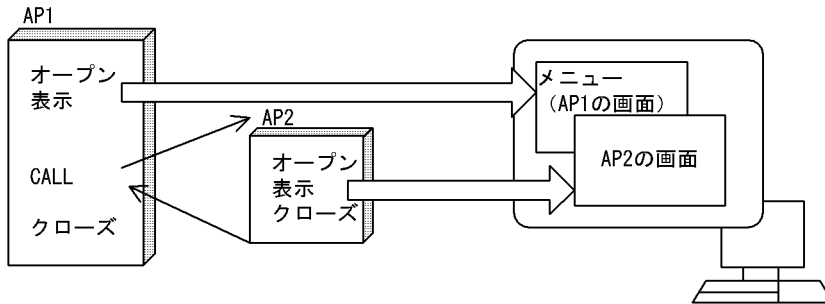


### (2) 新しい画面を表示するとき前の画面を消さないで表示しておくケース

AP1 から AP2 ~ n を呼び出すとき、AP1 の画面を消さないで表示しておくケースを次の図に示します。ただし、この方法では動きが遅くなります。

- AP1 で画面をオープン AP2 ~ n を呼び出す
- AP2 ~ n ではオープン ~ クローズを行う

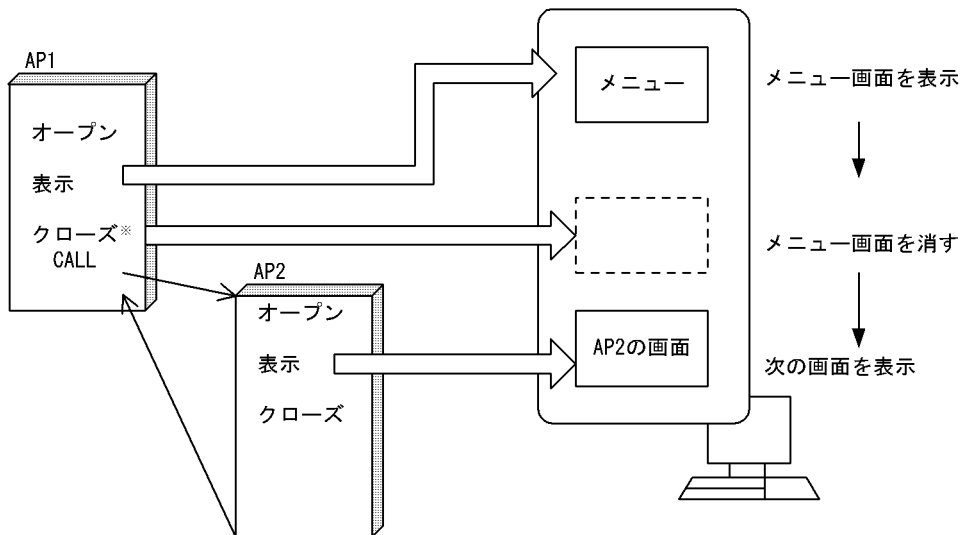
図 7-5 AP1 から AP2 ~ n を呼び出すとき、AP1 の画面を消さないで表示しておくケース



### (3) 新しい画面を表示するとき前の画面を消す場合

AP1 から AP2 ~ n を呼び出すとき、AP1 の画面を消して AP2 を表示するケースを次の図に示します。ただし、この方法では動きが遅くなります。

図 7-6 AP1 から AP2 ~ n を呼び出すとき、AP1 の画面を消して AP2 を表示するケース



注※ DISABLE文を発行 (COBOLだけ)

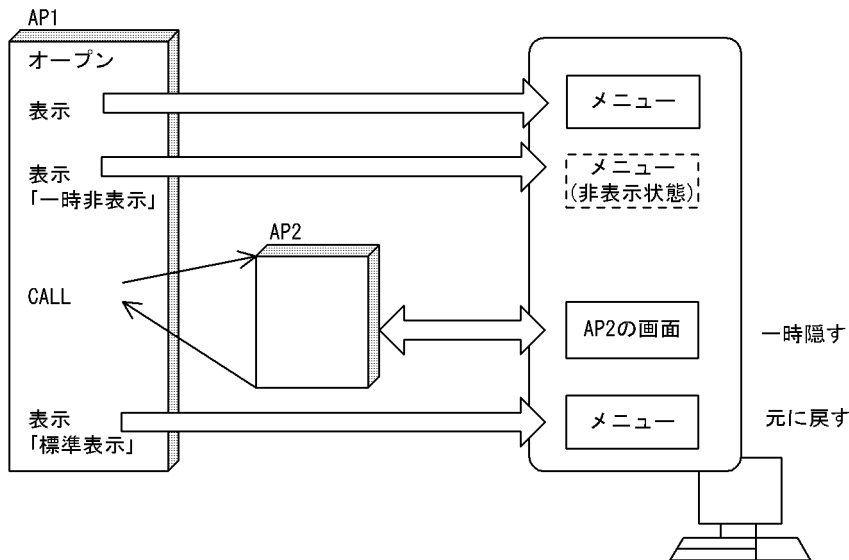
### (4) 画面をクローズしないで一時的に表示中の画面を消す場合

画面をクローズすると、次表示時には必ずオープンしなければならないので、この間のオープン/クローズの時間が掛かります。このような時間を節約したいときには、GUI画面の画面属性ダイアログの「Z位置」で「後ろに表示」か「一時非表示」を指定します。これらの指定をすると、一時的に表示中の画面を見えないようになります。いちば

## 7. AP のコーディング方法

ん後ろに表示された画面を前に表示させたり、一時的に非表示になった画面を元に戻す場合は、ドロースेटアップの「表示属性の動的変更」の「位置属性」タブで「XY 位置」を「手前に表示」と指定します。

図 7-7 画面をクローズしないで一時的に表示中の画面を消す場合



### 7.1.4 AP での入力論理マップ上のデータチェック

#### (1) イベント通知コードのチェック

画面確定時のイベント通知コード（例えば、ファンクションキー 1 では PF01，ボタンでは A001）を返します。イベント通知コードの値は、ドロースेटアップで変更できません。なお、イベント通知コードが AP に渡るタイミングは入力単位の指定（画面確定のタイミング）によって異なります。それぞれの詳細については、「1.5.8 入力単位」を参照してください。

画面属性ダイアログの入力単位で「画面」を指定している場合

画面上で、一つだけ活性になっているオブジェクトを選択したり、データを入力したりしたあと、画面確定を設定しているメニューバー、プッシュボタンまたは画面確定キーを押すと、AP に制御が渡ります。また、「自動送信」を指定しているオブジェクトからフォーカス・カーソルが離脱したタイミングでも AP に制御が渡ります。

画面属性ダイアログの入力単位で「フィールド」を指定している場合

画面上で活性になっている、フォーカス・カーソルのあるオブジェクトを選択したり、データを入力したりしたあと、フォーカス・カーソルが次のオブジェクトに移動するタイミングで AP に制御が渡ります。



画面属性ダイアログの入力単位で「イベント」を指定している場合  
画面上で活性になっているすべてのオブジェクトから一つを選択したり、データを入力したりしたあと、フォーカス・カーソルがほかのフィールドに移動するタイミングで AP に制御が渡ります。

## (2) 入力データのチェック

XMAP3 では、画面から入力されたデータを使用目的に従ってチェックします。エラーがあれば、入力時にエラーがあると即座に画面に返す場合と、設定に従ったエラー通知文字やデータ有無コードを AP に返し、処理を続行する場合があります。

画面から入力したデータ以外で、AP に返すデータは次のとおりです。なお、データが返されているかどうかは「データ項目名 + H」のデータ名で参照します。

### (a) エラー通知

例えば、カナ文字を扱うテキスト・フィールドに対して全角文字（漢字）を入力した場合に、エラー通知文字（HIGH (X'FF')）を AP に返します。

エラー通知文字は、ドロースेटアップの「論理マップ属性」で変更できます。

### (b) データ消去の通知

例えば、フィールドキー（[ End ] キー）でデータ消去した場合に、データ消去通知文字（LOW (X'00')）を AP に返します。

データ消去通知文字は、ドロースेटアップの「論理マップ属性」で変更できます。

### (c) データ未入力の通知（初期値を返す場合）

例えば、テキスト・フィールドで初期値に ABC の文字を設定し、データの入力をしない場合に初期値の ABC を AP に返します。

## (3) データ未入力の通知（初期クリア文字を返す場合）

初期値の文字を設定せず、データの入力をしない場合に初期クリア文字（埋字）を返します。

例えば、使用目的が「英数」の場合は埋字（スペース）を、「数字」の場合は埋字（ゼロ）を AP に返します。

初期クリア文字は、ドロースेटアップの「論理マップ属性」で変更できます。

## (4) フォーカス・カーソル位置のチェック

画面確定時に、どのオブジェクトにフォーカス・カーソルが位置づいているかをチェックできます。なお、フォーカス・カーソルについては「2.4.26 フォーカスとカーソル」を参照してください。

## 7. AP のコーディング方法

### (a) フォーカス位置の通知 (GUI 画面の場合)

画面確定時のフォーカス位置を、フォーカス定数で返します。

フォーカス定数は、例えばドロで AP が受け取る項目のデータ名を MAP001-FIELD0001- に設定した場合は、MAP001-FIELD0001-T の名称で論理マップ中に生成されます。フォーカス定数と入力フォーカス項目の値が一致するオブジェクトの位置が、画面上に位置づいたフォーカスの位置として AP で認識されます。

### (b) カーソル位置の通知

画面確定時のカーソル位置を、カーソル定数で返します。

カーソル定数は、例えばドロで AP が受け取る項目のデータ名を MAP001-FIELD0001- に設定した場合は、MAP001-FIELD0001-T の名称で論理マップ中に生成されます。カーソル定数と入力カーソル項目の値が一致するオブジェクトの位置が、画面上に位置づいたカーソルの位置として AP で認識されます。

ドロセットアップの「カーソルとフォーカス」で行列 (2 進) カーソルに変更できません。ドロセットアップでの設定についてはマニュアル「XMAP3 開発・実行ガイド」を参照してください。

## 7.1.5 AP 作成時の注意事項

### (1) [ 閉じる ] ボタンを使用する場合

XMAP3 の画面で [ 閉じる ] ボタンまたは [ 閉じる ] メニューが操作された場合、XMAP3 では、直接ウィンドウを閉じずに AP に対して [ 閉じる ] ボタンまたは [ 閉じる ] メニューが操作されたことを通知します。このため、AP では、このイベントに対して、次のような処理をしてください。

なお、通知するイベント種別は、[ Break ] キーに対応するイベント通知コードです。

#### (処理の例)

- 通知コードが返ってきたとき、AP で必要な終了処理をする。
- 直ちに終了すると不都合が発生する場合は、サブウィンドウで確認をして、その応答内容によって、終了するかどうかを決定する。

イベントに対応する処理がない場合は、[ 閉じる ] ボタンを使用しても、AP を終了できません。また、「閉じる」処理に対応していない AP を使用する場合は、[ 閉じる ] ボタンおよび [ 閉じる ] メニューを表示しないで運用してください。

なお、ウィンドウ上の [ 閉じる ] ボタンやコントロールメニューの [ 閉じる ] を使用する場合は、表示・印刷セットアップの「デザイン 1」タブで指定します。

## (2) [PF] キーなどに「中断」を割り当てた場合

表示・印刷セットアップの「キー操作1」タブの「強制確定キーの動作」で、「入力済みデータを送らない」を指定した場合、その設定は [PA1] ~ [PA3] キー, [BREAK] キー, [SCREEN] キーだけが対象となります。

ドローで [PF] キーなどに「中断」を割り当てた場合は、そのキーを押すと画面上のデータも送られます。入力必須項目や、日付・時刻・金額項目など、通常はデータの妥当性がチェックされる項目についても、チェックがされていない状態のデータが返るので、「中断」を使用する場合には、必ず AP で「中断」イベントが起こっていないことを確認してからデータを処理してください。

## (3) AP に制御が渡る前の画面を表示する場合

表示画面から強制確定キーによって AP に制御が渡るとき、金額、数値、日付、時刻の入出力テキストおよびフィールドについては、表示されていた画面上のデータは保証されません。再び同じ画面を表示する場合は、明示的にそれぞれのオブジェクトのデータを設定するか、画面属性ダイアログの「入力・選択状態の扱い」を「初期状態」にして表示してください。

## (4) ポップアップメニューの動的変更をする場合

ポップアップメニューの通知コードを変更する場合やメニューの数を少なくする場合で、動的変更を実施する前に選択されていたメニューに対する通知コードが動的変更後になくなるようなときは、AP から設定する通知コードをクリアするか、存在するメニューの通知コードを設定して動的変更を実施してください。

## 7.2 COBOL の画面入出力命令

---

ここでは、XMAP3 を使用した、スタンドアロンまたは C/S システムでの COBOL の AP のコーディング方法について説明します。コーディングをする前に、次の準備をしてください。

1. AP を格納するためのフォルダを用意する

ソースプログラムやコンパイル後に生成される作業ファイルを格納するフォルダを用意します。開発環境に合わせてフォルダを分類し、作成位置や名称を決めます。

2. XMAP3 の標準提供ファイルから、必要なファイルをコピーする

XMAP3 では、AP の作成に必要な登録集原文 を標準提供しています。1. で作成したフォルダにコピーして使用することをお勧めします。このとき、コーディングのひな型である AP パターンもコピーしておきます。

### 注

標準提供の登録集原文を次に示します。

- 動的変更テーブル (X3MODTBL.CBL)

動的変更用の修飾名と出力論理マップの初期化に使用する定数が指定されています。

- インタフェース領域 (JSVWATBL.CBL)

CALL 文で画面を送受信するときにパラメタとして使用するインタフェース領域が指定されています。

### 7.2.1 仮想端末の自動割当て

C/S システム構成の環境で、サーバ上の AP から複数のクライアント PC へ画面表示をする場合、仮想端末の自動割当て機能を利用できます。この機能は、C/S システム構成の場合だけ利用できます。

仮想端末の自動割当て機能は、サーバ AP が各クライアント PC に対応した仮想端末名を意識することなく一つの仮想端末名だけを意識していれば、各クライアント PC にある XMAP3 の表示サービスが起動されたタイミングで、AP から送信された情報を画面表示するようにできます。仮想端末の自動割当て機能を利用するときは、サーバ側の C/S セットアップで、AP で指定した自動割当て用の仮想端末名を設定します。

仮想端末の自動割当ての詳細については、マニュアル「XMAP3 開発・実行ガイド」を参照してください。

### 7.2.2 論理マップの取り込み方法 (COBOL)

AP 中に論理マップを取り込む場合、COBOL の WORKING-STORAGE SECTION、または LINKAGE SECTION に COPY 文を指定します。ただし、論理マップ中に定数を展

開している場合、LINKAGE SECTION には取り込めません。

AP に論理マップを取り込む例を次に示します。

(取り込みの例)

```
WORKING-STORAGE SECTION.
COPY MAP003O. *1 ..... 出力論理マップの取り込み
COPY MAP003I. *1 ..... 入力論理マップの取り込み
COPY X3MODTBL. *2 ..... 動的変更テーブルの取り込み
```

注 1

新規作成ダイアログのマップ名に、出力用は「O」を入力用は「I」を付けたものです。ここではマップ名に MAP003 を指定した場合の例になります。

注 2

AP から文字色などの表示属性を変更するとき使用するデータが格納されたテーブルです。

### 7.2.3 画面表示命令 ( COBOL )

スタンドアロン環境、および C/S システム環境下では、次に示す方法で AP から PC に対して、画面の入出力ができます。

1. 日立 COBOL の SEND 文、RECEIVE 文、および TRANSCEIVE 文による方法
2. COBOL の CALL 文による方法

#### ( 1 ) SEND 文、RECEIVE 文、および TRANSCEIVE 文による方法

日立 COBOL の SEND 文、RECEIVE 文、および TRANSCEIVE 文によって画面の入出力をする方法について指定例を使用して説明します。

詳細については、マニュアル「COBOL2002 言語 拡張仕様編」またはマニュアル「COBOL85 言語」を参照してください。

#### ( a ) 通信記述項

画面の入出力に必要な情報を COMMUNICATION SECTION 中に指定します。

## 7. AP のコーディング方法

(例)

COMMUNICATION SECTION.	
*	
CD DSP FOR I-O WS .....	1
MAP NAME IS 画面マップ名 .....	3
SYMBOLIC TERMINAL IS 画面端末名. ....	4
MAPPING MODE IS マッピングモード .....	5
[STATUS KEY IS 画面-RC] .....	6
[DATA ABSENCE CODE IS データ有無コード] .....	7

注

1, 2, 3 および 5 は必ず指定してください。ほかの項目は、必要に応じて指定してください。

(例の説明)

1. 通信記述名 (DSP) (必須)

SEND 文, RECEIVE 文, および TRANSCIVE 文で使用する名称です。

2. 通信種別 (I - O WS) (必須)

ディスプレイ用であることを示します。

3. 物理マップ名称格納エリア (必須)

SEND 文, RECEIVE 文, および TRANSCIVE 文で出力, または入力したい場合に物理マップ名称を格納するエリアです。格納するマップ名には, 次に示すデバイス ID を付ける必要があります。

GUI 画面の場合

マップ名 6 文字: ND

マップ名 7 文字: O

CUI 画面の場合

マップ名 6 文字: NC

マップ名 7 文字: S

4. 仮想端末名称格納エリア

最初の SEND 文を実行する前に, 仮想端末名を格納しておくエリアです。正しい名称を指定しないと, 実行時に表示先が決定できなくなりエラーになります。スタンドアロンの場合は, 標準の「DSP001」を指定します。C/S システムの場合は, C/S セットアップで指定した仮想端末名を指定してください。

AP 中で仮想端末名を指定していない場合, 仮想端末名の先頭 1 バイト目にスペースを指定している場合, または「SYMBOLIC TERMINAL IS」を省略している場合, 環境変数「CBLTERMID」で指定された仮想端末名が仮定されます。環境変数「CBLTERMID」も指定されていない場合は, サーバ PC 上に画面を表示します。

また, 環境変数「CBLTERM\_xxx」を指定することで, 出力先の仮想端末名を変更することもできます。

CBLTERMID, CBLTERM\_xxx の詳細については、マニュアル「COBOL2002 ユーザーズガイド」またはマニュアル「COBOL85 ユーザーズガイド」を参照してください。

5. マッピングオプション（必須）

論理マップ、または物理マップのマッピング方式を指定します。詳細については、「7.2.3(1)(f) MAPPING MODE 句の扱い（マッピングオプション）」および「1.5.3(1) マッピングオプション」を参照してください。

6. リターンコード格納エリア

SEND 文、RECEIVE 文、および TRANSCEIVE 文のリターンコードを格納するエリアです。SEND 文、RECEIVE 文、および TRANSCEIVE 文が正常に処理されたかを判定する場合、このエリアを参照します。詳細については、「7.2.3(1)(g) STATUS KEY 句のリターンコード」を参照してください。

7. データ有無コード格納エリア

データ有無コードを変更したい場合には、最初の SEND 文を実行する前にデータ有無コードを格納しておく必要があります。「DATA ABSENCE CODE IS」を省略すると、データ有無コードとして標準の (1F)<sub>16</sub> が仮定されます。なお、データ有無コードはすべての AP にかかわるコードのため、できるだけ標準値を利用することをお勧めします。

(b) SEND 文

ディスプレイに画面を表示します。画面表示時に変更したいデータは、出力論理マップへ格納したあとに実行します。

画面の送信から受信までの間に AP でほかの処理をする場合などに使用します。この場合、表示画面への入力操作を抑止するために、ドロウの定義で、画面属性の「キーボードのロック状態を解除する」のチェックを外した画面を作成してください。

この画面を最初に SEND 文で表示し、AP 側の処理が終了して表示画面への入力操作ができるようになったとき、AP からウィンドウ属性の「キーボードのロック状態を解除する」の修飾名を指定して、SEND / RECEIVE 文（または TRANSCEIVE 文）を発行してください。

なお、二次ウィンドウ、または二次ウィンドウがある一次ウィンドウの場合の送受信には、SEND 文を使用せずに TRANSCEIVE 文を使用してください。

(例)

PROCEDURE DIVISION.	
：	
MOVE 'MAP003ND' TO 画面マップ名	… 1. マップ格納エリアにマップ名 'MAP003ND' を格納
SEND DSP	… 2. 通信記述名を指定 (ディスプレイ用)
FROM MAP0030	… 3. 出力用論理マップ名称
WITH EMI	… 4. 必ず指定すること

## 7. AP のコーディング方法

### (c) RECEIVE 文

ディスプレイ端末からの画面入力を行います。画面入力の結果は、入力論理マップに格納されます。SEND 文と対で指定してください。

#### (例)

```
PROCEDURE DIVISION.  
  .  
  SEND DSP  
  .  
  RECEIVE DSP ..... 1. SEND文と同じ通信記述名を指定  
    FIRST SEGMENT ..... 2. 必ず指定すること  
    INTO MAP003I ..... 3. 入力用論理マップ名称
```

### (d) TRANSCEIVE 文

ディスプレイ端末での画面の入出力をまとめて行います。SEND 文と RECEIVE 文の機能を持ちます。通常の画面の送受信には、TRANSCEIVE 文を使用するようにしてください。

#### (例)

```
PROCEDURE DIVISION.  
  .  
  MOVE 'MAP003ND' TO 画面マップ名 ... 1. マップ名称格納エリアに物理マップ名  
    'MAP003ND' を格納  
  TRANSCEIVE DSP ... 2. 通信記述名を指定(ディスプレイ用)  
    FROM MAP003O ... 3. 出力用論理マップ名称を指定  
    INTO MAP003I ... 4. 入力用論理マップ名称を指定
```

### (e) DISABLE 文

AP 中の最初の SEND 文または、TRANSCEIVE 文が発行されたとき、ディスプレイをオープンし、プログラム終了までディスプレイを解放(クローズ)しません。

画面表示をしていないときは、ディスプレイ端末を解放し、ほかのプログラムを表示できるようにしたい場合は DISABLE 文を使用します。

「例」のようにコーディングすると、次のオープン命令が発行された場合、ディスプレイ上に表示されているウィンドウは消え、新たにウィンドウが表示されます。



(例)

PROCEDURE DIVISION.	
⋮	
SEND DSP	
⋮	
DISABLE DSP	… 1. 通信記述名を指定(ディスプレイ用)

(f) MAPPING MODE 句の扱い (マッピングオプション)

通信記述項の MAPPING MODE 句には、画面を表示するときのオプションとしてディスプレイ用に次に示すものが指定できます。なお、指定値(「マージ」「論理マップだけ」「物理マップだけ」)の詳細については「1.5.3(1) マッピングオプション」を参照してください。

目的	MAPPING MODE 句の指定値
2 回目または 1 回目でユーザデータを表示したいとき	, または 0 マージ
同じ画面に対して一部上書をしたいとき	2 論理マップだけ
ユーザデータなしで 1 回目(初期)表示用	3 物理マップだけ

(g) STATUS KEY 句のリターンコード

STATUS KEY 句のリターンコードの意味を次に示します。

コード	意味	状態 / 要因 / 対処方法
'00000'	正常終了	データの入出力が正常に終了した。
'00008'	論理エラー	COBOL の AP に誤りがある。AP に誤りがないかを調べて修正する。
'10004' '10008'	マッピングエラー	画面データのマッピング中にエラーが発生した。STATUS KEY 句の指定がない場合、実行を中止する。STATUS KEY 句の指定がある場合、処理を続行する。画面入出力時のリターンコードの内容を調査して対処する必要がある。
'20008'	入出力エラー	画面データの入出力中にエラーが発生した。STATUS KEY 句の指定がない場合、実行を中止する。STATUS KEY 句の指定がある場合、処理を続行する。画面入出力時のリターンコードの内容を調査して対処する必要がある。

注

画面入力時のリターンコードの内容については、実行時のトラブル対処方法のヘルプを参照してください。

なお、これらのリターンコードは日立 COBOL のコンソール画面に表示されます。

## 7.2.4 CALL 文による方法

ここでは、COBOL の CALL インタフェースによってマッピングライブラリを使用する方法を説明します。

### (1) 環境部

CALL インタフェースを使用する場合の環境部 ( ENVIRONMENT DIVISION ) の定義を次に示します。

```
ENVIRONMENT          DIVISION.
CONFIGURATION        SECTION.
    SPECIAL-NAMES.
    STDCALL IS 一意名1.
EXTERNAL-PROGRAM     SECTION.
    CALL-CONVENTION.
    ' jsvwadr' IS 一意名1.
```

### 注

一意名 1 は、任意に設定してください。

環境部に上記の指定をした場合には、コンパイル時に次の指定をする必要があります。

1. コンパイラオプションに次の指定をする。

COBOL2002 の場合

'-Comp5'

'-JPN,Alnum'

COBOL85 の場合

'-X5'

'-Na'

2. リンカオプションに次の指定をする。

リンカオプション：指定しない

インポートライブラリ / ユーザ指定ライブラリ：

XMAP3 インストール先 ¥LIB¥X3MWDR32.LIB

( COBOL2002, COBOL85 Version 7 および Version 6 の場合 )

XMAP3 インストール先 ¥LIB¥V3Compatible¥X3MWDR32.LIB

( COBOL85 Version 5.0 の場合 )

AP に上記のコーディングを追加しない場合は、コンパイル時に次の指定をする必要があります。

1. エディタを使用してファイルを作成する。

ファイル内容           : jsvwadr

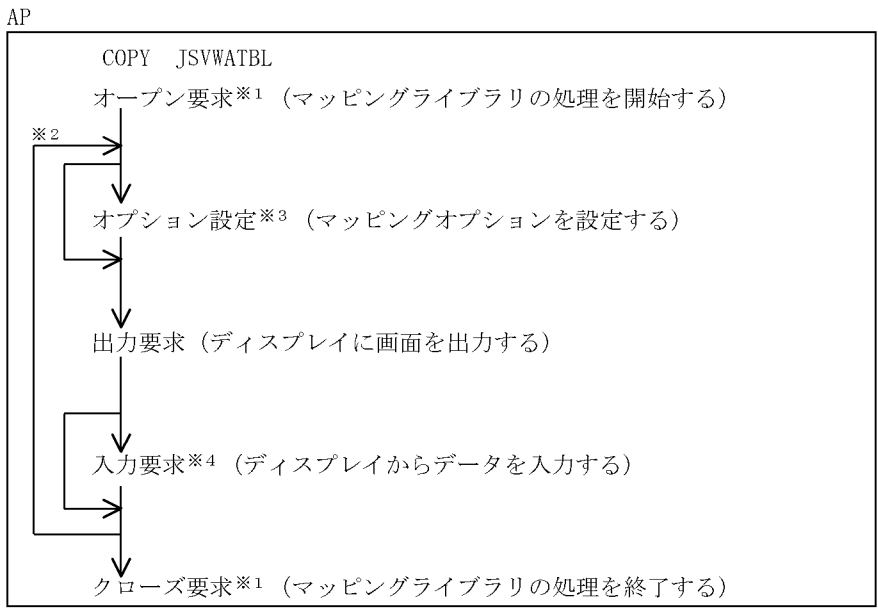
ファイル名             : XXXX.cbw ( XXXX は任意 )

- ファイルの格納場所：AP と同じフォルダ
2. コンパイラオプションに次の指定をする。  
COBOL2002 の場合
    - '-Comp5'
    - '-StdCall'
    - '-JPN,Alnum'
 COBOL85 の場合
    - '-X5'
    - '-Cp'
    - '-Na' ,
  3. リンカオプションに次の内容を指定する。  
リンカオプション：指定しない  
インポートライブラリ/ユーザ指定ライブラリ：
    - XMAP3 インストール先 ¥LIB¥X3MWDR32.LIB  
( COBOL2002 , COBOL85 Version 7 および Version 6 の場合 )
    - XMAP3 インストール先 ¥LIB¥V3Compatible¥X3MWDR32.LIB  
( COBOL85 Version 5.0 の場合 )

## (2) CALL 文の発行順序

マッピングライブラリの機能と発行順序を次の図に示します。

図 7-8 マッピングライブラリの機能と発行順序



- 注※1 ディスプレイとプリンタを使用する場合、それぞれに必要です。
- 注※2 画面数分を繰り返します。
- 注※3 マッピングオプション変更時に発行します。
- 注※4 ディスプレイからデータを入力する場合に必要です。

### (3) CALL 命令

CALL 命令の形式と発行順序を次に示します。

```
CALL ' jsvwdrv' USING XMAP-COM XMAP-REQ データ名3 データ名4.
```

#### 1. XMAP-COM

共通インタフェース領域を指定します。形式の詳細を次の表に示します。この領域は、画面を出力したい端末単位に作成し、オープンしたときの領域をクローズ要求まで引き継いで使用します。また、次に示す各要求時にマッピング、および入出力に必要な情報を代入します。

- オープン要求時 : 仮想端末名, 通信種別を指定する。
- 出力要求時 : 物理マップ名を指定する。
- 入力要求時 : 入力論理マップ長を指定する。

表 7-3 共通インタフェース領域の形式 (XMAP-COM, COBOL)

データ項目名	長さ (位置)	データ形式	指定内容 (データ名)
アイキャッチャ	4	X(4)	' * XP 'とする (XMAP-COM-ID)
リターンコード	2(4)	9(4)COMP-5	2進形式で格納 (XMAP-COM-RTN)
リターンコード 詳細 (画面入出力時の リターンコード)	2(6)	9(4)COMP-5	2進形式で格納 (XMAP-COM-RSN)
未使用	3(8)	X(3)	(00) <sub>16</sub> とする (XMAP-COM-RSV1)
id 区分	1(11)	X	必ず 'I' とする (XMAP-COM-ITYPE)
仮想端末名称	8(12)	X(8)	スタンドアロンの場合、標準の「DSP001」を指定する。C/S システムの場合、出力先の仮想端末名を指定する (XMAP-COM-TNAME)
未使用	4(20)	X(4)	(00) <sub>16</sub> とする (XMAP-COM-RSV2)
通信種別	4(24)	X(4)	ディスプレイ要求時は 'BWS 'とする (XMAP-COM-MSG)
未使用	44(28)	X(44)	(00) <sub>16</sub> とする (XMAP-COM-RSV3)
マップ名称	8(72)	X(8)	物理マップ名を左詰めでデバイス ID 付きで指定する。残りは空白 (XMAP-COM-MAPNAME)
未使用	8(80)	X(8)	(00) <sub>16</sub> とする (XMAP-COM-RSV4)
入力論理マップ 長	4(88)	S9(8) COMP-5	入力論理マップ長を指定する (XMAP-COM-INLNG)
未使用	68(92)	X(68)	(00) <sub>16</sub> とする (XMAP-COM-RSV5)

## 2. XMAP-REQ

要求インタフェース領域を指定します。形式を次の表に示します。この領域は、必ず指定してください。この領域を指定した内容によって、要求種別が決まります。

表 7-4 要求インタフェース領域の形式 (XMAP-REQ, COBOL)

データ項目名	長さ (位置)	データ形式	指定内容 (データ名)
要求種別	4	X(4)	'OPEN': オープン要求 'CLOS': クローズ要求 'SEND': 出力要求 'RECV': 入力要求 'MDO ' : オプション設定要求 (XMAP-REQ-TYPE)
RECEIVE オプション	1(4)	X	'F': 入力要求時 ' ': 上記以外 (XMAP-REQ-OPT1)

## 7. AP のコーディング方法

データ項目名	長さ (位置)	データ形式	指定内容(データ名)
SEND オプション	1(5)	X	'1': 出力要求時 ' ': 上記以外 (XMAP-REQ-OPT2)
未使用	2(6)	X(2)	' ' とする (XMAP-REQ-RSV)

### 3. データ名 3

- 出力要求時 : 出力論理マップを指定する。
- 入力要求時 : 入力論理マップを指定する。
- オープン要求時 : オープンインタフェース領域を指定する。

形式の詳細を次の表に示します。この領域は必ず指定してください。

表 7-5 オープンインタフェース領域の形式 (XMAP-OPN, COBOL)

データ項目名	長さ (位置)	データ形式	指定内容(データ名)
データ有無 コード 使用選択	1	X	'1' : データ有無コードを指定する その他 : データ有無コードを指定しない (標準値 「1F」を仮定) (XMAP-OPN-DCODE-SET)
データ有無 コード	1(1)	X	データ有無コードを指定する場合, 2桁の16進数 (00 ~ FF) で指定する (XMAP-OPN-DCODE)
未使用	2(2)	X(2)	' ' とする (XMAP-OPN-RSV)

なお, データ有無コード使用選択項目でデータ有無コードを無効にする指定をした場合は, 標準値「1F」が仮定されます。

### 4. データ名 4

オプション設定要求時は, マッピングインタフェース領域を指定します。形式の詳細を次の表に示します。なお, マッピングオプションを指定しない場合は, 「0」を仮定します。

表 7-6 マッピングインタフェース領域の形式 (XMAP-MDO, COBOL)

データ項目名	長さ (位置)	データ形式	指定内容(データ名)
マッピング オプション 大分類	4	9(8)COMP-5	XMAP-MDO-SFLD の内容を代入する (XMAP-MDO-OPT1)
マッピング オプション 小分類 (MAPPING MODE 句相当)	4(4)	9(8)COMP-5	XMAP-MDO-MAPFLD XMAP-MDO-PHFLD XMAP-MDO-LOGFLD (XMAP-MDO-OPT2)

## 注

データ名とマッピングオプションは次のように対応しています。なお、マッピングオプションの詳細については「1.5.3(1) マッピングオプション」を参照してください。

- ・MAPFLD：マージ
- ・PHFLD：物理マップだけ
- ・LOGFLD：論理マップだけ

## インタフェース領域の取り込み方法

XMAP3 が提供しているインタフェース領域 (JSVWATBL.cbl) を AP に取り込む場合、COBOL の WORKING-STORAGE SECTION、または LINKAGE SECTION に COPY 文を指定します。また、コンパイルする前に JSVWATBL.cbl のフォルダ名を CBLLIB に設定しておく必要があります。

インタフェース領域を次の図に示します。

図 7-9 インタフェース領域

```

*
*COMMON INTERFACE AREA
01 XMAP-COM.
  03 XMAP-COM-ID          PIC X(4).
  03 XMAP-COM-RTN        PIC 9(4) COMP-5.
  03 XMAP-COM-RSN        PIC 9(4) COMP-5.
  03 XMAP-COM-RSV1       PIC X(3).
  03 XMAP-COM-ITYPE      PIC X.
  03 XMAP-COM-TNAME      PIC X(8).
  03 XMAP-COM-RSV2       PIC X(4).
  03 XMAP-COM-MSG        PIC X(4).
  03 XMAP-COM-RSV3       PIC X(44).
  03 XMAP-COM-MAPNAME    PIC X(8).
  03 XMAP-COM-RSV4       PIC X(8).
  03 XMAP-COM-INLNG      PIC S9(8) COMP-5.
  03 XMAP-COM-RSV5       PIC X(68).
*
* REQUEST INTERFACE AREA
01 XMAP-REQ.
  03 XMAP-REQ-TYPE       PIC X(4).
  03 XMAP-REQ-OPT1       PIC X.
  03 XMAP-REQ-OPT2       PIC X.
  03 XMAP-REQ-RSV        PIC X(2).
*
* OPEN INTERFACE AREA
01 XMAP-OPN.
  03 XMAP-OPN-DCODE-SET  PIC X.
  03 XMAP-OPN-DCODE      PIC X.
  03 XMAP-OPN-RSV        PIC X(2).
*
* MAPPING OPTION INTERFACE AREA
01 XMAP-MDO.
  03 XMAP-MDO-OPT1       PIC 9(8) COMP-5.
  03 XMAP-MDO-OPT2       PIC 9(8) COMP-5.
*
* MAPPING OPTION VALUE
77 XMAP-MDO-SFLD        PIC 9(8) COMP VALUE 3.
77 XMAP-MDO-MAPFLD      PIC 9(8) COMP VALUE 13.
77 XMAP-MDO-PHFLD       PIC 9(8) COMP VALUE 14.
77 XMAP-MDO-LOGFLD      PIC 9(8) COMP VALUE 15.

```

### オープン要求

CALL 命令を使用する場合、画面を表示したい端末に付けた仮想端末名単位にオープン要求をします。複数の仮想端末（ディスプレイ）をオープン要求する場合、インタフェース領域は、仮想端末ごとに準備してください。一つの AP（プロセス）内で同時にオープンできる端末数の上限は 15 個です。



(例)

```

PROCEDURE DIVISION.
.
.
MOVE 0    TO XMAP-COM.           ...1. 共通エリアをクリアする
MOVE SPACE TO XMAP-REQ.         ...2. 要求エリアをクリアする
*
MOVE 'OPEN' TO XMAP-REQ-TYPE.   ...3. オープン要求を代入する
MOVE '*XP△' TO XMAP-COM-ID.    ...4. IDを代入する
MOVE 'I'   TO XMAP-COM-ITYPE.   ...5. ID区分を代入する
MOVE 'BWS△' TO XMAP-COM-MSG.   ...6. ディスプレイの通信種別を代入する
MOVE 'DSP001△△' TO XMAP-COM-TNAME. ...7. 仮想端末名を代入する

MOVE '1'   TO XMAP-OPN-DCODE-SET.
MOVE X'1F' TO XMAP-OPN-DCODE.   } ...8. データ有無コードを代入する
MOVE SPACE TO XMAP-OPN-RSV.
CALL 'jsvwadrv' USING XMAP-COM
      XMAP-REQ           ...9. 共通インタフェース領域
      XMAP-OPN           ...10. 要求インタフェース領域
      BY VALUE 0.         ...11. オープンインタフェース領域

```

## オプション設定要求

仮想端末へ画面表示をする場合、マッピングオプションを指定できます。オープン要求で使用した共通インタフェースを引き継いで使用します。

(例)

```

MOVE 'MDO△' TO XMAP-REQ-TYPE.   ...1. オプション設定要求を代入する
MOVE '△'   TO XMAP-REQ-OPT1.    ...2. 必ず空白を代入する
MOVE '△'   TO XMAP-REQ-OPT2.    ...3. 必ず空白を代入する
MOVE XMAP-MDO-SFLD TO XMAP-MDO-OPT1. ...4. マッピングオプション
      大分類を代入する
MOVE XMAP-MDO-LOGFLD TO XMAP-MDO-OPT2. ...5. マッピングオプション
      小分類を代入する
CALL 'jsvwadrv' USING XMAP-COM   ...6. 共通インタフェース領域
      XMAP-REQ                   ...7. 要求インタフェース領域
      BY VALUE 1                 ...8. 必ず1を指定する
      BY REFERENCE XMAP-MDO.     ...9. マッピングオプションインタフェース領域

```

## 出力要求

オープン要求した仮想端末へ画面を表示します。オープン要求で使用した共通インタフェース領域を引き継いで使用します。

## 7. AP のコーディング方法

(例)

```
MOVE 'SEND' TO XMAP-REQ-TYPE. ...1. 出力要求を代入する
MOVE '△' TO XMAP-REQ-OPT1. ...2. 出力要求時に必ず空白を代入する
MOVE '1' TO XMAP-REQ-OPT2. ...3. 出力要求時に必ず'1'を代入する
MOVE 'MAP003NC' TO XMAP-COM-MAPNAME ...4. 物理マップ名を代入する
MOVE データ TO MAP0030. ...5. 画面表示したいデータを
      出力論理マップへ代入する
CALL 'jswadrv' USING XMAP-COM ...6. 共通インタフェース領域
      XMAP-REQ ...7. 要求インタフェース領域
      MAP0030 ...8. 出力論理マップ
      BY VALUE 0.
```

### 入力要求

入力要求によって、画面から入力した情報を AP へ代入します。画面表示要求で使  
用したインタフェース領域を引き継いで使用します。

(例)

```
MOVE 'RECV' TO XMAP-REQ-TYPE. ...1. 入力要求を代入する
MOVE 'F' TO XMAP-REQ-OPT1. ...2. 入力要求時に必ず'F'を代入する
MOVE '△' TO XMAP-REQ-OPT2. ...3. 入力要求時に必ず空白を代入する
MOVE map003S TO XMAP-COM-INLNG. ...4. 入力論理マップ長を代入する
*
CALL 'jswadrv' USING XMAP-COM ...5. 共通インタフェース
      XMAP-REQ ...6. 要求インタフェース
      MAP003I ...7. 入力論理マップ
      BY VALUE 0.
IF MAP003-INC1 ...8. 入力論理マップのデータを判定する
```

### クローズ要求

仮想端末をクローズします。オープン要求、出力要求、および入力要求で使用した  
インタフェースを引き継いで使用します。

(例)

```
MOVE 'CLOS' TO XMAP-REQ-TYPE. ...1. クローズ要求を代入する
MOVE '△' TO XMAP-REQ-OPT1. ...2. 必ず空白を代入する
CALL 'jswadrv' USING XMAP-COM ...3. 共通インタフェース領域
      XMAP-REQ ...4. 要求インタフェース領域
      BY VALUE 0 0.
```

## 7.2.5 COBOL 開発マネージャを使用したコンパイルと実行のポイント

COBOL 開発マネージャを使用したコンパイルと実行のポイントについて説明します。

## (1) COBOL 開発マネージャの概要

COBOL 開発マネージャは、日立 COBOL から提供されている COBOL プログラムの統合的な開発環境です。COBOL で AP を開発するときに必要な COBOL ソースや登録集原文などの資産をその依存関係に従って管理し、コンパイルなどの作業を自動化します。

COBOL 開発マネージャを使用する場合、次の手順で AP を作成します。

1. プロジェクトの作成
2. 資産の登録, 定義
3. ビルド, リビルド

次に、それぞれの手順について説明します。

### (a) プロジェクトの作成

COBOL 開発マネージャでは、一つの .EXE または .DLL ファイルの開発単位を「プロジェクト」と呼んでいます。COBOL 開発マネージャを使用して AP を作成する場合、このプロジェクトを定義する必要があります。

### (b) 資産の登録, 定義

COBOL 開発マネージャでは、プロジェクトで作成される実行のファイル (.EXE, または .DLL) やそれを構成する要素を「資産」と呼んでいます。また、COBOL 開発マネージャでは、その依存関係を定義する必要があります。

資産の依存関係を定義すると、COBOL 開発マネージャではどのように表されるかを次の図に示します。

図 7-10 COBOL2002, COBOL85 Version 7 および Version 6 の場合

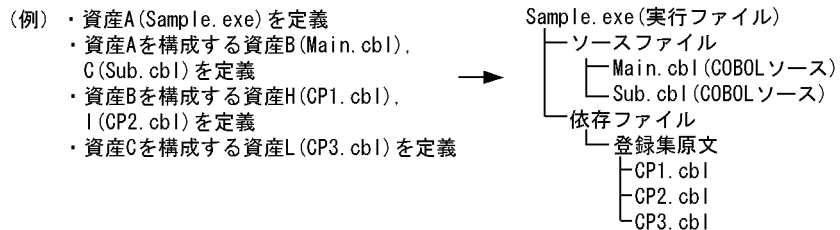
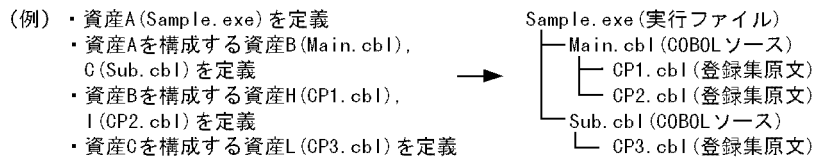


図 7-11 COBOL85 Version 5.0 の場合



## 7. AP のコーディング方法

### (c) ビルド, リビルド

COBOL 開発マネージャでは, (b) で定義した資産の依存関係に基づいてコンパイルとリンケージをします。

その方法として, 各資産の依存関係とタイムスタンプ (ファイル作成 / 修正日時) の前後関係に基づいてコンパイルとリンケージをする「ビルド」があります。例えば, 「実行ファイルの作成 / 修正日時」より実行ファイルが取り込んでいる「COBOL ソースの作成 / 修正日時」の方が新しい場合にコンパイルとリンケージが実行されます。

また, タイムスタンプには関係なくコンパイルとリンケージをする「リビルド」もあります。

### (2) コンパイル時のポイント

#### (a) XMAP3 と連携するためのコンパイラオプションを指定する

日立 COBOL を使用したとき, 指定が必要なコンパイルオプションを次に示します。

##### COBOL2002 の場合

- Comp5 : COBOL プログラム中の COMP-5 を利用できるようにするオプション。
- JPN,Alnum : 論理マップ内で日本語項目を扱えるようにするオプション。
- StdCall : CALL インタフェースを使用し, 環境部に 'jsvwadrv' の指定をしない場合に必要なオプション。

##### COBOL85 の場合

- X5 : COBOL プログラム中の COMP-5 を利用できるようにするオプション。
- Na : 論理マップ内で日本語項目を扱えるようにするオプション。
- Cp : CALL インタフェースを使用し, 環境部に 'jsvwadrv' の指定をしない場合に必要なオプション。

#### (b) AP 間でオープンを引き継ぐ

SEND / RECEIVE / TRANSCEIVE を使用して AP を分割してコンパイルする場合, 共通画面を表示する個所を別 DLL にします。AP を分割してコンパイルするときは, コンパイル単位で XMAP3 の画面オープン・クローズ要求が毎回発行されないようにします。日立 COBOL の実行支援の環境変数で, 「CBLTERMSHAR=YES」を指定します。「CBLTERMSHAR=YES」は, SEND / RECEIVE / TRANSCEIVE インタフェースのときだけ有効です。CALL インタフェースでは無効になります。

CALL インタフェースの場合, または「CBLTERMSHAR」をサポートしていないバージョンの日立 COBOL を使用している場合で, AP 間のオープン引き継ぎをするときは, ユーザが, XMAP3 インタフェースエリアの情報を引き継ぐようなコーディングをする必要があります。サンプルコーディングとして, AP パターンの GENDSP02 および GENDSP03 が利用できます。

(c) 登録集原文，動的变化テーブル (X3MODTBL) を確認する

- AP の COPY 文に，論理マップ (マップ生成時に付けられた名称) や動的变化テーブル (X3MODTBL) が間違っ指定されていないか確認してください。
- 論理マップや動的变化テーブルがフォルダ中に用意されているか確認してください。ソースプログラムと登録集原文が同じフォルダで管理されている場合は，論理マップや動的变化テーブルが同一フォルダに格納されていないことが考えられます。また，ソースプログラムと登録集原文が別のフォルダで管理されている場合は，環境変数に指定した登録集原文のフォルダに誤りがあったり，指定したフォルダに論理マップや動的变化テーブルが格納されていなかったりすることが考えられます。動的变化テーブルおよび XMAP3 インタフェース領域テーブルは，「XMAP3 インストール先 ¥INCLUDE」の下に格納されていますので，必要なフォルダにコピーするか，または登録集原文が格納されているフォルダを環境変数 CBLLIB に指定して使用してください。
- データ名または変数名に対し，不当な文字を AP で指定していないか確認してください。
- CALL インタフェースで XMAP3 を利用する場合には，XMAP3 インタフェース領域テーブル (JSVWATBL) が AP の COPY 文に指定されているか確認してください。

### (3) リンケージ時のポイント

(a) XMAP3 使用時のリンケージオプションを指定する

CALL インタフェースで XMAP3 を利用する場合，インポートライブラリ / ユーザ作成ライブラリに，次のように指定します。

COBOL2002，COBOL85 Version 7 および Version 6 の場合  
XMAP3 インストール先 ¥LIB¥X3MWDR32.LIB

COBOL85 Version 5.0 の場合  
XMAP3 インストール先 ¥LIB¥V3Compatible¥X3MWDR32.LIB

(b) XMAP3 のライブラリを設定する

コンパイル環境 (CALL 文のインタフェースを使用する AP の場合は，リンケージ環境) の PC には，XMAP3 の開発環境をインストールしておく必要があります。

### (4) 実行時のポイント

AP を実行する前に，物理マップやグラフィックデータ (グラフィックを使用する場合) を実行可能ファイルと同じフォルダに格納してください。

また，マップパスやグラフィックパスを指定する方法もあります。マップパスおよびグラフィックパスは，表示・印刷セットアップで指定します。表示・印刷セットアップについては，マニュアル「XMAP3 開発・実行ガイド」を参照してください。

AP を分割してコンパイルした場合は，実行支援の環境変数が必要になります。「7.1.2 AP 間でオープンを引き継ぐ場合」を参照してください。

## 7.2.6 COBOL 開発マネージャでの XMAP3 の利用方法

COBOL 開発マネージャと XMAP3 との両方をインストールしている場合、COBOL 開発マネージャから XMAP3 の機能呼び出せます。XMAP3 の資産を登録、定義すれば、ビルド、またはリビルドによってコンパイルとリンケージをして AP を作成できます。

また、AP 作成時にソースプログラムや登録集原文を格納したりするフォルダを作成しておいてください。

次に、COBOL 開発マネージャから XMAP3 を利用する方法について説明します。

### (1) COBOL 開発マネージャ上での資産の登録、定義

COBOL 開発マネージャ上では、XMAP3 の資産を扱う場合、XMAP3 でのファイル名の規則に従います。また、XMAP3 の「マップ名」を基に作成します。

XMAP3 の資産を COBOL 開発マネージャで利用するときのファイル名称の規則を、次の表に示します。

表 7-7 XMAP3 の資産を COBOL 開発マネージャで利用するときのファイル名称の規則

XMAP3 の資産	COBOL 開発マネージャでの名称	COBOL 開発マネージャで付けるファイル名称	内容
マップ定義	マップ定義	マップ名 .imp	ユーザが定義した画面の情報を格納している。定義を修正するときにはこのファイル名を指定。
論理マップ	登録集原文	マップ名 + O.cbl	AP で取り込む出力情報の登録原文集。
		マップ名 + I.cbl	AP で取り込む入力情報の登録原文集。
物理マップ	-	マップ名 + XX .pmp	ユーザが定義した画面のうち、AP ではアクセスしない、固定部分の情報。COBOL 開発マネージャの管理対象外。

(凡例)

- : 該当しない。

注

XX は、デバイス ID を示します。

また、COBOL 開発マネージャから XMAP3 の資産を登録、定義した場合、どのように表されるかを次の図に示します。

図 7-12 COBOL2002 , COBOL85 Version 7 および Version 6 の場合

```

ソースプログラム
├── ユーザプログラム (COBOLソース) ←XMAP3の機能で定義した画面を利用するAP
└── マップ名. imp (マップ定義ファイル)
依存ファイル
├── マップ名+1. cbl (登録集原文)
└── マップ名+0. cbl (登録集原文)

```

図 7-13 COBOL85 Version 5.0 の場合

```

ユーザプログラム (COBOLソース) ←XMAP3の機能で定義した画面を利用するAP
├── マップ名+0. cbl (登録集原文)
│   └── マップ名. imp (マップ定義ファイル)
└── マップ名+1. cbl (登録集原文)
    └── マップ名. imp (マップ定義ファイル)

```

これを基にして、XMAP3 で作成した 2 種類の画面を COBOL 開発マネージャで利用して AP を作成する場合の例を次の図に示します。

図 7-14 COBOL2002 , COBOL85 Version 7 および Version 6 の場合

(例) XMAP3で作成した画面1 (マップ名:MENU) と画面2 (マップ名:MENU) を利用してAPを作成する場合

```

APP. exe (実行ファイル)
├── ソースファイル
│   ├── Samp le. cbl (COBOLソース)          ←1. AP
│   ├── MENU. imp (マップ定義ファイル)     ←2. 画面 1 用 画面定義ファイル
│   └── MENU. imp (マップ定義ファイル)     ←3. 画面 2 用 画面定義ファイル
└── 依存ファイル
    ├── 登録集原文
    │   ├── MENU1. cbl (登録集原文)        ←4. 画面 1 用 入力論理マップ
    │   ├── MENU0. cbl (登録集原文)        ←5. 画面 1 用 出力論理マップ
    │   ├── MENU1. cbl (登録集原文)        ←6. 画面 2 用 入力論理マップ
    │   └── MENU0. cbl (登録集原文)        ←7. 画面 2 用 出力論理マップ

```

(凡例) —: 画面 1 のマップ名  
 ==: 画面 2 のマップ名

図 7-15 COBOL85 Version 5.0 の場合

(例) XMAP3で作成した画面1(マップ名:MENU)と画面2(マップ名:MENU)を利用してAPを作成する場合

APP. exe (実行ファイル)		
Sample. cbl (COBOLソース)		←1. AP
└─MENU1. cbl (登録集原文)		←2. 画面 1 用 入力論理マップ
└─┬─MENU. imp (マップ定義ファイル)		←3. 画面 1 用 画面定義ファイル
└─MENU0. cbl (登録集原文)		←4. 画面 1 用 出力論理マップ
└─┬─MENU. imp (マップ定義ファイル)		←5. 画面 1 用 画面定義ファイル
└─MENU1. cbl (登録集原文)		←6. 画面 2 用 入力論理マップ
└─┬─MENU. imp (マップ定義ファイル)		←7. 画面 2 用 画面定義ファイル
└─MENU0. cbl (登録集原文)		←8. 画面 2 用 出力論理マップ
└─┬─MENU. imp (マップ定義ファイル)		←9. 画面 2 用 画面定義ファイル

(凡例) — : 画面 1 のマップ名  
 == : 画面 2 のマップ名

### 《注意》

- ファイル名は、表 7-7 の規則に従って定義する必要があります。
- ファイル名は、XMAP3 のマップ名を使用して作成してください。XMAP3 のファイル名を使用しないと、上記に示す資産定義と一致しくなくなります。
- XMAP3 で入出力画面の定義をする場合、3.、5. または 7.、9. は同じファイル名で定義してください。

このように資産を登録、定義しておけば、定義ファイルに変更があった場合、ビルドによってコンパイル、リンケージされ、自動的に登録集原文(論理マップ)が生成し直されます。

## (2) マップ定義ファイルの作成と修正

マップ定義ファイルを作成、または修正したい場合、次に示すどちらかの操作をすることでXMAP3の画面定義ができます。

COBOL 開発マネージャ上で、マップ定義ファイル (xxx.imp) をダブルクリックします。

COBOL 開発マネージャ上で、マップ定義ファイル (xxx.imp) をクリックして、[ ツール ] メニューから [ マップ定義 ] を選びます (COBOL85 Version 5.0 の場合だけ)。

## (3) マップ生成

COBOL 開発マネージャのビルド、またはリビルドを使用しないで単独にマップ生成をする場合、次の操作をすることでマップ生成ができます。生成される論理マップや物理マップのファイル名については、表 7-7 を参照してください。

マップ生成時に使用するリトルエンディアン、およびビッグエンディアンは、オプションで指定してください。



## 《操作方法》

COBOL2002, COBOL85 Version 7 および Version 6 の場合

- 特定のマップ定義ファイルからマップ生成をするときにだけ有効とする場合  
COBOL 開発マネージャ上で, マップ定義ファイル (xxx.imp) をクリックして, [ファイル] - [ファイルの設定] を選択し, 必要なコンパイルオプションを指定します。
- すべてのマップ定義ファイルからマップ生成をするときに有効とする場合  
COBOL 開発マネージャ上で, [プロジェクト] - [プロジェクトの設定] を選択し, 「製品連携」タブ で必要なコンパイルオプションを指定します。

## 注

COBOL2002 の場合, 「最適化」タブとなります。

COBOL85 Version 5.0 の場合

- 特定のマップ定義ファイルからマップ生成をするときにだけ有効とする場合  
COBOL 開発マネージャ上で, マップ定義ファイル (xxx.imp) をクリックして, [プロジェクト] - [単一生成オプション] を選択し, 必要なコンパイルオプションを指定します。
- すべてのマップ定義ファイルからマップ生成をするときに有効とする場合  
COBOL 開発マネージャ上で, [オプション] - [マップ生成] を選択します。

## 《設定形式》

[ -BigEndian, Bin ] ( COBOL2002 の場合 )

[ -Bb ] ( COBOL85 の場合 )

## 《説明》

-BigEndian, Bin または -Bb は, ビッグエンディアンを指定することを示します。指定がない場合, リトルエンディアンが仮定されます。

## 7.3 C 言語の画面入出力命令

ここでは、XMAP3 を使用したシステムでの C 言語の AP のコーディング方法について説明します。コーディングをする前に、次の準備をしてください。従来の C 言語にないクラスなどの概念を持った C++ 言語で AP を作成する場合には、「7.4 汎用 AP インタフェース」を参照してください。

### 1. AP を格納するためのディレクトリを用意する

ソースプログラムやコンパイル後に生成される作業ファイルを格納するディレクトリを用意します。開発環境に合わせてディレクトリを分類し、作成位置や名称を決めます。

### 2. XMAP3 の標準提供ファイルから、必要なファイルをコピーする

XMAP3 では、AP の作成に必要なヘッダファイルを標準提供しています。1. で作成したディレクトリにコピーして使用することをお勧めします。このとき、コーディングのひな型である AP パターンもコピーしておきます。

#### 注

標準提供のヘッダファイルを次に示します。

- 動的変更テーブル (X3MODTBL.H)

動的変更用の修飾名と、出力論理マップの初期化に使用する定数が指定されています。

- インタフェース領域 (JSVWATBL.H)

jsvwadrv 関数で帳票を送信するときにパラメタとして使用するインタフェース領域が指定されています。

### 7.3.1 仮想端末の自動割当て

C/S システム構成の環境で、サーバ上の AP から複数のクライアント PC へ画面表示をする場合、仮想端末の自動割当て機能を利用できます。この機能は、C/S システム構成の場合だけ利用できます。

仮想端末の自動割当て機能は、サーバ AP が各クライアント PC に対応した仮想端末名を意識することなく一つの仮想端末名だけを意識していれば、各クライアント PC にある XMAP3 の表示サービスが起動されたタイミングで、AP から送信された情報を画面表示するようにできます。仮想端末の自動割当て機能を利用するときは、サーバ側の C/S セットアップで、AP で指定した自動割当て用の仮想端末名を設定します。

仮想端末の自動割当ての詳細については、マニュアル「XMAP3 開発・実行ガイド」を参照してください。

### 7.3.2 論理マップの取り込み方法 (C 言語)

AP 中に論理マップを取り込むには、`#include` 指示語を使用します。

AP に論理マップ `MAP003O.h`、および `MAP003I.h` を取り込む例を次に示します。

```
#include "MAP003O.h" ... 出力論理マップの取り込み
#include "MAP003I.h" ... 入力論理マップの取り込み
```

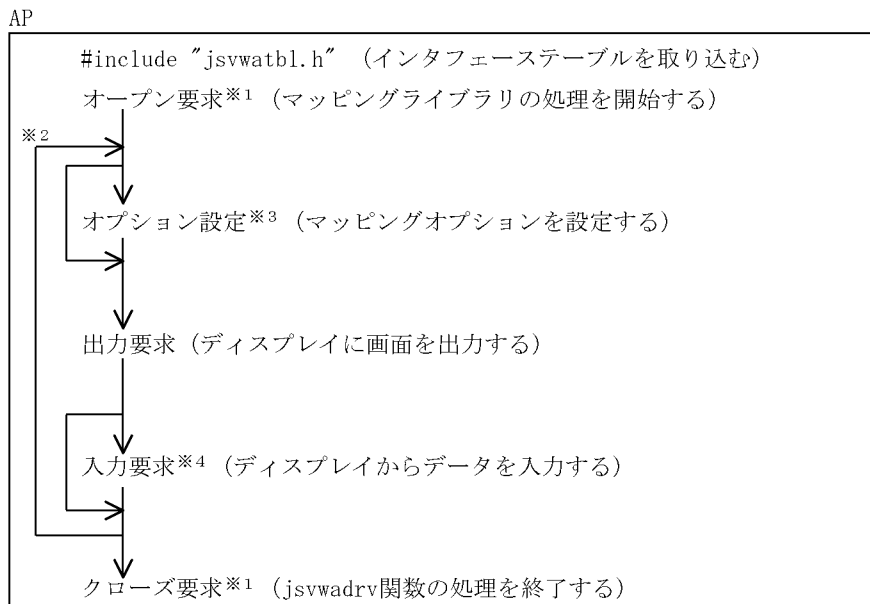
`#include` 指示語によって AP に実領域が取られます。

### 7.3.3 画面表示命令 (C 言語)

XMAP3 を呼び出すための C 言語インターフェースは、次に示す `jsvwadv` 関数を提供します。

#### (1) `jsvwadv` 関数の発行順序

`jsvwadv` 関数の機能と発行順序を次に示します。



注※1 必ず指定します。

注※2 画面数分を繰り返します。

注※3 マッピングオプション変更時に発行します。

注※4 ディスプレイからデータを入力する場合に必要です。

## (2) jsvwadrv 関数

jsvwadrv 関数の形式を次に示します。

```
long APIENTRY jsvwadrv ( XMAP_COMアドレス, XMAP_REQアドレス,
                        パラメタ3, パラメタ4 )
```

## 1. XMAP\_COM アドレス

共通インタフェース領域のアドレスを指定します。形式の詳細を次の表に示します。この領域は、画面を出力したい端末単位に作成し、オープンしたときの領域をクローズ要求まで引き継いで使用します。また、次に示す各要求時にマッピング、および入出力に必要な情報を代入します。

- オープン時 : 仮想端末名、通信種別を指定する。
- 出力要求時 : 物理マップ名称を指定する。
- 入力要求時 : 入力論理マップ長を指定する

表 7-8 共通インタフェース領域の形式 (XMAP\_COM, C 言語)

項目	長さ (位置)	データ形式	格納内容	データ名
アイキャッチャ	4	char[4]	' * XP ' とする	xmap_com_id
リターンコード 1	2(4)	unsigned short	2 進形式で XMAP3 が格納	xmap_com_rtn
画面入出力時のリターンコード 1	2(6)	unsigned short	2 進形式で XMAP3 が格納	xmap_com_rsn
未使用	3(8)	char[3]	(00) <sub>16</sub> とする	xmap_com_rsv1
id 区分	1(11)	char	必ず 'I' とする	xmap_com_itype
仮想端末名	8(12)	char[8]	出力先の仮想端末名を指定する (標準は 'DSP001 ' を指定する)	xmap_com_tname
未使用	4(20)	char[4]	(00) <sub>16</sub> とする	xmap_com_rsv2
通信種別	4(24)	char[4]	'BWS ' とする	xmap_com_msg
未使用	44(28)	char[44]	(00) <sub>16</sub> とする	xmap_com_rsv3
マップ名称	8(72)	char[8]	物理マップ名を左詰めでデバイス ID 付きで指定する。残りは空白 <sup>2</sup> 。	xmap_com_mapname
未使用	8(80)	char[8]	(00) <sub>16</sub> とする	xmap_com_rsv4
入力論理マップ長	4(88)	long	入力論理マップ長を指定する	xmap_com_inlng
未使用	68(92)	char[68]	(00) <sub>16</sub> とする	xmap_com_rsv5

## 注 1

論理マップ名については、「7.2.3(1)(a) 通信記述項」を参照してください。

## 注 2

リターンコードについては「7.2.3(1)(g) STATUS KEY 句のリターンコード」を参照してください。

## 2. XMAP\_REQ アドレス

要求インタフェース領域のアドレスを指定します。形式の詳細を次の表に示します。この領域は、必ず指定してください。この領域で指定した内容によって要求種別が決まります。

表 7-9 要求インタフェース領域の形式 (XMAP\_REQ, C 言語)

項目	長さ (位置)	データ形式	指定内容	データ名
要求種別	4	char[4]	'OPEN': オープン要求 'CLOS': クローズ要求 'RECV': 入力要求 'SEND': 出力要求 'MDO ': オプション設定要求	xmap_req_type
RECEIVE オプション	1(4)	char	'F': 入力要求時 ' ': 上記以外	xmap_req_opt1
SEND オプション	1(5)	char	'1': 出力要求時 ' ': 上記以外	xmap_req_opt2
未使用	2(6)	char[2]	' ' とする	xmap_req_rsv

## 3. パラメタ 3

- 出力要求時：出力論理マップのアドレスを指定する。
- 入力要求時：入力論理マップのアドレスを指定する。
- クローズ要求時：0L を指定する。
- オプション設定要求時：1L を指定する。
- オープン要求時：オープンインタフェース領域のアドレスを必ず指定する。

オープンインタフェース領域の詳細を次の表に示します。

表 7-10 オープンインタフェース領域の形式 (XMAP\_OPN, C 言語)

項目	長さ (位置)	データ形式	指定内容 (データ名)	データ名
データ有無 コード使用選 択	1	char	'1': データ有無コードを指定する その他：データ有無コードを指定 しない (標準値「1F」を仮定)	xmap_opn_dcode _set

## 7. AP のコーディング方法

項目	長さ (位置)	データ形式	指定内容(データ名)	データ名
データ有無 コード	1(1)	unsigned char	データ有無コードを指定する場 合, 2桁の16進数(00 ~ FF)で 指定する	xmap_opn_dcode
未使用	2(2)	char[2]	' 'とする	xmap_opn_rsv

なお, データ有無コード使用選択項目でデータ有無コードを無効にする指定をした場合は, 標準値「1F」が仮定されます。

### 4. パラメタ 4

オプション設定要求時は, マッピングインタフェース領域を指定します。形式の詳細を次の表に示します。なお, マッピングオプションを指定しない場合には, 0 を指定します。

表 7-11 マッピングインタフェース領域の形式 (XMAP\_MDO, C 言語)

項目	長さ (位置)	データ形式	指定内容(データ名)	データ名
マッピングオ プション大分 類	4	unsigned long	XMAP_MDO_SFLD の内容を セットする	xmap_mdo_opt1
マッピングオ プション小分 類	4(4)	unsigned long	XMAP_MDO_MAPFLD XMAP_MDO_PHFLD XMAP_MDO_LOGFLD	xmap_mdo_opt2

### 注

データ名とマッピングオプションは次のように対応しています。

XMAP\_MDO\_MAPFLD : マージ

XMAP\_MDO\_PHFLD : 物理マップだけ

XMAP\_MDO\_LOGFLD : 論理マップだけ

なお, マッピングオプションの指定値の詳細については「1.5.3(1) マッピングオプション」を参照してください。

### (a) リターン情報

共通インタフェース領域の xmap\_com\_rtn にリターン値が設定されます。リターン値を次に示します。

0 : 正常終了

4, 8 : 異常終了 (詳細については, 共通インタフェース領域の xmap\_com\_rsn に設定される)

12 : パラメタ不正

## (b) インタフェーステーブルの取り込み方法

XMAP3 が提供しているインタフェーステーブル (jsvwatbl.h) を AP に取り込む場合、`#include` 指示語を使用します。

```
#include "jsvwatbl.h" ... インタフェーステーブルの取り込み
```

ただし、jsvwatbl.h 中のインタフェーステーブルは、構造体の形式などを定義しており、`#include` 指示語によって AP に実領域は確保されません。このため、AP 中でインタフェース領域を用意する必要があります。例を次に示します。

(例)

```
#include "jsvwatbl.h"

XMAP_COM com:    /* 共通インタフェース領域      */ * /
XMAP_REQ req:    /* 要求インタフェース領域      */ * /
XMAP_OPN opn:    /* オープンインタフェース領域  */ * /
XMAP_MDO mdo:    /* マッピングインタフェース領域 */ * /
```

インタフェーステーブルを次の図に示します。

図 7-16 インタフェーステーブル

```

/**      COMMON INTERFACE AREA      *****/
typedef struct {
    char        xmap_com_id[4];
    unsigned short  xmap_com_rtn;
    unsigned short  xmap_com_rsn;
    char        xmap_com_rsv1[3];
    char        xmap_com_itype;
    char        xmap_com_tname[8];
    char        xmap_com_rsv2[4];
    char        xmap_com_msg[4];
    char        xmap_com_rsv3[44];
    char        xmap_com_mapname[8];
    char        xmap_com_rsv4[8];
    long        xmap_com_inlng;
    char        xmap_com_rsv5[68];
}    XMAP_COM;

/**      REQUEST INTERFACE AREA      *****/
typedef struct {
    char        xmap_req_type[4];
    char        xmap_req_opt1;
    char        xmap_req_opt2;
    char        xmap_req_rsv[2];
}    XMAP_REQ;

/**      OPEN INTERFACE AREA      *****/
typedef struct {
    char        xmap_opn_dcode_set;
    unsigned char  xmap_opn_dcode;
    char        xmap_opn_rsv[2];
}    XMAP_OPN;

/**      MAPPING OPTION INTERFACE AREA      *****/
typedef struct {
    unsigned long  xmap_mdo_opt1;
    unsigned long  xmap_mdo_opt2;
}    XMAP_MDO;

/**      MAPPING OPTION VALUE      *****/
#define        XMAP_MDO_SFLD        3
#define        XMAP_MDO_MAPFLD        13
#define        XMAP_MDO_PHFLD        14
#define        XMAP_MDO_LOGFLD        15

```

## (c) オープン要求

jsvwadvr 関数を使用する場合、画面を表示したい端末に付けた仮想端末名単位にオープン要求をします。複数の仮想端末をオープン要求する場合、インタフェース領域は仮想端末ごとに用意する必要があります。一つの AP (プロセス) 内で同時にオープンできる端末数の上限は 15 個です。





## 7. AP のコーディング方法

(例)

```
memcpy(req.xmap_req_type, "SEND", sizeof(req.xmap_req_type)); ...1. 出力要求の代入
req.xmap_req_opt1='△'; ...2. 必ず空白を代入する
req.xmap_req_opt2='1'; ...3. 出力要求時1を代入する
memcpy(com.xmap_com_mapname, "MAP003NC", sizeof(com.xmap_com_mapname));
...4. 物理マップ名を代入する

jsvwdrv(&com, &req, &MAP0030, 0L);
      ↑      ↑      ↑
      出力論理マップのアドレス
      要求インタフェース領域アドレス
      共通インタフェース領域アドレス
```

注

jsvwdrv 関数を呼ぶ前に、表示したいデータを出力論理マップにセットしてください。

(f) 入力要求

入力要求によって、画面から入力した情報を AP へ入力します。画面表示要求で使  
したインタフェース領域を引き継いで使用します。

(例)

```
memcpy(req.xmap_req_type, "RCV", sizeof(req.xmap_req_type)); ...1. 入力要求を代入
req.xmap_req_opt1='F'; ...2. 入力要求時必ず"F"を代入
req.xmap_req_opt2='△'; ...3. 必ず空白を代入
com.xmap_com_inlmg=sizeof(MAP003I); ...4. 入力論理マップ長の代入
jsvwdrv(&com, &req, &MAP003I, 0L);
      ↑      ↑      ↑
      入力論理マップのアドレス
      要求インタフェース領域アドレス
      共通インタフェース領域アドレス
```

(g) オプション設定要求

仮想端末へ画面を表示する場合、マッピングオプションを指定できます。オープン要求  
で使った共通インタフェース領域を引き継いで使用します。

(例)

```

memcpy (req. xmap_req_type, "MDO△", sizeof (req. xmap_req_type));
req. xmap_req_opt1='△';
req. xmap_req_opt2='△';
mdo. xmap_mdo_opt1=XMAP_MDO_SFLD;
mdo. xmap_mdo_opt2=XMAP_MDO_LOGFLD;

jsvwadv (&com, &req, 1L, &mdo);

```

…1. オプション設定要求の代入  
 …2. 必ず空白を代入  
 …3. 必ず空白を代入  
 …4. マッピングオプション大分類を代入  
 …5. マッピングオプション小分類に論理データだけを代入

↑ マッピングインタフェース領域アドレス  
 ↑ 必ず"1"を指定する  
 ↑ 要求インタフェース領域アドレス  
 ↑ 共通インタフェース領域アドレス

### (3) リンケージ時の注意事項

ライブラリには、次に示すファイルを指定する必要があります。

Visual C++ .NET, Visual C++ 6.0 の場合

XMAP3 インストール先 ¥LIB¥X3MWDR32.LIB

Visual C++ 5.0 の場合

XMAP3 インストール先 ¥LIB¥V3Compatible¥X3MWDR32.LIB

## 7.3.4 C 言語固有のコーディング例

C 言語には COBOL のような編集項目 (PIC Z, 9), 文字項目 (PIC X) という概念がないため、数を扱う場合、そのデータが文字 (数字) か、または数値かをユーザが意識して AP を作成する必要があります。また、コンマを付けたりする出力数字編集も AP である必要があります。

ここでは、C 言語で AP を作成する場合に注意が必要な固有のコーディングについて説明します。

### (1) エリアの一括クリアには、memset を使用します。

(例)

出力項目を空白でクリアする。なお、エリアサイズを求めるには、sizeof 演算子を使用する。

```
memset (MAP0030. MAP003_ERRMSG_0, ' ', sizeof (MAP0030. MAP003_ERRMSG_0));
```

(2) データの一括コピーには、memcpy を使用します。

(例)

項目属性に 'PROT' を設定する。

```
memcpy (MAP0030. MAP003_NYUKIN_A, "PROT", sizeof (MAP0030. MAP003_NYUKIN_A));
```

(3) 配列データ、またはメモリ領域の比較には、memcmp を使用します。

(例)

INC コードが 'PF01' であるかどうかを判定する。

```
if (0==memcmp (MAP0031. MAP003_INC1, "PF01", sizeof (MAP0031. MAP003_INC1)));
```

(4) 出力論理マップ中のカーソル位置 (行, 列) のクリアには (00)<sub>16</sub> を使用します。

(例)

出力カーソルの行列位置エリアをクリアする。

```
#define NULLDATA 0x00
memset (&MAP0030. MAP003_OUTCURS_LINL, NULLDATA, sizeof (MAP0030. MAP003_OUTCURS_LINL));
memset (&MAP0030. MAP003_OUTCURS_LINC, NULLDATA, sizeof (MAP0030. MAP003_OUTCURS_LINC));
```

(5) 数字から数値への変換には、文字列変換関数 atoi, または atof を使用します。

(例)

入力論理マップ mapli 中の項目 (char MAP003\_SU1\_I[5];) に設定されている数字データを数値データに変換する。

```
double atof();
double kazu; /*変換後の数値を格納するエリア*/
char wk_s[10]; /*変換前の文字配列を文字列に変換するエリア*/
long lng; /*文字列長*/

lng=sizeof (MAP0031. MAP003_SU1_I);
memcpy (wk_s, MAP0031. MAP003_SU1_I, lng);
wk_s[lng]='¥0';
kazu=atof (wk_s);
```

(6) 数値から数字への変換には書式付き出力関数 `sprintf` を使用します。

(例)

数値を文字に変換して出力論理マップの項目 (`char MAP003_GOUKEI_O[15];`) に格納して計算結果などを画面に表示する。

```
double fdata; /* 数値エリア */
long ldata; /* 数字エリア */
char wk[20]; /* 変換用ワークエリア */

fdata=35540400;
sprintf(wk, "%15.0f", fdata);
memcpy(MAP003O.MAP003_GOUKEI_O, wk, 15);
ldata=123;
sprintf(wk, "%15.0d", ldata);
memcpy(MAP003O.MAP003_GOUKEI_O, wk, 15);
```

## 7.4 汎用 AP インタフェース

Windows 専用の汎用入力関数について説明します。また、汎用関数を使用して AP (Visual Basic および Delphi) で表示する方法について説明します。汎用関数は、C 言語からも使用できます。

なお、OLTP サーバ構成の環境では、汎用 AP インタフェースは使用できません。

### 7.4.1 関数の機能概要

Windows 専用の汎用入出力関数とそれぞれの機能概要を表 7-12 に、Windows 専用の汎用入出力関数の発行順序を図 7-17 に示します。

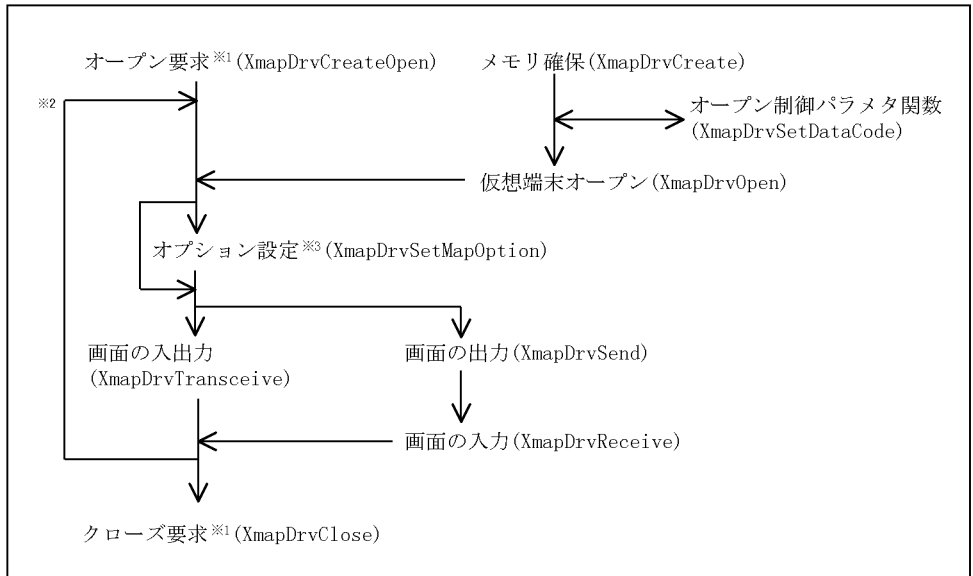
表 7-12 Windows 専用の汎用入出力関数

関数	機能概要	備考
XmapDrvCreateOpen	メモリ確保, 仮想端末オープン (ディスプレイ)	OPEN 要求関数
XmapDrvClose	メモリ解放, 仮想端末クローズ	CLOSE 要求関数
XmapDrvSend	画面の出力	-
XmapDrvReceive	画面の入力	-
XmapDrvTransceive	画面の入出力	-
XmapDrvGetError	エラーコードの詳細を取得	-
XmapDrvCreate	メモリ確保	-
XmapDrvOpen	仮想端末オープン	OPEN 要求関数
XmapDrvSetMapOption	マッピングオプションを設定	出力制御パラメタ関数
XmapDrvSetDataCode	データ有無コードを設定	オープン制御パラメタ関数

(凡例)

- : 該当しない。

図 7-17 Windows 専用の汎用入出力関数の発行順序



- 注※1 必ず指定します。  
 注※2 画面数分を繰り返します。  
 注※3 マッピングオプション変更時に発行します。

## (1) XmapDrvCreateOpen 関数

インタフェース用メモリを確保したり、仮想端末をオープンしたりします。

形式

```
long APIENTRY XmapDrvCreateOpen(tname)
unsigned char *tname; /* 仮想端末名称 */
```

仮想端末名称 (tname): 仮想端末定義ファイルで指定した名称を指定します。

処理内容

XMAP\_COM, XMAP\_REQ, XMAP\_OPN および XMAP\_MDO テーブルを確保し、仮想端末をオープンする。

リターンコード

- 0: 異常終了
- 1 以上: 正常終了 (仮想端末識別子: 確保した領域の先頭アドレス)

注意事項

仮想端末名称は 8 バイト以内で指定し、最後に NULL で終わる文字列を指定してください。

一つの AP (プロセス) 内で同時にオープンできる端末数の上限は 15 個です。

データ有無コードには標準値「1F」が仮定されます。

## (2) XmapDrvClose 関数

仮想端末をクローズします。

形式

```
long APIENTRY XmapDrvClose (termid)
long                termid;          /* 仮想端末識別子 */
```

仮想端末識別子 (termid) : オープン要求関数の戻り値を指定します。

処理内容

1. 仮想端末をクローズする。
2. XMAP\_COM, XMAP\_REQ, XMAP\_OPN, XMAP\_MDO テーブル領域を解放する。

リターンコード

- 0 : 正常終了
- 1 : 異常終了

注意事項

クローズ済みの仮想端末識別子を指定した場合、アクセス違反を起こす場合があります。

## (3) XmapDrvSend 関数

画面を表示します。

形式

```
long APIENTRY XmapDrvSend (termid, pmpname, olog);
long                termid;          /* 仮想端末識別子 */
unsigned char       *pmpname;       /* 物理マップ名称 */
unsigned char       *olog;          /* 出力論理マップ */
```

仮想端末識別子 (termid) : オープン要求関数の戻り値を指定する。  
 物理マップ名称 (pmpname) : 出力する物理マップ名称を指定する。  
 出力論理マップ (olog) : 出力論理セグメントを指定する。

処理内容

画面を表示します。

リターンコード

- 0 : 正常終了
- 1 : 異常終了



**注意事項**

クローズ済みの仮想端末識別子を指定した場合、アクセス違反を起こす場合があります。

エラーが発生した場合、自動的にクローズ処理が実行されます。

**(4) XmapDrvReceive 関数**

画面に入力されたデータを取得します。

**形式**

```
long APIENTRY XmapDrvReceive (termid, ilog);
long          termid;          /* 仮想端末識別子 */
unsigned char *ilog;          /* 入力論理マップ */
```

仮想端末識別子 (termid): オープン要求関数の戻り値を指定する。

入力論理マップ (ilog): 入力論理マップを指定する。

入力論理マップの全体長を入力論理マップの先頭 (2 バイト) に指定する。

**処理内容**

画面に入力されたデータを取得します。

**リターンコード**

0: 正常終了

- 1: 異常終了

**注意事項**

クローズ済みの仮想端末識別子を指定した場合、アクセス違反を起こす場合があります。

エラーが発生した場合、自動的にクローズ処理が実行されます。

入力論理マップ長として実際より短い値を指定した場合、エラーを返します。

**(5) XmapDrvTransceive 関数**

画面を表示したり、画面に入力されたデータを取得したりします。

**形式**

```
long APIENTRY XmapDrvTransceive (termid, pmpname, olog, ilog);
long          termid;          /* 仮想端末識別子 */
unsigned char *pmpname;       /* 物理マップ名称 */
unsigned char *olog;          /* 出力論理マップ */
unsigned char *ilog;          /* 入力論理マップ */
```

仮想端末識別子 (termid): オープン要求関数の戻り値を指定する。

物理マップ名称 (pmpname): 出力する物理マップ名称を指定する。

出力論理マップ (olog): 出力論理マップを指定する。

入力論理マップ (ilog): 入力論理マップを指定する。

## 7. AP のコーディング方法

入力論理マップの全体長を入力論理マップの先頭（2 バイト）に指定する。

### 処理内容

画面を表示したり、画面に入力されたデータを取得したりします。

### リターンコード

- 0 : 正常終了
- 1 : 異常終了

### 注意事項

クローズ済みの仮想端末識別子を指定した場合、アクセス違反を起こす場合があります。

エラーが発生した場合、自動的にクローズ処理が実行されます。

入力論理マップ長として実際より短い値を指定した場合、エラーを返します。

## (6) XmapDrvGetError 関数

エラーの詳細を取得します。

### 形式

```
long APIENTRY XmapDrvGetError ()
```

### 処理内容

直前に発行された関数のエラー詳細を取得します。

### リターンコード

- 0 以上 : 正常終了（詳細エラーコード）

## (7) XmapDrvCreate 関数

メモリを確保します。

### 形式

```
long APIENTRY XmapDrvCreate ()
```

### 処理内容

XMAP\_COM, XMAP\_REQ, XMAP\_OPN, XMAP\_MDO テーブル領域を確保します。

### リターンコード

- 0 : 異常終了
- 1 以上 : 正常終了（仮想端末識別子：確保した領域の先頭アドレス）

## (8) XmapDrvOpen 関数

仮想端末をオープンします。

形式

long APIENTRY XmapDrvOpen(termid, tname)		
long	termid;	/* 仮想端末識別子 */
unsigned char	*tname;	/* 仮想端末名称 */

仮想端末識別子 ( termid ): XmapDrvCreate の戻り値を指定する。

仮想端末名称 ( tname ): 仮想端末定義ファイルで指定した名称を指定する。

処理内容

仮想端末をオープンします。

リターンコード

0 : 正常終了  
 - 1 : 異常終了

注意事項

クローズ済みの仮想端末識別子を指定した場合、アクセス違反を起こす場合があります。

仮想端末名称には、最後にヌルで終わる文字列を指定してください。9 バイト以上の場合、9 バイト以降は切り捨てられます。

一つの AP ( プロセス ) 内で同時にオープンできる端末数の上限は 15 個です。

## (9) XmapDrvSetMapOption 関数

マッピングオプションを指定する。

形式

long APIENTRY XmapDrvSetMapOption(termid, mode)		
long	termid;	/* 仮想端末識別子 */
long	mode;	/* マッピングオプション */

1. 仮想端末識別子 ( termid )  
 オープン要求関数の戻り値を指定する。
2. マッピングオプション ( mode )  
 マッピングオプションを指定する。

指定できるマッピングオプションは、次のとおりです。

## 7. AP のコーディング方法

内容	指定値
マージ	0
論理マップだけ	2
物理マップだけ	3

### 注

mode の内容の詳細については、「1.5.3(1) マッピングオプション」を参照してください。

### 処理内容

マッピングオプションを設定します。

### リターンコード

0 : 正常終了

### 注意事項

クローズ済みの仮想端末識別子を指定した場合、アクセス違反を起こす場合があります。

mode に不当な値を指定した場合、「マージ」を仮定します。

## (10) XmapDrvSetDataCode 関数

データ有無コードを設定する。

```
long APIENTRY XmapDrvSetDataCode(termid, code)
long          termid;          /* 仮想端末識別子 */
unsigned char code;           /* データ有無コード */
```

### 1. 仮想端末識別子 (termid)

オープン要求関数の戻り値を指定する。

### 2. データ有無コード (code)

データ有無コードを指定する。

### 処理内容

データ有無コード ( $(00)_{16}$  ~  $(1F)_{16}$  または  $(FF)_{16}$ ) を設定します。データ有無コードはすべての AP にかかわるコードのため、できるだけ標準値 ( $(1F)_{16}$ ) を利用することをお勧めします。

### リターンコード

0 : 正常終了

### 注意事項

クローズ済みの仮想端末識別子を指定した場合、アクセス違反を起こす場合があります。

この関数を発行しない場合、データ有無コードには標準値「1F」が仮定されます。

## 7.4.2 C 言語での表示方法

C 言語での表示方法について説明します。

### (1) DLL の呼び出し

汎用 AP インタフェースでは、XMAP3 が提供する DLL を使用するため、インタフェース宣言ファイルおよびリンケージライブラリを取り込む必要があります。

- インタフェース宣言ファイルを取り込みます

インタフェース宣言ファイルの格納先

XMAP3 インストール先 ¥INCLUDE¥x3mwd32.h

- リンケージライブラリを取り込みます。リンケージライブラリの格納先を次に示します。

Visual C++ .NET, Visual C++ 6.0 の場合

XMAP3 インストール先 ¥LIB¥x3mwd32.lib

Visual C++ 5.0 の場合

XMAP3 インストール先 ¥LIB¥V3Compatible¥x3mwd32.lib

## 7.4.3 Visual Basic および Access での表示方法

Visual Basic および Access での表示方法について説明します。

### (1) 論理マップの再定義

XMAP3 では、Visual Basic および Access 用の論理マップは生成できません。このため、C 言語の論理マップを生成し、これを参照して Visual Basic または Access 用に論理マップを再定義する必要があります。ただし、Visual Basic と Access の論理マップの再定義内容は同じです。

XmapdrvSend, XmapDrvTransceive 関数を使用する場合、再定義した論理マップの先頭 2 バイトには、正しい論理マップの長さが指定されている必要があります。論理マップの長さについては、再定義元の論理マップを参考にしてください。

C 言語と Visual Basic および Access の変数型の互換表を次の表に示します。

表 7-13 C 言語と Visual Basic および Access の対応

C 言語	Visual Basic, Access
unsigned char 項目名;	項目名 As Byte
short 項目名;	項目名 As Integer

## 7. AP のコーディング方法

C 言語	Visual Basic , Access
long 項目名 ;	項目名 As Long
struct { 項目定義 : } 実体名 ;	Type 構造体名 項目定義 : End Type 実体名 As 構造体名
unsigned char 項目名 [10];	項目名 (9) As Byte

### (2) DLL の呼び出し

汎用 AP インタフェースでは、XMAP3 が提供する DLL を使用するため、インタフェース宣言ファイルを取り込む必要があります。Visual Basic での、インタフェース宣言ファイルの取り込む例を次に示します。

1. [プロジェクト]メニューから [ファイルの追加] を選択し、インタフェーステーブルを指定します。

x3mwgd32.bas・・・インタフェース宣言ファイル

## 7.4.4 Delphi での表示方法

Delphi での表示方法について説明します。

### (1) 論理マップの再定義

XMAP3 では、Delphi 用の論理マップは生成できません。このため、C 言語の論理マップを生成し、これを参照して Delphi 用の論理マップを再定義する必要があります。

C 言語と Delphi の変数型の互換表を次の表に示します。

表 7-14 C 言語と Delphi の対応

C 言語	Delphi
unsigned char 項目名 ;	項目名 : Char;
short 項目名 ;	項目名 : Short;
long 項目名 ;	項目名 : Integer;
struct { 項目定義 : } 実体名 ;	構造体 = packed record 項目定義 : End; 実体名 : 構造体名 ;
unsigned char 項目名 [10];	実体名 : array[0..9] of Char;

注

構造体名は、Type セクションに定義する必要があります。

## (2) DLL の呼び出し

汎用 AP インタフェースでは、XMAP3 が提供する DLL を使用するため、インタフェース宣言ファイルを取り込む必要があります。インタフェース宣言ファイルの取り込む例を次に示します。

- インタフェース宣言ファイル「x3mwgd32.pas」をカレントフォルダにコピーします。
- [プロジェクト]メニューから [オプション] を選択し、「ディレクトリ/条件」の検索パスを指定します。

## 7.4.5 文字編集 DLL

### (1) ファイルの取り込み

#### (a) 汎用関数を使用した場合の C 言語でのファイルの取り込み

汎用 AP インタフェースでは、XMAP3 が提供する DLL を使用するため、インタフェース宣言ファイルおよびリンケージライブラリを取り込む必要があります。

- インタフェース宣言ファイルを取り込みます。  
インタフェース宣言ファイルの格納先  
XMAP3 インストール先 ¥INCLUDE¥x3mwst32.h
- リンケージライブラリを取り込みます。リンケージライブラリの格納先は次のとおりです。

Visual C++ .NET, Visual C++ 6.0 の場合

XMAP3 インストール先 ¥LIB¥x3mwst32.lib

Visual C++ 5.0 の場合

XMAP3 インストール先 ¥LIB¥V3Compatible¥x3mwst32.lib

#### (b) Visual Basic および Access でのファイルの取り込み

文字編集 DLL では、XMAP3 が提供する DLL を使用するため、インタフェース宣言ファイルを取り込む必要があります。Visual Basic および Access での、インタフェース宣言ファイルを取り込む例を次に示します。

1. [プロジェクト]メニューから [ファイルの追加] を選択し、インタフェーステーブルを指定します。  
x3mwst32.bas・・・インタフェース宣言ファイル

#### (c) Delphi でのファイルの取り込み

文字編集 DLL では、XMAP3 が提供する DLL を使用するため、インタフェース宣言ファイルを取り込む必要があります。インタフェース宣言ファイルを取り込む例を次に

## 7. AP のコーディング方法

示します。

1. インタフェース宣言ファイル「x3mwst32.pas」をカレントフォルダにコピーします。
2. [プロジェクト]メニューから[オプション]を選択し、「ディレクトリ/条件」の検索パスを指定します。

### (2) 数字から数値に変換

数字データを数値データに変換します。変換には、文字列変換関数「XmapStrAtol」を使用します。

```
long APIENTRY XmapStrAtol(lpszIn, nInLng)
```

#### (a) 引数

- unsigned char \*lpszIn: 入力領域の先頭アドレス
- long nInLng: 入力文字の有効桁数

#### (b) 戻り値 (4 バイト)

long: 数字から数値に変換した結果

#### (c) 補足説明

\*lpszIn が示すアドレスから nInLng バイト分で、「0 ~ 9」「+」「-」「¥」「,」の文字だけを抜き出し、数字から数値変換します。不当な文字が検出された時点で、処理を中断し値を返します。変換結果が、long 型で表現できない数値の場合、0 を返します。

次に、変換例を示します。

入力数字	出力数値
"¥123,456"	123456
"-¥123,456"	-123456
"123456"	123456

### (3) 数値から数字に変換

数値データを数字データに変換します。または、数値データを数字データに変換し、変換と同時に文字編集を設定します。変換には、文字列変換関数「XmapStrItoA」を使用します。

```
long APIENTRY XmapStrItoA(lpszOut, nOutLng, nIn, nFormat, cData)
```



## (a) 引数

unsigned char	*lpszOut	出力領域の先頭アドレス
long	nOutLng	出力領域の長さ
long	nIn	入力数値
long	nFormat	編集形式
	XMAPSTR_SUM	=0x00800000:コンマ編集
	XMAPSTR_SYMBOL	=0x00400000:先頭に「¥」付加
	XMAPSTR_FRONT	=0x00080000:先頭に符号付加
	XMAPSTR_BACK	=0x00040000:最後に符号付加
	XMAPSTR_PLUS	=0x00020000:プラス値も付加
	XMAPSTR_LEFT	=0x00008000:左寄せ
	XMAPSTR_RIGHT	=0x00004000:右寄せ
unsigned char	cData	埋字

## (b) 戻り値

0:正常

1以上:異常(必要な出力領域の桁数)

## (c) 補足説明

nIn の数値を数字に変換し、nFormat が示す編集を行います。\*lpszOut から nOutLng の領域へ結果を格納します。余った領域を cData で埋めます。変換結果よりも出力領域の桁数が短い場合、必要な桁数を返します。

次に、変換例を示します。

入力数値	nFormat	出力数字
123456	XMAPSTR_SUM	"123,456"
123456	XMAPSTR_SYMBOL	"¥123456"
123456	XMAPSTR_SUM XMAPSTR_SYMBOL	"¥123,456"
123456	XMAPSTR_FRONT XMAPSTR_PLUS	" +123456"
-123456	XMAPSTR_FRONT XMAPSTR_PLUS	" -123456"
123456	XMAPSTR_FRONT	"123456"
-123456	XMAPSTR_FRONT	"-123456"
-123456	XMAPSTR_BACK	"123456-"

nOutLng = 10 で cData = " " の場合

入力数値	nFormat	出力数字
123456	XMAPSTR_LEFT	"123456 "
123456	XMAPSTR_RIGHT	" 123456"

#### (4) 右寄せ配置の指定

指定文字を指定領域へ、右寄せ配置します。右寄せ配置は、「XmapStrMoveRight」関数を使用します。

```
long APIENTRY XmapStrMoveRight(1pszOut, nOutLng, 1pszIn, nInLng, cData)
```

##### (a) 引数

unsigned char	*lpszOut	出力領域の先頭アドレス
long	nOutLng	出力領域の長さ
unsigned char	*lpszIn	入力領域の先頭アドレス
long	nInLng	入力文字列の有効桁数
unsigned char	cData	埋字

##### (b) 戻り値

0: 正常

##### (c) 補足説明

\*lpszIn が示す領域の文字を、\*lpszOut から nOutLng バイト分の領域へ右寄せ配置します。左側へは、cData 埋められます。nInLng > nOutLng の場合、桁寄せはしません。

次に、右寄せの配置例を示します。

入力文字	出力文字 (nOutLng=10, cData = " ")
"123456"	" 123456"

#### (5) 左寄せ配置の指定

指定文字を指定領域へ、左寄せ配置します。左寄せ配置は、「XmapStrMoveLeft」関数を使用します。

```
long APIENTRY XmapStrMoveLeft(1pszOut, nOutLng, 1pszIn, nInLng, cData)
```

##### (a) 引数

unsigned char	*lpszOut	出力領域の先頭アドレス
long	nOutLng	出力領域の長さ
unsigned char	*lpszIn	入力領域の先頭アドレス
long	nInLng	入力文字列の有効桁数
unsigned char	cData	埋字

##### (b) 戻り値

0: 正常

## (c) 補足説明

\*lpzIn が示す領域の文字を, \*lpzOut から nOutLng バイト分の領域へ左寄せ配置します。右側へは, cData 埋められます。nInLng > nOutLng の場合, 桁寄せはしません。

次に, 左寄せの配置例を示します。

入力文字	出力文字 ( nOutLng=10,cData = " ")
"123456"	"123456 "

## 7.5 OLTP サーバ構成での画面表示

ここでは、OLTP サーバ構成用のプログラムの作成について説明します。TP1/NET/XMAP3 を使用する AP は問い合わせ応答型、または継続問い合わせ応答型で作成してください。

OpenTP1 と連携する基本的な AP の構造は次のような処理形態になります。

1. TP1/NET/XMAP3 によって AP が起動されます。
2. 受信命令によって、画面の入力論理マップが取得されます。
3. AP からの送信命令によって、画面が表示されます。
4. AP の処理が終了します。

### 7.5.1 COBOL での画面表示命令

#### (1) SEND 文、および RECEIVE 文による画面表示

日立 COBOL の SEND 文、および RECEIVE 文によって画面を入出力する方法を説明します。

##### (a) 論理マップの取り込み

```
DATA DIVISION.
WORKING-STORAGE SECTION.
COPY 入力論理マップ名. .... 使用する論理マップを指定する
COPY 出力論理マップ名. .... 使用する論理マップを指定する
```

##### (b) 通信記述項

```
DATA DIVISION.
COMMUNICATION SECTION.
CD 通信記述名 FOR I-O ..... 1
MAP NAME IS データ名1 ..... 2
[STATUS KEY IS データ名2] ..... 3
[SYMBOLIC TERMINAL IS データ名3] ..... 4
[MESSAGE DATE IS データ名4] ..... 5
[MESSAGE TIME IS データ名5]. ..... 6
```

#### 注

1 および 2 は必ず指定してください。ほかの項目は、必要に応じて指定してください。

1. 通信記述名、および通信種別 (I-O) (必須)

通信記述名には SEND 文、および RECEIVE 文で使用する名称を指定します。

通信種別には「I-O」を指定します。

2. 物理マップ名称格納エリア（必須）  
SEND 文および RECEIVE 文で表示したい物理マップ名称を格納するエリアです。
3. リターンコード格納エリア  
SEND 文、RECEIVE 文、および TRANSCEIVE 文のリターンコードを格納するエリアです。SEND 文、RECEIVE 文、および TRANSCEIVE 文が正常に処理されたかを判定する場合、このエリアを参照します。詳細については、「7.2.3(1)(g) STATUS KEY 句のリターンコード」を参照してください。
4. 仮想端末名格納エリア  
最初の SEND 文を実行する前に、仮想端末名を格納しておくエリアです。正しい名称を指定しないと、実行時に表示先が決定できなくなりエラーになります。TP1/NET/XMAP3 で指定した仮想端末名を指定してください。
5. 受信日付格納エリア  
受信日付を格納しておくエリアです。受信日付の形式は YYMMDD（YY：西暦下 2 桁、MM：月、DD：日）の形式となります。
6. 受信時間格納エリア  
受信時間を格納しておくエリアです。受信時間の形式は HHMMSS（HH：時、MM：分、SS：秒）の形式となります。

#### (c) RECEIVE 文

```

PROCEDURE          :
                   DIVISION.

RECEIVE  通信記述名  FIRST SEGMENT INTO  入力論理マップ名.  ..... 1

```

1. 通信記述名、および入力論理マップ名  
通信記述名には通信記述項の通信記述名に定義した名称を指定します。  
入力論理マップ名には、WORKING-STORAGE SECTION の COPY 文で取り込んだ、入力論理マップを指定します。

#### (d) SEND 文

```

PROCEDURE DIVISION.
:
MOVE  物理マップ名 TO  データ名1.  ..... 1
SEND  通信記述名 FROM  出力論理マップ名 WITH EMI.  ..... 2

```

1. 物理マップ名、およびデータ名 1  
物理マップ名には、表示させたい物理マップの名称を指定します。このときデバイス ID を含めない名称を指定してください。RECEIVE 文で受信した画面が次画面の場合

## 7. AP のコーディング方法

合、物理マップ名の指定は不要です。

データ名 1 には、通信記述項の物理マップ格納エリアに定義した名称を指定します。

### 2. 通信記述名、および出力論理マップ名

通信記述名には、通信記述項の通信記述名に定義した名称を指定します。

出力論理マップ名には、WORKING-STORAGE SECTION の COPY 文で取り込んだ、表示したい画面の出力論理マップを指定します。

## (2) CALL 文による画面表示

ここでは、COBOL の CALL インタフェースによってマッピングライブラリを使用する方法を説明します。あらかじめ、WORKING-STORAGE SECTION の COPY 文を指定し、使用したい論理マップ名（入力論理マップ、および出力論理マップ）を取り込んでおきます。論理マップの取り込みについては「7.2.2 論理マップの取り込み方法 (COBOL)」を参照してください。

### (a) インタフェース領域の作成

TP1/NET/XMAP3 とのインタフェース用領域を作成します。インタフェース領域については、マニュアル「OpenTP1 プロトコル TP1/NET/XMAP3 編」を参照してください。

DATA	DIVISION.
WORKING-STORAGE	SECTION.
01 一意名1.	
02 データ名1	PIC X(8). ..... 先頭
	:
	:
02 データ名2	PIC X(8). ..... 9個目
	:
	:
01 一意名2.	
	:
	:
02 データ名3	PIC X(8). ..... 3個目
	:
	:

また、上記で注釈のついているデータ項目は、以降に説明する命令文で使用するデータ項目です。それ以外のデータ項目については、マニュアル「OpenTP1 プロトコル TP1/NET/XMAP3 編」を参照してください。

## (b) 受信命令 (CALL 'CBLDCMCF')

```

:
PROCEDURE DIVISION.

MOVE 'RECEIVE ' TO データ名1. .... 1
CALL 'CBLDCMCF' USING 一意名1 一意名2 入力論理マップ名. .... 2

```

1. 'RECEIVE' (必須), およびデータ名 1  
受信を命令するため, 'RECEIVE' を指定します。また, データ名 1 には, インタフェース領域の一意名 1 に定義した先頭のデータ項目名を指定します。
2. 一意名 1, 一意名 2, および入力論理マップ名  
一意名 1, および一意名 2 にはインタフェース領域に定義した一意名 1 および一意名 2 を指定します。  
入力論理マップ名には WORKING-STORAGE SECTION の COPY 文で取り込んだ入力論理マップ名を指定します。

## (c) 送信命令 (CALL 'CBLDCMCF')

```

:
PROCEDURE DIVISION.

MOVE 'REPLY ' TO データ名1. .... 1
MOVE 'EMI ' TO データ名2. .... 2
MOVE 物理マップ名 TO データ名3. .... 3
CALL 'CBLDCMCF' USING 一意名1 一意名2 出力論理マップ名. .... 4

```

1. 'REPLY' (必須), およびデータ名 1  
送信を命令するため, 'REPLY' を指定します。また, データ名 1 には, インタフェース領域の一意名 1 に定義した先頭のデータ項目名を指定します。
2. 'EMI' (必須), およびデータ名 2  
データ名 2 には, インタフェース領域の一意名 1 に定義した 9 個目のデータ項目名を指定します。
3. 物理マップ名, およびデータ名 3  
物理マップ名には, 表示させたい物理マップの名称を指定します。このときデバイス ID を含めない名称を指定してください。RECEIVE 文で受信した画面が次画面の場合, 物理マップ名の指定は不要です。  
データ名 3 には, インタフェース領域の一意名 2 に定義した 3 個目のデータ項目名を指定します。
4. 一意名 1, 一意名 2, および出力論理マップ名  
一意名 1 および一意名 2 には, インタフェース領域に定義した一意名 1 および一意名 2 を指定します。  
出力論理マップ名には, WORKING-STORAGE SECTION の COPY 文で取り込んだ

## 7. AP のコーディング方法

出力論理マップ名を指定します。

### (3) コンパイル時のポイント

日立 COBOL を使用して OLTP サーバ構成用のプログラムをコンパイルするとき、コンパイラオプション「-OpenTP1」(COBOL2002 の場合)または「-Ot」(COBOL85 の場合)を指定する必要があります。

## 7.5.2 C 言語での画面表示命令

### (1) C 言語による画面の入出力

#### (a) 論理マップの取り込み

```
#include "入力論理マップ名" ..... 使用する論理マップ名を指定する  
#include "出力論理マップ名" ..... 使用する論理マップ名を指定する
```

#### (b) OpenTP1 のインタフェーステーブルの取り込み

OpenTP1 と連携するためのインタフェーステーブル (dcmcf.h) は #include 指示語を使用します。

```
      :  
#include <dcmcf.h>  
      :
```



## (c) 受信命令 (dc\_mcf\_receive)

```

:
long    action;
long    commform;
char    termnam[9];
char    mapname[9];
long    rdataleng;
long    time;

int     ret;

:
action  = DCMCFRST | DCMCFBUF2;
commform = DCNOFLAGS;

ret = dc_mcf_receive(action,
                    commform,
                    termnam,
                    mapname,
                    (char *)&入力論理マップ名,
                    &rdataleng,
                    入力論理マップ長,
                    &time);
:

```

## (d) 送信命令 (dc\_mcf\_reply)

```

:
long    action;
long    commform;
char    resv01[9];
char    mapname[9];
char    nextap[9];
long    opcd;

int     ret;

:
action  = DCMCFEMI | DCMCFBUF2;
commform = DCNOFLAGS;
resv01[0] = 0x00;
strcpy(mapname, "物理マップ名");
nextap[0] = 0x00;
opcd = DCNOFLAGS;

ret = dc_mcf_reply(action,
                  commform,
                  resv01,
                  mapname,
                  (char *)&出力論理マップ名,
                  出力論理マップ長-4,
                  nextap,
                  opcd);
:

```

### 7.5.3 AP の実行

OLTP サーバ構成用のプログラムを実行するには、次に示す環境を設定しておく必要があります。

#### (1) TP1/NET/XMAP3 での環境設定

TP1/NET/XMAP3 側で設定しておく内容を次に示します。マニュアル「OpenTP1 プロトコル TP1/NET/XMAP3 編」もあわせて参照してください。

TP1/NET/XMAP3 上での TP1 コネクション名や論理端末名と、XMAP3 上の表示サービス名が対応するよう、あらかじめ設定しておきます。

データ有無コードを使用している場合、XMAP3 での設定内容を基に、TP1/NET/XMAP3 にも同じ内容を設定しておきます。

AP で表示させたい画面について、トランザクションコードと AP の関連を定義しておきます。

#### (2) XMAP3 での環境設定

C/S セットアップで、XMAP3 の表示サービス名に対し、サーバ側の実行ファイルを起動できるよう設定しておきます。なお、サーバ側の実行ファイルとは、OpenTP1 との接続を確立するために、その命令処理を発行する実行ファイルを指します。

表示・印刷セットアップで、オペレータインジケータの表示形式を確認します。設定値が「表示しない」、または「拡張形式」となっている場合は「上に表示」、または「下に表示」を設定してください。

表示・印刷セットアップで、バージョン間の互換オプションを確認します。「C/S の場合、アプリケーションサーバのセットアップだけが有効」の設定が有効になっている場合、チェックボックスをオフにして設定を無効にしてください。

# 8

## 標準パターンを使用した AP の作成

標準パターンとは、XMAP3 が用意している「定義サンプル」「レイアウトパターン」および「AP パターン」の三つを指します。これらのパターンを組み合わせることで、画面の入出力を含めた AP を作成できます。

この章では、画面の定義サンプルや AP パターンにユーザ独自の処理を加える際のポイントを、目的別の例題に沿って説明します。

---

8.1 標準パターン

---

8.2 標準パターンを使用した機能別の作成例 ( COBOL )

---

8.3 各種例題プログラム

---

## 8.1 標準パターン

---

XMAP3 では、AP を効率良く開発する目的で標準パターンを用意しています。標準パターンには、画面の定義時に利用できる「定義サンプル」、「レイアウトパターン」と、AP 作成時に利用できる「AP パターン」とがあります。

ここでは、使用目的別に XMAP3 が標準提供している画面の定義サンプルと AP パターンの使用目的別の組み合わせを紹介します。8.2 以降では、標準パターンを利用した例題を通じて、AP 作成時に留意する点を説明しています。

### 8.1.1 レイアウトパターンとアプリケーションのパターン

#### (1) 定義サンプル

XMAP3 では、画面レイアウトのサンプルを「定義サンプル」として標準提供していますので、画面レイアウトの参考にしてください。定義サンプルは、マップ定義ファイルとして提供されているため、ユーザ独自の変更を加えることで流用して利用することもできます。

定義サンプルを利用するには、新規/更新ダイアログで「画面・帳票を更新する」を選び、COBOL の場合は「XMAP3 インストール先 ¥SAMPLES¥DSPMAP」の下にある定義サンプルのファイル名を指定してください。C 言語で利用する場合は、COBOL 用の定義サンプルのマップ定義ファイルをユティリティのコンバート機能で C 言語用に変換してから、利用してください。ユティリティの詳細についてはマニュアル「XMAP3 開発・実行ガイド」を参照してください。

#### (2) レイアウトパターン

XMAP3 では、ドローのレイアウト領域の標準的なサイズやメッセージ、メニューなど、用途に合わせたパターンを「レイアウトパターン」として用意しています。画面レイアウトが決まったら、このレイアウトパターンを基にして画面を定義します。作成する画面サイズを統一するために使用することをお勧めします。

レイアウトパターンを利用するには、新規/更新ダイアログで「画面・帳票を新規作成する」を選び、新規作成ダイアログの「レイアウトパターンの選択」でパターン名を指定してください。パターンの種類については、「付録 B XMAP3 で提供している定義パターン」を参照してください。

#### (3) AP パターン

XMAP3 では、AP の定型的な型として、COBOL 用および C 言語用の AP パターンを用意しています。AP パターンには、AP パターンと AP 部品とがあります。

## AP パターン

### COBOL の場合

COBOL ソースプログラムの見出し部、データ部、手続き部など、プログラム全体の標準的な骨組みが記述されていますので、メインプログラムとして利用できます。

### C 言語の場合

C 言語ソースプログラムの標準的な骨組みが記述されていますので、メインプログラムとして利用できます。

### AP 部品

AP パターン共通に、頻繁に使用すると考えられる処理の手続きが記述されていますので、サブルーチンとして利用できます。

XMAP3 で作成した画面を取り込むソースプログラム作成時には、これらのパターンを組み合わせて使用することで、AP を効率良く作成できるほか、定型的なコーディングの統一を図れます。

コーディング時には、@@@XXX (XXX は数字) で示されているマップ名を 6 文字以内の半角英数字に変更するほか、ユーザ独自の処理を加えてください。

AP 部品を AP パターンへ組み込む方法については、「8.1.2 COBOL エディタからの AP パターンの利用」または「8.1.3(2) C 言語用パターンの利用方法」を参照してください。

## (4) 画面サンプルと AP パターンの組み合わせ

目的に合った画面の定義サンプルと AP パターンの組み合わせを次の表に示します。定義サンプルは「¥SAMPLES¥DSPMAP または ¥PRTMAP」に、AP パターンは「¥Patterns」にそれぞれ格納されています。

なお、OLTP サーバ構成用の AP に、その他の AP パターンを流用することはできません。OLTP サーバ構成の環境で使用する AP には、専用の AP パターンを利用してください。

表 8-1 画面の定義サンプルと AP パターンの組み合わせ (COBOL)

使用目的		定義サンプルのマップ名	AP パターンファイル名	例題の説明箇所
メニュー画面表示	ユーザ登録確認	GPASW1.imp	BTMENU01.cbl	8.2.1
	初期メニュー表示	BMENU1.imp SMENU1.imp		
	スプラッシュ画面の表示	SPLASH.imp		8.2.2
メニュー画面からの次画面表示		SMENU1.imp GOPTN1.imp	GENDSP01.cbl NXTDSP01.cbl	8.2.3

## 8. 標準パターンを使用した AP の作成

使用目的	定義サンプルのマップ名	AP パターンファイル名	例題の説明個所	
データの入出力	フィールド単位 (CUI)	FIE1CC.imp	GENDSP01.cbl	8.2.4
	フィールド単位 (GUI)	FIE1GC.imp	GENFLD01.cbl	8.2.5
	イベント単位	EVE1GC.imp	GENEVN01.cbl	8.2.6
	行単位	LIN1GC.imp	GENDSP01.cbl PATWRT01.cbl	8.2.7
	入力フィールドクリア	GFILD3.imp	GENDSP01.cbl CLRINP01.cbl	8.2.8
	フィールド属性変更	GPASW1.imp	GENDSP01.cbl MODATR01.cbl	8.2.9
一覧リストの表示	表形式の一覧表示	GCTBL1.imp	GENDSP01.cbl	8.2.10
	可変項目の再表示			8.2.11
オプション画面の表示	ヘルプ画面の表示	GFILD3.imp HELP01.imp	GENDSP01.cbl FLDHLP01.cbl	8.2.12
	ダイアログの表示	GFILD3.imp GLBOX3.imp		8.2.13
	掲示板の表示	BOARD.imp		8.2.14
帳票印刷機能との連携	画面表示と帳票印刷	LTABL1.imp BMENU1.imp	DSPPR01.cbl	8.2.15
OLTP サーバ構成での画面表示と帳票印刷	画面表示と帳票印刷	MCF1G.imp MCF1P.imp MCF2G.imp MCF3G.imp MCF4G.imp MCF5G.imp	MCFDP01.cbl	8.2.16

### 8.1.2 COBOL エディタからの AP パターンの利用

日立 COBOL を利用する場合、COBOL エディタを使用して XMAP3 の AP パターンおよび AP 部品を編集できます。エディタウィンドウでメニューバーの [ ファイル ] から [ 開く ] を選び、XMAP3 インストール先 ¥PATTERNS¥COBOL の下にあるファイル名 ( 拡張子が .cbl ) を指定するとソースプログラムを編集できる状態になります。なお、エディタウィンドウでの操作方法については、マニュアル「COBOL2002 操作ガイド」またはマニュアル「COBOL85 操作ガイド」を参照してください。

また、COBOL エディタで XMAP3 が標準提供しているプログラムテンプレート ( 拡張子が .cet のファイル ) を利用できます。プログラムテンプレートの内容は、XMAP3 標準提供の AP パターン ( 拡張子が .cbl のファイル ) と同じです。

## (1) プログラムテンプレートを使ったファイルの編集方法

COBOL エディタで XMAP3 のプログラムテンプレートを使用してファイルを編集する方法を説明します。

なお、この方法でプログラムテンプレートを使用すると、そのつどファイル名を指定して参照する必要があります。COBOL エディタが提供しているテンプレート以外のファイルを、常にテンプレートファイルのリストに表示したい場合の設定方法については、マニュアル「COBOL2002 操作ガイド」またはマニュアル「COBOL85 操作ガイド」を参照してください。

### (a) COBOL2002, COBOL85 Version 7 および Version 6 の場合

1. COBOL エディタのエディタウィンドウで、[ ファイル ] メニューから [ 新規作成 ] を選ぶ。  
新規作成ダイアログが表示されます。
2. 作成したいファイルの種類を選ぶ。  
必要に応じてファイルの拡張子も選びます。
3. ファイルの種類に対応するプログラムテンプレートの [ ... ] ボタンをクリックする。
4. ファイルを開くダイアログで、「ファイルの場所」に「XMAP3 インストール先¥ PATTERNS¥ COBOL」を指定する。  
XMAP3 のプログラムテンプレートが表示されます。
5. 使用したいプログラムテンプレートのファイルを選ぶ。  
テンプレートファイルのリストに選択したファイル名が追加されます。
6. [ OK ] ボタンをクリックする。  
新規にファイルが作成され、プログラムテンプレートの内容がソースウィンドウに表示されます。

### (b) COBOL85 Version 5.0 の場合

1. COBOL エディタのエディタウィンドウで、[ ファイル ] メニューから [ 新規作成 ] を選ぶ。  
新規作成ダイアログが表示されます。
2. エディタウィンドウのテンプレートで、[ 参照 ... ] ボタンを選ぶ。
3. フォルダの参照ダイアログで、フォルダに「XMAP3 インストール先 ¥PATTERNS¥COBOL」を指定する。  
XMAP3 のプログラムテンプレートが表示されます。
4. 使用したいプログラムテンプレートのファイルを選ぶ。  
テンプレートファイルのリストに選んだファイル名が追加されます。
5. [ OK ] ボタンを選ぶ。  
新規にファイルが作成され、プログラムテンプレートの内容がソースウィンドウに表

## 8. 標準パターンを使用した AP の作成

示されます。

### (2) COBOL エディタでの AP 部品の利用方法

XMAP3 の AP 部品を COBOL エディタで利用する方法を説明します。

AP 部品は、拡張子が .cbl のファイルとして提供されています。このため、COBOL エディタでは、AP 部品を一つのソースファイルとして扱います。

1. ソースウィンドウで、AP 部品を挿入したい位置にカーソルを移動する。
2. [挿入]メニューから [ファイル] を選ぶ。  
挿入するソースファイルを指定するダイアログが表示されます。
3. 「XMAP3 インストール先 ¥PATTERNS¥COBOL」を指定し、使用したい AP 部品のファイルを選ぶ。
4. [OK] ボタンをクリックする。  
指定したソースファイルのすべての内容が、カーソル位置に挿入されます。

## 8.1.3 C 言語用のサンプルと AP パターン

### (1) 画面サンプルと AP パターンの組み合わせ

目的別の定義サンプルと AP パターンの組み合わせを次の表に示します。なお、定義サンプルについては C 言語用としては提供していません。そのため、COBOL 用として提供しているものを、ユティリティ機能の「コンバート機能」を使用して、C 言語用定義サンプルに変換してお使いください。

AP パターンは、「XMAP3 インストール先 ¥PATTERNS¥C」に格納されています。

なお、OLTP サーバ構成用の AP に、その他の AP パターンを流用することはできません。OLTP サーバ構成の環境で使用する AP には、専用の AP パターンを利用してください。

表 8-2 画面の定義サンプルと AP パターンの組み合わせ (C 言語)

	使用目的	定義サンプルの マップ名	AP パターン ファイル名
メニュー画面表示	ユーザ登録画面	GPASW1.imp	BTMENU01.c
		BMENU1.imp	
	SMENU1.imp		
	スプラッシュ画面の表示	SPLASH.imp	BTMENU01.c
		BMENU1.imp	
		SMENU1.imp	
メニュー画面からの次画面表示	SMENU1.imp	GENDSP01.c	



使用目的		定義サンプルの マップ名	AP パターン ファイル名
		GOPTN1.imp	NXTDSP01.c
データの入力	フィールド単位 (CUI)	FIE1CH.imp	GENDSP01.c
	フィールド単位 (GUI)	FIE1GH.imp	GENFLD01.c
	行単位 (GUI)	LIN1GH.imp	GENDSP01.c
	画面単位 (GUI)	GAM1GH.imp	GENDSP01.c
	イベント単位	EVE1GH.imp	GENEVN01.c
一覧リストの表示	表形式の一覧表示	GCTBL1.imp	GENDSP01.c
	可変項目の再表示		
オプション画面の表示	ヘルプ画面の表示	HELPO1.imp	
	ダイアログの表示	GFILD3.imp	
		GLBOX3.imp	
帳票印刷機能との連携	画面表示と帳票印刷	BMENU1.imp	DSPPRT01.c
		LTABL1.imp	
OLTP サーバ構成での画面表示と帳票印刷	画面表示と帳票印刷	MCF1G.imp MCF1P.imp MCF2G.imp MCF3G.imp MCF4G.imp MCF5G.imp	MCFDP01.c

## (2) C 言語用パターンの利用方法

XMAP3 の AP 部品を Visual C++.NET, Visual C++ で利用する方法を説明します。

AP 部品は、拡張子が .c のファイルとして提供されています。AP 部品は一つのソースファイルとして扱われます。

1. ソースウィンドウで、AP 部品を挿入したい位置にカーソルを移動する。
2. [挿入]メニューから [ファイル] を選ぶ。  
挿入するソースファイルを指定するダイアログが表示されます。
3. 「XMAP3 インストール先 ¥PATTERNS¥C」を指定し、使用したい AP 部品のファイルを選ぶ。
4. [OK] ボタンをクリックする。  
指定したソースファイルのすべての内容が、カーソル位置に挿入されます。

## 8.2 標準パターンを使用した機能別の作成例 (COBOL)

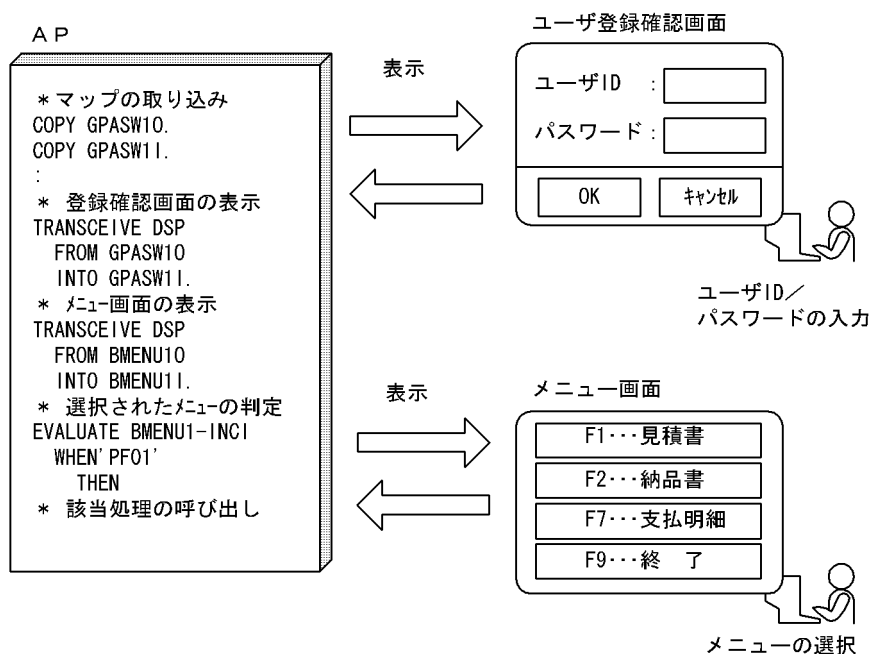
### 8.2.1 BTMENU01 パターンを使用したメニュー画面の表示

#### (1) 処理の概要

ユーザ ID とパスワードを入力するためのユーザ登録確認画面を表示します。ユーザ ID とパスワードを照合して登録があれば、次にメニュー画面を表示します。このメニュー画面で、プッシュボタンまたはファンクションキーを使用して次の処理を選ぶと、選ばれたメニューに対応した使用ごとの AP を起動します。

メニュー画面の表示処理の概要を次の図に示します。

図 8-1 メニュー画面の表示処理



#### (2) 画面定義上のポイント

作成する画面の基になる定義サンプル，または画面のレイアウトパターンを次に示します。

作成する画面	適用できるパターン	
	画面の定義サンプル名	画面のレイアウトパターン名
ユーザ登録確認画面	GPASW1	ログインウィンドウ
メニュー画面	BMENU1	一段ボタンメニュー

ユーザ登録確認画面には次の特長があります。

- パスワードを入力する項目の使用目的を「パスワード」と指定すると、パスワード入力時は入力データを「\*」や「空白」で隠せる。
- ユーザ ID またはパスワードに誤りがあった場合に利用できるエラーメッセージ用の項目を用意している。

### (3) AP 作成上のポイント

作成する AP の基になる AP パターンを次に示します。

- メインプログラムで使用する AP パターン名：BTMENU01

メインプログラムには、次の特長があります。

- 対応しないファンクションキーや [ Enter ] が押されたときの処理が用意されている。
- メニュー画面からは、[ F1... 見積書 ] ボタンまたは [ F1 ] を選んだときの業務の AP を CALL 文で呼び出す仕様となっている。
- メニュー画面から業務ごとの AP を呼び出すとき、メニュー画面を DISABLE 文でクローズしている。

なお、この AP パターンを使用するメインプログラム作成時には次の注意が必要です。

- ユーザ ID およびパスワードの照合処理は含まれていないため、ユーザ独自のコーディングが必要となる。

### (4) コーディング例

この例題のソースプログラムを 'MENUOUT.cbl' として標準提供しています。ソースプログラムからコーディング上のポイントを抜粋して次に示します。

## 8. 標準パターンを使用した AP の作成

```
WORKING-STORAGE          SECTION.
* 論理マップのCOPY展開
* ユーザ登録確認画面用
COPY  GPASW10.            ..... 1.
COPY  GPASW11.
* メニュー画面用
COPY  BMENU10.           ..... 1.
COPY  BMENU11.
COPY  X3MODTBL.
01 EXEC-NAME1.
02 FILLER                PIC X(12) ..... 2.
                           VALUE 'DSPCLEAR.exe'
PROCEDURE                 DIVISION.
* マップ名の設定
MOVE 'GPASW1ND' TO 画面マップ名
* ユーザ登録確認画面の表示
TRANSCEIVE DSP FROM GPASW10 INTO GPASW11.
:
* メニュー画面の表示
* マップ名の設定
MOVE 'BMENU1ND' TO 画面マップ名.
* 画面の表示と入力
TRANSCEIVE DSP FROM BMENU10 INTO BMENU11.
* ボタン・ファンクションキーのチェックと該当プログラムの呼び出し
EVALUATE BMENU1-INC1
  WHEN 'PF01'
    DISABLE DSP
    MOVE '0' TO PROC-IND
    CALL 'CBLEXEC' USING EXEC-NAME-LEN ..... 3.
                           EXEC-NAME1 EXEC-PARM
END-EVALUATE.
```

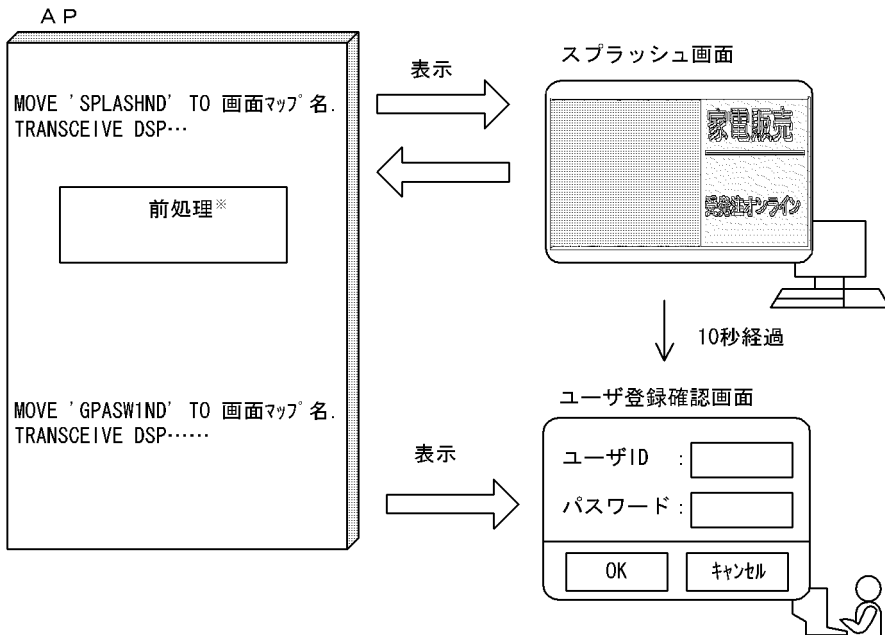
1. マップ名を変更する。  
プログラム中の '@@@111' を 'GPASW1' に、 '@@@444' を 'BMENU1' に変更します。
2. 実行ファイル名を変更する。  
プログラム中の '###EXE01.exe' を 'DSPCLEAR.exe' に変更します。
3. CALL 文を使用して AP を呼び出す。  
CALL 'DSPCLEAR.exe'

### 8.2.2 BTMENU01 の AP 実行前に表示するスプラッシュ画面の追加

#### (1) 処理の概要

AP の前処理をしている間に、これから実行する AP の内容を示すようなスプラッシュ画面を表示します。ここでは、スプラッシュ画面が 10 秒間表示されたあと、BTMENU01 パターンによる「ユーザ登録確認画面」に画面が切り替わります。なお、このコーディング例では、BTMENU01 での AP 前処理はされません。AP からスプラッシュ画面を表示させ、「ユーザ登録確認画面」に切り替える処理の概要を次の図に示します。

図 8-2 スプラッシュ画面の表示と画面切り替え



注※ 例題プログラムでは、前処理は省略されています。

## (2) 画面定義上のポイント

スプラッシュ画面には次のような内容を定義します。

- タイトルバー、メニューバーおよびオペレータインジケータは表示できない。
- 全面に壁紙を使用できる。
- 画面属性ダイアログの「入力単位」を「表示直後」に設定し、表示後すぐに AP に制御が渡るようにする。

## (3) AP 作成上のポイント

スプラッシュ画面を表示すると、表示直後に AP に制御が戻るので、AP の前処理をします。前処理が終了したら、同じ仮想画面に「ユーザ登録確認画面」を表示します。このとき、表示形態は「全面書換」または「自動」です。

8. 標準パターンを使用した AP の作成

(4) コーディング例

```

WORKING-STORAGE SECTION.
* 論理マップのCOPY展開
* スプラッシュ画面用
  COPY          SPLASHO.
  COPY          SPLASHI.
* ログイン画面用
  COPY          GPASW10.
  COPY          GPASW11.
  .
PROCEDURE DIVISION.
*****
*      スプラッシュ画面の表示      *
*****
*      端末名の設定
      MOVE 'DSP001' TO 画面端末名.          ..... 1.

*      出カエリアのクリア
      MOVE ALL X'1F' TO SPLASHG.
*      スプラッシュ画面のマップ名の設定
      MOVE 'SPLASHND' TO 画面マップ名.      ..... 2.
*      スプラッシュ画面の表示
      TRANSCEIVE DSP FROM SPLASHO INTO SPLASHI.  ..... 2.

*      結果確認
      .
*****
*-----ログイン画面表示用のユーザ前処理を追加して下さい-----*
*-----本例では、前処理の代わりに数10秒間APを待ち状態にします-----*
      MOVE 1 TO 処理待ち回数.
      PERFORM UNTIL (処理待ち回数 > 処理待ちMAX)
        MOVE 1 TO 処理待ち領域(処理待ち回数)
        COMPUTE 処理待ち回数 = 処理待ち回数 + 1
      END-PERFORM.
*****

*****
*      画面表示処理：ログイン画面      *
*****
*      端末名の設定
      MOVE 'DSP001' TO 画面端末名.

*      マップ名の設定
      MOVE 'GPASWIND' TO 画面マップ名      .....4.
      .

*      画面の表示と入力
      TRANSCEIVE DSP FROM GPASW10 INTO GPASW11  .....4.
      .

*      結果確認
      IF 画面-RC = 正常
  
```

1. 画面の表示先を、仮想端末名 'DSP001' にする。
2. スプラッシュ画面を表示する。

3. スプラッシュ画面が表示されている間に、ログイン画面を表示するための前処理をする。  
この例題では、前処理は省略しています。その代わりに、数 10 秒間実行待ち状態にする処理を入れてあります。
4. ログイン画面を表示する。

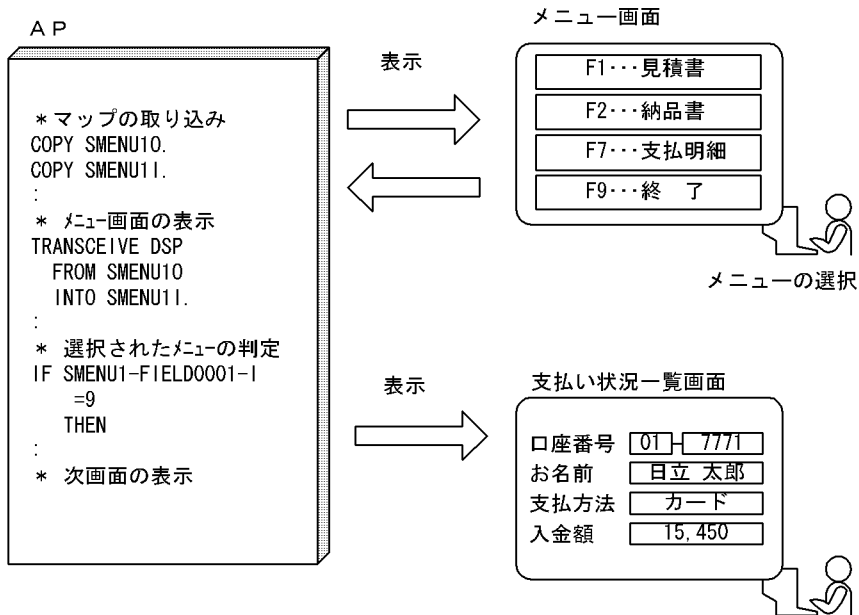
## 8.2.3 GENDSP01 パターンを使用したメニュー画面からの次画面表示

### (1) 処理の概要

処理を選ぶためのメニュー画面を表示します。この画面でメニューを選び、[ OK ] ボタンを選ぶと、選ばれたメニューに対応する業務ごとの画面を表示します。

メニュー画面から次画面（この例題では「支払い状況一覧画面」）を表示する処理の概要を次の図に示します。

図 8-3 メニュー画面からの次画面表示処理



### (2) 画面定義上のポイント

作成する GUI 画面の基になる定義サンプルを次に示します。

- メニュー画面の定義サンプル：SMENU1
- 支払い状況一覧画面の定義サンプル（次画面）：GOPTN1

## 8. 標準パターンを使用した AP の作成

### (3) AP 作成上のポイント

作成する AP の基になる AP パターンと AP 部品を次に示します。

- メインプログラムで使用する AP パターン名：GENDSP01
- サブルーチンで使用する AP 部品名：NXTDSP01

このパターンを使用する AP 作成時には、次の注意が必要です。

- メニュー画面でエラーチェックをするためには、コーディングを追加する必要がある。
- メニューに対応して表示される画面の数に合わせて、AP 部品 'NXTDSP01' を使用して画面表示処理を追加する。
- メニュー画面でエラーが発生した場合の、後続画面の表示スキップ処理を必要に応じて追加する。
- 一次ウィンドウから一次ウィンドウへ遷移するため、サブルーチンでの画面表示時のマッピングオプションは、最初は常に初期表示の指定をする。

### (4) コーディング例

この例題のソースプログラムを 'OUTDSP01.cbl' として標準提供しています。ソースプログラムからコーディング上のポイントを抜粋して次に示します。



```

WORKING-STORAGE SECTION.
* 論理マップのCOPY展開
* メニュー画面用
COPY SMENU10. .... 1.
COPY SMENU11.
* 支払い状況一覧画面用
COPY GOPTN10. .... 2.
COPY GOPTN11.
COPY X3MODTBL.
PROCEDURE DIVISION.
* マップ名の設定
MOVE 'SMENU1ND' TO 画面マップ名
* メニュー画面の表示
TRANSCIEVE DSP FROM SMENU10 INTO SMENU11.
:
* 支払い状況一覧画面の表示 ..... 3.
* マップ名の設定
MOVE 'GOPTN1ND' TO 画面マップ名.
* 画面の表示と入力
MOVE +0 TO 画面2-フラグ.
PERFORM UNTIL 画面2-フラグ = 終了 OR
              画面2-フラグ = キャンセル OR
              画面1-フラグ = キャンセル
              TRANSCEIVE DSP FROM GOPTN10 INTO GOPTN11
* ボタン。ファンクションキーのチェックと該当プログラムの呼び出し
EVALUATE GOPTN-INCI ..... 3.
    WHEN 'ENTER'
        PERFORM 画面2-セクション
    WHEN 'A002'
        MOVE キャンセル TO 画面2-フラグ
END-EVALUATE.
END-PERFORM.
* ----- キャンセルのときの処理を追加 ----- *
画面1-セクション SECTION.
IF SMENU1-FIELD0001-1 = 9
    THEN
* メニュー画面で【F9...終了】が選ばれた場合
MOVE キャンセル TO 画面1-フラグ
    ELSE
* メニュー画面で【F9...終了】以外のメニューが選ばれた場合
MOVE 終了 TO 画面1-フラグ
END-IF.
画面2-セクション SECTION. .... 3.
MOVE 終了 TO 画面2-フラグ.

```

1. マップ名を変更する。  
プログラム中の '@@@555' を 'SMENU1' に、変更します。
2. 'GENDSP01' に 'NXTDSP01' を取り込み、 '@@@666' を 'GOPTN1' に変更する。
3. 選択されたメニューの判定と、該当画面を表示する処理を追加する。

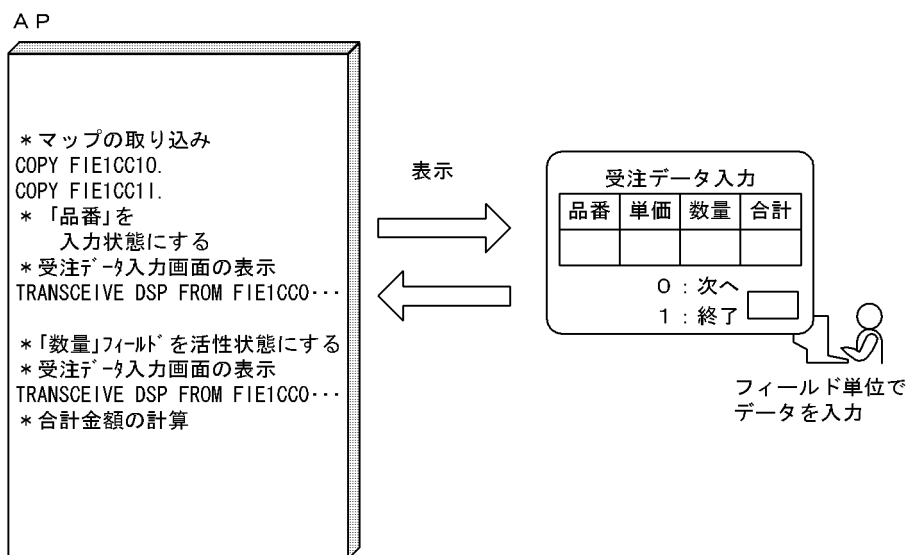
## 8.2.4 GENDSP01 パターンを使用した CUI 画面でのフィールド単位のデータ入力

### (1) 処理の概要

受注データ入力画面を表示します。ファンクションキーなどを使用しないで、テンキーだけを操作してデータ入力をし、フィールド単位でデータの処理をします。

受注データ入力画面でのデータ入力処理の概要を次の図に示します。

図 8-4 受注データ入力画面でのデータ入力処理



### (2) 画面定義上のポイント

作成する CUI 画面の基になる定義サンプルを次に示します。

- 受注データ入力画面の定義サンプル：FIE1CC

定義サンプルを使用しない場合は、レイアウトパターンの「フリー」を選んで定義してください。

受注データ入力画面の作成時には、次の注意が必要です。

- すべての入出力フィールドに「自動スキップ（選択したら次のフィールドにフォーカスを移す）」の属性を指定する。これによって、フィールドの領域いっぱいまで文字を入力すると、自動的に次の入出力フィールドにカーソルが移動される。
- すべての入出力フィールドに「自動送信（入力直後に AP へ制御を渡す）」の属性を指定する。これによって、データ入力後にほかのフィールドにカーソルが移動したときに、このフィールドは送信（実行）キーが押された状態になり、AP へ自動的に通知される。

- 処理済みのフィールドを書き込み禁止状態にするために、すべての入出力フィールドに「動的変更 (AP から表示属性を変更する)」の属性を指定する。

### (3) AP 作成上のポイント

作成する AP の基になる AP パターンを次に示します。

- メインプログラムで使用する AP パターン名 : GENDSP01

この AP パターンを使用する AP 作成時には次の注意が必要です。

- フィールド単位で、順にデータ入力、表示処理をする。このとき処理済みのフィールドに、AP から「入力不可」属性を指定することで、フィールドが書き込み禁止の状態になる。
- ファイルの入出力処理は、必要に応じて追加する。

### (4) コーディング例

この例題のソースプログラムを 'FIELDCC.cbl' として標準提供しています。ソースプログラムからコーディング上のポイントを抜粋して次に示します。

## 8. 標準パターンを使用した AP の作成

```
WORKING-STORAGE          SECTION.
* 論理マップの取り込み
  COPY   FIE1CC0.          ..... 1.
  COPY   FIE1CC1.
* 動的変更テーブルの取り込み
  COPY   X3MODTBL.
  :
PROCEDURE                 DIVISION.
* マップ名の設定
  MOVE   'FIE1CCNC'       TO   画面マップ名

* 品番フィールドだけ入力できる状態にする
  MOVE   XMAP-IN-ATTR3    TO   FIE1CC-数量-A.    ..... 2.
  MOVE   XMAP-IN-ATTR3    TO   FIE1CC-判定-A.

* 画面の表示とデータの入力
  TRANSCEIVE DSP FROM FIE1CC0 INTO FIE1CC1.    .... 3.
  :
* 品番フィールドに対する処理を行う
  :
* 数量フィールドだけ入力できる状態にする
  MOVE   XMAP-IN-ATTR3    TO   FIE1CC-品番-A.    ..... 2.
  MOVE   X'1F1F'          TO   FIE1CC-数量-A.
  MOVE   XMAP-IN-ATTR3    TO   FIE1CC-判定-A.

* 画面の表示とデータの入力
  TRANSCEIVE DSP FROM FIE1CC0 INTO FIE1CC1.    .... 3.
  :
* 合計の項目を計算する処理
  COMPUTE FIE1CC-合計-0 =
          FIE1CC-単価-0 * FIE1CC-数量-1.
  :
* 判定フィールドだけを入力できる状態にする
  MOVE   XMAP-IN-ATTR3    TO   FIE1CC-品番-A.    ..... 2.
  MOVE   XMAP-IN-ATTR3    TO   FIE1CC-数量-A.
  MOVE   X'1F1F'          TO   FIE1CC-判定-A.
* 受注データ入力画面の表示とデータの入力
  TRANSCEIVE DSP FROM FIE1CC0 INTO FIE1CC1.    .... 3.
  :
```

1. マップ名を変更する。  
プログラム中の '@@@555' を 'FIE1CC' に変更します。
2. AP から表示属性を変更するための修飾名を設定する。
3. 画面の表示とデータの入力。

### 8.2.5 GENFLD01 パターンを使用した GUI 画面でのフィールド単位のデータ入力

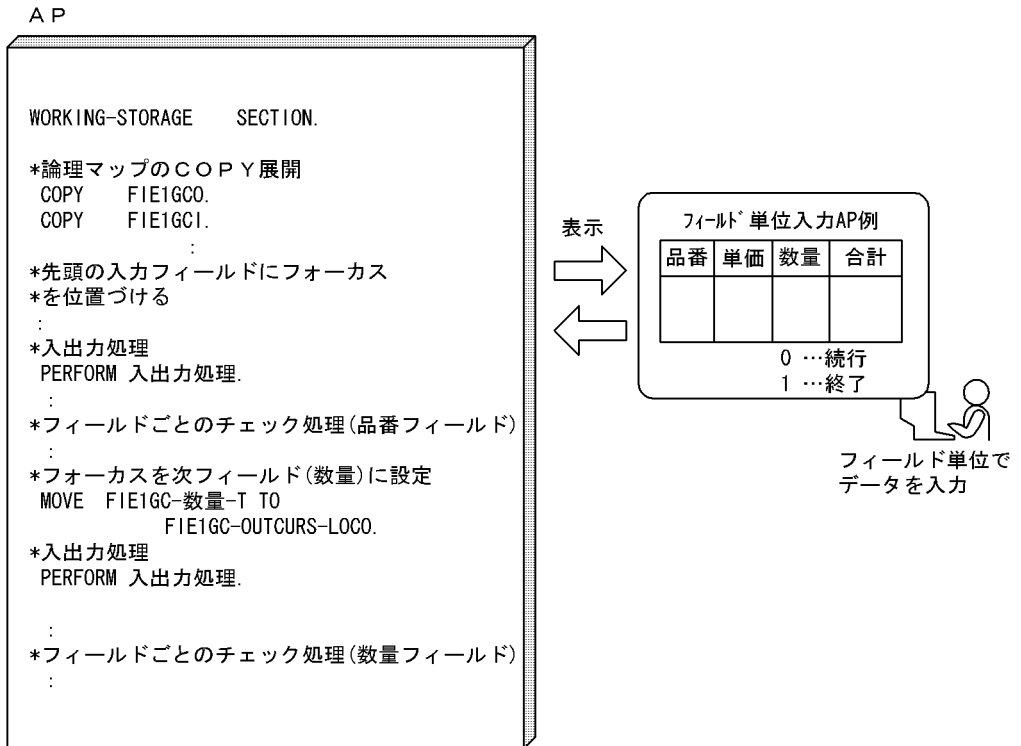
#### (1) 処理の概要

フィールド単位入力画面を表示します。ファンクションキーなどを使用しないで、デー

タ入力をし、フィールド単位でデータの処理をします。入力画面上では、フォーカス・カーソルが位置づいているオブジェクトだけが入力できる状態になります。どのオブジェクトを入力対象とするかは、AP から指定します。

フィールド単位入力画面でのデータ入力処理の概要を次の図に示します。

図 8-5 フィールド単位入力画面でのデータ入力処理



## (2) 画面定義上のポイント

作成する GUI 画面の基になる定義サンプルを次に示します。

- フィールド単位入力画面の定義サンプル：FIE1GC

定義サンプルを使用しない場合は、レイアウトパターンの「フリー」を選んで定義してください。

受注データ入力画面の作成時には、次の注意が必要です。

- 画面属性ダイアログの「入力単位」で「フィールド」を指定する。

## (3) AP 作成上のポイント

作成する AP の基になる AP パターンを次に示します。

## 8. 標準パターンを使用した AP の作成

- メインプログラムで使用する AP パターン名 : GENFLD01

この AP パターンを使用しての AP 作成時には次の注意が必要です。

- フィールド単位で順に、データ表示と入力、および入力チェック処理をする。

このとき、次に入力するフィールドに AP からフォーカス・カーソルを設定することで、そのフィールドだけが入力できる状態になります。

- ファイルの入出力処理は、必要に応じて追加する。

### (4) コーディング例

この例題のソースプログラムを 'FIELDGC.ebl' として標準提供しています。ソースプログラムからコーディング上のポイントを抜粋して次に示します。

```
WORKING-STORAGE SECTION.
* 論理マップの取り込み
COPY FIE1GCO. . . . 1.
COPY FIE1GCI.
* 動的変更テーブルの取り込み -- "XMAP3インストール先¥INCLUDE"にあります
COPY X3MODTBL.

*
PROCEDURE DIVISION.
***** マップ名の設定 *****
MOVE 'FIE1GCND' TO 画面マップ名.

*
***** 品番フィールドのみを入力可能状態にする *****
MOVE FIE1GC-品番-T TO FIE1GC-OUTCURS-LOC0. . . . 2.

*
***** 画面の表示と入力 *****
PERFORM 画面入出力.

*
***** 品番処理のチェック処理 ***** . . . 3.
*----- ユーザ任意のチェック処理を追加します -----*
:

*
***** 数量フィールドのみを入力可能状態にする *****
MOVE FIE1GC-数量-T TO FIE1GC-OUTCURS-LOC0. . . . 4.

***** 画面の表示と入力 *****
PERFORM 画面入出力.

*
***** 数量処理のチェック処理 *****
*----- ユーザ任意のチェック処理を追加します -----*
:

画面入出力 SECTION.

***** 画面の表示と入力 ***** . . . 5.
TRANSCIVE DSP FROM FIE1GCO INTO FIE1GCI.
:
```

1. マップ名を変更する。  
プログラム中の '@@@FFF' を 'FIE1GC' に変更します。
2. 入力するフィールドに、フォーカス・カーソルを位置づける。
3. 入力したフィールドに対応する処理をする。
4. 次に入力するフィールドに、フォーカス・カーソルを移動させる。
5. 画面の入出力処理をする。

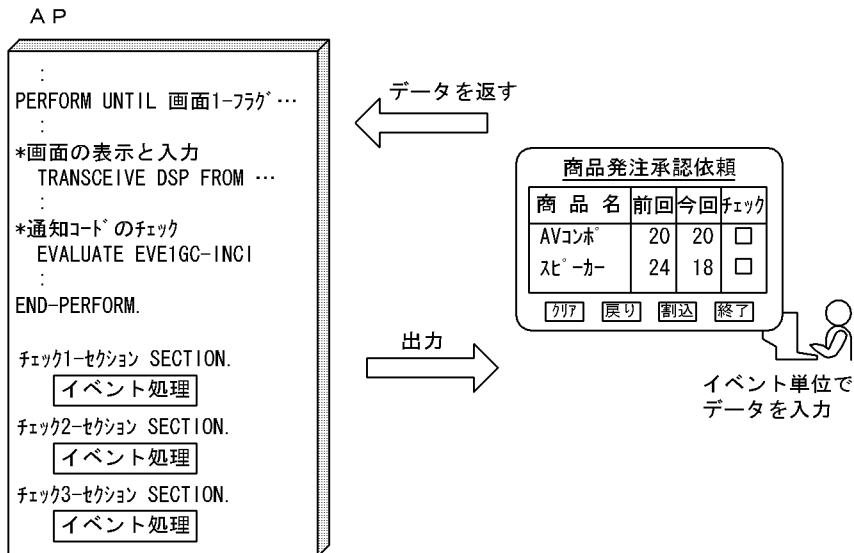
## 8.2.6 GENEVN01 パターンを使用した GUI 画面でのイベント単位でのデータ入力

### (1) 処理の概要

GUI 画面を表示します。ファンクションキーなどで画面を確定したタイミングで AP に制御を返すのではなく、オブジェクト単位で AP に制御を返しデータの処理をします。入力画面上では、入力または選択の制限を指定しているオブジェクト以外は、すべて入力・選択できる状態になります。

イベント単位入力画面でのデータ入力処理の概要を次の図に示します。

図 8-6 イベント単位入力画面でのデータ入力処理



### (2) 画面定義上のポイント

作成する GUI 画面の基になる定義サンプルを次に示します。

- イベント単位入力画面の定義サンプル：EVE1GC

定義サンプルを使用しない場合は、レイアウトパターンの「フリー」を選んで定義してください。

## 8. 標準パターンを使用した AP の作成

イベント単位入力画面の作成時には、次の注意が必要です。

- 画面属性ダイアログの「入力単位」で「イベント」を指定する。

### (3) AP 作成上のポイント

作成する AP の基になる AP パターンを次に示します。

- メインプログラムで使用する AP パターン名：GENEVN01

この AP パターンを使用する AP 作成時には次の注意が必要です。

- イベントごとの処理のセクションを作成する。
- 画面表示および入力した後、画面から AP に返されたイベント通知コードを判定する処理を作成する。入力または選択可能なオブジェクトから返されるイベント通知コードには、イベント定数が格納される。

### (4) コーディング例

この例題のソースプログラムを 'EVENTGC.cbl' として標準提供しています。ソースプログラムからコーディング上のポイントを抜粋して次に示します。



```

002100 WORKING-STORAGE SECTION.
002200*
003900*      論理マップの取り込み
004000      COPY     EVE1GC0.
004100      COPY     EVE1GC1.
004200*
004300*  動的変更テーブルの取り込み --"XMAP3インストール先%INCLUDE"にあります
004400      COPY     X3MODTBL.
004500*
      :

006000 PROCEDURE DIVISION.
006100
008700***** マップ名の設定 *****
008800      MOVE 'EVE1GCND' TO 画面マップ名.          . . . 1
008900
009000      MOVE      +0      TO 画面1-フラグ.
009100      PERFORM UNTIL 画面1-フラグ = 終了 OR
009200                  画面1-フラグ = キャンセル
009300
009400***** 画面の表示と入力 *****
009500      TRANSCEIVE DSP FROM EVE1GC0 INTO EVE1GC1 . . . 2
009600
010600*****
010700* イベント通知コードとして格納される値により処理を振り分ける *
010800*     イベント定数    ---> トグルフィールドからのイベント *
010900*                   フィールドごとのルーチンへ *
011000*     INC定数      ---> ボタン・PFキーからのイベント *
011100*                   該当する処理へ *
011200*****
011300      EVALUATE EVE1GC-INC1
011400      WHEN EVE1GC-チェック-V          . . . 3
011500**----- トグルフィールドからのイベント -----**
011600      PERFORM フィールド-イベント処理
011700*
011800**----- ボタン・PFキーからのイベント -----**
011900      WHEN 'PF04'          . . . 4
012000      MOVE 終了          TO 画面1-フラグ
012100      WHEN 'PF03'
012200      MOVE キャンセル TO 画面1-フラグ
012300      WHEN 'PF02'
012400      MOVE 終了          TO 画面1-フラグ
013600      WHEN OTHER
013700      CONTINUE
013800      END-EVALUATE
013900
014000      END-PERFORM.
014100
      :

```

(続く)

## 8. 標準パターンを使用した AP の作成

```
(続き)
015400*
015500*---- 各オブジェクトごとの処理 -----*
016000 フィールドイベント処理 SECTION.
016100
016700***** トグルフィールドのカーソル定数を判定 *****
016800 EVALUATE EVE1GC-INC1
016900 WHEN EVE1GC-チェック-T(01)
017000 PERFORM チェック1イベント処理
017100 WHEN EVE1GC-チェック-T(02)
:
018900 END-EVALUATE.
019000
019100 フィールドイベント処理-END.
019200 EXIT.
019300
019400*****
019500* チェックを判定して発注承認を行う -チェック1- *
019600*****
019700 チェック1イベント処理 SECTION. . . . 5
019800
019900* トグルフィールドがチェックされたかの判定
020000 IF EVE1GC-チェック-I(1) = '01'
020100 THEN
020200 MOVE N'発注' TO EVE1GC-チェック結果-0(1)
020300 ELSE
020400 MOVE ALL SPACE TO EVE1GC-チェック結果-0(1)
020500 END-IF.
020600
020700*----- ユーザ処理の追加 -----
020800
020900 チェック1イベント処理-END.
021000 EXIT.
023000*
023100*****
023200*** 以下, トグルフィールドの個数分, 処理を繰り返して下さい **
023300*****
```

1. マップ名を変更する。  
プログラム中の '@@@EEE' を 'EVE1GC' に変更します。
2. 画面の表示, およびデータ入力。
3. 入力または選択可能なオブジェクトからイベントが返ると, 通知コードとして, そのオブジェクトのイベント定数が格納される。
4. ボタンやファンクションキーからイベントが返ると, 通知コードとして, そのボタンやファンクションキーに対応する INC 定数が格納される。
5. イベントを返したオブジェクトに対する処理をする。

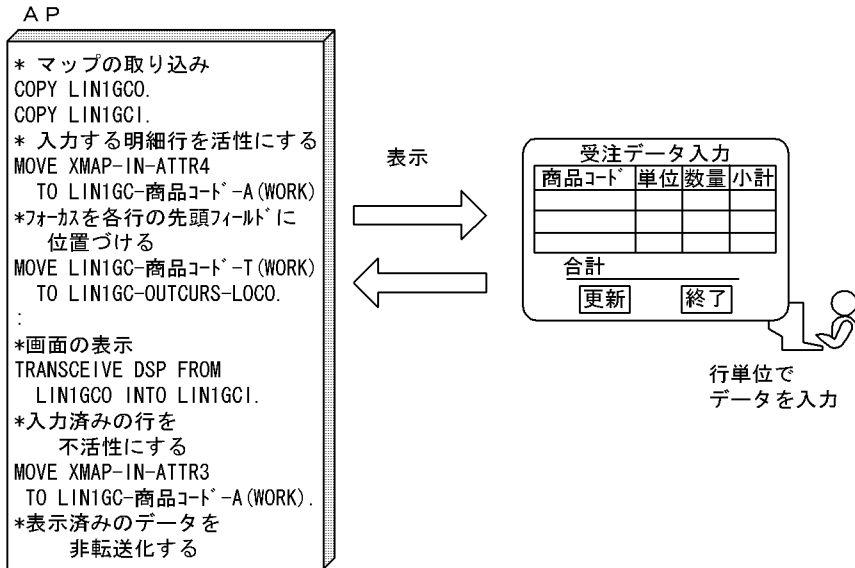
## 8.2.7 GENDSP01 パターンを使用した行単位のデータ入力

### (1) 処理の概要

受注データ入力画面を表示します。1行ずつデータが入力できる状態になり、行単位で入力を確認して、入力データを処理します。

受注データ入力画面での、データ入力処理の概要を次の図に示します。

図 8-7 受注データ入力画面でのデータ入力処理



### (2) 画面定義上のポイント

作成する GUI 画面の基になる定義サンプルを次に示します。

- 受注データ入力画面の定義サンプル：LIN1GC

定義サンプルを使用しない場合は、レイアウトパターンの「フリー」を選んで定義してください。

受注データ入力画面の作成時には、次の注意が必要です。

- すべての入出力フィールドに「入力不可 標準表示」の表示属性を指定する。
- すべての入出力フィールドに「動的変更 (AP から表示属性を変更する)」の属性を指定する。
- 行の最後の入出力フィールド (数量) には「自動送信 (入力直後に AP へ制御を渡す)」の属性を指定する。

## 8. 標準パターンを使用した AP の作成

### (3) AP 作成上のポイント

作成する AP の基になる AP パターンと AP 部品を次に示します。

- メインプログラムで使用する AP パターン名：GENDSP01
- サブルーチンで使用する AP 部品名：PATWRT01

この AP パターンを使用する AP 作成時には、次の注意が必要です。

- 行単位で、順にデータ入力、表示処理をする。このとき、1 行ずつ入力できるように行単位で「入力可能 標準表示」の表示属性を動的に変更する。
- データ処理後の表示済みのフィールドは 2 回目の画面表示以降ではデータを送らないようにして、必要以上に画面がちらつかないようにする。
- ファイルの入出力処理は、必要に応じて追加する。

### (4) コーディング例

この例題のソースプログラムを 'LINEGC.cbl' として標準提供しています。ソースプログラムからコーディング上のポイントを抜粋して次に示します。

```

WORKING-STORAGE SECTION.
* 論理マップのコピー展開
COPY LIN1GCO. . . . 1
COPY LIN1GCI.
* 動的変更テーブルのコピー展開
COPY X3MODTBL.
*****
PROCEDURE DIVISION.
* 端末名の設定
MOVE 'DSP001' TO 画面端末名.
* 出力エリアのクリア
MOVE ALL X'1F' TO LIN1GCG.
*****
行単位に処理を行う *****
MOVE 1 TO WORK.
PERFORM UNTIL ( WORK > GYO-CNT )
* 入力する明細行の入力フィールドを活性とする . . . 2
MOVE XMAP-IN-ATTR4 TO LIN1GC-商品コード-A(WORK)
MOVE XMAP-IN-ATTR4 TO LIN1GC-数量-A(WORK)
* フォーカスの設定 (各行の先頭フィールドに設定) . . . 3
MOVE LIN1GC-商品コード-T(WORK) TO LIN1GC-OUTCURS-LOC
* マップ名の設定
MOVE 'LIN1GCND' TO 画面マップ名
* 画面の表示と入力 . . . 4
TRANSCIEVE DSP FROM LIN1GCO INTO LIN1GCI
:
* ボタン・PFキーのチェックと該当処理の呼び出し
EVALUATE LIN1GC-INCI
WHEN 'ENTR'
***** 一行ごとの処理 *****
PERFORM 明細行計算処理
***** 次行の処理をするため、行カウンタを更新 *****
COMPUTE WORK = WORK + 1
WHEN 'PF02'
***** 全ての行が処理し終わったら *****
PERFORM 更新処理
WHEN 'PF10'
MOVE +0 TO RETURN-CODE
GO TO 正常終了
WHEN OTHER
CONTINUE
END-EVALUATE
END-PERFORM.
:
*****
* 明細部の計算処理 *
*****
明細行計算処理 SECTION.
* 入力した明細行の処理を行う
:
* 入力した明細行の入力フィールドを不活性に戻す . . . 5
MOVE XMAP-IN-ATTR3 TO LIN1GC-商品コード-A(WORK)
MOVE XMAP-IN-ATTR3 TO LIN1GC-数量-A(WORK)
* ひとつ前の行の出力データをクリアする . . . 6
PERFORM 前行クリア処理.
*
明細部計算処理-END.
EXIT.

```

## 8. 標準パターンを使用した AP の作成

1. マップ名を変更する。  
プログラム中の '@@555' を 'LIN1GC' に変更してください。
2. 入力する行の入力フィールドを活性に変更する。
3. フォーカスの位置づけ。
4. 画面の表示とデータの入力。
5. 入力した行の入力フィールドを不活性に戻す。
6. 表示済みのフィールドに対するデータを転送しないようにする。

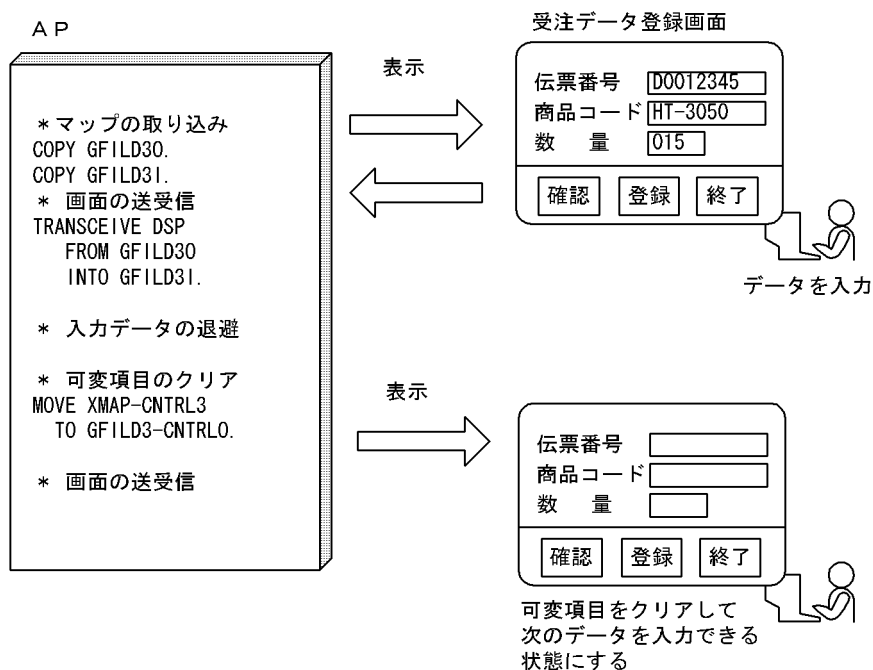
### 8.2.8 GENDSP01 パターンを使用した入力フィールドのクリア

#### (1) 処理の概要

受注データ登録画面を表示します。受注伝票の内容を入力後に [ 確認 ] ボタンまたは [ 登録 ] ボタンを選ぶと、表示画面はそのままにして入力したデータをすべてクリアし、次の伝票を入力できる状態にします。

受注データ登録画面での、データ入力処理の概要を次の図に示します。

図 8-8 受注データ登録画面でのデータ入力処理



#### (2) 画面定義上のポイント

作成する GUI 画面の基になる定義サンプルを次に示します。

- ・ 受注データ登録画面の定義サンプル：GFILD3

受注データ登録画面の作成時には、次の注意が必要です。

- ・ 各入出力フィールドに初期値は指定しない。

### (3) AP 作成上のポイント

作成する AP の基になる AP パターンと AP 部品を次に示します。

- ・ メインプログラムで使用する AP パターン名：GENDSP01
- ・ サブルーチンで使用する AP 部品名：CLRINP01L

この AP パターンを使用する AP 作成時には、次の注意が必要です。

- ・ 2 回目以降の画面表示ではマッピングオプションを「マージ」に、画面属性ダイアログのウィンドウ属性の「入力・選択状態の扱い」を「初期状態」に変更して送信する。
- ・ 入力データのエラーチェックは必要に応じて追加する。
- ・ 入力データをデータベースなどに書き込む処理を追加する。

### (4) コーディング例

この例題のソースプログラムを 'DSPCLEAR.cbl' として標準提供しています。ソースプログラムからコーディング上のポイントを抜粋して次に示します。

```

WORKING-STORAGE          SECTION.
* 論理マップのCOPY展開
* メニュー画面用
COPY  GFILD30.            ..... 1.
COPY  GFILD31.
COPY  X3MODTBL.

PROCEDURE                 DIVISION.
* マップ名の設定
MOVE  'GFILD3ND' TO 画面マップ名
* 画面の表示とデータの入力
TRANSCEIVE DSP FROM GFILD30 INTO GFILD31
画面1-セクション        SECTION. .... 2.
* 入力項目をクリアして入力項目だけを一部上書
MOVE  XMAP-CNTRL3      TO GFILD3-CNTRLO. .... 3.
MOVE  SPACE TO マッピングモード.

```

1. マップ名を変更する。  
プログラム中の '@@@555' を 'GFILD3' に変更します。
2. AP パターン 'GENDSP01' に AP 部品 'CRLINP01' を取り込む。
3. 2 回目の表示でマッピングオプションおよびウィンドウ属性を変更する処理を追加する。

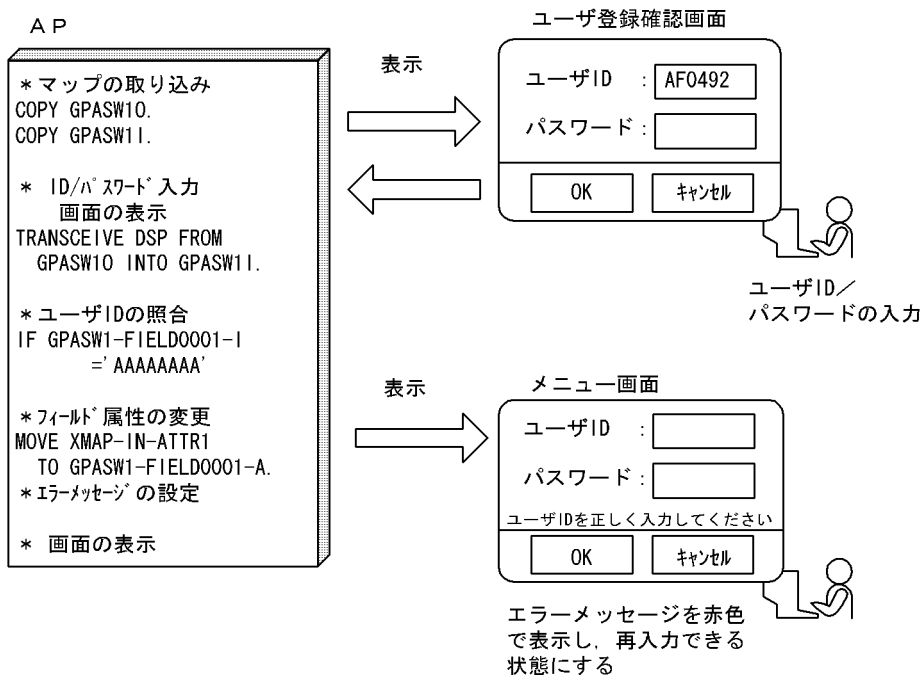
## 8.2.9 GENDSP01 パターンを使用した AP からのフィールド属性の変更

### (1) 処理の概要

入力したデータに誤りがあった場合、エラーメッセージを表示します。このときエラーとなったフィールドを赤色にしてカーソルを設定し、データの再入力を促します。

エラー発生時のフィールド表示の変更処理の概要を次の図に示します。

図 8-9 エラー発生時のフィールド表示の変更処理



### (2) 画面定義上のポイント

作成する GUI 画面の基になる定義サンプルを次に示します。

- ユーザ登録画面の定義サンプル : GPASW1

画面作成時には次の注意が必要です。

- エラーメッセージ表示用の出力フィールドの使用目的を「日本語」に変更する。

### (3) AP 作成上のポイント

作成する AP の基になる AP パターンと AP 部品を次に示します。

- メインプログラムで使用する AP パターン名 : GENDSP01



- サブルーチンで使用する AP 部品名 : MODATR01

この AP パターンを使用する AP 作成時には、次の注意が必要です。

- 画面の初期表示のときと、エラー表示のときにはマッピングオプションを変更している。
- 入力したユーザ ID が 'AAAAAAAA' のときエラーにするという仕様は、ユーザ独自の仕様に変更する。
- エラーメッセージは 1 種類だけ用意している。

#### (4) コーディング例

この例題のソースプログラムを 'FILDCHG1.cbl' として標準提供しています。ソースプログラムからコーディング上のポイントを抜粋して次に示します。

```

WORKING-STORAGE SECTION.
* 論理マップのCOPY展開
* メニュー画面用
COPY GPASW10. .... 1.
COPY GPASW11.
COPY X3MODTBL.
PROCEDURE DIVISION.
* マップ名の設定
MOVE 'GPASW1ND' TO 画面マップ名
* 画面の表示とデータの入力
TRANSCIEVE DSP FROM GPASW10 INTO GPASW11
画面1-セクション SECTION. .... 2.
IF GPASW1-FIELD0001-I = 'AAAAAAAA' .... 3.
THEN ....4.
MOVE XMAP-IN-ATTR1 TO GPASW1-FIELD0001-A
MOVE 'ユーザ ID を正しく入力してください'
TO GPASW1-FIELD0003-0
ELSE
MOVE SPACE TO GPASW1-FIELD0003-0
END-IF.

```

1. マップ名を変更する。  
プログラム中の '@@@555' を 'GPASW1' に変更します。
2. AP パターン 'GENDSP01' に AP 部品 'MODATR01' を取り込む。
3. エラーチェック処理を追加する。
4. エラー表示、フィールド表示属性を変更するコーディングを追加する。

### 8.2.10 GENDSP01 パターンを使用した表形式の一覧表示

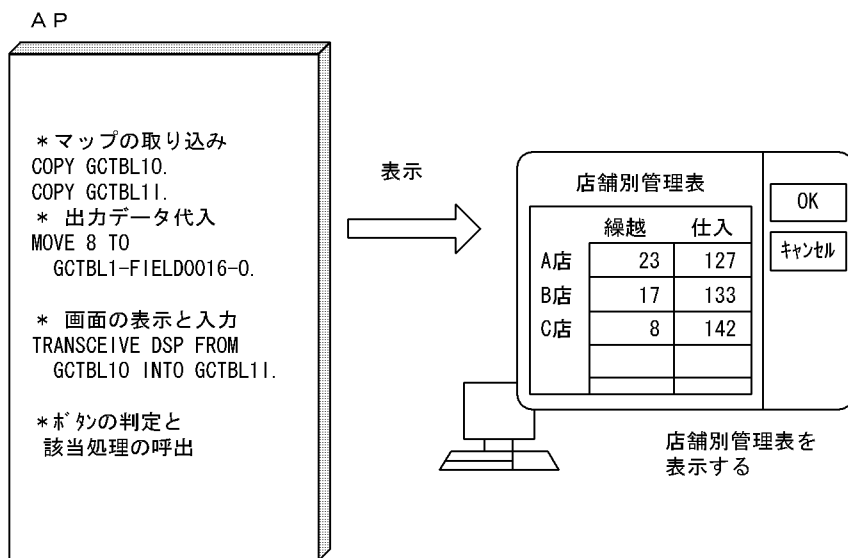
#### (1) 処理の概要

店舗別の在庫管理表を一覧で表示します。メニューバーの [ ファイル ] - [ 終了 ], または [ キャンセル ] ボタンを選ぶことで画面表示を終了します。

## 8. 標準パターンを使用した AP の作成

表形式の一覧の表示の処理概要を次の図に示します。

図 8-10 表形式の一覧の表示処理



### (2) 画面定義上のポイント

作成する GUI 画面の基になる定義サンプルを次に示します。

- 店舗別管理表の定義サンプル：GCTBL1

定義サンプルを使用しない場合は、レイアウトパターンの「表」を選んで定義してください。

店舗別管理表の定義サンプルには、次の特長があります。

- 各項目はすべて出力フィールドで定義しているため、画面からの入力はない。
- 出力フィールドには「動的変更 (AP から表示属性を変更する)」の属性を指定していない。

### (3) AP 作成上のポイント

作成する AP の基になる AP パターンを次に示します。

- メインプログラムで使用する AP パターン名：GENDSP01

この AP パターンを使用する AP 作成時には次の注意が必要です。

- 該当しないファンクションキーが押された場合は、画面を再表示するだけの仕様のため、エラー処理は必要に応じて追加する。
- ファイルの入出力処理は、必要に応じて追加する。

#### (4) コーディング例

この例題のソースプログラムを 'TABLEOUT.cbl' として標準提供しています。ソースプログラムからコーディング上のポイントを抜粋して次に示します。

```

WORKING-STORAGE SECTION.
* 論理マップのCOPY展開
* メニュー画面用
COPY GCTBL10. .... 1.
COPY GCTBL11.
COPY X3MODTBL.
PROCEDURE DIVISION.
PERFORM 前頁データ設定. .... 2.
* マップ名の設定
MOVE 'GCTBL1ND' TO 画面マップ名
* 画面の表示と入力
TRANSCIEVE DSP FROM GCTBL10 INTO GCTBL11
* ボタンの判定と該当処理の呼び出し
EVALUATE GCTBL1-INC1 ..... 3.
  WHEN 'PF10'
    MOVE キャンセル TO 画面1-フラグ
  WHEN 'A002'
    MOVE キャンセル TO 画面1-フラグ
END-EVALUATE
:
前頁データ設定 SECTION. ....2.
MOVE 11 TO GCTBL1-FIELD0001-0(1).
MOVE 12 TO GCTBL1-FIELD0001-0(2).
:
MOVE 7 TO GCTBL1-FIELD0018-0.
前頁データ設定-END.

```

1. マップ名を変更する。  
プログラム中の '@@555' を 'GCTBL1' に変更します。
2. 出力データの代入処理をする。
3. 終了の判定を変更する。

### 8.2.11 GENDSP01 パターンを使用した出力フィールドの再表示

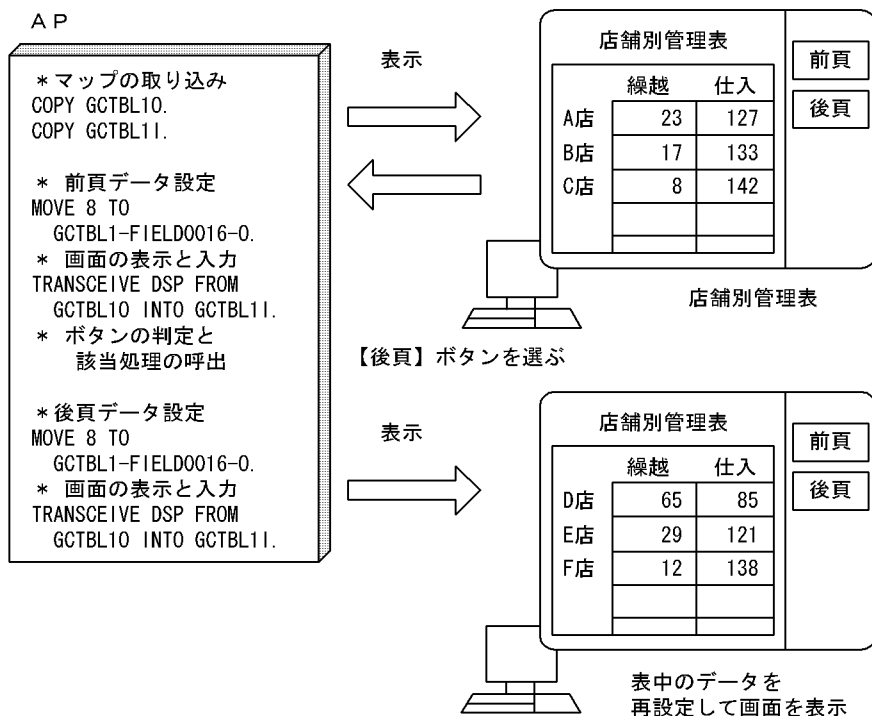
#### (1) 処理の概要

店舗別の在庫管理表を一覧で表示します。[ 後頁 ] のプッシュボタンをクリックすると、表中に次のデータを表示します。

表中の変変データだけを入れ換えて表示する処理の概要を次の図に示します。

## 8. 標準パターンを使用した AP の作成

図 8-11 表中の可変データだけを再表示する処理



### (2) 画面定義上のポイント

作成する GUI 画面の基になる定義サンプルを次に示します。

- 店舗別管理表の定義サンプル：GCTBL1

定義サンプルを使用しない場合は、レイアウトパターンの「表」を選んで定義してください。

店舗別管理表の画面サンプルには、次の特長があります。

- 各項目はすべて出力フィールドで定義しているため、画面からの入力はない。
- 出力フィールドには「動的変更（AP から表示属性を変更する）」の属性は指定していない。
- [キャンセル],[後頁],[前頁] ボタンがクリックされたときだけの処理をサポートしている。ほかのボタンがクリックされた場合は画面を再表示する。

### (3) AP 作成上のポイント

作成する AP の基になる AP パターンを次に示します。

- メインプログラムで使用する AP パターン名：GENDSP01

この AP パターンを使用する AP 作成時には、次の注意が必要です。

- 2度目の画面表示では、マッピングオプションを「論理マップだけ」に指定する。
- サポートしていないプッシュボタンをクリックされた場合は、画面の再表示だけをする。エラー処理は、必要に応じて追加する。
- ファイル入出力処理は行っていないため、必要に応じて追加する。表示データは固定となっている。

#### (4) コーディング例

この例題のソースプログラムを 'DSPWRITE.cbl' として提供しています。ソースプログラムからコーディング上のポイントを抜粋して次に示します。

```

WORKING-STORAGE SECTION.
* 論理マップのCOPY展開
* メニュー画面用
COPY GCTBL10. ....1.
COPY GCTBL11.
COPY X3MODTBL.
PROCEDURE DIVISION.
PERFORM 前頁データ設定. ....2.
* マップ名の設定
MOVE 'GCTBL1ND' TO 画面マップ名
* 画面の表示と入力
TRANSCEIVE DSP FROM GSTBL10 INTO GCTBL11
* ボタンの判定と該当処理の呼び出し
EVALUATE GCTBL1-INC1 .....3.
  WHEN 'PF10'
    MOVE キャンセル TO 画面1-フラグ
  WHEN 'A002'
    MOVE キャンセル TO 画面1-フラグ
  WHEN 'A003'
    PERFORM 画面1-セクション
  WHEN 'A004'
    PERFORM 画面1-セクション
END-EVALUATE
:
画面1-セクション SECTION.
EVALUATE GCTBL1-INC1 .....2.
  WHEN 'A003'
    PERFORM 前頁データ設定
  WHEN 'A004'
    PERFORM 後頁データ設定
END-EVALUATE.
前頁データ設定 SECTION.
MOVE '2' TO マッピングモード. .... 4.
MOVE 8 TO GCTBL-FIELD0016-0.....2.
MOVE 5 TO GCTBL-FIELD0017-0
後頁データ設定 SECTION.
MOVE '2' TO マッピングモード. ....4.
MOVE 8 TO GCTBL-FIELD0016-0.....2.
MOVE 6 TO GCTBL-FIELD0017-0
MOVE 2 TO GCTBL-FIELD0018-0

```

## 8. 標準パターンを使用した AP の作成

1. マップ名を変更する。  
プログラム中の '@@@555' を 'GCTBL1' に変更します。
2. 出力データの代入処理。
3. ボタンの判定処理。
4. マッピングオプションを設定する。

### 8.2.12 GENDSP01 パターンを使用したヘルプ画面の表示

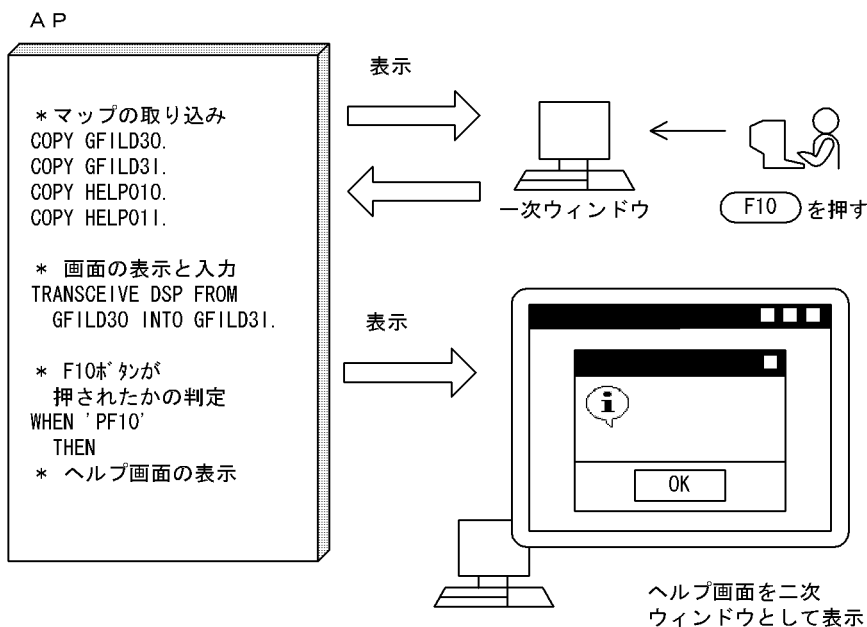
#### (1) 処理の概要

受注データ登録画面を一次ウィンドウとして表示します。「顧客コード」または「商品コード」にカーソルがあるときに [ F10 ] を押すと、それぞれの項目の入力を補助するためのヘルプ画面を二次ウィンドウとして表示します。

ヘルプ画面から一次ウィンドウに戻ると、ヘルプ画面は消えて、一次ウィンドウが活性状態になります。入力途中でヘルプ画面を表示した場合でも、そのデータは保証されません。

顧客コードの項目に対するヘルプ画面を表示する場合の処理の概要を次の図に示します。

図 8-12 ヘルプ画面の表示処理



#### (2) 画面定義上のポイント

作成する GUI 画面の基になる定義サンプルを次に示します。

- 受注データ登録画面の定義サンプル : GFILD3

- ヘルプ画面の定義サンプル：HELP01

ヘルプ画面のレイアウトは、任意に定義してください。

一次ウィンドウ作成時には、次の注意が必要です。

- 一次ウィンドウに対する二次ウィンドウの表示位置をあらかじめ指定しておく。
- 一次ウィンドウには、ヘルプ画面を表示するためのファンクションキー、またはプッシュボタンを割り当てておく。

二次ウィンドウであるヘルプ画面を作成するときには、次の注意をする必要があります。

- サンプルは一次ウィンドウのため、二次ウィンドウとして指定し直す必要がある。

またヘルプ画面には、次の特長があります。

- [OK] ボタンのプッシュボタンダイアログで「二次ウィンドウを閉じる」の属性が指定してある。
- 画面に入出力フィールドは配置していない。

### (3) AP 作成上のポイント

作成する AP の基になる AP パターンと AP 部品を次に示します。

- メインプログラムで使用する AP パターン：GENDSP01
- サブルーチンで使用する AP 部品：FLDHLP01

この AP 部品を使用する AP 作成時には、次の注意が必要です。

- 一次ウィンドウの再表示ではマッピングオプションを変更する。
- ヘルプ画面消去後も一次ウィンドウでの入力データを保持するために、入力フォーカス項目とフォーカス定数を比較して、フォーカスのあるフィールドを特定する。

### (4) コーディング例

この例題のソースプログラムを 'HELPOUT.cbl' として標準提供しています。ソースプログラムからコーディング上のポイントを抜粋して次に示します。

## 8. 標準パターンを使用した AP の作成

```
WORKING-STORAGE                                SECTION.
* 論理マップのCOPY展開
* 受注データ登録画面用
COPY   GFILD30. .... 1.
COPY   GFILD31.
* ヘルプ画面用
COPY   HELP010. .... 1.
COPY   HELP011.
COPY   X3MODTBL.
PROCEDURE                                       DIVISION.
* マップ名の設定
MOVE 'GFILD3ND' TO 画面マップ名
* ボタンの判定と該当処理の呼び出し
EVALUATE GCTBL1-INC1 ..... 2.
  WHEN 'PF10'
    PERFORM H E L P 処理
END-EVALUATE
:
H E L P 処理 SECTION. .... 3.
EVALUATE GFILD3-INCURS-LOC1 ..... 4.
* 顧客コードにカーソルがある場合
  WHEN GFILD3-KCODE-T
* マップ名の設定
  MOVE 'HELP01ND' TO 画面マップ名
* 「顧客コード」に対応するヘルプ画面を表示
  TRANSCEIVE DSP FROM HELP010 INTO HELP011..... 5.
* 「商品コード」にカーソルがある場合
  WHEN GFILD3-CODE-T
* マップ名の設定
  MOVE 'HELP01ND' TO 画面マップ名
* 「顧客コード」に対応するヘルプ画面を表示
  TRANSCEIVE DSP FROM HELP010 INTO HELP011.....5.
END-EVALUATE.
```

1. マップ名を変更する。  
プログラム中の '@@@555' を 'GFILD3' に変更し, 'HELP010' と 'HELP011' の取り込み処理を追加します。
2. [ F10 ] を押したかどうかのチェック処理を追加する。
3. AP パターン 'GENDSP01' に AP 部品 'FLDHLP01' を取り込み, '@@@333' を 'HELP01' に変更する。
4. カーソルのあるフィールドを特定する処理を追加する。
5. ヘルプ画面を表示する処理を追加する。

### 8.2.13 GENDSP01 パターンを使用したダイアログの表示

#### (1) 処理の概要

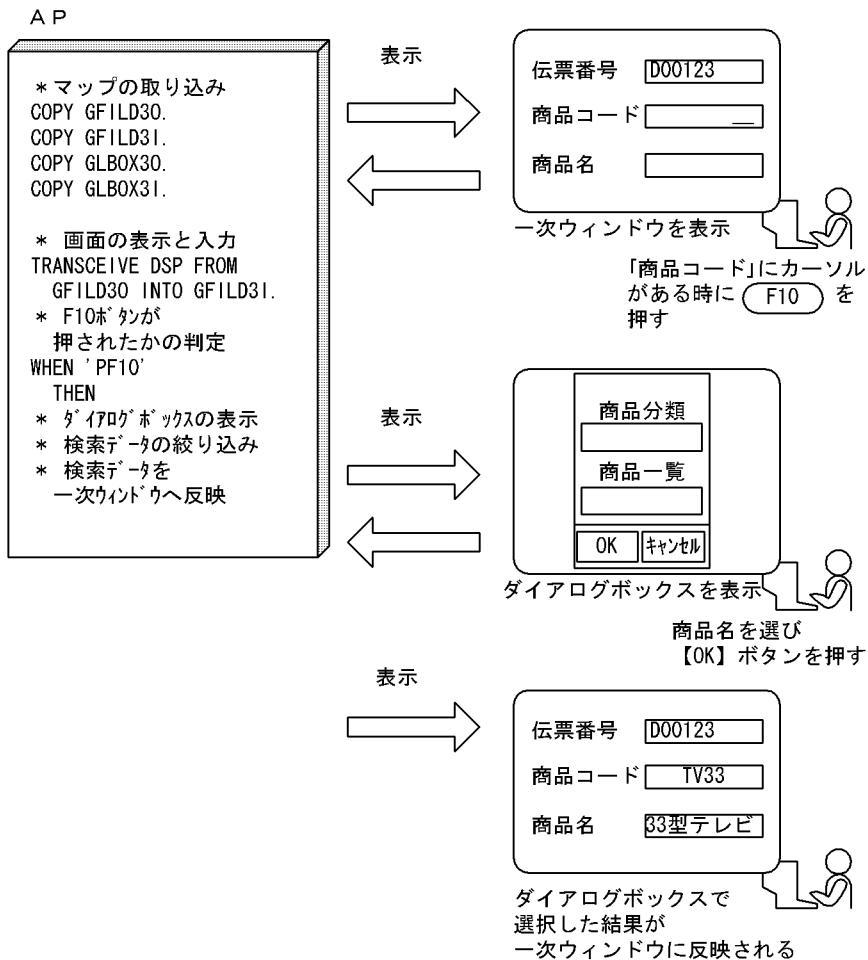
受注データ登録画面を一次ウィンドウとして表示します。[ F10 ] を押すと, 商品を検索できるダイアログを二次ウィンドウとして表示します。



ダイアログは、受注データ登録画面に重ねて表示されます。ダイアログでは、はじめに「商品分類」のリストで商品の候補を絞り込みます。次に「商品一覧」のリストから商品名を選んで [OK] ボタンをクリックすると、ダイアログが消えて、受注データ入力画面の「商品名」の欄に選んだ商品名が反映されます。

ダイアログを使用した入力補助の概要を次の図に示します。

図 8-13 ダイアログを使用した入力補助



## (2) 画面定義上のポイント

作成する GUI 画面の基になる定義サンプルを次に示します。

- 受注データ登録画面の定義サンプル：GFILD3

定義サンプルを使用しない場合は、レイアウトパターンの「中汎用ウィンドウ」を使用して定義してください。

## 8. 標準パターンを使用した AP の作成

- ダイアログのサンプル：GLBOX3

複数選択リストボックスを使用した「商品一覧」のダイアログです。画面のサンプルを使用しない場合は、レイアウトパターンの「二次ウィンドウ/二段リストボックス（複数選択）」を使用して定義してください。

一次ウィンドウの作成時には、次の注意が必要です。

- 一次ウィンドウに対する二次ウィンドウの表示位置をあらかじめ指定しておく。
- 一次ウィンドウにダイアログを表示するためのファンクションキー、またはプッシュボタンを割り当てておく。

二次ウィンドウの作成時には、次の注意が必要です。

- サンプルは一次ウィンドウのため、二次ウィンドウとして指定し直す必要がある。

また、二次ウィンドウには次の特長があります。

- [OK] ボタンのプッシュボタンダイアログで「二次ウィンドウを閉じる」の属性が指定してある。
- 二次ウィンドウの結果は固定的に一次ウィンドウに反映される。

### (3) AP 作成上のポイント

作成する AP の基になる AP パターンを次に示します。

- メインプログラムで使用する AP パターン：GENDSP01

この AP パターンを使用する AP 作成には、次の注意が必要です。

- 一次ウィンドウの再表示ではマッピングオプションを設定する。
- ダイアログを表示する場合と、一次ウィンドウを表示する場合との区別をする。
- ダイアログの再表示ではマッピングオプションを設定する。
- 一次ウィンドウや二次ウィンドウでのデータ入力のチェック処理は含まれていないため、必要があれば追加する。
- ダイアログに表示するデータは、固定となっている。必要があれば、マスタファイルから入力して代入する処理などに変更する。

### (4) コーディング例

この例題のソースプログラムを 'DAILOG.CBL' として標準提供しています。ソースプログラムからコーディング上のポイントを抜粋して次に示します。

```

WORKING-STORAGE SECTION.
* 論理マップのCOPY展開
* 受注データ登録画面用
COPY GFILD30. .... 1.
COPY GFILD31.
* ダイアログボックス用
COPY GLBOX30. .... 1.
COPY GLBOX31.
COPY X3MODTBL.
PROCEDURE DIVISION.
* マップ名の設定
MOVE 'GFILD3ND' TO 画面マップ名
* 画面の表示と入力
TRANSCERVE DSP FROM GFILD30 INTO GFILD31
* ボタンの判定と該当処理の呼び出し
EVALUATE GFILD3-INCI .....2.
  WHEN 'PF10'
    PERFORM ダイアログ-セクション
    MOVE '2' TO マッピングモード
    MOVE XMAP-CNTRL1 TO GFILD-CNTRLO
  WHEN 'PF12'
    MOVE キャンセル TO 画面1-フラグ
END-EVALUATE
:
ダイアログ-セクション SECTION. .... 3.
EVALUATE GFILD3-INCURS-LOC1 ..... 4.
  WHEN 'ENTER'
    MOVE 'カラーテレビ33型' TO GLBOX3-ILABL-0(1)
    MOVE 'TV33' TO GLBOX3-ICODE-0(1)
    MOVE GLBOX3-INAME-T(1) TO GLBOX3-OUTCURS-LOC0
END-EVALUATE.
画面2-セクション SECTION. .... 5.
IF GLBOX3-INAME-I(1) = 'TV33'
  MOVE 'カラーテレビ33型' TO GFILD3-NAME-0
END-IF.

```

1. マップ名を変更する。  
プログラム中の '@@@555' を 'GFILD3' に変更します。
2. ファンクションキーの判定処理。
3. AP パターン 'GENDSP01' に AP 部品 'NXTDSP01' を取り込み, '@@@666' を 'GLBOX3' に変更する。
4. 二次ウィンドウの再表示処理 (絞り込み後の選択)。
5. 二次ウィンドウの選択結果の一次ウィンドウへの反映。

## 8.2.14 GENDSP01 パターンを使用した掲示板の表示

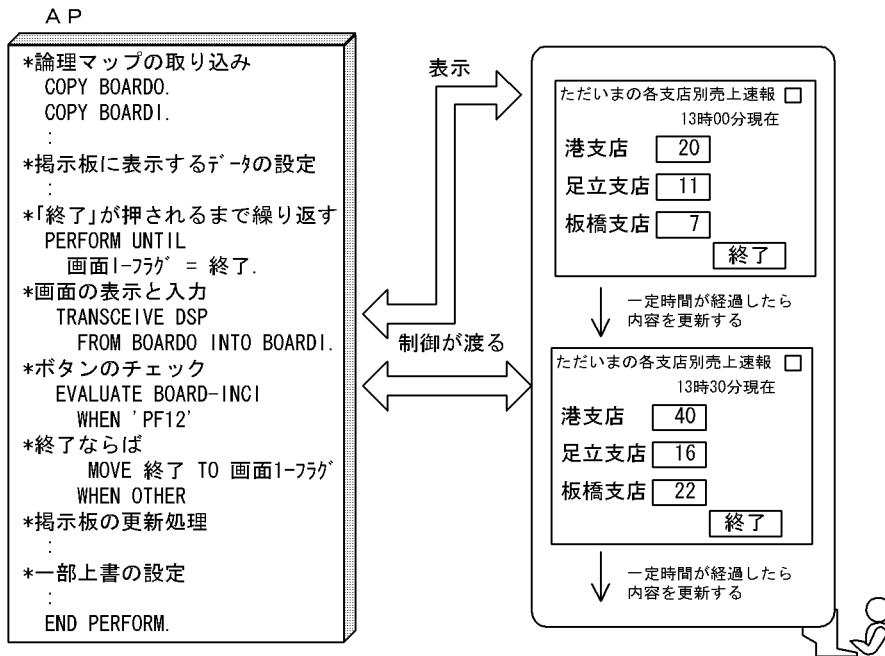
### (1) 処理の概要

画面の右上に、常に手前に表示されるウィンドウ (掲示板) を設定します。

表示内容は、定期的に AP から書き換えます。また、ユーザが強制的に終了させることもできます。掲示板の処理概要を次の図に示します。

## 8. 標準パターンを使用した AP の作成

図 8-14 掲示板の表示と書き換え，強制終了



### (2) 画面定義上のポイント

作成する GUI 画面の基になる定義サンプルを次に示します。

- 掲示板画面の定義サンプル：BOARD

定義サンプルを使用しないで掲示板画面を作成するときには，次の注意が必要です。

- [ 終了ボタン ] を定義する。
- 画面属性ダイアログで「入力単位」の「表示直後」を指定する。「表示直後」指定時は制御が AP に渡るまでの待機時間を設定することもできる。待機時間はユーザ任意。
- 画面属性ダイアログで「XY 位置」の「右上」と、「Z 位置」の「手前に表示」をそれぞれ指定する。

### (3) AP 作成上のポイント

作成する AP の基になるプログラムパターンを次に示します。

- メインプログラムで使用する AP パターン：GENDSP01

この AP パターンを使用しての AP 作成時には，次の注意が必要です。

- AP に制御が渡ったとき，まず [ 終了 ] ボタンがクリックされたかどうかを判定する。
- [ 終了 ] ボタンがクリックされていないければ，掲示板に表示する最新のデータを入力し，画面を再表示する。このとき，表示形態では「一部上書」を指定する。表示処理

を [ 終了 ] ボタンがクリックされるまで繰り返すようにする。

#### (4) コーディング例

```

WORKING-STORAGE SECTION.
*
*      01 画面 1-フラグ PIC S9(2).
*
*      論理マップの COPY 展開
COPY   BOARD0. ....1.
COPY   BOARD1.
      :
PROCEDURE DIVISION.
      :
*      マップ名の設定
MOVE 'BOARDND' TO 画面マップ名
*
*----- 掲示板に表示したいデータを設定して下さい -----*
*----- データベースなどから最新データを取り出して -----*
*----- 更新します。ここでは、固定値を設定しています -----*
*
MOVE +0 TO 画面 1-フラグ.
PERFORM UNTIL 画面 1-フラグ = 終了
*----- 画面の表示と入力 -----*
TRANSCEIVE DSP FROM BOARD0 INTO BOARD1 .....2.
*----- ボタン・Fキーのチェックと該当処理の呼び出し -----*
EVALUATE BOARD-INC1
WHEN 'PF12' .....3.
*----- 掲示板の表示を終了します -----*
MOVE 終了 TO 画面 1-フラグ
WHEN OTHER .....4.
*----- 掲示板の表示内容を更新します -----*
PERFORM 掲示板更新-セクション
END-EVALUATE
END-PERFORM.
      :
*****
*      掲示板更新-セクション
*      掲示板の表示内容を更新します
*****
掲示板更新-セクション SECTION. ....4.
*----- 出力論理マップの初期化 -----*
MOVE ALL 'X' '1F' TO BOARDG.
*
*----- 掲示板の更新処理を追加して下さい -----*
*----- データベースなどから最新データを取り出して -----*
*----- 更新します。ここでは、固定値を設定しています -----*
*----- 画面の再表示 (一部上書) -----*
MOVE XMAP-CNTRL1 TO BOARD-CNTRLO .....5.
MOVE 2 TO マッピングモード.

掲示板更新-セクション-END.
EXIT.
*****

```

1. マップ名を変更する。  
プログラム中の '@@@555' を 'BOARD' に変更します。

## 8. 標準パターンを使用した AP の作成

2. 画面の表示と入力処理。
3. [ 終了 ] ボタンがクリックされたら、終了する。
4. [ 終了 ] ボタンがクリックされていないならば、掲示板の更新処理をする。
5. 表示形態を「一部上書」にして再表示する。

### 8.2.15 DSPPRT01 パターンを使用した画面表示と帳票印刷

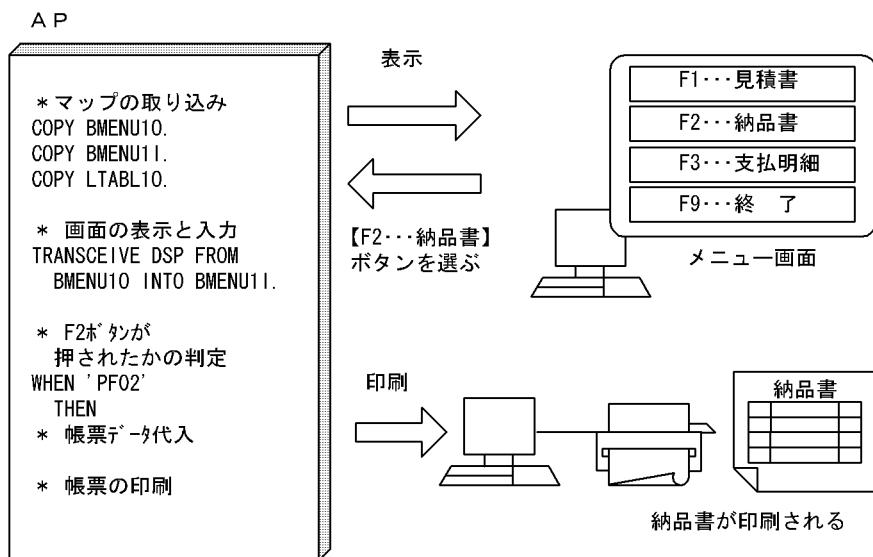
XMAP3/NET を使用すると、画面と帳票を定義できます。一つの AP で画面の表示と帳票の印刷をする業務を想定した例題を説明します。

#### (1) 処理の概要

各帳票を印刷するためのメニュー画面を表示します。メニューから [ F2... 納品書 ] のプッシュボタンを選べると、納品書の帳票を印刷します。このとき、帳票はシリアルインパクトプリンタへ印刷されます。

メニュー画面の表示と帳票印刷の処理の概要を次の図に示します。

図 8-15 メニュー画面の表示と帳票印刷



#### (2) 画面定義上のポイント

作成する画面・帳票の基になる定義サンプルを次に示します。

- メニュー画面の定義サンプル：BMENU1

画面のサンプルは、プッシュボタンのメニューだけから成る GUI 画面です。

- 納品書の帳票の定義サンプル：LTABL1

帳票のサンプルは、XMAP3 インストール先 ¥SAMPLES¥PRTMAP の下に格納されています。納品書の帳票はシリアルインパクトプリンタ用のものです。

### (3) AP 作成上のポイント

作成する AP の基になる AP パターンを次に示します。

- メインプログラムで使用する AP パターン：DSPPRT01

この AP では、[ F2... 納品書 ] ボタンが押されたときの処理だけを記述しています。「納品書」を 1 ページ印刷すると、プログラムが終了します。

この AP 作成時には、次の点に注意してください。

- [ F2... 納品書 ] 以外のボタンが押されたときの処理を追加する。
- 印刷データの代入処理を追加する。
- メニュー画面でのエラーチェックを必要に応じて追加する。

### (4) コーディング例

この例題のソースプログラムを 'NOHIN.cb1' として標準提供しています。ソースプログラムからコーディング上のポイントを抜粋して次に示します。

```

WORKING-STORAGE          SECTION.
* 論理マップのCOPY展開
* メニュー画面用
COPY   BMENU10. .... 1.
COPY   BMENU11.
* ダイアログボックス用
COPY   LTABL10. .... 1.
PROCEDURE                DIVISION.
* マップ名の設定
MOVE 'BMENU1ND'          TO 画面マップ名
* 画面の表示と入力
TRANSCEIVE DSP FROM BMENU10 INTO BMENU11
* ボタンの判定と該当処理の呼び出し
EVALUATE BMENU1-INCL ..... 2.
  WHEN 'PF02'
    PERFORM 帳票処理
  WHEN 'PF09'
    MOVE キャンセル TO 画面1-フラグ
END-EVALUATE
:
帳票処理                SECTION.
  PERFORM 出力データ代入. .... 3.
  SEND PRT FORM LTABL10 WITH EMI.
出力データ代入          SECTION.
  MOVE 'D0012345'        TO LTABL1-SHEETNO-0.
  MOVE '○×A1家電'      TO LTABL1-CUSTOMER-0.
  MOVE 'K0012345'        TO LTABL1-USERNO-0.
出力データ代入-END.

```

## 8. 標準パターンを使用した AP の作成

1. マップ名を変更する。  
プログラム中の '@@@888' を BMENU1 に変更し、'¥¥¥222' を 'L'TABL1' に変更します。
2. ファンクションキーの判定処理。
3. 印刷データの代入処理。

### 8.2.16 MCFDP01 パターンを使用した画面表示と帳票印刷

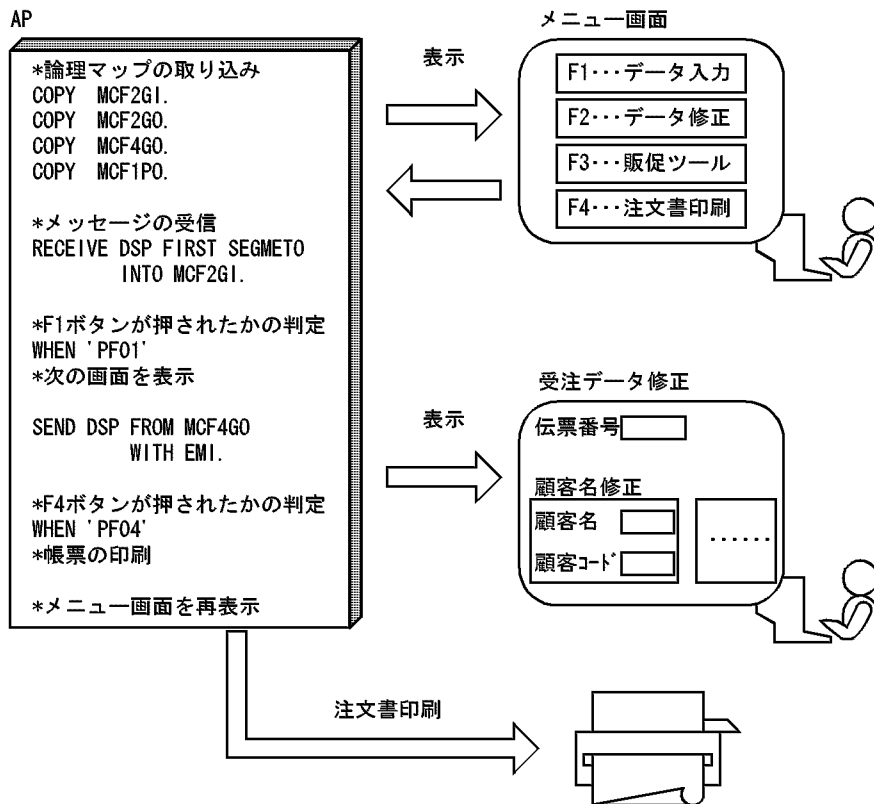
#### (1) 処理の概要

利用者名入力画面にてログインしたあと、業務を選択するためのメニュー画面を表示します。この画面で各メニューのプッシュボタンを選ぶと、選ばれたメニューに対応する業務の画面を表示します。

なお、[注文書印刷] プッシュボタンを選ぶと納品書が印刷されます。

メニュー画面から次画面（この例題では「受注データ修正画面」）を表示する処理の概要を次の図に示します。

図 8-16 メニュー画面からの次画面表示処理





## (2) 画面定義上のポイント

作成する GUI 画面や帳票の基になる定義サンプルを次に示します。

- メニュー画面の定義サンプル：MCF2G
- 次に表示する画面の定義サンプル：MCF4G
- 注文書の帳票の定義サンプル：MCF1P

## (3) AP 作成上のポイント

作成する AP の基になる AP パターンを次に示します。

- メインプログラムで使用する AP パターン名：MCFDP01

この AP パターンを使用した AP 作成時には次の注意が必要です。

- [ F1：受注データ修正 ] ボタン, [ F4：注文書印刷 ] ボタンが押されたときの処理  
( [ F1：受注データ修正 ] ボタンを押したときは次の画面表示し, [ F4：注文書印刷 ] ボタンを押したときは注文書を印刷したあと, メニュー画面を再表示する ) を追加する。
- 次の画面, 帳票に出力するデータの代入処理を追加する。
- エラー処理を必要に応じて追加する。

## (4) コーディング例

この例題のソースプログラムを 'MCF2G.cbl' として標準提供しています。ソースプログラムからコーディング上のポイントを抜粋して次に示します。

8. 標準パターンを使用した AP の作成

```

IDENTIFICATION          DIVISION.
PROGRAM-ID.             MCF2G.          . . . . . 1.
:
WORKING-STORAGE         SECTION.
COPY                   MCF2G1.         . . . . . 2.
COPY                   MCF2GO.         . . . . . 2.
COPY                   MCF3GO.         . . . . . 2.
COPY                   MCF1PO.         . . . . . 2.
:
PROCEDURE               DIVISION.
*   メッセージの受信
    RECEIVE DSP FIRST SEGMENT INTO MCF2G1.
:
    EVALUATE MCF2G-INCI . . . . . 3.
    WHEN 'PF01'
*   次の画面を表示します
        MOVE 'MCF3G'      TO DSP-MAP-NAME
        SEND DSP FROM MCF3GO WITH EMI
:
    WHEN 'PF04'
*   帳票を印刷したあと、受信した画面を再表示します
*   帳票を印刷します
        MOVE 'XPNLE201'  TO PRT-TERM . . . . . 4.
        MOVE 'MCF1P'     TO PRT-MAP-NAME
        SEND PRT FROM MCF1PO WITH EMI
*   画面を再表示します
        MOVE 'MCF2G'     TO DSP-MAP-NAME
        SEND DSP FROM MCF2GO WITH EMI
    WHEN OTHER
*   受信した画面を再表示します
        MOVE 'MCF2G'     TO DSP-MAP-NAME
        SEND DSP FROM MCF2GO WITH EMI
    END-EVALUATE
*   プログラムの終了
    EXIT PROGRAM.

```

1. 入り口名を変更する。  
プログラム中の '@@@111' を 'MCF2G' に変更します。
2. マップ名を変更する。  
プログラム中の以下の項目をそれぞれ変更します。  
'@@@222' を 'MCF2G' に変更  
'@@@333' を 'MCF3G' に変更  
'@@@444' を 'MCF1P' に変更
3. ボタンの判定処理。
4. 論理端末名を変更する。  
プログラム中の '@@@555' を 'XPNLE201' (プリンタに割り当てた論理端末名) に変更します。

## 8.3 各種例題プログラム

### 8.3.1 AP サンプルのコンパイルと実行手順

AP サンプルについて説明します。

#### (1) COBOL を使用した COBOL 開発マネージャでのサンプルプログラムの実行

業務サンプルを作成、実行するためのサンプルファイルの準備をします。

##### (a) COBOL 用サンプルの準備

サンプルプログラムファイルおよび動的変更テーブルは次のフォルダに格納されています。使用するときは、別の任意のフォルダにコピーしてください。なお、サンプルプログラムファイルと動的変更テーブルは同じフォルダにコピーしてください。

- サンプルプログラムファイルの格納先  
XMAP3 インストール先 ¥SAMPLES¥COBOL
- 動的変更テーブルの格納先  
XMAP3 インストール先 ¥INCLUDE¥x3modtbl.cbl

##### (b) プログラムサンプルのファイル名一覧 (COBOL)

提供しているプログラムサンプルのソースファイル名と関連するマップ定義ファイル名を示します。

種別	プログラム サンプル名称	プロジェクトファイル 名	ソースファイル名	定義ファイル名	
GUI	GUI 入門例題	JYUTUGC.PMI	JYUTUGC.CBL	JYU1GC.IMP	
	画面単位入力例題	GAMENGC.PMI	GAMENGC.CBL	GAM1GC.IMP	
	行単位入力例題	LINEGC.PMI	LINEGC.CBL	LIN1GC.IMP	
	イベント単位入力例題	EVENTGC.PMI	EVENTGC.CBL	EVE1GC.IMP EVE2GC.IMP	
	フィールド単位入力例題	FIELDGC.PMI	FIELDGC.CBL	FIE1GC.IMP FIE2GC.IMP	
	受発注業務例題			KADN1GC.CBL	KADxGC.IMP (x:1 ~ 9, A ~ G)
				KADN1PC.CBL	KAD1PC.IMP KAD1SC.IMP
一連の受発注業務をサーバ側の OpenTP1 に処理要求する例題	-	-	OLTP1GC.CBL	-	

## 8. 標準パターンを使用した AP の作成

種別	プログラム サンプル名称	プロジェクトファイ ル名	ソースファイル名	定義ファイル名
	OLTP サーバ構成での 受発注業務の例題	-	MCFxG.CBL (x : 1 ~ 5)	MCFxG.IMP (x : 1 ~ 5)
CUI	CUI 入門例題	JYUTUCC.PMI	JYUTUCC.CBL	JYU1CC.IMP
	画面単位入力例題	GAMENCC.PMI	GAMENCC.CBL	GAM1CC.IMP
	行単位入力例題	LINECC.PMI	LINECC.CBL	LIN1CC.IMP
	フィールド単位入力例 題	FIELDCC.PMI	FIELDCC.CBL	FIE1CC.IMP
	受発注業務例題	KADN1CC.PMI	KADN1CC.CBL	KADxCC.IMP (x : 1 ~ 4)
KADN1PC.CBL			KAD1PC.IMP KAD1SC.IMP	

(凡例)

- : 該当しない。

### (c) プログラムサンプルのコンパイルと実行手順 (COBOL)

#### コンパイル

日立 COBOL の COBOL 開発マネージャを使用して、プログラムサンプルをコンパイルします。

- COBOL2002, COBOL85 Version 7 および Version 6 の場合

COBOL 開発マネージャのツリービューウィンドウで、各プログラムサンプルに応じたプロジェクト名を選び、右クリックで表示されるメニューから [ 選択中のプロジェクトのビルド ] を選ぶと、自動的にコンパイルが実行され、実行ファイルが生成されます。

- COBOL85 Version 5.0 の場合

各プログラムサンプルに応じたプロジェクトファイル名を選び、COBOL 開発マネージャの [ プロジェクト ] メニューの [ ビルド ] を選ぶと、自動的にコンパイルが実行され、実行ファイルが生成されます。

コンパイル時には、次のフォルダにあるインポートライブラリを使用してください。

XMAP3 インストール先 ¥LIB¥V3Compatible

#### 実行

日立 COBOL の COBOL 開発マネージャを使用して、プログラムサンプルを実行します。実行手順を次に示します。

- COBOL2002, COBOL85 Version 7 および Version 6 の場合

COBOL2002, COBOL85 Version 7 および Version 6 でプログラムを実行するには、実行支援を使用して実行環境を設定して実行する方法と、実行環境を使用しないでそのまま実行する方法があります。

< 実行環境を設定して実行する >

COBOL 開発マネージャのウィンドウの [ ビルド ] - [ 実行支援 ] を選び、表示されるダイアログで実行環境を設定して実行します。

<実行環境を設定しないで実行する>

COBOL 開発マネージャのウィンドウの [ビルド] - [実行] を選び、表示されるダイアログで実行プログラム名を指定して実行します。

- COBOL85 Version 5.0 の場合

各プログラムサンプルに応じたプロジェクトファイル名を選び、COBOL 開発マネージャの [実行] メニューの [実行] を選ぶと、実行ダイアログ上のコマンドライン上に実行ファイル名が表示されるので、[OK] を選んで、プログラムサンプルを実行します。

なお、各プログラムサンプルの実行ファイル名は、プロジェクトファイル名と同じです。

すでに実行ファイルを作成しているのであれば、エクスプローラまたはファイルマネージャから、実行ファイルを起動することで、プログラムサンプルを実行できます。

## (2) C 言語を使用したプログラムサンプルの実行

業務サンプルを作成、実行するためのサンプルファイルを準備します。

### (a) C 言語用サンプルの準備

サンプルプログラムファイル、動的変更テーブルおよびインタフェーステーブルは次のフォルダに格納されています。使用するときは、別の任意のフォルダにコピーしてください。なお、サンプルプログラムファイルと動的変更テーブル、インタフェーステーブルは同じフォルダにコピーしてください。

- サンプルプログラムファイルの格納先 (MAKE ファイル, ソースファイル, 定義ファイル)  
XMAP3 インストール先 ¥SAMPLES¥C
- 動的変更テーブル, インタフェーステーブルの格納先  
XMAP3 インストール先 ¥INCLUDE¥x3modtbl.h  
XMAP3 インストール先 ¥INCLUDE¥jsvwatbl.h

### (b) プログラムサンプルのファイル名一覧 (C 言語)

提供しているプログラムサンプルのソースファイル名と関連するマップ定義ファイル名を示します。

種別	プログラムサンプル名称	実行ファイル名	ソース ファイル名	定義ファイル名
GUI	GUI 入門例題	JYUTUGH.EXE	JYUTUGH.C	JYU1GH.IMP
	画面単位入力例題	GAMENGH.EXE	GAMENGH.C	GAM1GH.IMP
	行単位入力例題	LINEGH.EXE	LINEGH.C	LIN1GH.IMP
	イベント単位入力例題	EVENTGH.EXE	EVENTGH.C	EVE1GH.IMP EVE2GH.IMP

## 8. 標準パターンを使用した AP の作成

種別	プログラムサンプル名称	実行ファイル名	ソース ファイル名	定義ファイル名
	フィールド単位入力例題	FIELDGH.EXE	FIELDGH.C	FIE1GH.IMP FIE2GH.IMP
	受発注業務例題	KADN1GH.EXE	KADN1GH.C	KADxGH.IMP (x:1 ~ , A ~ G)
			KADN1PH.C	KAD1PH.IMP KAD1SH.IMP
	OLTP サーバ構成での受発注業務の例題	-	MCFxG.C (x:1 ~ 5)	MCFxG.IMP (x:1 ~ 5)
CUI	CUI 入門例題	JYUTUCH.EXE	JYUTUCH.C	JYU1CH.IMP
	画面単位入力例題	GAMENCH.EXE	GAMENCH.C	GAM1CH.IMP
	行単位入力例題	LINECH.EXE	LINECH.C	LIN1CH.IMP
	フィールド単位入力例題	FIELDCH.EXE	FIELDCH.C	FIE1CH.IMP
	受発注業務例題	KADN1CH.EXE	KADN1CH.C	KADxCH.IMP (x:1 ~ 4)
KADN1PH.C			KAD1PH.IMP KAD1SH.IMP	

(凡例)

- : 該当しない。

### (c) プログラムサンプルで使用するマップの生成

プログラムサンプルを実行する場合、プログラムサンプルに対応したマップを作成する必要があります。実行プログラムサンプルとマップ定義ファイルとの対応は、「8.3.1(2)(b) プログラムサンプルのファイル名一覧 (C 言語)」を参照してください。

### (d) プログラムサンプルのコンパイルと実行手順 (C 言語)

コンパイル

コンパイルには、MAKE ファイル「MAKENET」を使用します。

コンパイル手順を次に示します。

1. 各製品に応じた MAKE ファイル中の、「XMAP3INSTALL=.%」を、XMAP3 インストール先に変更します。
2. MS-DOS プロンプトを開いて、プログラムサンプルのファイルをコピー

(例)

```
cd c:¥temp¥sample
```

3. 実行したいプログラムサンプルに応じて、次のコマンドを実行します。

```
NMAKE / F MAKENET xxxxxxx.EXE
```

「xxxxxxx.EXE」には、実行したいプログラムサンプルの実行ファイル名を入れてください。

## 注

Visual C++ 5.0 でコンパイルする場合は、次のフォルダに提供されているインポートライブラリを使用するように、MAKE ファイルの「XMAPLIBDIR」を変更してください。

XMAP3 インストール先 ¥LIB¥V3Compatible

## 実行

プログラムサンプルを実行します。実行手順は次のうちどれかです。なお、実行ファイル名については、「8.3.1(2)(b) プログラムサンプルのファイル名一覧 (C 言語)」を参照してください。

1. タスクバーの「ファイルを指定して実行」で、各プログラムサンプルに応じた実行ファイル名を指定して、実行します。
2. エクスプローラまたはファイルマネージャから、プログラムサンプルをコピーしたフォルダに移動し、実行ファイルを起動すると、プログラムサンプルを実行します。

## (3) プログラムサンプルを実行するための設定

プログラムサンプルを実行するための設定について、以下に示します。

## ディスプレイのカラーパレットの設定

「コントロールパネル」の「画面」を起動し、表示されるダイアログの「カラーパレット」(Windows 98 の場合は「色」、Windows 2000、Windows XP、Windows Server 2003、および Windows Server 2003 x64 の場合は「画面の色」)で、「256 色」以上 (Windows XP、Windows Server 2003、および Windows Server 2003 x64 の場合は「中 (16 ビット)」以上) を設定してください。

### 8.3.2 COBOL による GUI 画面を使用した画面単位入力例題

画面単位に AP に制御を戻し、入力したデータを一回の画面入力で一括して処理する方法です。標準的な画面入出力の業務に適しています。なお、この例では GUI 画面を想定していますが、CUI 画面でも同じような処理が実現できます。CUI 画面を使用する場合は、プッシュボタンの代わりに、ファンクションキーを使用します。

#### (1) 想定するエントリ画面

図 8-17 エントリ画面 (GUI : 画面単位の入力)

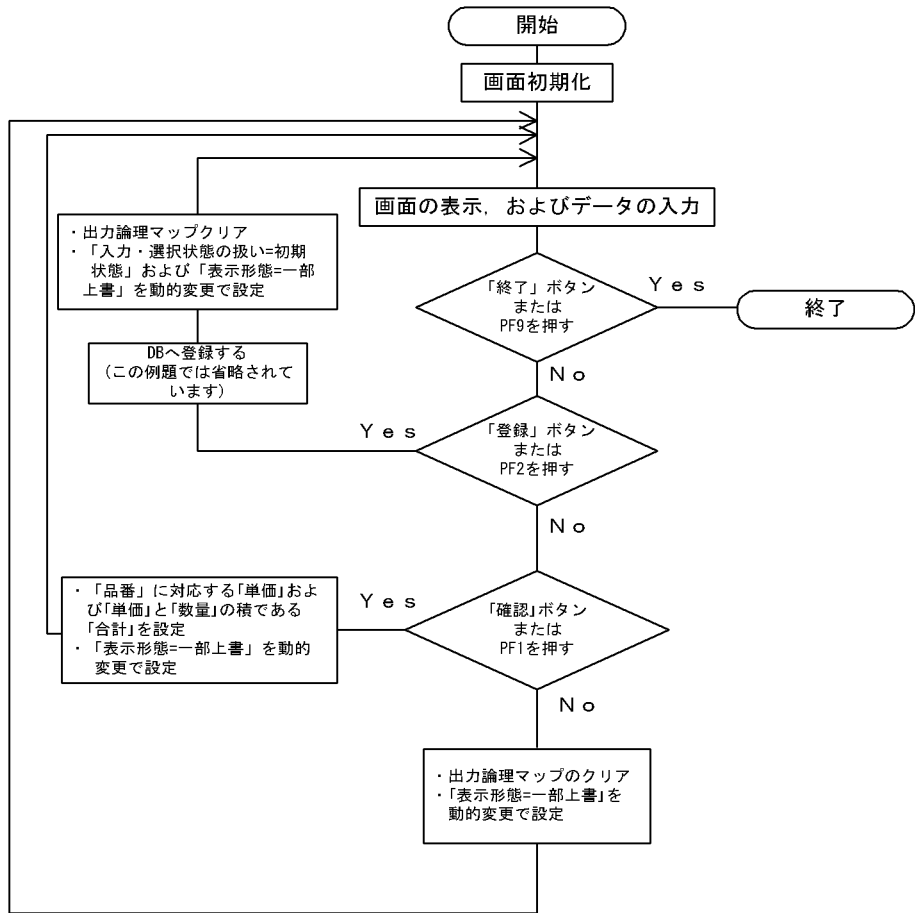
画面単位入力	
品番	XX-XXXX
単価	¥533,000
数量	10
合計	¥5,330,000
確認    登録    終了	

注

物理マップサイズの目安は、約 4KB です。



## (2) 処理の流れ



## (3) コーディング

画面からのデータが入力されたあと、AP 中ではどのプッシュボタン（CUI 画面の場合、ファンクションキー）が押されたかを判定します。

「終了」の場合は AP を終了します。「確認」の場合は、入力したデータ（「品番」「数量」）に対応した出力データを設定し、「表示形態 = 一部上書」で画面を再表示します。「登録」の場合は、出力論理マップをクリアしたあと「入力・選択状態の扱い = 初期状態」および「表示形態 = 一部上書」で画面を再表示します。

コーディングパターンの例を次の図に示します。マップ名は「GAM1GC」です。プログラムサイズは約 12KB です。

8. 標準パターンを使用した AP の作成

図 8-18 画面単位にデータを入力するコーディング ( 1 / 3 )

```

000100*****
000200*      プログラム名 :  G U I 画面一画面単位入力例題      *
000300*      ファイル名 :  G A M E N G C                      *
000400*      説明      :  画面単位に A P とやり取りする例題です *
000500*****
000600 IDENTIFICATION      DIVISION.
000700 PROGRAM-ID.        GAMENGC.
000800*
000900 ENVIRONMENT          DIVISION.
001000*
001100 DATA                DIVISION.
001200*
001300 WORKING-STORAGE     SECTION.
001400*
001500      77 画面 1-フラグ PIC S9(2).
001600*
001700      77 正常          PIC S9(5) VALUE +0.
001800      77 終了          PIC S9(2) VALUE +1.
001900      77 単価WK       PIC S9(6) COMP.
002000      77 数量WK       PIC S9(4) COMP.
002100*
002200*****
002300*      論理マップと動的変更テーブルの取込み      *
002400*      出力用 : マップ名+0, 入力用 : マップ名+1      *
002500*****
002600*      論理マップの取り込み      *
002700      COPY      GAM1GCO.
002800      COPY      GAM1GCI.
002900*
003000* 動的変更テーブルの取り込み --"XMAP3インストール先¥INCLUDE"にあります
003100      COPY      X3MODTBL.
003200*
003300*****
003400*      通信記述      *
003500*****
003600 COMMUNICATION          SECTION.
003700*      ディスプレイに対する通信記述項
003800 CD      DSP FOR I-O WS
003900      MAP NAME          IS 画面マップ名
004000      SYMBOLIC TERMINAL IS 画面端末名
004100      MAPPING MODE      IS マッピングモード
004200      STATUS KEY        IS 画面-RC.
004300*
004400*****
004500*      プログラム開始      *
004600*****
004700 PROCEDURE              DIVISION.
004800
004900*      端末名の設定
005000      MOVE 'DSP001' TO 画面端末名.
005100
005200*      出力エリアのクリア
005300      MOVE ALL '1F' TO GAM1GCG.

```

図 8-19 画面単位にデータを入力するコーディング (2 / 3)

```

005400
005500*   マッピングオプションの設定 (マージで表示)
005600   MOVE SPACE TO マッピングモード
005700*
005800   MOVE +0 TO 画面 1-フラグ.
005900   PERFORM UNTIL 画面 1-フラグ = 終了
006000
006100*   マップ名の設定
006200   MOVE 'GAM1GCND' TO 画面マップ名
006300*   画面の表示と入力
006400   TRANSCEIVE DSP FROM GAM1GCO INTO GAM1GCI
006500
006600*   結果確認
006700   IF 画面-RC = 正常
006800       THEN
006900           CONTINUE
007000       ELSE
007100           DISABLE DSP
007200           MOVE 12 TO RETURN-CODE
007300           STOP RUN
007400       END-IF
007500
007600*   ボタン・P Fキーのチェックと該当処理の呼び出し
007700   EVALUATE GAM1GC-INCI
007800       WHEN 'PF01'
007900           PERFORM 確認処理
008000       WHEN 'PF02'
008100           PERFORM 登録処理
008200       WHEN 'PF09'
008300           MOVE 終了 TO 画面 1-フラグ
008400       WHEN OTHER
008500           CONTINUE
008600       END-EVALUATE
008700
008800   END-PERFORM.
008900
009000   GO TO 正常終了.
009100
009200*****
009300*   プログラムの終了 *
009400*****
009500 正常終了.
009600
009700*   XMAP3のクローズ
009800   DISABLE DSP.
009900   MOVE +0 TO RETURN-CODE
010000   STOP RUN.
010100
010200*****
010300*   確認処理 *
010400*   品番・数量を入力し、単価と合計を計算します *
010500*   その後、画面を「一部上書」で再表示します *
010600*****

```

8. 標準パターンを使用した AP の作成

図 8-20 画面単位にデータを入力するコーディング ( 3 / 3 )

```

010700 確認処理 SECTION.
010800     IF  GAM1GC-FIELD0006-I = ALL SPACE
010900     THEN
011000         MOVE  "0"      TO  GAM1GC-FIELD0007-0
011100         MOVE  GAM1GC-FIELD0007-0 TO  単価WK
011200         MOVE  GAM1GC-FIELD0008-I TO  数量WK
011300         COMPUTE GAM1GC-FIELD0009-0 = 単価WK * 数量WK
011400         MOVE  XMAP-CNTRL1 TO  GAM1GC-CNTRLO
011500     ELSE
011600         MOVE  "533000" TO  GAM1GC-FIELD0007-0
011700         MOVE  GAM1GC-FIELD0007-0 TO  単価WK
011800         MOVE  GAM1GC-FIELD0008-I TO  数量WK
011900         COMPUTE GAM1GC-FIELD0009-0 = 単価WK * 数量WK
012000         MOVE  XMAP-CNTRL1 TO  GAM1GC-CNTRLO
012100*      マッピングオプションの設定 (論理マップだけ)
012200         MOVE  2          TO  マッピングモード
012300     END-IF.
012400 確認処理-END.
012500     EXIT.
012600
012700*****
012800*      登録処理 *
012900*      DBへの登録処理を行います *
013000*      その後、入力データを消去して画面をリフレッシュします *
013100*****
013200 登録処理 SECTION.
013300
013400*----- DBへの登録処理を行います (本例では省略) -----*
013500
013600
013700     MOVE ALL X'1F' TO GAM1GCG.
013800     MOVE ALL ZERO TO GAM1GC-FIELD0007-0
013900     MOVE ALL ZERO TO GAM1GC-FIELD0009-0
014000     MOVE XMAP-CNTRL3 TO GAM1GC-CNTRLO
014100     MOVE ' ' TO マッピングモード.
014200
014300 登録処理-END.
014400     EXIT.
014500
014600*****
014700 END PROGRAM GAMENGC.

```

### 8.3.3 COBOL による GUI 画面を使用したフィールド単位入力例題

フィールド単位に AP に制御を戻して、入力や確認に必要な情報を画面に少しずつ表示していく方法です。フィールド単位に処理するので、少量のデータを頻繁に入出力するようなデータエントリ業務に適しています。なお、フィールド単位に処理をするため、C/S システムのように逐一、サーバとの送受信が発生するようなシステムでは、処理動作が遅くなりますので、ご注意ください。

AP から、ある入力フィールドに対してフォーカス・カーソルを位置づけると入力可能となります。その他の入力フィールドには入力できません。

#### (1) 想定するエントリ画面

図 8-21 エントリ画面（フィールド単位にデータを入力する）

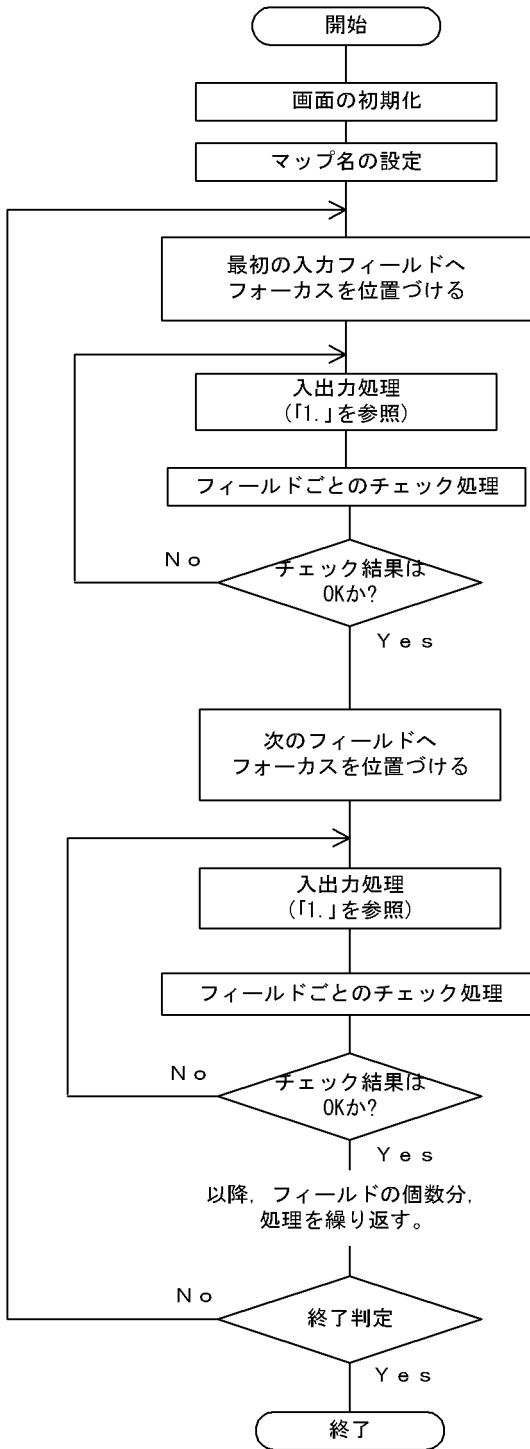
品番	単価	数量	合計
222	100	222	¥22,200

1 以外・・・続行  
1・・・終了

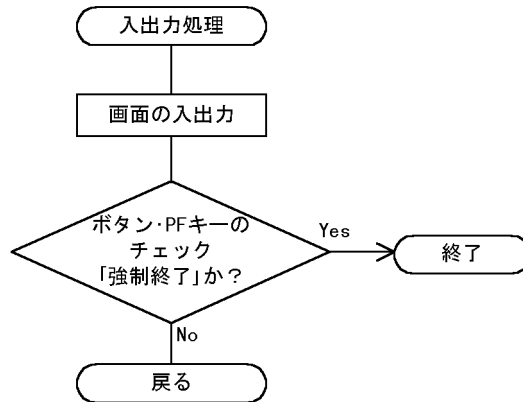
PF12: 強制終了

品番に'999' 入力するとエラーメッセージを表示します

(2) 処理の流れ



## 1. 入出力処理の詳細



## (3) コーディング

画面表示の際に、入力したいフィールドに対してフォーカス・カーソルを位置づけることでフィールド単位入力を実現します。AP には、フォーカス・カーソルを位置づけたフィールドからの入力データだけが返り、そのデータに対応する処理が呼び出されます。処理が終わったら、次に入力するフィールドにフォーカス・カーソルを位置づけて画面を再表示します。以降、入力フィールドの個数分、これらの処理を繰り返します。

コーディングパターン例を次の図に示します。マップ名は「FIE1GC」です。プログラムサイズは約 10KB です。

8. 標準パターンを使用した AP の作成

図 8-22 フィールド単位にデータを入力するコーディング例 ( 1 / 4 )

```

000100*****
000200*   プログラム名   :   G U I 画面-フィールド単位入力例題   *
000300*   ファイル名   :   F I E L D G C   *
000400*   説明       :   フィールド単位に A P とやり取りする例題です *
000500*****
000600 IDENTIFICATION   DIVISION.
000700 PROGRAM-ID.     FIELDGC.
000800*
000900 ENVIRONMENT      DIVISION.
001000*
001100 DATA             DIVISION.
001200*
001300 WORKING-STORAGE  SECTION.
001400*
001500   77  正常      PIC X(5) VALUE '00000'.
001600*
001700*****
001800*   論理マップと動的変更テーブルの取込み   *
001900*****
002000   COPY   FIE1GCO.
002100   COPY   FIE1GCI.
002200
002300*  動的変更テーブルの取込み -- "XMAP3インストール先¥INCLUDE"にあります
002400   COPY   X3MODTBL.
002500
002600*****
002700*   通信記述   *
002800*****
002900 COMMUNICATION     SECTION.
003000*   ディスプレイに対する通信記述項
003100 CD   DSP   FOR   I-O   WS
003200   MAP   NAME      IS   画面マップ名
003300   SYMBOLIC   TERMINAL   IS   画面端末名
003400   MAPPING   MODE      IS   マッピングモード
003500   STATUS   KEY      IS   画面-RC.
003600*****
003700*   プログラム開始   *
003800*****
003900 PROCEDURE           DIVISION.
004000
004100*   画面端末名の設定
004200   MOVE 'DSP001' TO 画面端末名.
004300
004400*   出力エリアのクリア
004500   MOVE ALL X'1F' TO FIE1GCG.
004600
004700*   マップ名の設定
004800   MOVE 'FIE1GCND' TO 画面マップ名.
004900*
005000 フィールド単位処理.
005100*****
005200*   フィールド 1 (品番) の処理   *
005300*****
005400*   品番フィールドを活性にする

```



図 8-23 フィールド単位にデータを入力するコーディング例 (2 / 4)

```

005500      MOVE FIE1GC-品番-T TO FIE1GC-OUTCURS-LOC0.
005600*
005700 品番処理.
005800*      画面の表示と入力
005900      PERFORM 画面入出力.
006000*
006100**      品番処理のチェック処理
006200*----- ユーザ任意のチェック処理を追加します -----*
006300*----- 本例題では、3けた入力のみOKとします -----*
006400      IF FIE1GC-品番-I < 100
006500          THEN
006600**          チェックでエラーありの場合の処理
006700              MOVE N' 3けた分の入力を行って下さい'
006800                  TO FIE1GC-メッセージ-0
006900*          品番フィールドの再入力へ
007000          GO TO 品番処理
007100      ELSE
007200**          チェックでエラーなしの場合の処理
007300              MOVE SPACE TO FIE1GC-メッセージ-0
007400*          単価&品番の設定
007500              MOVE 100 TO FIE1GC-単価-0
007600              MOVE FIE1GC-品番-I TO FIE1GC-品番-0
007700      END-IF.
007800*
007900*****
008000*      フィールド2(数量)の処理 *
008100*****
008200*      数量フィールドを活性にする
008300      MOVE FIE1GC-数量-T TO FIE1GC-OUTCURS-LOC0.
008400
008500 数量処理.
008600*      画面の表示と入力
008700      PERFORM 画面入出力.
008800*
008900*      数量処理のチェック処理
009000*----- ユーザ任意のチェック処理を追加します -----*
009100*----- 本例題では、3けた入力のみOKとします -----*
009200      IF FIE1GC-数量-I < 100
009300          THEN
009400**          チェックでエラーありの場合の処理
009500              MOVE N' 3けた分の入力を行って下さい'
009600                  TO FIE1GC-メッセージ-0
009700          GO TO 数量処理
009800      ELSE
009900**          チェックでエラーなしの場合の処理
010000              MOVE SPACE TO FIE1GC-メッセージ-0
010100*          合計の計算
010200              MOVE FIE1GC-数量-I TO FIE1GC-数量-0
010300              COMPUTE FIE1GC-合計-0 = FIE1GC-単価-0 * FIE1GC-数量-I
010400      END-IF.

```

8. 標準パターンを使用した AP の作成

図 8-24 フィールド単位にデータを入力するコーディング例 ( 3 / 4 )

```

010500*
010600*****
010700*   最終フィールド(判定)の処理 *
010800*****
010900*   最終フィールド(判定フィールド)を活性にする
011000   MOVE FIE1GC-判定-T TO FIE1GC-OUTCURS-LOC0.
011100
011200 最終フィールド処理.
011300*   画面の表示と入力
011400   PERFORM   画面入出力.
011500
011600**   最終フィールドのチェック処理
011700*   終了判定
011800   IF FIE1GC-判定-I NOT = 1
011900     THEN
012000*   元に戻る
012100*   出力エリアのクリア
012200     MOVE ALL LOW-VALUE TO FIE1GCG
012300     GO TO   フィールド単位処理
012400     ELSE
012500*   終了へ
012600     CONTINUE
012700     END-IF.
012800
012900*****
013000*   プログラムの終了 *
013100*****
013200 正常終了.
013300*   X M A P 3 のクローズ
013400   DISABLE   DSP.
013500
013600   MOVE     +0 TO RETURN-CODE.
013700   STOP   RUN.
013800
013900*****
014000*   入出力処理 -- 画面入出力とイベントの判定 -- *
014100*****
014200 画面入出力 SECTION.
014300
014400*   画面の表示と入力
014500   TRANSCEIVE DSP FROM FIE1GCO INTO FIE1GCI.
014600
014700*   結果確認
014800   IF 画面-RC = 正常
014900     THEN
015000     CONTINUE
015100     ELSE
015200     DISABLE   DSP
015300     MOVE     12 TO RETURN-CODE
015400     STOP   RUN
015500     END-IF.

```

図 8-25 フィールド単位にデータを入力するコーディング例 (4 / 4)

```
015600
015700*   ボタン・PFキーのチェックと該当処理の呼び出し
015800   EVALUATE FIELDGC-INCI
015900   WHEN 'PF01'
016000*   PERFORM 次画面表示-セクション
016100   WHEN 'PF02'
016200*   GO TO 正常終了
016300   WHEN 'BREK'
016400*   GO TO 正常終了
016500   WHEN 'PF12'
016600*   強制終了へ
016700   GO TO 正常終了
016800   WHEN OTHER
016900   CONTINUE
017000   END-EVALUATE.
017100
017200 画面入出力-END.
017300   EXIT.
017400
017500*****
017600 END PROGRAM FIELDGC.
```

### 8.3.4 COBOL による GUI 画面を使用したイベント単位入力例題

画面でイベントが発生するたびに AP に制御を返して、入力データの処理や確認に必要な情報を画面に少しずつ表示していく方法です。イベントが発生したときは、イベント通知コードとして、プッシュボタンやファンクションキーに対応した INC 定数が、各オブジェクトに対応したイベント定数が AP に返されます。このイベント定数を AP で判定して、各オブジェクトの処理を行います。

イベント単位入力では、フィールド単位入力と異なり、画面上のどの活性オブジェクトも使用できる状態になっています。

なお、イベント単位入力は、GUI 画面でだけ使用できます。また、イベント単位に処理をするため、C/S システムのように逐一、サーバとの送受信が発生するようなシステムでは、処理動作が遅くなりますのでご注意ください。

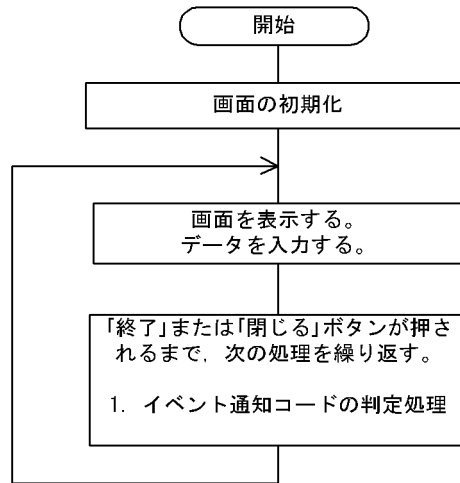
#### (1) 想定するエントリ画面

図 8-26 エントリ画面 (GUI: イベント単位の入力)

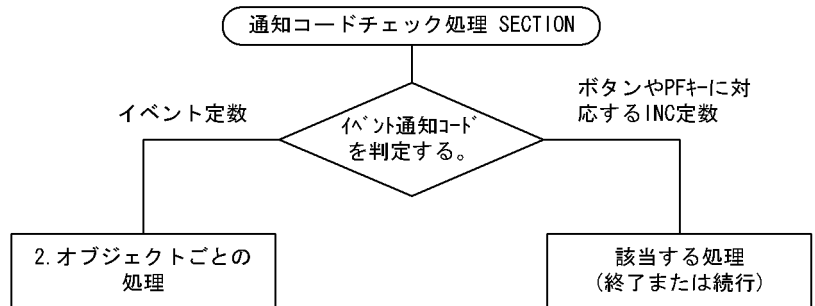
商品名	前回	今回	チェック
AVコンポ	20	20	発注 <input checked="" type="checkbox"/>
スピーカー	24	18	<input type="checkbox"/>
食器洗い機	11	5	<input type="checkbox"/>
衣類乾燥機	15	8	<input type="checkbox"/>
冷蔵庫	7	15	<input type="checkbox"/>
加湿器	2	1	<input type="checkbox"/>
液晶テレビ	27	19	<input type="checkbox"/>
			<input type="checkbox"/>
			<input type="checkbox"/>
			<input type="checkbox"/>

コメント

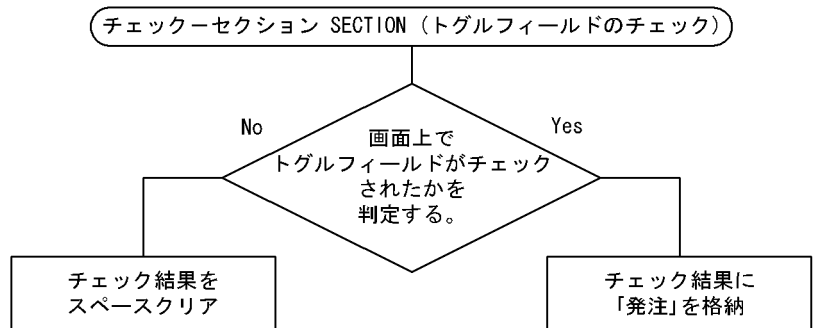
## (2) 処理の流れ



## 1. イベント通知コードの判定処理



## 2. オブジェクトごとの処理



## (3) コーディング

画面を表示してデータを入力したあと、APでイベント通知コードを判定し、処理を振り分けます。イベント定数が返された場合には、オブジェクトからイベントが返されたと判断し、オブジェクトごとの処理を行います。また、プッシュボタンやファンクション

8. 標準パターンを使用した AP の作成

キーに対応する INC 定数が返された場合には、ボタンやキーからイベントが返されたかと判断し、該当する処理を行います。コーディングパターンの例を次の図に示します。マップ名は「EVE1GC」です。プログラムサイズは約 14KB です。

図 8-27 イベント単位にデータを入力するコーディング例 ( 1 / 5 )

```

000100*****
000200*   プログラム名 :   G U I 画面-イベント単位入力例題           *
000300*   ファイル名 :   E V E N T G C                               *
000400*   説明      :   イベント単位に A P とやり取りする例題です *
000500*****
000600 IDENTIFICATION      DIVISION.
000700 PROGRAM-ID.        EVENTGC.
000800*
000900 ENVIRONMENT        DIVISION.
001000*
001100 DATA              DIVISION.
001200*
001300 WORKING-STORAGE    SECTION.
001400*   発注商品データファイル
001500     01 売上データ.
001600     02 売上データファイル.
001700     03 FILLER PIC X(24)
001800         VALUE ' A V コンボ      20  20  ' .
001900     03 FILLER PIC X(24)
002000         VALUE ' スピーカー    24  18  ' .
002100     03 FILLER PIC X(24)
002200         VALUE ' 食器洗い機    11   5  ' .
002300     03 FILLER PIC X(24)
002400         VALUE ' 衣類乾燥機    15   8  ' .
002500     03 FILLER PIC X(24)
002600         VALUE ' 冷蔵庫        7  15  ' .
002700     03 FILLER PIC X(24)
002800         VALUE ' 加湿器        2   1  ' .
002900     03 FILLER PIC X(24)
003000         VALUE ' 液晶テレビ   27  19  ' .
003100     02 FILLER
003200         REDEFINES 売上データファイル OCCURS 7.
003300     03 売上データレコード PIC X(24).
003400
003500     77 商品カウンタ PIC 9(4).
003600     77 トグルカウンタ PIC 9(4).
003700     77 画面 1-フラグ PIC S9(2).
003800     77 正常 PIC S9(5) VALUE +0.
003900     77 終了 PIC S9(2) VALUE +1.
004000     77 キャンセル PIC S9(2) VALUE -1.
004100
004200*****
004300*   論理マップと動的変更テーブルの取込み           *
004400*****
004500*   論理マップの取り込み
004600     COPY   EVE1GC0.
004700     COPY   EVE1GC1.
004800*
004900*   動的変更テーブルの取り込み --"XMAP3インストール先¥INCLUDE"にあります
005000     COPY   X3MODTBL.

```

図 8-28 イベント単位にデータを入力するコーディング例 (2 / 5)

```

005100*****
005200*   通信記述 *
005300*****
005400 COMMUNICATION SECTION.
005500*   ディスプレイに対する通信記述項
005600 CD   DSP FOR I-O WS
005700     MAP NAME           IS 画面マップ名
005800     SYMBOLIC TERMINAL IS 画面端末名
005900     MAPPING MODE       IS マッピングモード
006000     STATUS KEY        IS 画面-RC.
006100*****
006200*   プログラム開始 *
006300*****
006400 PROCEDURE DIVISION.
006500
006600*   端末名の設定
006700     MOVE 'DSP001' TO 画面端末名.
006800
006900*   マッピングオプションの設定 (マージで表示)
007000     MOVE SPACE TO マッピングモード.
007100*
007200*   出力エリアのクリア
007300     MOVE ALL X'1F' TO EVE1GCG.
007400
007500*----- 発注商品データの設定
007600     PERFORM WITH TEST AFTER VARYING 商品カウンタ
007700         FROM 1 BY 1 UNTIL (商品カウンタ >= 7)
007800         MOVE 売上データレコード(商品カウンタ)
007900             TO EVE1GC-商品明細-0(商品カウンタ)
008000     END-PERFORM.
008100
008200*   マップ名の設定
008300     MOVE 'EVE1GCND' TO 画面マップ名.
008400
008500     MOVE +0 TO 画面1-フラグ.
008600     PERFORM UNTIL 画面1-フラグ = 終了 OR
008700         画面1-フラグ = キャンセル
008800
008900*   画面の表示と入力
009000     TRANSCEIVE DSP FROM EVE1GCO INTO EVE1GCI
009100
009200*   結果確認
009300     IF 画面-RC = 正常
009400         THEN
009500             CONTINUE
009600         ELSE
009700             DISABLE DSP
009800             MOVE 12 TO RETURN-CODE
009900             STOP RUN
010000     END-IF
010100

```

8. 標準パターンを使用した AP の作成

図 8-29 イベント単位にデータを入力するコーディング例 ( 3 / 5 )

```

010200*****
010300* イベント通知コードとして格納される値により処理を振り分けます *
010400* イベント定数 ---> 入力・選択可能なオブジェクトからのイベント *
010500* オブジェクトごとのルーチンへ *
010600* INC定数 ---> ボタン・PFキーからのイベント *
010700* 該当する処理へ *
010800*****
010900 EVALUATE EVE1GC-INC1
011000 WHEN EVE1GC-チェック-V
011100**----- トグルフィールドからのイベント -----*
011200 PERFORM トグルフィールドーイベント処理
011300* コメントフィールドの表示属性を標準に戻す
011400 MOVE XMAP-IN-ATTR0 TO EVE1GC-コメント-0
011500 WHEN EVE1GC-コメント-V
011600**----- コメントフィールドからのイベント -----*
011700 PERFORM コメントフィールドーイベント処理
011800
011900**----- ボタン・PFキーからのイベント -----*
012000 WHEN 'PF04'
012100 MOVE 終了 TO 画面1-フラグ
012200 WHEN 'PF03'
012300 MOVE キャンセル TO 画面1-フラグ
012400 WHEN 'PF02'
012500 MOVE 終了 TO 画面1-フラグ
012600 WHEN 'PF01'
012700* 画面再表示 (一部上書&論理マップ だけ) の準備
012800 MOVE ALL '1F' TO EVE1GCG.
012900 MOVE XMAP-CNTRL1 TO EVE1GC-CNTRLO
013000 MOVE 2 TO マッピングモード
013100* チェック結果フィールドのスペースクリア
013200 PERFORM WITH TEST AFTER VARYING 商品カウンタ
013300 FROM 1 BY 1 UNTIL (商品カウンタ >= 10)
013400 MOVE ALL SPACE TO
013500 EVE1GC-チェック結果-0(商品カウンタ)
013600 END-PERFORM
013700
013800* コメントフィールドの表示属性を標準に戻す
013900 MOVE XMAP-IN-ATTR0 TO EVE1GC-コメント-0
014000 END-EVALUATE
014100
014200 END-PERFORM.
014300
014400*****
014500* プログラムの終了 *
014600*****
014700 正常終了.
014800* XMAP3のクローズ
014900 DISABLE DSP.
015000*
015100 MOVE +0 TO RETURN-CODE.
015200 STOP RUN.
015300

```



図 8-30 イベント単位にデータを入力するコーディング例 (4 / 5)

```

015400*****
015500*      トグルフィールドからのイベントに対する処理      *
015600*****
015700 トグルフィールドーイベント処理 SECTION.
015800
015900*      画面再表示（一部上書&論理マップ だけ）の準備
016000      MOVE ALL X'1F' TO EVE1GCG.
016100      MOVE XMAP-CNTRL1 TO EVE1GC-CNTRLO
016200      MOVE 2          TO マッピングモード.
016300
016400*----- トグルフィールドのカーソル定数を判定 -----*
016500      MOVE 0 TO トグルカウンタ.
016600      EVALUATE EVE1GC-INCURS-LOC1
016700          WHEN EVE1GC-チェック-T(01)
016800              MOVE 1 TO トグルカウンタ
016900              PERFORM チェック-セクション
017000          WHEN EVE1GC-チェック-T(02)
017100              MOVE 2 TO トグルカウンタ
017200              PERFORM チェック-セクション
017300          WHEN EVE1GC-チェック-T(03)
017400              MOVE 3 TO トグルカウンタ
017500              PERFORM チェック-セクション
017600          WHEN EVE1GC-チェック-T(04)
017700              MOVE 4 TO トグルカウンタ
017800              PERFORM チェック-セクション
017900          WHEN EVE1GC-チェック-T(05)
018000              MOVE 5 TO トグルカウンタ
018100              PERFORM チェック-セクション
018200          WHEN EVE1GC-チェック-T(06)
018300              MOVE 6 TO トグルカウンタ
018400              PERFORM チェック-セクション
018500          WHEN EVE1GC-チェック-T(07)
018600              MOVE 7 TO トグルカウンタ
018700              PERFORM チェック-セクション
018800          WHEN EVE1GC-チェック-T(08)
018900              MOVE 8 TO トグルカウンタ
019000              PERFORM チェック-セクション
019100          WHEN EVE1GC-チェック-T(09)
019200              MOVE 9 TO トグルカウンタ
019300              PERFORM チェック-セクション
019400          WHEN EVE1GC-チェック-T(10)
019500              MOVE 10 TO トグルカウンタ
019600              PERFORM チェック-セクション
019700      END-EVALUATE.
019800
019900 トグルフィールドーイベント処理-END.
020000      EXIT.

```

## 8. 標準パターンを使用した AP の作成

図 8-31 イベント単位にデータを入力するコーディング例 (5 / 5)

```
020100*****
020200* トグルフィールドごとの処理 -チェックを判定して発注承認を行う -*
020300*****
020400 チェッカーセクション SECTION.
020500
020600* トグルフィールドがチェックされたかの判定
020700 IF EVE1GC-チェック-I(トグル-カウンタ) = '01'
020800 THEN
020900 MOVE N'発注' TO
021000 EVE1GC-チェック結果-0(トグル-カウンタ)
021100 ELSE
021200 MOVE ALL SPACE TO
021300 EVE1GC-チェック結果-0(トグル-カウンタ)
021400 END-IF.
021500
021600 チェッカーセクション-END.
021700 EXIT.
021800*****
021900* コメントフィールドからのイベントに対する処理 *
022000*****
022100 コメントフィールドーイベント処理 SECTION.
022200
022300* 画面再表示 (一部上書&論理マップ だけ) の準備
022400 MOVE ALL X'1F' TO EVE1GCG.
022500 MOVE XMAP-CNTRL1 TO EVE1GC-CNTRLO
022600 MOVE 2 TO マッピングモード.
022700
022800* コメントフィールドの表示属性を「赤色」に変更して再表示する
022900 MOVE XMAP-IN-ATTR1 TO EVE1GC-コメント-0
023000 MOVE EVE1GC-コメント-I TO EVE1GC-コメント-0.
023100
023200 コメントフィールドーイベント処理-END.
023300 EXIT.
023400
023500*****
023600 END PROGRAM EVENTGC.
```

### 8.3.5 COBOL による GUI 画面を使用した行単位入力例題

行単位に AP に制御を戻して、入力や確認に必要な情報を画面に少しずつ表示していく方法です。行単位のデータを転送しながら画面レベルで確認ができるので、まとまったデータをグループ処理したい場合に適しています。データを入力する領域として GUI 画面中のフィールドボックスを使用することをお勧めします。なお、この例では GUI 画面を想定していますが、CUI 画面でも同じような処理を実現できます。CUI 画面の場合は、押しボタンの代わりにファンクションキーが使用できます。

#### (1) 想定するエントリ画面

図 8-32 エントリ画面 (GUI: 行単位のデータ入力)

商品コード	商品名	単価	数量	小計
100201	ウォーターベッド	10000	002	20000
100701	シャンデリア (金)	9000	001	9000
100508	ベルシヤ絨毯	6500	002	13000
101004	ロッキングチェア	26000	004	104000
100705	桐たんす (3段)	66000	001	66000
101002	スイス製洋服ダンス	88000	002	172000
合計				384000

更新                      終了

行末のフィールドを「自動送信 (入力直後に AP へ制御を渡す)」属性に定義しておきます。

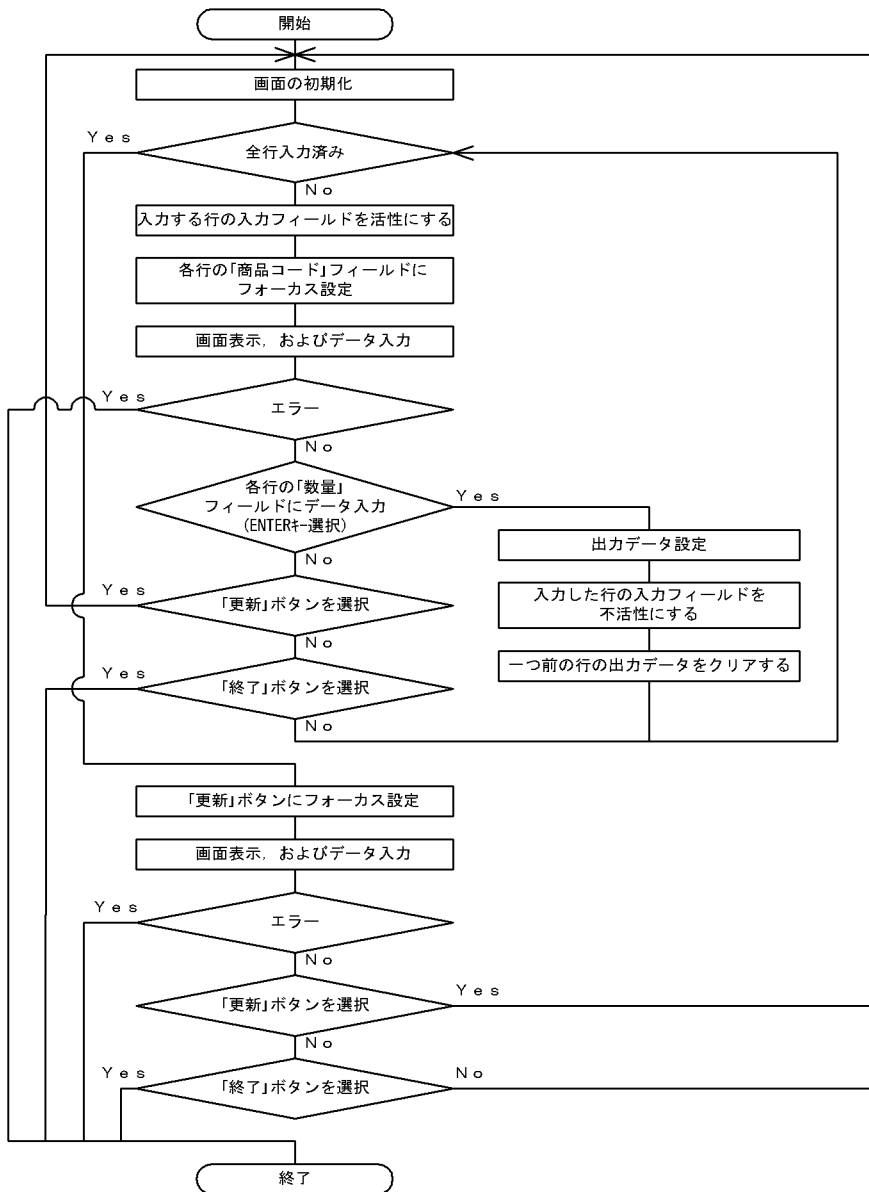
また、各入力フィールドは入力可否の動的変更ができるように「動的変更 (AP から表示属性を変更する)」を指定しておきます。

注

物理マップサイズの目安は、約 5.5KB です。

## 8. 標準パターンを使用した AP の作成

### (2) 処理の流れ



### (3) コーディング

行単位でフィールドの「入力可否」の切り替えをします。行末のフィールドにデータが入力されたら、自動的に送信を発生させ、次の行へ続行させます。処理済みの行については、余分なデータが送信されないようにデータ有無コードを設定しておきます。

コーディングパターンの例を次の図に示します。マップ名は「LIN1GC」です。プログラムサイズは約 20KB です。

図 8-33 行単位にデータを入力するコーディング (1 / 5)

```

000100*****
000200*   プログラム名 : G U I 画面一行単位入力例題 *
000300*   ファイル名 : L I N E G C *
000400*   説明      : 行単位に A P とやり取りする例題です *
000500*****
000600 IDENTIFICATION DIVISION.
000700 PROGRAM-ID.     LINEGC.
000800*
000900 ENVIRONMENT     DIVISION.
001000*
001100 DATA           DIVISION.
001200*
001300 WORKING-STORAGE SECTION.
001400*
001500      01 商品名 D B.
*          (以下、省略)
001600
001700      01 単価 D B.
*          (以下、省略)
001800
001900      77 画面 1-フラグ PIC S9(2).
002000      77 単価       PIC  9(18) COMP.
002100      77 小計       PIC  9(18) COMP.
002200      77 合計       PIC  9(18) COMP.
002300*
002400      77 正常       PIC  S9(5) VALUE  +0.
002500      77 終了       PIC  S9(2) VALUE  +1.
002600      77 カウンタ   PIC  9(5) VALUE  0 COMP.
002700      77 行カウンタ PIC  9(5) VALUE  6 COMP.
002800*
002900*****
003000*   論理マップと動的変更テーブルの取込み *
003100*****
003200*   論理マップの取込み
003300   COPY  LIN1GCO.
003400   COPY  LIN1GCI.
003500*
003600* 動的変更テーブルの取込み --"XMAP3インストール先¥INCLUDE"にあります
003700   COPY  X3MODTBL.
003800*
003900*****
004000*   通信記述 *
004100*****
004200 COMMUNICATION SECTION.
004300*   ディスプレイに対する通信記述項
004400 CD  DSP FOR I-O WS
004500   MAP NAME           IS 画面マップ名
004600   SYMBOLIC TERMINAL IS 画面端末名
004700   MAPPING MODE       IS マッピングモード
004800   STATUS KEY         IS 画面-RC.
004900*

```

8. 標準パターンを使用した AP の作成

図 8-34 行単位にデータを入力するコーディング ( 2 / 5 )

```

005000*****
005100*      プログラム開始 *
005200*****
005300 PROCEDURE      DIVISION.
005400
005500*      端末名の設定
005600      MOVE 'DSP001' TO 画面端末名.
005700      MOVE      +0 TO 画面1-フラグ.
005800*
005900 業務開始.
006000*      出力エリアのクリア
006100      MOVE ALL X'1F' TO LIN1GCG.
006200
006300*      マップ名の設定
006400      MOVE 'LIN1GCND' TO 画面マップ名
006500
006600***** 行単位の処理を行う *****
006700      PERFORM      行単位処理.
006800*
006900***** 全ての行にデータ入力後, 更新処理を行う *****
007000*      「更新」ボタンにフォーカスを設定
007100      MOVE LIN1GC-PBOX0001-T TO LIN1GC-OUTCURS-LOGO
007200
007300*      マッピングオプションの設定 (マージで表示)
007400      MOVE SPACE TO マッピングモード
007500      MOVE XMAP-CNTRL1 TO LIN1GC-CNTRL0
007600
007700*      画面の表示と入力
007800      TRANSCEIVE DSP FROM LIN1GCO INTO LIN1GCI
007900
008000*      結果確認
008100      IF 画面-RC = 正常
008200          THEN
008300              CONTINUE
008400          ELSE
008500              DISABLE DSP
008600              MOVE 12 TO RETURN-CODE
008700              STOP RUN
008800      END-IF.
008900
009000***** 処理の判定 -- 終了か更新(続行)か -- *****
009100      EVALUATE LIN1GC-INC1
009200          WHEN 'PF02'
009300              PERFORM 更新処理
009400          WHEN 'PF10'
009500              GO TO 正常終了
009600          WHEN OTHER
009700              CONTINUE
009800      END-EVALUATE.
009900

```

図 8-35 行単位にデータを入力するコーディング (3 / 5)

```

010000*****
010100*      プログラムの終了                      *
010200*****
010300 正常終了.
010400
010500*      XMAP3のクローズ
010600      DISABLE DSP.
010700*
010800      MOVE +0 TO RETURN-CODE
010900      STOP RUN.
011000
011100*****
011200*      行単位処理 -- 各行の処理を行います --      *
011300*****
011400 行単位処理      SECTION.
011500*
011600      MOVE 1      TO カウンタ.
011700      MOVE 0      TO 合計.
011800
011900      PERFORM UNTIL ( カウンタ > 行カウンタ )
012000
012100***** 入力する明細行の入力フィールドを活性とする *****
012200      MOVE XMAP-IN-ATTR4 TO LIN1GC-商品コード-A(カウンタ)
012300      MOVE XMAP-IN-ATTR4 TO LIN1GC-数量-A(カウンタ)
012400
012500***** フォーカスの設定 (各行の先頭フィールドに設定) *****
012600      MOVE LIN1GC-商品コード-T(カウンタ)
012700      TO LIN1GC-OUTCURS-LOC0
012800*
012900*      マッピングオプションの設定
013000      IF 画面1-フラグ = 終了
013100      THEN
013200      MOVE XMAP-CNTRL2 TO LIN1GC-CNTRL0
013300      MOVE +0      TO 画面1-フラグ
013400      ELSE
013500***** マッピングオプションを「論理マップだけ」に設定 ****
013600      MOVE '2'      TO マッピングモード
013700***** 表示形態を「一部上書」に設定 *****
013800      MOVE XMAP-CNTRL1 TO LIN1GC-CNTRL0
013900      END-IF
014000*
014100*      マップ名の設定
014200      MOVE 'LIN1GCND' TO 画面マップ名
014300*
014400*      画面の表示と入力
014500      TRANSCEIVE DSP FROM LIN1GC0 INTO LIN1GC1
014600*

```

8. 標準パターンを使用した AP の作成

図 8-36 行単位にデータを入力するコーディング (4 / 5)

```

014700*      結果確認
014800      IF 画面-RC = 正常
014900      THEN
015000      CONTINUE
015100      ELSE
015200      DISABLE DSP
015300      MOVE 12 TO RETURN-CODE
015400      STOP RUN
015500      END-IF
015600*
015700*      ボタン・PFキーのチェックと該当処理の呼び出し
015800      EVALUATE LIN1GC-INCI
015900      WHEN 'ENTR'
016000***** 行ごとの処理 *****
016100      PERFORM 明細処理
016200***** 次行の処理をするため、行カウンタを更新 *****
016300      COMPUTE カウンタ = カウンタ + 1
016400      WHEN 'PF02'
016500      PERFORM 更新処理
016600      WHEN 'PF10'
016700      GO TO 正常終了
016800      WHEN OTHER
016900      CONTINUE
017000      END-EVALUATE
017100
017200      END-PERFORM.
017300*
017400 行単位処理-END.
017500      EXIT.
017600*
017700*****
017800*      各行の明細の処理 *
017900*****
018000 明細処理      SECTION.
018100*
018200***** 入力した明細行の処理を行う *****
018300***** 本来ならばDBなどからデータを取得しますが、 *****
018400***** ここでは省略します *****
018500
018600***** 入力した明細行の入力フィールドを不活性に戻す *****
018700      MOVE XMAP-IN-ATTR3 TO LIN1GC-商品コード-A(カウンタ)
018800      MOVE XMAP-IN-ATTR3 TO LIN1GC-数量-A(カウンタ).
018900*
019000***** ひとつ前の行の出カデータをクリアする *****
019100      PERFORM 前行クリア処理.
019200*
019300 明細処理-END.
019400      EXIT.

```



図 8-37 行単位にデータを入力するコーディング (5 / 5)

```

019500*
019600*****
019700*      データベース更新処理                      *
019800*****
019900 更新処理                      SECTION.
020000*
020100*----- データベース更新処理を追加（ここでは省略します） -----
020200*
020300      MOVE  終了  TO  画面1-フラグ
020400      GO    TO  業務開始.
020500*
020600 更新処理-END.
020700      EXIT.
020800*
020900*****
021000*      前行クリア処理                      *
021100*      ひとつ前の行の出力データを送信しないようにします      *
021200*****
021300 前行クリア処理                      SECTION.
021400*
021500      IF  カウンタ  >  1
021600      THEN
021700      MOVE  ALL X'1F'  TO  LIN1GC-明細-0(カウンタ - 1)
021800      ELSE
021900      CONTINUE
022000      END-IF.
022100*
022200 前行クリア処理-END.
022300      EXIT.
022400*
022500*****
022600 END  PROGRAM  LINEGC.

```

### 8.3.6 COBOL による CUI 画面を使用したフィールド単位入力例題

フィールド単位に AP に制御を戻して、入力や確認に必要な情報を画面に少しずつ表示していく方法です。フィールド単位に処理するので、少量のデータを頻繁に入出力するようなデータエントリ業務に適しています。なお、フィールド単位に処理をするため、C/S システムのように逐一、サーバとの送受信が発生するようなシステムでは、処理動作が遅くなりますのでご注意ください。

#### (1) 想定するエントリ画面

図 8-38 エントリ画面

品番	単価	数量	合計
102	100	002	200

0 .....次へ

1 .....終了

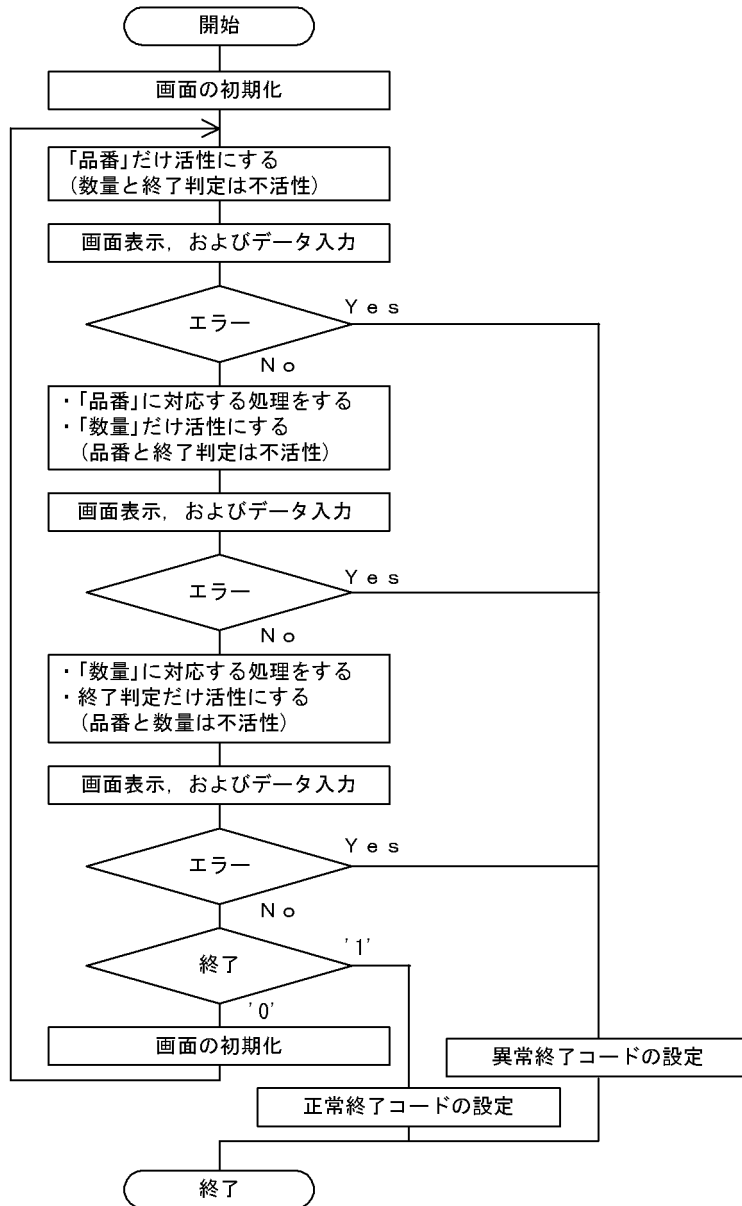
F10K F7/P

品番と数量の合計フィールドを「自動送信（入力直後に AP へ制御を渡す）」属性に定義しておき、単価と合計金額は出力フィールドとして定義しておきます。

#### 注

物理マップサイズの目安は、約 1.5KB です。

## (2) 処理の流れ



## (3) コーディング

最初に画面を表示するときは、「品番」だけ入力できるようにしておきます。

「品番」を入力すると AP に制御が移り、「単価」が表示されます。次に「数量」を入力できるようにします。「数量」が入力されたら、「金額」を計算して「終了判定」にカーソルを移動させ、終了するか、次のデータ入力を始めるかを番号で指定させます。この処理を、「終了判定」に終了の合図が入力されるまで繰り返します。

8. 標準パターンを使用した AP の作成

コーディングパターンの例を次の図に示します。マップ名は「FIE1CC」です。プログラムサイズは約 10KB です。

図 8-39 フィールド単位にデータを入力するコーディング ( 1 / 3 )

```

000100*****
000200*   プログラム名   : CUI画面-フィールド単位入力例題   *
000300*   ファイル名   : FIE1DCC   *
000400*   説明       : フィールド単位にAPとやり取りする例題です *
000500*****
000600 IDENTIFICATION   DIVISION.
000700 PROGRAM-ID.     FIELDCC.
000800*
000900 ENVIRONMENT     DIVISION.
001000*
001100 DATA           DIVISION.
001200*
001300 WORKING-STORAGE SECTION.
001400*
001500 77 正常 PIC X(5) VALUE '00000'.
001600*
001700*****
001800*   論理マップと動的変更テーブルの取込み   *
001900*****
002000* 論理マップの取込み
002100 COPY FIE1CC0.
002200 COPY FIE1CC1.
002300* 動的変更テーブルの取込み--"XMAP3インストール先¥INCLUDE"にあります
002400 COPY X3MODTBL.
002500*****
002600*   通信記述   *
002700*****
002800 COMMUNICATION   SECTION.
002900*   ディスプレイに対する通信記述項
003000 CD DSP FOR I-O WS
003100 MAP NAME IS 画面マップ名
003200 SYMBOLIC TERMINAL IS 画面端末名
003300 MAPPING MODE IS マッピングモード
003400 STATUS KEY IS 画面-RC.
003500*****
003600*   プログラム開始   *
003700*****
003800 PROCEDURE       DIVISION.
003900*
004000*   表示先の設定
004100 MOVE 'DSP001' TO 画面端末名.
004200*   出力エリアのクリア
004300 MOVE ALL X'1F' TO FIE1CCG.
004400
004500 画面処理.
004600*   マップ名の設定
004700 MOVE 'FIE1CCNC' TO 画面マップ名.
004800*

```

図 8-40 フィールド単位にデータを入力するコーディング (2 / 3)

```

004900*-----品番のみ活性にする-----
005000 MOVE XMAP-IN-ATTR3 TO FIE1CC-数量-A
005100 MOVE XMAP-IN-ATTR3 TO FIE1CC-判定-A.
005200*
005300* 画面の表示と入力
005400 TRANSCEIVE DSP FROM FIE1CC0 INTO FIE1CC1.
005500* 結果確認
005600 IF 画面-RC = 正常
005700 THEN
005800 CONTINUE
005900 ELSE
006000 DISABLE DSP
006100 MOVE 12 TO RETURN-CODE
006200 STOP RUN
006300 END-IF.
006400*
006500
006600* 品番に対する処理を行う
006700 MOVE FIE1CC-品番-I TO FIE1CC-品番-0
006800 MOVE 100 TO FIE1CC-単価-0.
006900
007000*-----次フィールド(数量)にフォーカスを位置づけ-----
007100 MOVE FIE1CC-数量-T TO FIE1CC-OUTCURS-LOC0.
007200
007300*-----数量のみ活性にする-----
007400 MOVE XMAP-IN-ATTR3 TO FIE1CC-品番-A
007500 MOVE X'1F1F' TO FIE1CC-数量-A
007600 MOVE XMAP-IN-ATTR3 TO FIE1CC-判定-A.
007700
007800* 表示形態を「一部上書」に設定
007900 MOVE XMAP-CNTRL1 TO FIE1CC-CNTRL0.
008000* マッピングオプションの設定(マージで表示)
008100 MOVE '0' TO マッピングモード.
008200
008300*
008400* 画面の表示と入力
008500 TRANSCEIVE DSP FROM FIE1CC0 INTO FIE1CC1.
008600* 結果確認
008700 IF 画面-RC = 正常
008800 THEN
008900 CONTINUE
009000 ELSE
009100 DISABLE DSP
009200 MOVE 12 TO RETURN-CODE
009300 STOP RUN
009400 END-IF.
009500*
009600*
009700* 数量に対する処理を行う
009800 COMPUTE FIE1CC-合計-0 =
009900 FIE1CC-単価-0 * FIE1CC-数量-I
010000 MOVE FIE1CC-数量-I TO FIE1CC-数量-0.
010100
010200*-----次フィールド(判定)にフォーカスを位置づけ-----
010300 MOVE FIE1CC-判定-T TO FIE1CC-OUTCURS-LOC0.

```

8. 標準パターンを使用した AP の作成

図 8-41 フィールド単位にデータを入力するコーディング (3 / 3)

```

010400
010500*-----終了判定のみ活性にする-----
010600  MOVE XMAP-IN-ATTR3      TO FIE1CC-品番-A
010700  MOVE XMAP-IN-ATTR3      TO FIE1CC-数量-A
010800  MOVE X'1F1F'          TO FIE1CC-判定-A.
010900
011000*      表示形態を「一部上書」に設定
011100  MOVE XMAP-CNTRL1        TO FIE1CC-CNTRL0.
011200*      マッピングオプションの設定 (マージで表示)
011300  MOVE '0' TO マッピングモード.
011400*
011500*
011600*      画面の表示と入力
011700  TRANSCEIVE DSP FROM FIE1CC0 INTO FIE1CC1.
011800*      結果確認
011900  IF 画面-RC = 正常
012000      THEN
012100      CONTINUE
012200      ELSE
012300      DISABLE DSP
012400      MOVE 12 TO RETURN-CODE
012500      STOP RUN
012600  END-IF.
012700*
012800*----- 終了判定 -----
012900  IF FIE1CC-判定-I NOT = 1
013000      THEN
013100*----- 再表示の準備
013200*      出力エリアのクリア
013300  MOVE LOW-VALUE TO FIE1CCG
013400*      先頭処理に戻る
013500  GO TO 画面処理
013600      ELSE
013700*----- 終了へ
013800  GO TO 正常終了
013900  END-IF.
014000*
014100*****
014200*      プログラムの終了 *
014300*****
014400  正常終了.
014500* XMAP3 のクローズ
014600  DISABLE DSP.
014700
014800  MOVE +0 TO RETURN-CODE
014900  STOP RUN.
015000
015100*****
015200 END PROGRAM FIELDCC.

```

### 8.3.7 C 言語による GUI 画面を使用した GUI 入門例題

マニュアル「XMAP3 入門」で紹介している「受注データ入力業務」を、C 言語で作成した例題です。

一つの商品を受注するごとに AP に制御を戻し、登録していく方法です。得意先コードおよび数量以外は、マウスでの操作ができます。そのため、キーボード入力が少なくてすみ、キーボードが不慣れな方でも簡単に受注データ入力が行えます。

#### (1) 想定する受注データ入力画面

図 8-42 受注データ入力画面

入力区分、商品名を「選択後次のボックスにフォーカスを移す」属性に定義しておきます。数量は数字として定義します。

注

物理マップサイズの目安は、約 2.5KB です。





リックして AP に制御を移します。次に、入力されたデータを代入して、次の入力を行います。

この処理を、[ 終了 ] ボタンをクリックするか、メニューから「終了」が選択されるまで繰り返します。

コーディングパターン例を次の図に示します。マップ名は「JYU1GH」です。プログラムサイズは約 9.5KB です。

図 8-43 C 言語での GUI 画面設計のコーディング ( 1 / 8 )

```

/*****
**   プログラム名   :   GUI 画面-GUI 入門例題           **
**   ファイル名   :   JYU1GH                           **
**   説明         :   入門マニュアルで紹介している例題です **
*****/
#include <stdio.h>

/* XMAP3が生成した論理マップの取り込み          */
#include "JYU1GH.i.h"
#include "JYU1GH.o.h"

/* 動的変更テーブル、インタフェーステーブルの取り込み          */
/* (XMAP3インストール先¥INCLUDEに入っています)          */
#include "x3modtbl.h"
#include "jsvwtatbl.h"

/* 変数宣言 */
unsigned char code_work[8];          /* 得意先コードワーク */
short input_flag;                   /* 得意先入力フラグ */
char map_name[9];                   /* 画面マップ名 */
char term_name[9];                  /* 画面端末名 */

long rtn_code;                       /* リターンコード */
int disp_flag;                       /* 画面状況 */

XMAP_COM com;                        /* 共通インタフェース領域 */

/* 画面表示用商品情報 */
typedef struct tagSYOHIN_DB {
    char    *scode;
    char    *sname;
} SYOHIN_DB;
SYOHIN_DB syohin_db[] = {
    "0001",    "BS/CSチューナー付28型テレビ",    /* 商品名 テレビ 1 */
    "0002",    "35型高精細テレビ",              /* 商品名 テレビ 2 */
    "0003",    "再生専用MDプレーヤー",           /* 商品名 オーディオ 1 */
    "0004",    "CDフェンジャー付オーディオ",     /* 商品名 オーディオ 2 */
    "0005",    "BS/CSチューナー付S-VHSビデオ",   /* 商品名 ビデオ 1 */
    "0006",    "8ミリビデオ",                   /* 商品名 ビデオ 2 */
    NULL,      NULL;                              /* 終了値 */

```

8. 標準パターンを使用した AP の作成

図 8-44 C 言語での GUI 画面設計のコーディング (2 / 8)

```

/* 関数宣言 */
static void init();
static void input_code();
static void output_file();
static void input_next();
/* 仮想端末オープン処理 */
static long xmap_open(XMAP_COM *cp,
                    char *termname,
                    char datacode,
                    char devkind);

/* 仮想端末クローズ処理 */
static long xmap_close(XMAP_COM *cp);
/* モードの設定処理 */
static long xmap_mdo(XMAP_COM *cp,
                    char opt1,
                    char opt2);

/* 画面入出力処理 */
static long xmap_trans(XMAP_COM *cp,
                    char *mapname,
                    char *outlmap,
                    char *inlmap,
                    long inlmaplen);

/* マクロ関数 */
#define MSET(a, b) memset((a), (b), sizeof(a)) /* MSET(a, b) aにbを設定 */
#define MCPY(a, b) memcpy((a), (b), sizeof(a)) /* MCPY(a, b) bをaに複写 */
#define MCMP(a, b) memcmp((a), (b), sizeof(a)) /* MCMP(a, b) aとbを比較 */

/* マクロ定義 */
#define NORMAL 0 /* 正常 */
#define ABNORMAL 12 /* 異常 */

/*****
** 受注データ入力プログラムメイン処理
**
*****/
int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance,
                  LPSTR lpCmdLine, int nShowCmd)
{
    /* 初期設定 */
    input_flag = 0;
    memset(code_work, XMAP_NODATA, sizeof(code_work));

    /* 仮想端末名の設定 */
    strcpy(term_name, "DSP001");

    /* マップ名の設定 */
    strcpy(map_name, "JYU1GHND");

    /* オープン要求 */
    rtn_code = xmap_open(&com, term_name, XMAP_NODATA, 0);

```

図 8-45 C 言語での GUI 画面設計のコーディング (3 / 8)

```

/* 結果確認 */
if (NORMAL != rtn_code) {
    return (ABNORMAL);
}

MCPY(JYU1GHI, JYU1GH_FIELD0003_I, "01");
disp_flag = 0;
/* プッシュボタンにより次データ入力又は終了処理を行う */
while (0 == disp_flag) {
    /* 論理マップの設定 */
    init();

    /* 出力要求 */
    rtn_code = xmap_trans(&com, map_name, (char *)&JYU1GH0,
                        (char *)&JYU1GHI, JYU1GHI, JYU1GHS);

    /* 結果確認 */
    if (NORMAL != rtn_code) {
        /* クローズ要求 */
        xmap_close(&com);
        return (ABNORMAL);
    }

    if ((0 == MCMP(JYU1GHI, JYU1GH_INCI, "PF03")) ||
        (0 == MCMP(JYU1GHI, JYU1GH_INCI, "BREK"))) {
        disp_flag = 1;
        continue;
    } else if ((0 == MCMP(JYU1GHI, JYU1GH_INCI, "PF02")) ||
               (0 == MCMP(JYU1GHI, JYU1GH_INCI, "ENTR"))) {
        input_next();
    }

    if (0 == input_flag) {
        input_code();
    }
}

/* クローズ要求 */
rtn_code = xmap_close(&com);

/* 結果確認 */
if (NORMAL != rtn_code) {
    return (ABNORMAL);
} else {
    return (NORMAL);
}
}

```

## 8. 標準パターンを使用した AP の作成

図 8-46 C 言語での GUI 画面設計のコーディング (4 / 8)

```
/*
*****
** init : 初期化処理
*****
static void init()
{
    SYOHIN_DB          *sdp;
    int                i;

    /* 出力論理マップの初期化を行います */
    memset(&JYU1GH0, XMAP_NODATA, sizeof(JYU1GH0));
    JYU1GH0.JYU1GHL = JYU1GHT;
    MSET(JYU1GH0.JYU1GHZ, 0x00);

    /* 入金区分を判定し、該当するラジオボタンを選択された */
    /* 状態にします (動的変更) */
    MCPY(JYU1GH0.JYU1GH_RADIO0004_A, xmap_button_sel3);
    MCPY(JYU1GH0.JYU1GH_RADIO0005_A, xmap_button_sel3);
    MCPY(JYU1GH0.JYU1GH_RADIO0006_A, xmap_button_sel3);
    if (0 == MCMP(JYU1GH1.JYU1GH_FIELD0003_I, "01")) {
        MCPY(JYU1GH0.JYU1GH_RADIO0004_A, xmap_button_sel2);
    } else if (0 == MCMP(JYU1GH1.JYU1GH_FIELD0003_I, "02")) {
        MCPY(JYU1GH0.JYU1GH_RADIO0005_A, xmap_button_sel2);
    } else if (0 == MCMP(JYU1GH1.JYU1GH_FIELD0003_I, "03")) {
        MCPY(JYU1GH0.JYU1GH_RADIO0006_A, xmap_button_sel2);
    }

    /* リストボックスにラベルとコードを設定します */
    /* 本来はデータベースから入力してループで設定します */
    /* が、ここでは固定値を直接設定しています */
    for (i = 0, sdp = syohin_db; NULL != sdp->scode; i++, sdp++) {
        memcpy(JYU1GH0.JYU1GH_LIST0007_0[i].JYU1GH_LIST_LABEL0007_0,
            sdp->sname, strlen(sdp->sname));

        memcpy(JYU1GH0.JYU1GH_LIST0007_0[i].JYU1GH_LIST_CODE0007_0,
            sdp->scode, strlen(sdp->scode));
    }

    MCPY(JYU1GH0.JYU1GH_FIELD0001_0, code_work);

    /* 得意先コードが正しく入力されていれば商品名選択 */
    /* リストボックスに、入力されていなければ得意先入力 */
    /* テキストボックスにフォーカスを位置づけます */
    if (1 == input_flag) {
        MCPY(JYU1GH0.JYU1GH_OUTCURS_LOC0, JYU1GHD.JYU1GH_FIELD0007_T);
    } else {
        MCPY(JYU1GH0.JYU1GH_OUTCURS_LOC0, JYU1GHD.JYU1GH_FIELD0001_T);
    }

    return;
}
*/
```

図 8-47 C 言語での GUI 画面設計のコーディング (5 / 8)

```

/*****
** input_code : 得意先コード検索処理
**
*****/
static void input_code()
{
    input_flag = 0;
    memset(code_work, XMAP_NODATA, sizeof(code_work));

    /* 正しい得意先コードが入力されたらそれをワークに退避します */
    if ((0 == MCMP(JYU1GHI. JYU1GH_FIELD0001_I, "SA001010")) ||
        (0 == MCMP(JYU1GHI. JYU1GH_FIELD0001_I, "SB002020")) ||
        (0 == MCMP(JYU1GHI. JYU1GH_FIELD0001_I, "SC003030")) ||
        (0 == MCMP(JYU1GHI. JYU1GH_FIELD0001_I, "SD004040")) ||
        (0 == MCMP(JYU1GHI. JYU1GH_FIELD0001_I, "SE005050"))) {
        input_flag = 1;
        MCPY(code_work, JYU1GHI. JYU1GH_FIELD0001_I);
    }

    return;
}

/* ファイル出力処理 */
/*****
** output_file : ファイル出力処理
**
*****/
static void output_file()
{
    /* ここではファイルを用意していませんが、本当は
    /* 商品コード、数量、割引の有無をファイルに出力します */

    return;
}

```

## 8. 標準パターンを使用した AP の作成

図 8-48 C 言語での GUI 画面設計のコーディング (6 / 8)

```
/******  
** input_next : 次データの入力処理 **  
*****/  
static void input_next()  
{  
    SYOHIN_DB      *sdp;  
  
    /* 顧客コードが正しく入力されていない場合は処理を行いません */  
    if (0 == input_flag) {  
        return;  
    }  
  
    /* 商品コードが正しく入力されていれば受注データをファイルに出力します */  
    for (sdp = syohin_db; NULL != sdp->scode; sdp++) {  
        if (0 == MCMP(JYU1GHI.JYU1GH_FIELD0007_1, sdp->scode)) {  
            break;  
        }  
    }  
    if (NULL != sdp->scode) {  
        output_file();  
    }  
  
    return;  
}
```

図 8-49 C 言語での GUI 画面設計のコーディング (7 / 8)

```

/*****
** xmap_open : 仮想端末をオープンします
** dev_kind 0:画面
**          1:帳票
*****/
static long xmap_open(XMAP_COM *cp,
                    char *termname,
                    char *datacode,
                    char devkind)
{
    XMAP_REQ xmap_req;
    XMAP_OPN xmap_opn;

    long rc;

    memset(cp, 0x00, sizeof(XMAP_COM));
    memcpy(cp->xmap_com_id, "XP", 4);
    cp->xmap_com_itype = 'I';
    memcpy(cp->xmap_com_tname, termname, strlen(termname));
    if (devkind == 0)
        memcpy(cp->xmap_com_msg, "BWS", 4); /* 画面 */
    else
        memcpy(cp->xmap_com_msg, "OWS", 4); /* 帳票 */
    memset(&xmap_req, ' ', sizeof(XMAP_REQ));
    memcpy(xmap_req.xmap_req_type, "OPEN", 4);
    memset(&xmap_opn, ' ', sizeof(XMAP_OPN));
    xmap_opn.xmap_opn_dcode_set = 'I';
    xmap_opn.xmap_opn_dcode = datacode;

    rc = jsvwdrv(cp, &xmap_req, &xmap_opn, 0L);

    return (rc);
}

/*****
** xmap_mdo : モードの設定をします
*****/
static long xmap_mdo(XMAP_COM *cp,
                    char *opt1,
                    char *opt2)
{
    XMAP_REQ xmap_req;
    XMAP_MDO xmap_mdo;

    long rc;

    memset(&xmap_req, ' ', sizeof(XMAP_REQ));
    memcpy(xmap_req.xmap_req_type, "MDO", 4);
    xmap_mdo.xmap_mdo_opt1 = opt1;
    xmap_mdo.xmap_mdo_opt2 = opt2;
    rc = jsvwdrv(cp, &xmap_req, 1L, &xmap_mdo);

    return (rc);
}

```

## 8. 標準パターンを使用した AP の作成

図 8-50 C 言語での GUI 画面設計のコーディング (8 / 8)

```

/*****
** xmap_close : 仮想端末をクローズします **
*****/
static long xmap_close(XMAP_COM *cp)
{
    XMAP_REQ xmap_req;
    long rc;

    memset(&xmap_req, ' ', sizeof(XMAP_REQ));
    memcpy(xmap_req.xmap_req_type, "CLOS", 4);
    rc = jsvwdrv(cp, &xmap_req, 0L, 0L);

    return (rc);
}

/*****
** xmap_trans : 画面の入出力を行います **
*****/
static long xmap_trans(XMAP_COM *cp,
                      char *mapname,
                      char *outlmap,
                      char *inlmap,
                      long inlmaplen)
{
    XMAP_REQ xmap_req;
    long rc;

    /*** 画面の出力 ***/
    memset(cp->xmap_com_mapname, ' ', sizeof(cp->xmap_com_mapname));
    memcpy(cp->xmap_com_mapname, mapname, strlen(mapname));
    memset(&xmap_req, ' ', sizeof(XMAP_REQ));
    memcpy(xmap_req.xmap_req_type, "SEND", sizeof(xmap_req.xmap_req_type));
    xmap_req.xmap_req_opt2 = '1';
    rc = jsvwdrv(cp, &xmap_req, outlmap, 0L);
    if (rc != 0) return (rc);

    /*** 画面の入力 ***/
    cp->xmap_com_inlmg = inlmaplen;
    memset(&xmap_req, ' ', sizeof(XMAP_REQ));
    memcpy(xmap_req.xmap_req_type, "RECV", sizeof(xmap_req.xmap_req_type));
    xmap_req.xmap_req_opt1 = 'F';
    rc = jsvwdrv(cp, &xmap_req, inlmap, 0L);

    return (rc);
}

```



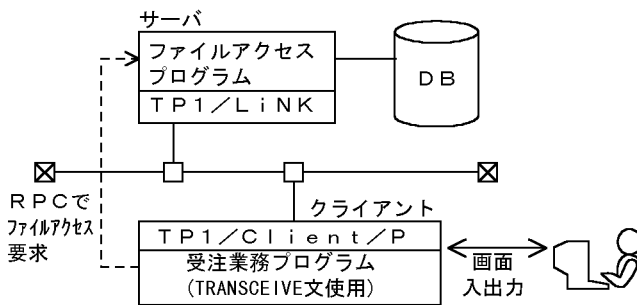
### 8.3.8 COBOL による OLTP クライアント AP 例 (OpenTP1 を使用した受注業務)

OpenTP1 を使用した C/S システム業務では、クライアントアプリケーション型の C/S システム業務となります。このとき、クライアントからサーバへの通信は RPC (Remote Procedure Call) コーディングを使用します。

#### (1) 処理の概要

クライアントにある AP が、RPC でサーバの OpenTP1 に対して処理を要求し、受注データ処理を行います。クライアントの AP は、TRANSCIVE 文で画面の入出力を要求し、RPC でサーバに対してファイルの参照・更新を要求します。

図 8-51 OLTP クライアント AP の処理の概要

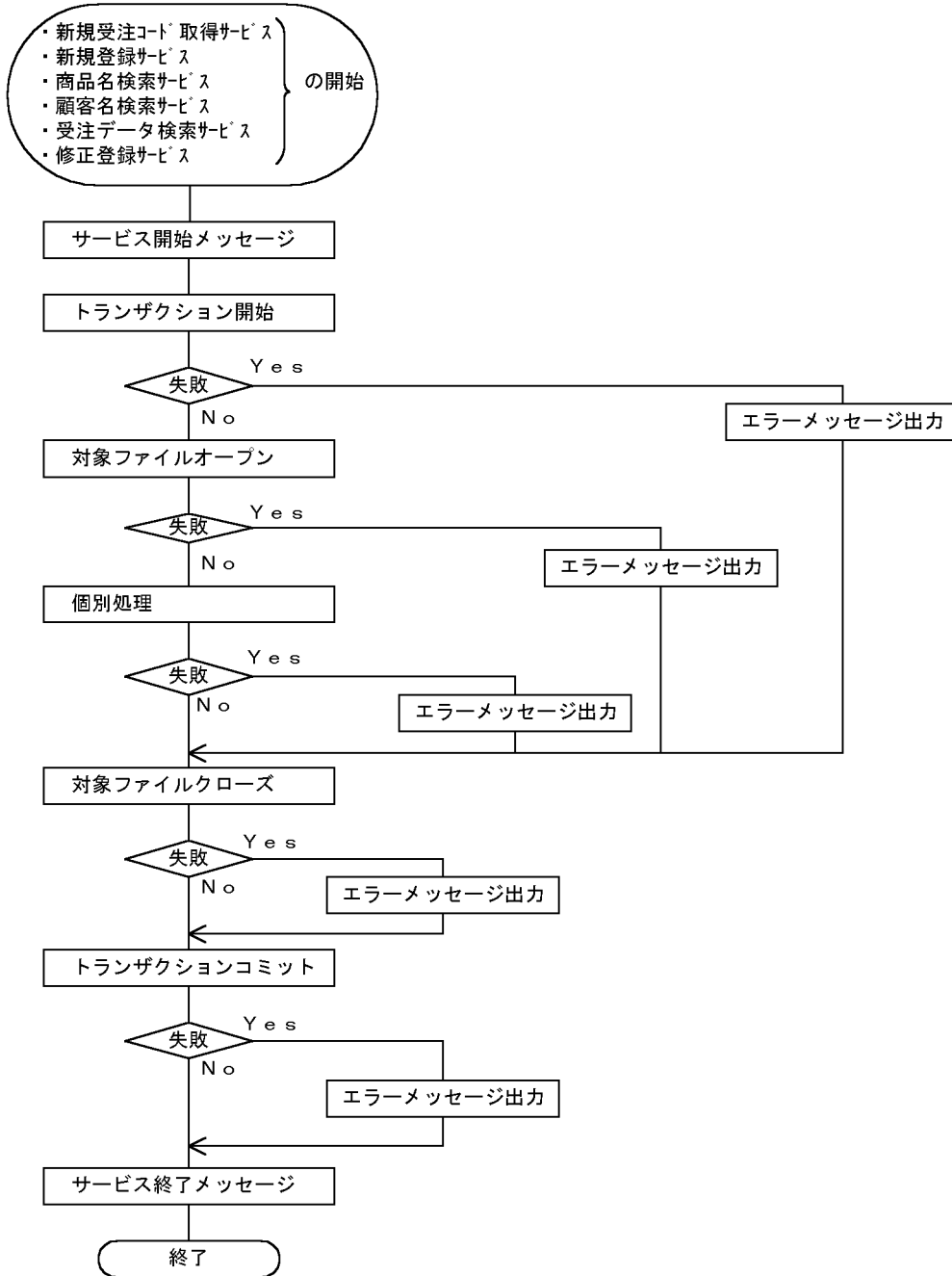


注

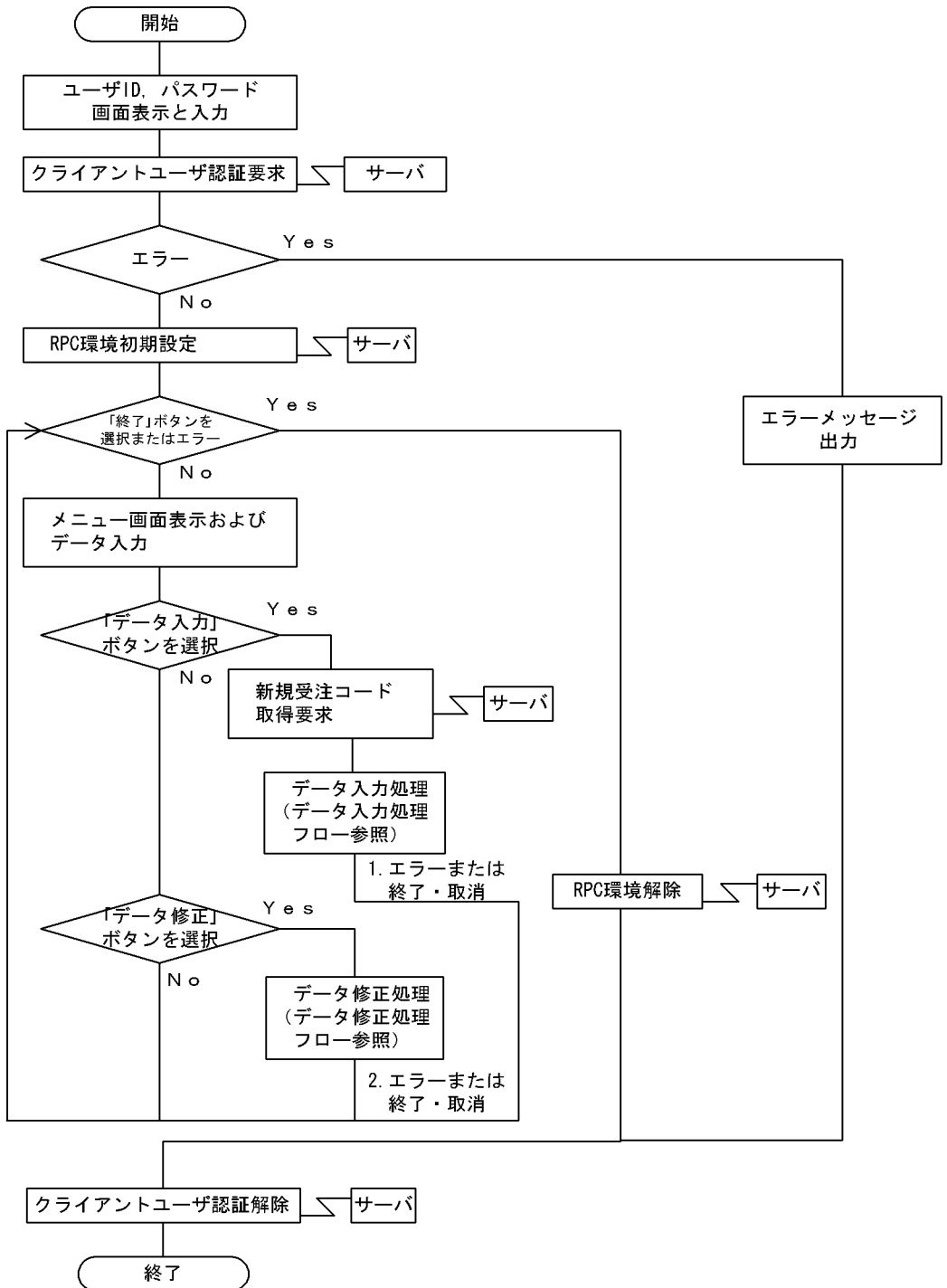
物理マップサイズの目安は、約 5.5KB です。

## (2) 処理の流れ

### ●サーバでの処理フロー

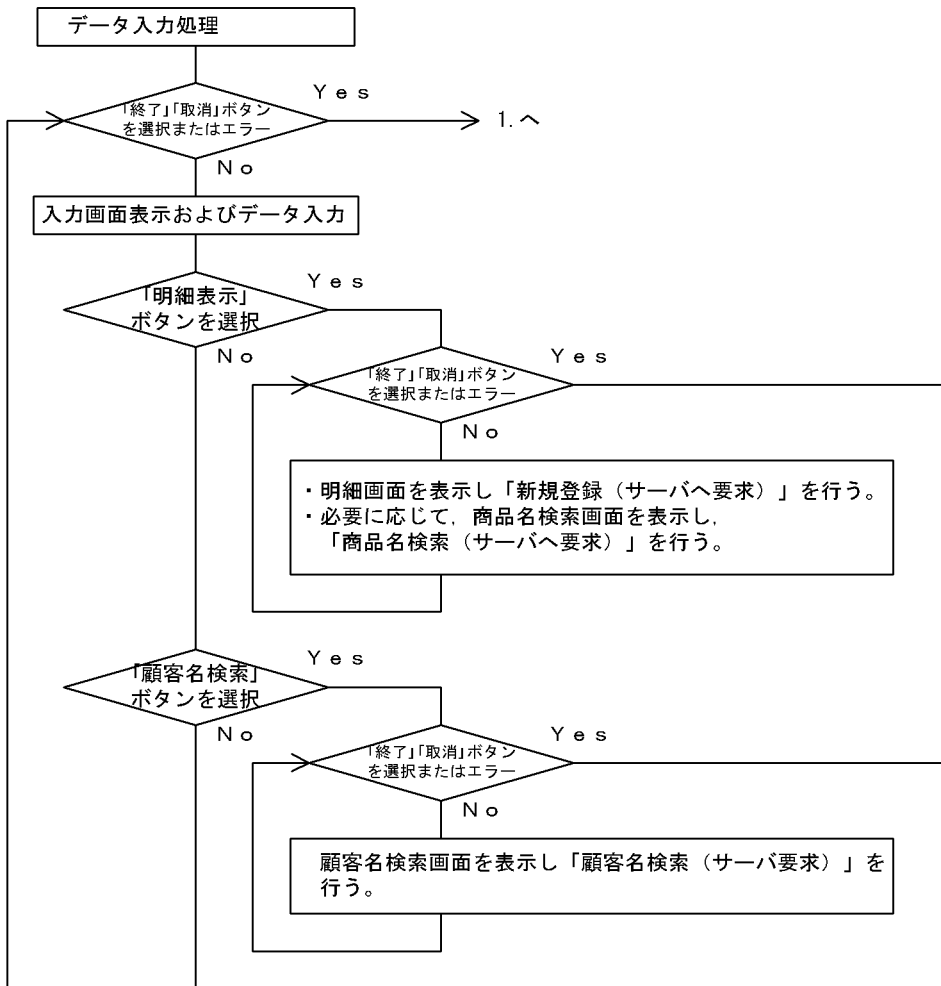


## ●クライアントでの処理フロー

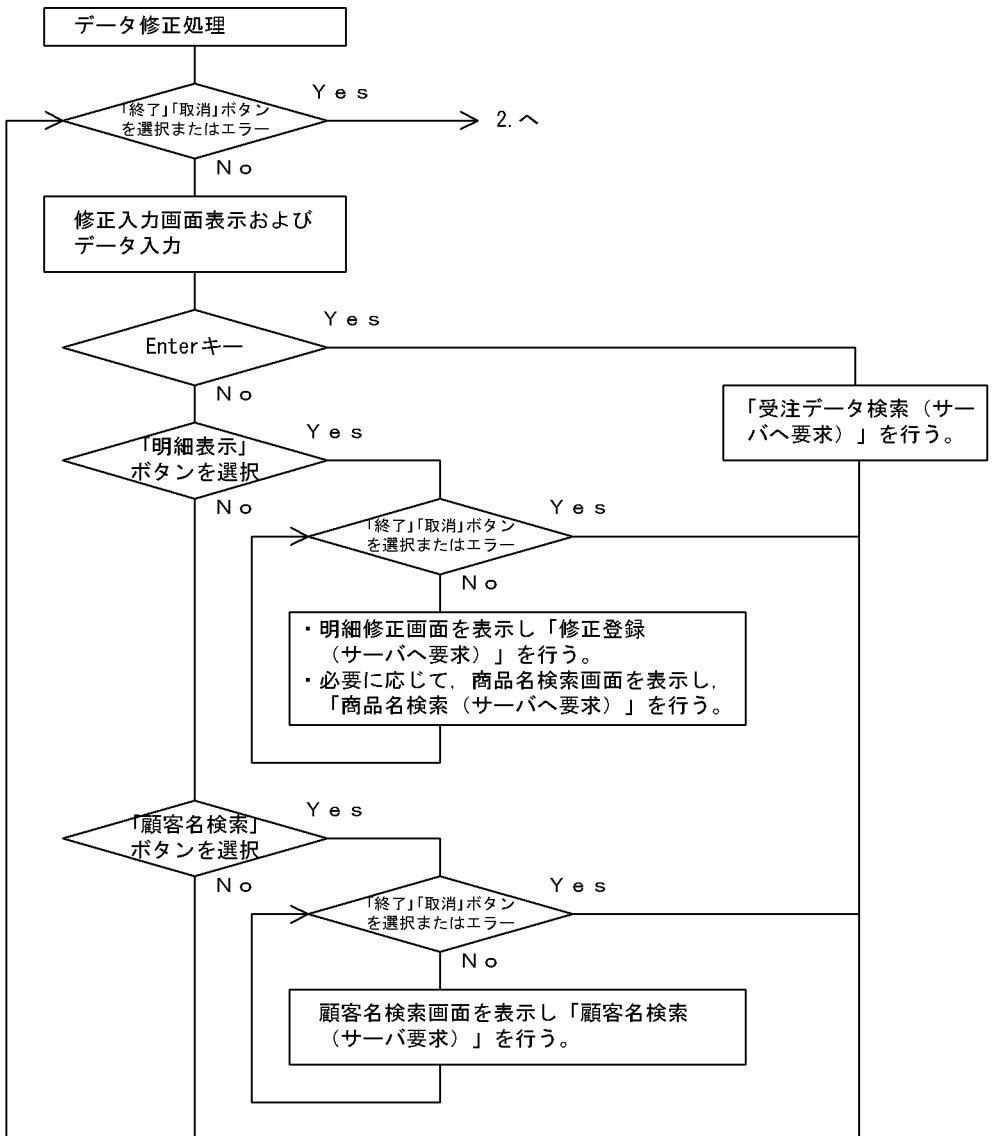


8. 標準パターンを使用した AP の作成

●データ入力処理フロー(クライアント)



## ●データ修正処理フロー(クライアント)



## (3) コーディング

サーバ側のコーディング例およびクライアント側のコーディング例を次の図に示します。

なお、このプログラムソースは、XMAP3 で提供されています。

## 8. 標準パターンを使用した AP の作成

図 8-52 サービスプログラム (1 / 4)

```
●サーバでの処理のコーディング

*
*****
*          ***サービスプログラム          *
*****
*
  IDENTIFICATION DIVISION.
  PROGRAM-ID.  ORDERSRV.
*
  ENVIRONMENT DIVISION.
*****
*          ISAM ファイルの定義          *
*****
  INPUT-OUTPUT SECTION.
  FILE-CONTROL.
    SELECT ORDER-FILE ASSIGN TO 'c:¥opentp1¥aplib¥ORDER'
      ORGANIZATION IS INDEXED
      ACCESS MODE IS DYNAMIC
      RECORD KEY IS DENPYOU-NO
      FILE STATUS IS ORDER-FS.
    SELECT CUST-FILE ASSIGN TO 'CUST'
      ORGANIZATION IS INDEXED
      ACCESS MODE IS DYNAMIC
      RECORD KEY IS KOUBAN-C
      ALTERNATE RECORD KEY IS CHIKU-NO-C
        WITH DUPLICATES
      ALTERNATE RECORD KEY IS CUST-NO-C
        WITH DUPLICATES
      FILE STATUS IS CUST-FS.
*
  DATA DIVISION.
  FILE SECTION.
  FD ORDER-FILE.
  COPY ORDERD.
  FD CUST-FILE.
```

図 8-53 サービスプログラム (2 / 4)

```

COPY CUSTD.
*
*****
*           データ領域の設定           *
*****
*
WORKING-STORAGE SECTION.
01 TRN-ARG1.
02 REQUEST                PIC X(008) VALUE SPACE.
02 STATUS-CODE            PIC X(008) VALUE SPACE.
01 LOOP-CNT                PIC 9(002).
01 ORDER-FS                PIC X(002).
01 CUST-FS                 PIC X(002).
01 FIND-CNT                PIC 9(004).
*
LINKAGE SECTION.
01 IN-DATA                PIC X(008).
01 IN-LENG                PIC S9(009) COMP.
01 OUT-DATA.
02 HEADER.
03 H-ORDER-CODE           PIC X(008).
03 H-CUST-CODE            PIC X(008).
03 H-CUST-NAME            PIC N(016).
03 H-CNTRCT-CODE          PIC X(004).
03 H-CNTRCT-NAME          PIC N(016).
03 H-BRANCH                PIC X(002).
03 H-DIVISIONS            PIC X(001).
03 H-SPECIAL              PIC X(001).
03 H-CARD                  PIC X(001).
03 H-SALE                  PIC X(001).
03 H-RETURN-CODES         PIC S9(009) COMP.
03 H-TOTAL                 PIC S9(009) COMP.
02 DETAILS OCCURS 40 TIMES.
03 D-ITEM-NUM             PIC X(002).
03 D-PRDCT-CODE           PIC X(010).
03 D-PRDCT-NAME           PIC N(010).
03 D-PRICE                 PIC S9(009) COMP.
03 D-NUMBERS               PIC S9(009) COMP.
03 D-SUBTOTAL             PIC S9(009) COMP.
01 OUT-LENG                PIC S9(009) COMP.
*
PROCEDURE DIVISION USING IN-DATA IN-LENG OUT-DATA OUT-LENG.
DISPLAY '受注検索を開始します。'          ... 開始メッセージ 出力
*
*****
*           送信バッファの初期化           *
*****
*
MOVE SPACE                TO OUT-DATA.
MOVE 1858                  TO OUT-LENG.
MOVE ZERO                  TO FIND-CNT.

```

## 8. 標準パターンを使用した AP の作成

図 8-54 サービスプログラム ( 3 / 4 )

```
*
*****
*          TRN-BEGIN (トランザクションの開始)          *
*****
*
  MOVE 'BEGIN' TO REQUEST OF TRN-ARG1.
  CALL 'CBLDCTRN' USING TRN-ARG1.          ... トランザクション開始
:
*
*****
*          受注ファイルを検索する          *
*****
*
  OPEN INPUT ORDER-FILE.                  ... 処理対象のファイルオープン
:
*
  MOVE IN-DATA TO DENPYOU-NO.
  READ ORDER-FILE KEY DENPYOU-NO.          ... 個別処理
  IF ORDER-FS NOT = '00' THEN
    DISPLAY '受注ファイルのレコード入力処理が'
            '失敗しました。' ORDER-FS
  MOVE -1 TO H-RETURN-CODES
:
*
  PROG-END.
  CLOSE ORDER-FILE.                        ... 処理対象のファイルクローズ
:
*
*****
*          トランザクションのコミット          *
*****
*
  MOVE 'U-COMMIT' TO REQUEST OF TRN-ARG1.
  CALL 'CBLDCTRN' USING TRN-ARG1.          ... トランザクション終了
:
*
*****
*          サービス関数の終了処理          *
*****
*
  DISPLAY 'サービス関数を終了します。'.    ... 終了メッセージ 出力
  END PROGRAM ORDERSRV.
```



図 8-55 サービスプログラム (4 / 4)

```

OPEN INPUT ORDER-FILE.                ... 処理対象のファイルオープン
:
*
MOVE IN-DATA TO DENPYOU-NO.
READ ORDER-FILE KEY DENPYOU-NO.        ... 個別処理
IF ORDER-FS NOT = '00' THEN
  DISPLAY '受注ファイルのレコード入力処理が'
    '失敗しました。' ORDER-FS
MOVE -1 TO H-RETURN-CODES
:
*
PROG-END.
CLOSE ORDER-FILE.                      ... 処理対象のファイルクローズ
:
*
*****
*           トランザクションのコミット           *
*****
*
MOVE 'U-COMMIT' TO REQUEST OF TRN-ARG1.
CALL 'CBLDCTRN' USING TRN-ARG1.        ... トランザクション終了
:
*
*****
*           サービス関数の終了処理           *
*****
*
DISPLAY 'サービス関数を終了します。' ... 終了メッセージ' 出力
END PROGRAM ORDERSRV.

```

8. 標準パターンを使用した AP の作成

図 8-56 OpenTP1 を使用した受注業務 ( 1 / 10 )

```

*****
*   プログラム名   :  G U I 画面 - O L T P クライアント例題   *
*   ファイル名    :  O L T P 1 G C                               *
*   説明          :  一連の受発注業務をサーバ側の           *
*                  O p e n T P 1 に処理要求する例題です     *
*****
IDENTIFICATION   DIVISION.
PROGRAM-ID.     OLTP1GC.

ENVIRONMENT      DIVISION.

DATA             DIVISION.
WORKING-STORAGE SECTION.

:
:

*   サーバとのインターフェース
    COPY          MACRO.
*****
*   論理マップと動的変更テーブルの取込み   *
*   出力用 : マップ名 + 0. 入力用 : マップ名 + 1 *
*****
*   ユーザ I D & パスワード入力画面
    COPY          OLTOGCO.
    COPY          OLTOGCI.
*   メニュー画面
    COPY          OLT1GCO.
    COPY          OLT1GCI.
*   受注データ入力及び修正画面 (ヘッダ)
    COPY          OLT2GCO.

:

*   動的変更テーブル
    COPY          X3MODTBL.
*****
*   通信記述   *
*****
COMMUNICATION SECTION.
*   ワークステーションディスプレイに対する通信記述項
    CD  DSP  FOR  I-O  WS
        MAP  NAME  IS  画面マップ名
        STATUS  KEY  IS  画面-RC.
*
*****
*   受注伝票発行業務開始   *
*****
PROCEDURE        DIVISION.

```

図 8-57 OpenTP1 を使用した受注業務 ( 2 / 10 )

```

業務開始.
****ユーザID&パスワード入力画面の表示&入力*****
*初期設定
  MOVE ALL X'1F' TO OLT0GCG.
*マップ名の設定
  MOVE 'OLT0GCND' TO 画面マップ名.
*画面の入出力
  TRANSCEIVE DSP FROM OLT0GCO INTO OLT0GCI.
*結果確認
  IF 画面-RC NOT '0000' AND 画面-RC NOT = '10004'
    THEN
      STOP RUN
    ELSE
      CONTINUE
*****
*
* クライアントユーザの認証要求 (demo/demo)
  MOVE 'demo' TO CI-L-NAME.
  MOVE 'demo' TO CI-P-NAME.
  CALL 'CBLDCCLT' USING ARG-CLTIN.
  IF CI-STATUS-CODE NOT = '00000' THEN
    DISPLAY 'クライアントユーザの認証に失敗しました'
    CI-STATUS-CODE
  GO TO 終了
  END-IF.
*
* RPC環境の初期設定
  CALL 'CBLDCRPC' USING ARG-RPC-OPEN.
  IF RO-STATUS-CODE NOT = '00000' THEN
    GO TO 業務終了
  END-IF.
*
* 主処理ループ
  PERFORM UNTIL OLT1GC-END-FLG = 'E'
*   メニュー画面表示
    MOVE 'OLT1GCND' TO 画面マップ名
    TRANSCEIVE DSP FROM OLT1GCO INTO OLT1GCI

    :
    :

*   選択業務判定 (正常ケースの場合)
    EVALUATE OLT1GC-INCI
      WHEN 'PF01'
        MOVE SPACE TO OLT1GC-CNTRL0
          OLT1GC-ERRMSG=0
        PERFORM 受注データ入力処理部
      WHEN 'PF02'
        :
      WHEN 'PF12'
        MOVE 'E' TO OLT1GC-END-FLG
      WHEN OTHER

```

8. 標準パターンを使用した AP の作成

図 8-58 OpenTP1 を使用した受注業務 ( 3 / 10 )

```

*           エラーメッセージ出力処理
              MOVE XMAP-CNTRL1 TO OLT1GC-CNTRL0
              MOVE ERRMSG-P     TO OLT1GC-ERRMSG-0
              END-EVALUATE
              END-IF
              END-PERFORM.

*****
* 受注伝票発行業務終了 *
*****
業務終了.
* R P C環境の解除
  CALL 'CBLDCRPC' USING ARG-RPC-CLOSE.
*
終了.
* クライアントユーザの認証解除
  CALL 'CBLDCCLT' USING ARG-CLTOUT.
  STOP RUN.
*
*****
* 業務メニューでの P F 1 : 受注データ入力処理 *
*****
受注データ入力処理部 SECTION.
*
* フラグ初期化
  MOVE SPACE          TO OLT2GC-END-FLG.
*
* 新規受注コード取得
  MOVE 'ORDERNUM'    TO RP-G-NAME.
  MOVE 'number'      TO RP-S-NAME.
  CALL 'CBLDCRPC' USING ARG-RPC-PRM ARG-OCODE-IN ARG-OCODE-OUT.
  IF RP-STATUS-CODE NOT = '00000'
* 異常処理の場合
  THEN
    MOVE 'E'          TO OLT2GC-END-FLG
* 正常処理の場合
  ELSE
* 入力画面初期化 (ヘッダ/明細画面)
    MOVE ALL X'1F' TO OLT2GC0
    MOVE OLT2GCT TO OLT2GCL
    MOVE +0      TO OLT2GCZ
    MOVE OLT2GC-KCODE-T TO OLT2GC-OUTCURS-LOC0
  END-IF.
*
* 受注コードを画面にセット
  MOVE OC-ORDER-CODE TO OLT2GC-DCODE-0.
*
* 入力処理ループ
  PERFORM UNTIL OLT2GC-END-FLG = 'E'
* 入力画面出力処理
    MOVE 'OLT2GCND' TO 画面マップ名
    TRANSCEIVE DSP FROM OLT2GC0 INTO OLT2GC1
  :
* 選択業務判定

```

図 8-59 OpenTP1 を使用した受注業務 ( 4 / 10 )

```

EVALUATE OLT2GC-INC1
WHEN 'PF01'
:
PERFORM 受注明細入力処理部
:
WHEN 'PF03'
PERFORM 顧客名検索部
MOVE OLT2GC-TCODE-T TO OLT2GC-OUTCURS-LOCO
WHEN 'PF12'
* 取消
MOVE 'E' TO OLT2GC-END-FLG
WHEN OTHER
CONTINUE
END-EVALUATE
END-IF
END-PERFORM.
受注データ入力処理部-END.
EXIT.
*
*****
* 受注データ入力画面でのPF1:受注明細入力処理部 *
*****
受注明細入力処理部 SECTION.
*
:
PERFORM UNTIL OLTBGC-END-FLG = 'E'
* 受注明細入力画面表示
MOVE 'OLTBGCND' TO 画面マップ名
TRANSCEIVE DSP FROM OLTBGC0 INTO OLTBGC1
:
*
* 選択業務判定
EVALUATE OLTBGC-INC1
WHEN 'PF01'
* 決定
PERFORM 計算処理部
PERFORM 新規受注登録処理
WHEN 'PF03'
PERFORM 商品名検索部

```

8. 標準パターンを使用した AP の作成

図 8-60 OpenTP1 を使用した受注業務 ( 5 / 10 )

```

        WHEN 'PF05'
*       印刷
        CONTINUE
        WHEN 'PF06'
*       印刷書式設定
        CONTINUE
        WHEN 'PF07'
*       印刷取消
        CONTINUE
        WHEN 'PF08'
        PERFORM 計算処理部
        WHEN 'PF12'
*       取消
        MOVE 'E' TO OLTBGC-END-FLG
        WHEN OTHER
        CONTINUE
        END-EVALUATE
        END-PERFORM.
        受注明細入力処理部-END.
        EXIT.
*
*****
* 受注データ入力画面での P F 1 : 新規受注登録処理 *
*****
        新規受注登録処理 SECTION.
*
* 画面データ→インターフェーステーブル
        MOVE OLT2GC-DCODE-0 TO OR-ORDER-CODE.
        MOVE OLT2GC-KCODE-1 TO OR-CUSTOMER-CODE.
        MOVE OLT2GC-KNAME-0 TO OR-CUSTOMER-NAME.
        MOVE OLT2GC-TCODE-1 TO OR-TANTO-CODE.
        MOVE OLT2GC-TNAME-0 TO OR-TANTO-NAME.
        MOVE OLT2GC-SCODE-1 TO OR-SITEN-NO.
        MOVE OLTBGC-RBT1-1 TO OR-NYUKIN.
        IF OLTBGC-CBT1-1 = 'Y'
        THEN
            MOVE 1 TO OR-TOKUTA1
        ELSE
            MOVE 0 TO OR-TOKUTA1
        END-IF.
        IF OLTBGC-CBT2-1 = 'Y'
        THEN
            MOVE 1 TO OR-CARD
        ELSE
            MOVE 0 TO OR-CARD
        END-IF.
        IF OLTBGC-CBT3-1 = 'Y'
        THEN
            MOVE 1 TO OR-SALE
        ELSE
            MOVE 0 TO OR-SALE
        END-IF.
    
```

図 8-61 OpenTP1 を使用した受注業務 ( 6 / 10 )

```

*
MOVE BACK-GOUKEI TO OR-GOUKEI.
MOVE 1 TO WORK.
PERFORM UNTIL WORK > 40
    MOVE BACK-SCODE(WORK) TO OR-SEIHIN-CODE(WORK)
    MOVE BACK-SNAME(WORK) TO OR-SEIHIN-NAME(WORK)
    MOVE BACK-TANKA(WORK) TO OR-TANKA(WORK)
    MOVE BACK-SURYO(WORK) TO OR-SU(WORK)
    MOVE BACK-SYOUKEI(WORK) TO OR-SYOUKEI(WORK)
    COMPUTE WORK = WORK + 1
END-PERFORM.
*
* 新規登録サービス要求
MOVE 'NEWORDER' TO RP-G-NAME.
MOVE 'new' TO RP-S-NAME.
CALL 'CBLDCRPC' USING ARG-RPC-PRM ARG-ORDER ARG-ORDER-OUT.
IF RP-STATUS-CODE = '00000'
THEN
*   正常終了メッセージ出力
MOVE 'OLTGGCND' TO 画面マップ名
TRANSCIVE DSP FROM OLTGGCO INTO OLTGGCI
MOVE 'E' TO OLT2GC-END-FLG
MOVE 'E' TO OLTBGC-END-FLG
ELSE
*   異常終了メッセージ出力
MOVE 'OLTFGCND' TO 画面マップ名
TRANSCIVE DSP FROM OLTFGCO INTO OLTFGCI
END-IF.
受注修正登録処理-END.
EXIT.
*
*****
*   業務メニューでのPF2 : 受注データ修正処理   *
*****
受注データ修正処理部 SECTION.
*
:
```

8. 標準パターンを使用した AP の作成

図 8-62 OpenTP1 を使用した受注業務 ( 7 / 10 )

```
受注データ修正処理部-END.  
EXIT.  
*  
*****  
* 受注データ修正画面でのENTER:受注データ検索処理部 *  
*****  
受注データ検索処理部 SECTION.  
*  
:  
  
受注データ検索処理部-END.  
EXIT.  
*  
*****  
* 受注データ修正画面でのPF1:受注明細修正処理部 *  
*****  
受注明細修正処理部 SECTION.  
*  
:  
  
受注明細修正処理部-END.  
EXIT.  
*  
*****  
* 受注データ修正画面でのPF1:受注修正登録処理 *  
*****  
受注修正登録処理 SECTION.  
*  
:  
  
受注修正登録処理部-END.  
EXIT.  
*  
*****  
* 受注データ入力・修正画面でのPF3:顧客名検索部 *  
*****  
顧客名検索部 SECTION.  
*  
:  
  
* 画面処理ループ  
MOVE SPACE TO OLTAGC-END-FLG.  
PERFORM UNTIL OLTAGC-END-FLG = 'E'  
* 顧客名検索画面表示  
MOVE 'OLTAGCND' TO 画面マップ名  
TRANSCIVE DSP FROM OLTAGCO INTO OLTAGCI  
IF 画面-RC NOT = '00000' AND 画面-RC NOT = '10004' THEN  
GO TO 顧客名検索部-END  
END-IF  
* 選択業務判定  
EVALUATE OLTAGC-INCI  
WHEN 'PF01'
```



図 8-63 OpenTP1 を使用した受注業務 ( 8 / 10 )

```

*      了解
      MOVE OLTAGC-KDATA-I TO WK111-INF
      MOVE WK111-CUST-CODE TO OLT2GC-KCODE-O
      MOVE WK111-CUST-NAME TO OLT2GC-KNAME-O
      MOVE 'E' TO OLTAGC-END-FLG
      WHEN 'PF12'
*      取消
      MOVE 'E' TO OLTAGC-END-FLG
      WHEN 'ENTR'
*      地区コード選択
      PERFORM 顧客名検索処理部
      WHEN OTHER
      CONTINUE
      END-EVALUATE
      END-PERFORM.
顧客名検索部-END.
EXIT.

*
*****
* 顧客名検索画面での P F 3 : 顧客名検索処理部 (顧客データ) *
*****
顧客名検索処理部 SECTION.
*
* 地区コードをインターフェーステーブルにセット
      MOVE OLTAGC-TKCODE-I TO C-CHIKU-CODE.
*
* 顧客名検索サービス要求
      MOVE 'CUST' TO RP-G-NAME.
      MOVE 'cust' TO RP-S-NAME.
      CALL 'CBLDCRPC' USING ARG-RPC-PRM ARG-CUST-IN ARG-CUST-OUT.
      IF RP-STATUS-CODE = '00000'
      THEN
*      検索結果を画面にセット
      MOVE 1 TO WORK
      PERFORM UNTIL WORK > C-RETURN
          MOVE C-CUST-CODE(WORK) TO WK111-CUST-CODE
          MOVE C-CUST-NAME(WORK) TO WK111-CUST-NAME
          MOVE WK111-INF TO OLTAGC-KCODE-O(WORK)
          MOVE C-CUST-NAME(WORK) TO OLTAGC-KNAME-O(WORK)
          COMPUTE WORK = WORK + 1
      END-PERFORM
      MOVE OLTAGC-KDATA-T TO OLTAGC-OUTCURS-LOCO

      :

      END-IF.
顧客名検索処理部-END.
EXIT.

```

8. 標準パターンを使用した AP の作成

図 8-64 OpenTP1 を使用した受注業務 ( 9 / 10 )

```

*
*****
* 受注明細入力・修正画面での P F 3 : 商品名検索部 *
*****
商品名検索部 SECTION.
*
* 画面データの初期化
  PERFORM 商品分類一覧取り出し.
*
* 画面処理ループ
  MOVE SPACE TO OLTGCG-END-FLG.
  PERFORM UNTIL OLTGCG-END-FLG = 'E'
* 商品名検索画面表示
  MOVE 'OLTGCGND' TO 画面マップ名
  TRANSCEIVE DSP FROM OLTGCGO INTO OLTGCGI
  :
* 選択業務判定
  EVALUATE OLTGCG-INCI
  WHEN 'PFO1'
* 了解
  PERFORM 商品データ反映処理部
  MOVE 'E' TO OLTGCG-END-FLG
  WHEN 'PF12'
* 取消
  MOVE 'E' TO OLTGCG-END-FLG
  WHEN 'ENTR'
* 商品分類選択
  PERFORM 商品名検索処理部
  WHEN OTHER
  CONTINUE
  END-EVALUATE
  END-PERFORM.
商品名検索部-END.
EXIT.
*
*****
* 商品名検索画面での P F 3 : 商品名検索処理部 *
*****
商品名検索処理部 SECTION.
*
* 商品分類コードをインターフェースにセット
  MOVE OLTGCG-BNAME-I TO P-BUN-BCODE.
*
* 商品検索サービス要求
  MOVE 'PRDCT' TO RP-G-NAME.
  MOVE 'prdct' TO RP-S-NAME.
  CALL 'CBLDCRPC' USING ARG-RPC-PRM ARG-PR-IN ARG-PR-OUT.
  IF RP-STATUS-CODE = '00000'
  THEN

```

図 8-65 OpenTP1 を使用した受注業務 ( 10 / 10 )

```

*      検索結果を画面にセット
MOVE 1 TO WORK
PERFORM UNTIL WORK > P-RETURN
    MOVE P-SEIHIN-CODE(WORK) TO WK-S-CODE
    MOVE P-SEIHIN-NAME(WORK) TO WK-S-NAME
    MOVE P-TANKA(WORK)       TO WK-TANKA
    MOVE WK-INF              TO OLTGCG-ICODE-O(WORK)
    MOVE P-SEIHIN-NAME(WORK) TO OLTGCG-ILABL-O(WORK)
    COMPUTE WORK = WORK + 1
END-PERFORM
MOVE OLTGCG-INAME-T(1) TO OLTGCG-OUTCURS-LOCO
*      一覧取得メッセージ出力
MOVE 'OLTDGCND' TO 画面マップ名
TRANSCIVE DSP FROM OLTDGCO INTO OLTDGCI
ELSE
*      異常終了メッセージ出力
MOVE 'OLTFGCND' TO 画面マップ名
TRANSCIVE DSP FROM OLTFGCO INTO OLTFGCI
END-IF.
商品名検索処理部-END.
EXIT.
*
*****
* 商品名検索画面での P F 1 : 商品データ反映処理部 *
*****
商品データ反映処理部 SECTION.
*
:

商品データ反映処理部-END.
EXIT.
*
*****
* 受注データ修正画面での P F 8 : 商品分類一覧取り出し *
*****
商品分類一覧取り出し SECTION.
*
:

商品分類一覧取り出し-END.
EXIT.
*
*****
* 受注明細入力・修正画面での P F 8 : 計算処理部 *
*****
計算処理部 SECTION.
*
:

顧客データ計算部-END.
EXIT.
*

```

### 8.3.9 COBOL によるトランザクション型 AP 例 (OpenTP1 を使用した受発注業務)

OLTP サーバ構成での受発注業務の例題です。

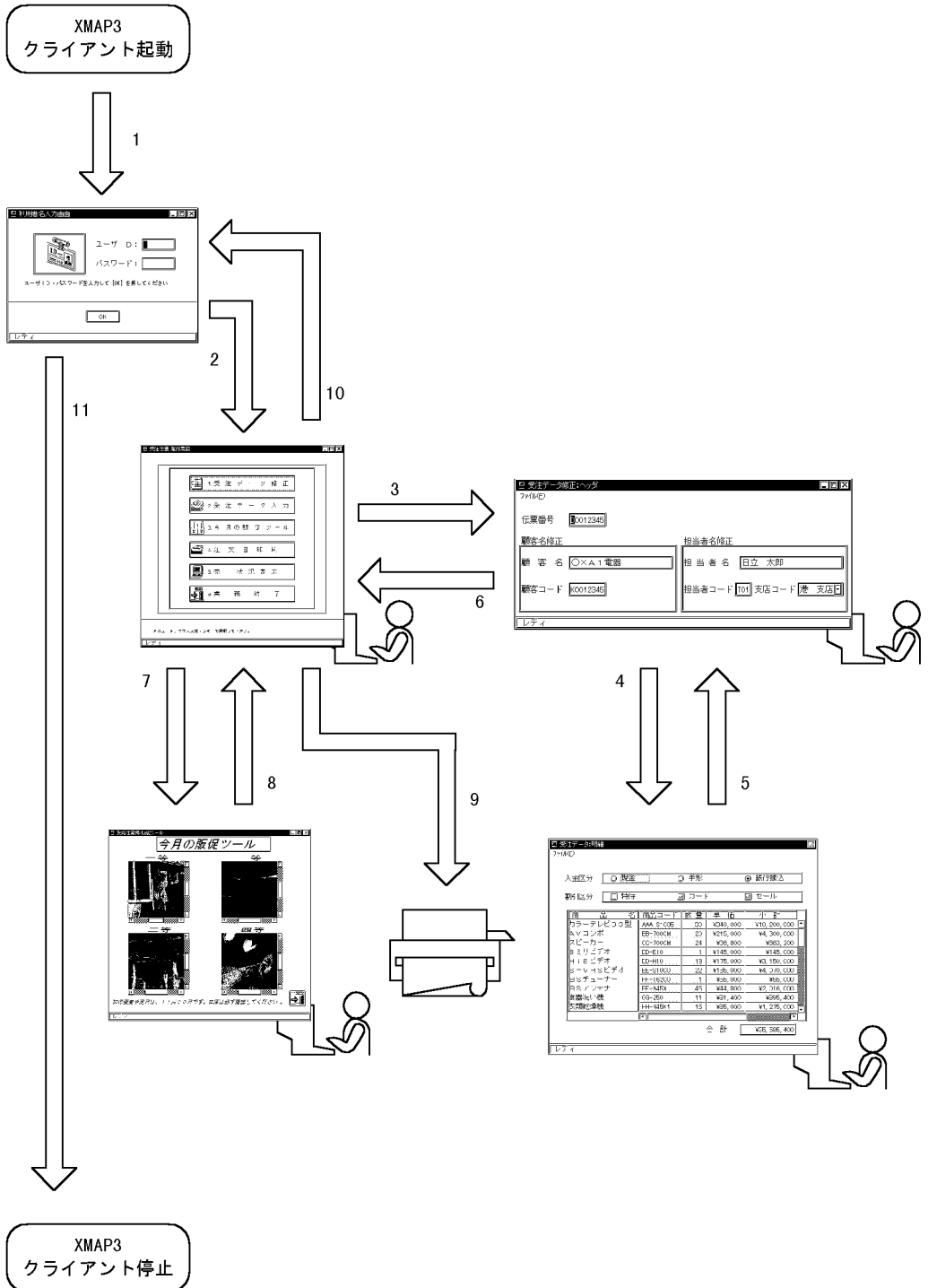
例題では XMAP3 クライアントを停止することで業務を終了 (初期画面を閉じる) しますが, AP からコネクションを解放するコマンド (mcftdcten) を発行することで業務を終了することもできます。この場合, 初期画面に [終了] ボタンを追加し, そのイベント処理でコマンドを発行します。業務を再開するときは, XMAP3 クライアントを終了させ, 再起動してください。

コネクションを解放するコマンドについては, マニュアル「OpenTP1 プロトコル TP1/NET/XMAP3 編」を参照してください。

#### (1) 想定する受発注業務の概要

この例題で想定する受発注業務の概要を次の図に示します。

図 8-66 受発注業務の概要



1. 「XMAP3 クライアント」を起動すると、初期画面（「利用者名入力画面」）が表示され

## 8. 標準パターンを使用した AP の作成

- る。
2. 初期画面で [OK] ボタンを選ぶと、メニュー画面（「受注伝票発行業務」画面）が表示される。
  3. メニュー画面で [受注データ修正] ボタンを選ぶと、修正ヘッダ画面（「受注データ修正：ヘッダ」画面）が表示される。
  4. 修正ヘッダ画面の [ファイル]・[明細] コマンドを選ぶと、修正明細画面（「受注データ：明細」画面）が表示される。
  5. 修正明細画面で [ファイル]・[終了] コマンド、または [ファイル]・[取消] コマンドを選ぶと、修正ヘッダ画面に戻る。
  6. 修正ヘッダ画面で [ファイル]・[終了] コマンドを選ぶと、メニュー画面に戻る。
  7. メニュー画面で [今月の販促ツール] ボタンを選ぶと、販促画面（「受発注業務：販促ツール」画面）が表示される。
  8. 販促画面で [EXIT] ボタンを選ぶと、メニュー画面に戻る。
  9. メニュー画面で [注文書印刷] ボタンを選ぶと、帳票が印刷される。
  10. メニュー画面で [業務終了] ボタンを選ぶと、初期画面に戻る。
  11. 「XMAP3 クライアント」を停止すると、初期画面が閉じられる。

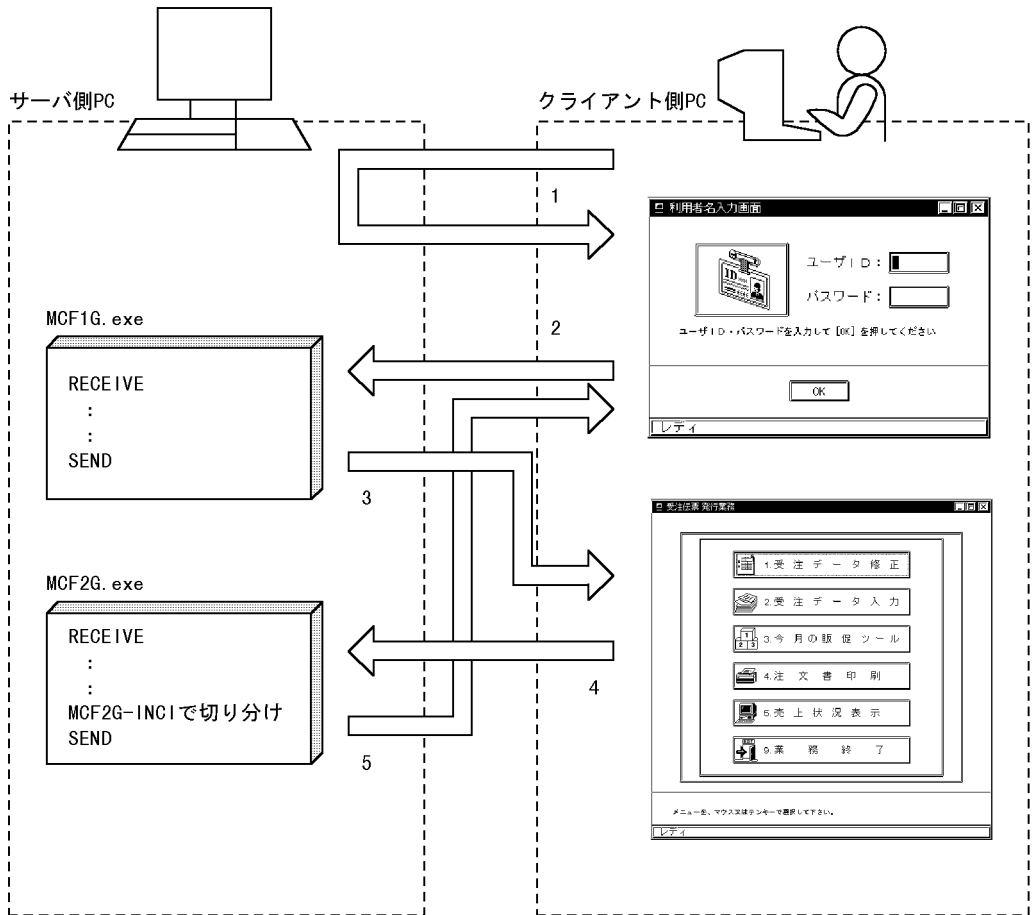
### （2）受発注業務の流れとサーバ AP の関連

受発注業務で使用する画面と、サーバ側 PC で実行する AP との関連を、業務の流れに基づいて説明します。

#### （a）XMAP3 クライアントの起動からメニュー画面の表示まで

XMAP3 クライアントを起動してからメニュー画面の表示処理、およびメニュー画面から初期画面への遷移処理について、トランザクションの流れを次の図に示します。

図 8-67 XMAP3 クライアントの起動～メニュー画面



1. クライアント側で「XMAP3 クライアント」を起動すると、連携するサーバ側の AP (またはコマンド) によって、OpenTP1 のコネクション要求が発行される。コネクションが確立されたあと、初期画面が表示される。
2. 初期画面でイベントを発生させると、初期画面のトランザクションコード (MCF1G) に対応した AP 「MCF1G.exe」が起動され、RECEIVE 命令が実行される。
3. 2 で発生したイベントが「[ OK ] ボタンの選択」の場合、「MCF1G.exe」からメニュー画面に対する SEND 要求が発行され、「MCF1G.exe」が終了する。
4. メニュー画面でイベントを発生させると、メニュー画面のトランザクションコード (MCF2G) に対応した AP 「MCF2G.exe」が起動され、RECEIVE 命令が実行される。
5. 4 で発生したイベントが「[ 業務終了 ] ボタンの選択」の場合、「MCF2G.exe」から初期画面に対する SEND 要求が発行される。その後、「MCF2G.exe」が終了する。

## 注

4 で発生したイベントが「未サポート業務 ([ 受注データ修正 ] ボタン、および [ 売

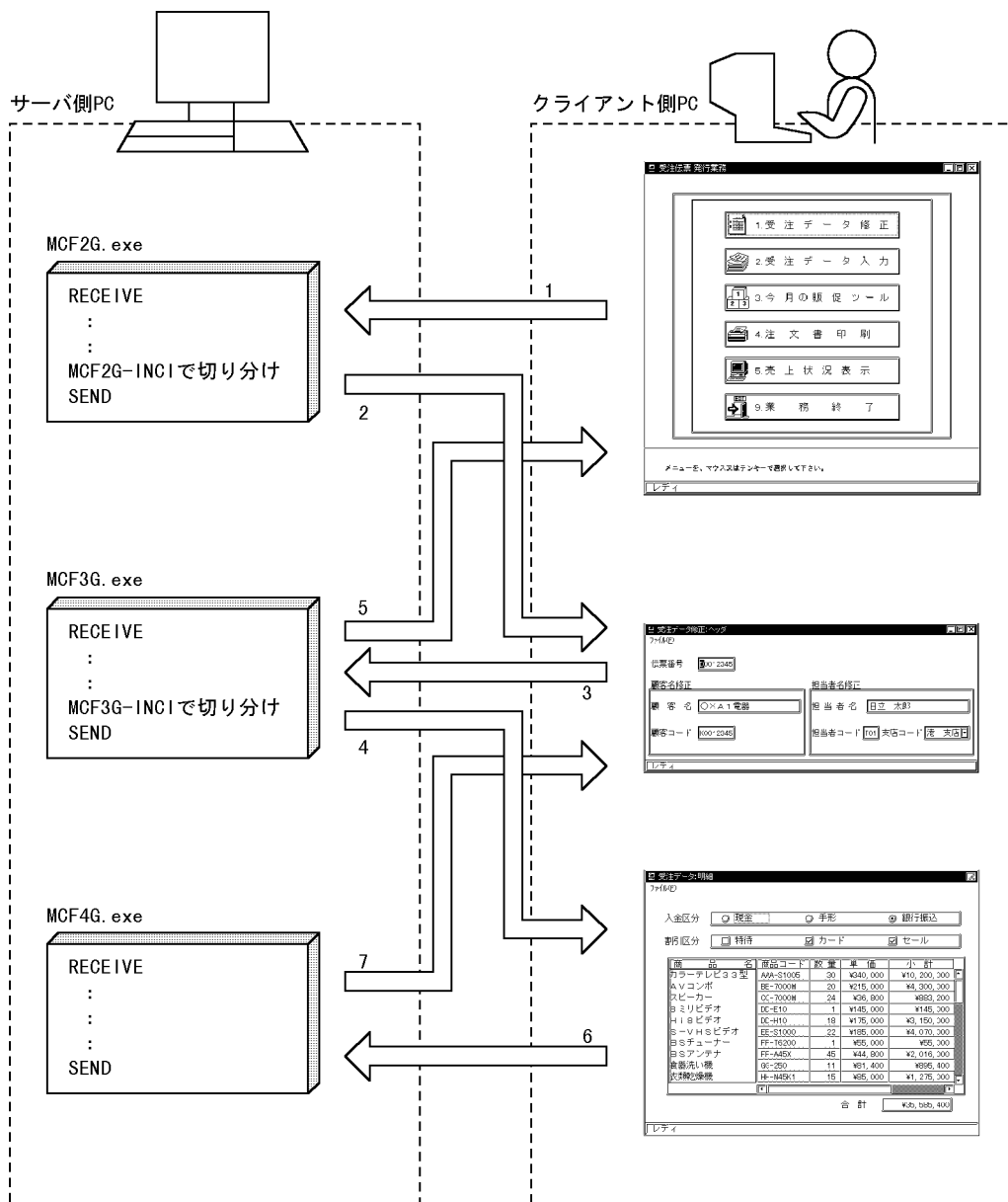
## 8. 標準パターンを使用した AP の作成

「上状況表示」ボタン)の選択」の場合、「MCF2G.exe」からメニュー画面の表示に対する SEND 要求が発行されます。その後、「MCF2G.exe」が終了します。

(b) メニュー画面から修正ヘッダ画面、修正明細画面の表示まで

メニュー画面から修正ヘッダ画面、および修正明細画面との遷移について、トランザクションの流れを次の図に示します。

図 8-68 メニュー画面～修正ヘッダ画面～修正明細画面





1. メニュー画面でイベントを発生させると、メニュー画面のトランザクションコード (MCF2G) に対応した AP「MCF2G.exe」が起動され、RECEIVE 命令が実行される。
2. 1 で発生したイベントが「[ 受注データ修正 ] ボタンの選択」の場合、「MCF2G.exe」から修正ヘッダ画面の表示に対する SEND 要求が発行される。その後、「MCF2G.exe」が終了する。
3. 修正ヘッダ画面でイベントを発生させると、修正ヘッダ画面のトランザクションコード (MCF3G) に対応した AP「MCF3G.exe」が起動され、RECEIVE 命令が実行される。
4. 3 で発生したイベントが「[ ファイル ] - [ 明細 ] コマンドの選択」の場合、「MCF3G.exe」から修正明細画面 (二次ウィンドウ) の表示に対する SEND 要求が発行される。その後、「MCF3G.exe」が終了する。
5. 3 で発生したイベントが「[ ファイル ] - [ 終了 ] コマンドの選択」の場合、「MCF3G.exe」からメニュー画面の表示に対する SEND 要求が発行される。その後、「MCF3G.exe」が終了する。
6. 修正明細画面でイベントを発生させると、修正明細画面のトランザクションコード (MCF4G) に対応した AP「MCF4G.exe」が起動され、RECEIVE 命令が実行される。
7. 6 で発生したイベントが「[ ファイル ] - [ 終了 ] コマンドの選択」、または「[ ファイル ] - [ 取消 ] の選択」の場合、「MCF4G.exe」から修正ヘッダ画面の表示に対する SEND 要求が発行される。その後、「MCF4G.exe」が終了する。

#### 注

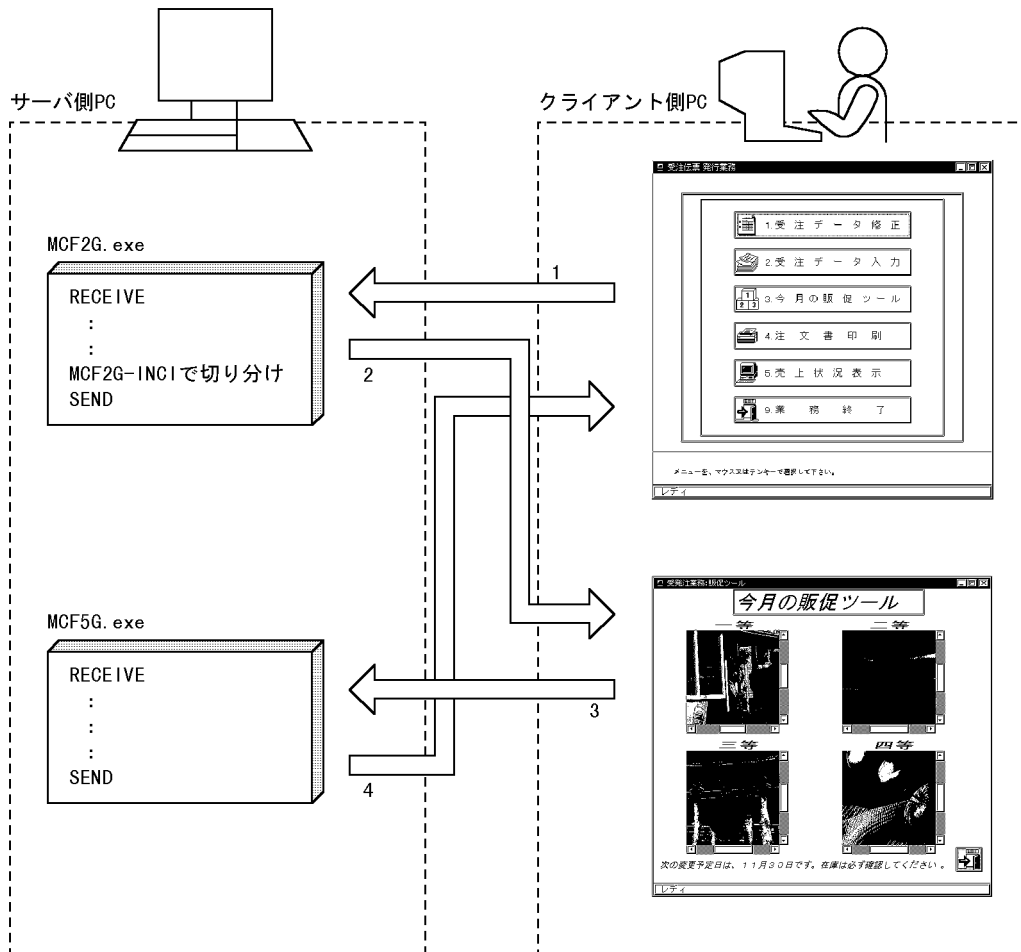
1 で発生したイベントが「未サポート業務 ([ 受注データ修正 ] ボタン、および [ 売上状況表示 ] ボタン) の選択」の場合、「MCF2G.exe」からメニュー画面の表示に対する SEND 要求が発行されます。その後、「MCF2G.exe」が終了します。

#### (c) メニュー画面から販促画面の表示まで

メニュー画面から販促画面との遷移までについて、トランザクションの流れを次の図に示します。

## 8. 標準パターンを使用した AP の作成

図 8-69 メニュー画面～販促画面



1. メニュー画面でイベントを発生させると、メニュー画面のトランザクションコード (MCF2G) に対応した AP「MCF2G.exe」が起動され、RECEIVE 命令が実行される。
2. 1 で発生したイベントが「[ 今月の販促ツール ] ボタンの選択」の場合、「MCF2G.exe」から販促画面の表示に対する SEND 要求が発行される。その後、「MCF2G.exe」が終了する。
3. 販促画面でイベントを発生させると、販促画面のトランザクションコード (MCF5G) に対応した AP「MCF5G.exe」が起動され、RECEIVE 命令が実行される。
4. 3 で発生したイベントが「[ EXIT ] ボタンの選択」の場合、「MCF5G.exe」からメニュー画面の表示に対する SEND 要求が発行される。その後、「MCF5G.exe」が終了する。

### 注

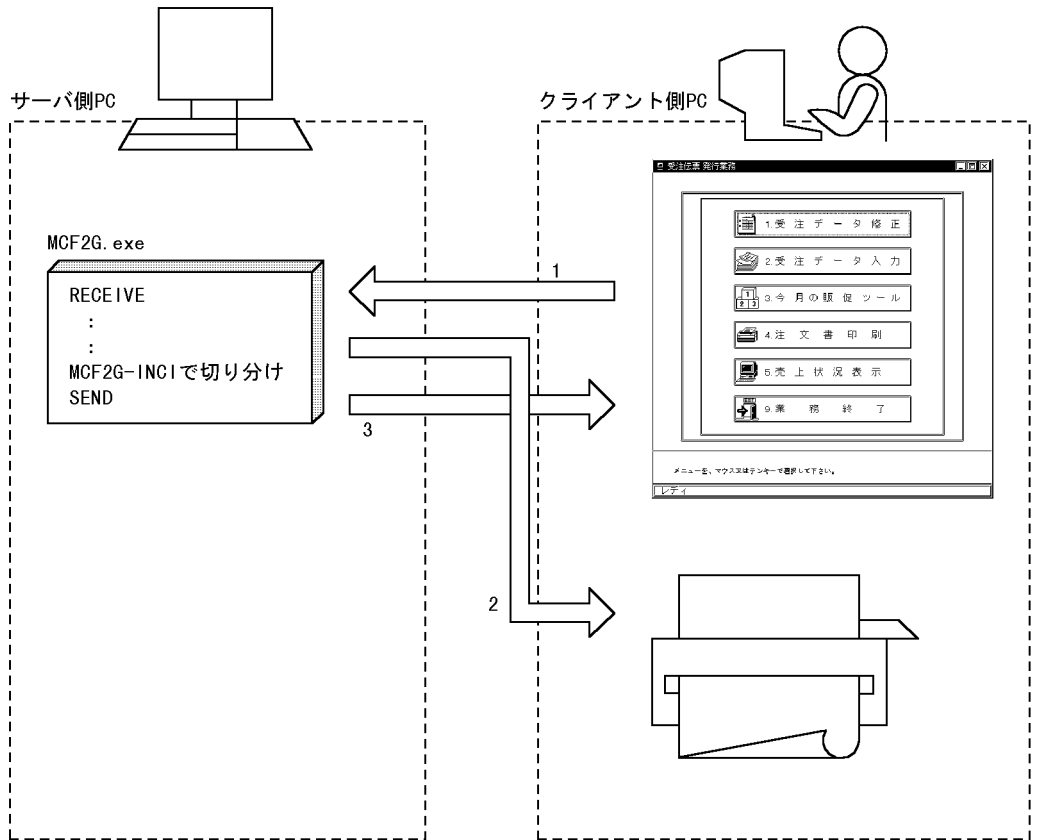
1 で発生したイベントが「未サポート業務 ([ 受注データ修正 ] ボタン, および [ 売

上状況表示] ボタン) の選択」の場合、「MCF2G.exe」からメニュー画面の表示に対する SEND 要求が発行されます。その後、「MCF2G.exe」が終了します。

(d) メニュー画面から帳票の出力まで

メニュー画面から帳票の出力までについて、トランザクションの流れを次の図に示します。

図 8-70 メニュー画面～帳票出力



1. メニュー画面でイベントを発生させると、メニュー画面のトランザクションコード (MCF2G) に対応した AP「MCF2G.exe」が起動され、RECEIVE 命令が実行される。
2. 1 で発生したイベントが「[ 注文書印刷 ] ボタンの選択」の場合、「MCF2G.exe」から帳票の出力命令が発行される。
3. 帳票の印刷終了後、「MCF2G.exe」からメニュー画面の表示に対する SEND 要求が発行され、「MCF2G.exe」が終了する。

注

1 で発生したイベントが「未サポート業務 ([ 受注データ修正 ] ボタン、および [ 売上状況表示 ] ボタン) の選択」の場合、「MCF2G.exe」からメニュー画面の表示に

## 8. 標準パターンを使用した AP の作成

対する SEND 要求が発行されます。その後、「MCF2G.exe」が終了します。

### (3) 受発注業務の動作環境

OLTP サーバ構成用の AP を実行する場合、あらかじめ TP1/NET/XMAP3 側や XMAP3 側での動作環境を設定しておく必要があります。ここでは、本例題を実行するために必要な、動作環境の設定例を説明します。

#### (a) TP1/NET/XMAP3 での環境設定

論理端末名、および TP1 コネクション名と XMAP3 の表示・印刷サービス名との対応 XMAP3 側の表示・印刷サービス名と対応するよう、TP1/NET/XMAP3 側で論理端末名や TP1 コネクション名を設定してください。

本例題ではそれぞれ次のように対応させます。

種別	TP1/NET/XMAP3		XMAP3
	TP1 コネクション名	論理端末名	表示・印刷サービス名
画面	XPNCN1	XPNLE101	xppDSP01
帳票	XPNCN2	XPNLE201	xppPRT01

#### データ有無コードの設定

本例題のように、データ有無コードを使用している場合は TP1/NET/XMAP3 側のデータ有無コードも同じ値を設定する必要があります。

本例題では TP1/NET/XMAP3 側のデータ有無コードの設定で、(1F)<sub>16</sub> を指定してください。

#### 初期画面の指定

論理端末名 (XPNLE101) の初期画面として (MCF1G) を指定します。

#### AP とトランザクションコードの関連の定義

AP とそれに対応するトランザクションコードを次に示します。

AP 名	トランザクションコード
MCF1G.exe	MCF1G
MCF2G.exe	MCF2G
MCF3G.exe	MCF3G
MCF4G.exe	MCF4G
MCF5G.exe	MCF5G

#### (b) XMAP3 側での環境設定

##### C/S セットアップ

C/S セットアップでは、xppDSP01 に呼応して起動するサーバ側の実行ファイルを指定します。

このサーバ側の実行ファイルとは、TP1 のコネクションを確立するための命令を発行するプログラムのことです。

#### 表示・印刷セットアップ

- オペレータインジケータの設定値の確認  
表示・印刷セットアップの「デザイン 1」タブで、「オペレータインジケータ」の設定値を確認します。  
設定できる値は「上に表示」、または「下に表示」になります。「表示しない」、および「拡張形式」は指定できません。
- 「バージョン間の互換オプション」の設定の確認  
表示・印刷セットアップの「アプリケーション 1」タブで、「バージョン間の互換オプション」の設定を確認します。  
「C/S の場合、アプリケーションサーバのセットアップだけが有効」チェックボックスがオンになっていた場合、オフにしてください。

#### (c) 実行環境の設定

##### 物理マップの作成と格納

本例題の実行前に、物理マップを作成しておきます。作成した物理マップは OpenTP1 のマップディレクトリへ格納してください。

##### グラフィックファイルの複写

本例題の実行前に、プログラムで使用するグラフィックファイルを次に示すフォルダへコピーしておきます。

- コピー元フォルダとコピーするグラフィックファイル  
「XMAP3 インストール先 ¥SAMPLES¥GRAPHICS」下の「1 等 .bmp」, 「2 等 .bmp」, 「3 等 .bmp」, 「4 等 .bmp」
- コピー先フォルダ  
「XMAP3 インストール先 ¥USER¥GRAPHICS」



# 9

## チューニングとトラブルの 対処方法

この章では、XMAP3 実行時の性能を向上させるためのポイント、およびトラブル発生時の対処方法について説明します。

---

9.1 性能向上のポイント

---

9.2 トラブルの対処方法

---

## 9.1 性能向上のポイント

---

ここでは、AP 実行の性能向上のためのテクニックについて説明します。

### (1) 定義時の性能を向上させるためのポイント

作成する画面は、必要最小限のオブジェクトにし、マップを小さくすることで実行性能が向上します。マップを小さくするには、次のことに注意してください。

#### (a) けい線の見直し

- むだなけい線は削除する。
- 矩形で囲んだフィールドは反転表示にして枠を取る。

#### (b) テキストボックスやフィールドの見直し

テキストなどの集約

テキストボックスやフィールドは、まとめて定義してください。例えば、「年」「月」「日」は三つに分けるのではなく、一つにまとめることでマップが小さくなります。

フィールドボックスの利用 (1)

テキストボックスで作成されている表は、フィールドボックスのフィールドとして定義してください。表示性能が向上します。

フィールドボックスの利用 (2)

GUI 画面でテキストボックスが多く繰り返されている場合には、フィールドボックス中にフレームを配置して、その中のフィールドとして定義してください。フィールドボックスのフレーム機能を使用すると、1 フィールドの定義で複数フィールドに相当する定義ができます。また、論理マップ上では OCCURS 展開されるため、AP のロジックも繰り返し指定で実現でき、効率が良くなります。

#### (c) グラフィックデータの見直し

不要なグラフィックデータの削除

グラフィックデータは、データ量が多くマップが大きくなってしまいます。グラフィックデータは、必要な個所にだけ使用するようしてください。

グラフィックデータの縮小化

固定グラフィックの場合はファイルサイズの総和が 31,000 バイト以内に収まるように作成してください。出力グラフィックの場合は 100KB 以内を目安にして作成することをお勧めします。容量の大きいグラフィックデータを小さくするには、次の方法で対処できます。

- ビットマップの縦横ドット数を小さくする

ビットマップは、中身の図柄がどのようなものであっても、同じ縦横サイズならばファイルサイズとして、常に同じバイト数を必要とします。したがって、縦横ドット数を小さくすることで、ファイルサイズが小さくなります。実際に絵を描いていない部分もビットマップの情報として格納されますので、必要最小限の領域を取るようにお勧めします。



次に既存のビットマップの必要な部分だけをペイント（またはペイントブラシ）を用いて切り出す方法を説明します。

1. ペイント（またはペイントブラシ）で基になるビットマップを表示する。
2. 四角形選択ツールで（または自動選択ツール）で必要な部分を選択する。
3. メニューバーから [ 編集 ] の [ ファイルへコピー ] を選択する。
4. ファイル名を指定して保存する。

- 使用する色を減らす

ビットマップの形式として、2色、16色、256色、1,600万色（フルカラー）の色数のものがあります。色数が増えると情報量が増えるので、それだけビットマップサイズが大きくなります。また、その表示色数をサポートしていないディスプレイに表示した場合、色化けが起きたり、表示が遅くなるなどの問題が起こります。既存のビットマップの色数を変更するには、画像フォーマットを変換する機能を持った市販のグラフィックソフトを使用する方法があります。しかし、色数を減らすことでビットマップが見にくくなる場合があります。したがって、初めから16色でビットマップを作成することをお勧めします。

#### (d) プレーンサイズの見直し

ボックスに隠れているプレーンは、データがなくても表示しています。不要なサイズになっていないか見直してください。

#### (e) 画面サイズの見直し

画面サイズが小さい方が表示性能は高くなります。画面サイズを見直してください。

## (2) AP 作成時のポイント

### (a) 同一マップを再表示するときの方法

同一マップの画面を再表示する場合は、一部上書を指定しないと、画面中の一つ（または一部）のオブジェクトだけを書き換えるときでも全画面の再表示となり、表示性能が低下します。

一部上書と全面書換とを使い分けることをお勧めします。特に一部上書は、フィールド単位にデータを入力し、APでチェックおよび表示していくような場合に効果的です。指定方法については「7.2.3(1)(f) MAPPING MODE 句の扱い（マッピングオプション）」を参照してください。

#### 前提となる定義

1. ドローで、画面属性の「入力・選択状態の扱い」に「未入力・未選択」または「状態を維持」を指定します。また、該当するテキスト・フィールドの属性の「初期値」に「なし」を指定します。
2. 1.の指定によって、データ未入力を埋字以外で判定する必要が生じた場合には、ドローセットアップの論理マップ属性ダイアログで、「初期クリア文字」に「埋字」以外を指定します。さらに、ユティリティの「セットアップ情報の反映」で、マップを再生成します。

## 9. チューニングとトラブルの対処方法

### <1 回目の表示時>

- プログラム中のマッピングオプションに、「マージ」を指定します。
- TRANSCEIVE 文, SEND 文などで画面を表示します。

### <同じ画面の再表示時>

1. プログラム中のマッピングオプションに、「論理マップだけ」を指定します。
2. 画面表示時に変更しない項目にデータ有無コードを格納します。
  - 動的変更属性の指定があれば、プログラム中の制御項目（修飾名を格納する論理マップのエリア）に格納します。
  - 項目データは、先頭 1 バイト目、またはすべての項目データに対して格納します。
  - 出力論理マップがデータ有無コードでクリアされるため、表示中のデータを記憶しておく場合には別のエリアに画面情報をバックアップしておくような対処が必要です。
3. TRANSCEIVE 文, SEND 文などで 1 回目と同じマップ名を再表示します。

このコーディングのサンプルとして、AP パターン CRLINP01.CBL を利用できます。

### (b) データ量の見直し

AP でむだなデータを送っていませんか。例えば、常にスペースやゼロを送ってしまうと実行性能が劣ってしまいます。したがって、出力時の性能を上げるにはデータの量を減らすことが重要です。

- データ有無コードで論理マップをクリアします。データ有無コードについては「6.2.7 特定の入出力テキストだけを書き換える（データ有無コードの利用）」を参照してください。
- むだな初期値は指定しないようにしてください。

### (c) AP 分割の見直し

XMAP3 のオープン要求がどのタイミングで出されているかを見直してください。コンパイル単位で、TRANSCEIVE 文または、SEND 文が発行されると、画面ごとにオープンが実行されるため、処理が遅くなります。不要に画面をオープンしないためには次の点に注意してください。

- 一つのコンパイル単位に TRANSCEIVE 文, SEND 文をまとめ、オープンの実行は必要最低限にします。
- 共通のルーチンにまとめて TRANSCEIVE 文, SEND 文を発行するように AP の構造を見直します。
- 日立 COBOL の場合、AP 間でオープンを引き継ぐ「CBLTERMSHAR=YES」を指定すると、AP 間でオープンの引き継ぎが行われます。

詳細については「7.1.2 AP間でオープンを引き継ぐ場合」を参照してください。

日立 COBOL では、明示的に CALL 文でオープンを要求する場合を除き、一つのコンパイル単位で最初の SEND 文または TRANSCEIVE 文が発行されると画面がオープンされます。そのため、1画面が1実行ファイル(.EXE)のような構成にすると、実行ファイルごとに画面をオープンすることになります。この場合、オープン・クローズ、SEND 文、TRANSCEIVE 文を発行する実行ファイル、ビジネス処理をする実行ファイルをそれぞれ分けた方が実行性能が高くなります。

また、複数のコンパイル単位のを合わせて一つの実行ファイルにするときは、各コンパイル単位でオープンを発行しないようにするため、日立 COBOL の実行支援の環境変数で、「CBLTERMSHAR=YES」を指定します。「CBLTERMSHAR=YES」は、SEND / RECEIVE / TRANSCEIVE インタフェースのときだけ有効です。CALL インタフェースでは無効になります。

CALL インタフェースの場合、または「CBLTERMSHAR」をサポートしていないバージョンの日立 COBOL を使用している場合で、AP間のオープン引き継ぎをするときは、ユーザが、XMAP3 インタフェースエリアの情報を引き継ぐようなコーディングをする必要があります。サンプルコーディングとして、AP パターンの GENDSP02 および GENDSP03 が利用できます。

#### (d) 画面クローズのタイミングの見直し

画面のクローズは、明示的に CALL 文でクローズするか、DISABLE 文を発行したタイミングです。ただし、クローズすると次に画面を表示するときには必ずオープンしなければなりません。画面を素早く消したり、再表示したりする場合は、画面属性ダイアログの「Z位置」で「一時非表示」を指定して、見た目上は表示されていないようにしておき、次に画面を表示するとき動的変更で「Z位置」の指定を「標準表示」または「手前に表示」に変更します。このようにすると、画面をクローズ、オープンするよりも早く画面を切り替えられます。

#### (e) AP との制御のやり取りの見直し

画面属性の定義で入力単位を「画面」にしているにもかかわらず、オブジェクトの属性で「自動送信」を指定して、オブジェクト単位で AP に制御を返すような AP 構造は、画面単位に AP に制御を返す AP 構造と比べ オーバーヘッドが大きくなり、オブジェクト遷移（フォーカス移動）が遅くなる場合があります。このため、AP との制御のやり取りは画面単位で行うことをお勧めします。オブジェクト単位で AP と制御のやり取りをする場合は、入力単位を「フィールド」や「イベント」に設定し、できるだけ画面中のオブジェクトの数を少なくすることがポイントです。

### (3) 実行時のポイント

#### (a) マップの常駐化サイズの見直し

使用する物理マップを常駐するバッファサイズは変更できます。詳細については、マ

## 9. チューニングとトラブルの対処方法

マニュアル「XMAP3 開発・実行ガイド」を参照してください。

### (b) C/S システム構成時の通信データ削減

C/S システム構成の場合、XMAP3 の通信データを圧縮する機能が利用できます。通信データの圧縮機能を利用すると、C/S 間の通信量が削減できるため、画面応答のレスポンス向上が期待できます。詳細については、マニュアル「XMAP3 開発・実行ガイド」を参照してください。

## 9.2 トラブルの対処方法

### 9.2.1 日立 COBOL 使用時のトラブル

#### ! 注意事項

エラーコードの「KCCC」は、COBOL85では「KCCB」となります。

#### (1) コンパイル時の主なエラー

表 9-1 コンパイル時の主なエラー

エラー	意味	対処方法
KCCC1076C-S	COPY 文を取り込もうとしているファイルが、AP のソースが格納されているフォルダ中になくことが考えられる。	ファイルが、AP のソースが格納されているフォルダ中にあるか見直し、ない場合はファイルをコピーしてコンパイルし直す。
	COPY 文を取り込もうとしているファイルの名称が誤っていることが考えられる。	XMAP3 で作成した論理マップファイルの名称であるか見直し、ファイルの名称が誤っているときは正しい名称に修正してコンパイルし直す。
KCCC3015C-S	データ名、または変数名に不当な文字を指定していることが考えられる。	データ名、および変数名を見直す。
	必要なテーブル「X3MODTBL」が AP を格納しているフォルダになくことが考えられる。	AP を格納しているフォルダに次のファイルをコピーして、コンパイルし直す。 XMAP3 インストール先 ¥INCLUDE¥X3MODTBL

## (2) リンク (MAKE) 時の主なエラー

表 9-2 リンク (MAKE) 時の主なエラー

エラー	意味	対処方法
LNK1181 LNK1104	TRANSCEIVE 文, SEND 文, および RECEIVE 文で XMAP3 を使用する場合に, 必 要な XMAP3 のライブ ラリを参照できない。	コンパイル環境の PC に XMAP3 の開発モデルがインストー ルされているか確認する (CALL 文のインタフェースで作成 した AP の場合はリンケージ環境)。
LNK2001 LNK1120	CALL 文で XMAP3 を 使用する場合に, リン ケージ時に必要な XMAP3 のライブラリを 参照できない。	日立 COBOL のリンケージオプションとして, 次の手順で設 定する。 《COBOL2002, COBOL85 Version 7 および Version 6 の場 合》 1. COBOL 開発マネージャの [プロジェクト] - [プロジェ クトの設定] コマンドを選ぶ。 2. 「リンク」タブの「ライブラリの指定」にチェックを入 れる。 表示されるダイアログの設定内容に, 次の内容を追加す る。 XMAP3 インストール先 ¥LIB¥X3MWDR32.LIB  《COBOL85 Version 5.0 の場合》 1. COBOL 開発マネージャの [オプション] - [リンカ] コ マンドを選ぶ。 2. リンカオプション設定ダイアログで, 次のように設定す る。  • リンカオプション: 何も指定しない • インポートライブラリ/ユーザ作成ライブラリ XMAP3 インストール先 ¥LIB¥V3Compatible¥X3MWDR32.LIB

## (3) 実行時の主なエラー

## (a) 日立 COBOL CD 項の STATUS KEY 句の実行状態コード

STATUS KEY 句の実行状態コードと意味, および対処方法を次の表に示します。

なお, 日立 COBOL のテストデバッガでコードのエリアを参照すると, 16 進データはバ  
イトが反転して表示されます。

表 9-3 STATUS KEY 句の実行状態コード

実行状態のコード	意味	対処方法
'00000'	正常終了	該当しない。
'00008'	論理エラー	対処: AP を見直す。

実行状態のコード	意味	対処方法
'10008'	マッピングエラー	日立 COBOL が出力するメッセージ KCCC5007R-S, または KCCC5008R-S の内容を調査する。AP が設定したデータに誤りがあることが考えられる。詳細は, KCCC5008R-S の詳細コードを調べ, ほかのリターンコードとともに確認すること。
'20008'	入出力エラー	日立 COBOL が出力するメッセージ KCCC5007R-S, または KCCC5008R-S の内容を調査する。入出力時の障害の詳細は, KCCC5008R-S の詳細コードを調べて, ほかのリターンコードとともに確認すること。リターンコード, 詳細コードについては関連項目を参照すること。

(b) 実行エラー時の主な XMAP3 のリターンコード・詳細コード

表 9-4 実行時の主なエラー (日立 COBOL のコンソール画面表示)

エラー	意味	対処方法
KCCC5008R-S	エラーコードで示すエラーが XMAP3 で発生した	画面入出力時のエラーコードを見て対処する。エラーコードの詳細は, 「付録 E XMAP3 のリターンコード・詳細コード」を参照のこと。
KCCC3417R-S	サービス名称不足	

## 9.2.2 旧バージョン提供ファイルのコンパイル時のトラブル

XMAP3 (02-04) での提供ヘッダファイル (jsvwatbl.h) をそのまま使用して, クラスなどの概念を持った C++ 言語で作成した AP に取り込むと, コンパイル時にエラーになります。この場合, ヘッダファイルをエディタで開き, 関数の記述を次のように書き換える必要があります。

変更前

```
#ifndef WIN32
extern long FAR PASCAL jsvwdrv ();
#else
extern long APIENTRY jsvwdrv ();
#endif
```

変更後

```
#ifndef WIN32
extern long FAR PASCAL jsvwdrv (XMAP_COM*, XMAP_REQ*, unsigned char*, unsigned char*);
#else
extern long APIENTRY jsvwdrv (XMAP_COM*, XMAP_REQ*, unsigned char*, unsigned char*);
#endif
```

ユーザは下線部を変更します。

## 9. チューニングとトラブルの対処方法

ただし、ヘッダファイルを修正すると、型不一致エラーになる場合があります。このような場合に対処するための型変換を次に示します。

```
XMAP_COM.....com;
```

```
XMAP_REQ.....req;
```

```
XMAP_OPN.....opn;
```

オープン時、出力時、クローズ時はそれぞれ次のようになります。

<オープン時>

```
jsvwdrv (&com, &req, (unsigned char*)&opn,0L) ;
```

<出力時>

```
jsvwdrv (&com, &req, (unsigned char*)&JYUGUI0,0L) ;
```

<クローズ時>

```
jsvwdrv (&com, &req, 0L,0L) ;
```

### (1) その他

その他のエラーについては、ヘルプのトラブルの対処方法を参照してください。

## 9.2.3 表示に関する対処方法

画面が表示されない、または表示状態が悪い場合の代表的な現象と、その対処方法を次の表に示します。なお、表中以外の現象については、ヘルプのトラブルの対処方法を参照してください。

表 9-5 表示に関する対処方法

現象	対処方法
画面がまったく表示されない。	ほかのウィンドウの後ろに隠れていないか確認し、その場合は、ウィンドウを前面に移動する。
	ほかの AP ですでに画面を表示していないか確認し、その場合は、ほかの AP を終了して、再実行する。
	SEND 文、または TRANSCEIVE 文の実行後のリターンコードを見て、AP でエラーが起きていないか確認する。その場合は、エラーを修正する。



現象	対処方法
	<p>実際に表示できるサイズよりも大きなウィンドウを指定していないか定義内容を見直し、その場合はサイズを修正する。</p> <p>表示サーバで指定できる最大オブジェクト数を超えていないか確認し、その場合は、オブジェクトの数を減らす。</p>
<p>画面がまったく表示されない。 (マッピングライブラリのリターンコードが「5120(1400)<sub>16</sub>」の場合)</p>	<ul style="list-style-type: none"> <li>• 物理マップ名称を正しく指定する。物理マップの名称は、「マップ名 + デバイス ID」 (例: GUI 画面の場合はマップ名 + ND またはマップ名 + 0)</li> <li>• ドローセットアップで設定したマップ名の長さを確認する。</li> </ul> <p>物理マップを AP と同じフォルダまたは物理マップロードパスに指定したフォルダの下に配置する。 なお、別フォルダから物理マップを参照するように XMAP3 の実行環境をカスタマイズしている場合 (物理マップロードパスを指定している場合) には、そのフォルダに物理マップがあるかを確認する。</p> <p>仮想端末名称を正しく指定する。</p> <p>ホスト名称が合っているか、また余分なドメイン名を指定していないか (C/S システム型の場合) サービス名ファイル、DNS 設定、HOSTS ファイルで指定しているホスト名称を合わせる。DNS 設定は、コントロールパネルのネットワークアイコンから「TCP/IP」を選んで設定する。 また、余分なドメイン名は削除する。</p>
<p>画面が途中までしか表示されない。 画面が壊れる。</p>	<p>ドローで、オブジェクトの属性設定用ダイアログの「表示方法」の指定を確認し、オブジェクトに「非表示」の属性を設定している場合は、解除する。また、属性の動的変更をしている場合は、修飾名の設定を確認する。</p> <p>データ有無コードに (1F)<sub>16</sub> 以下のコードを指定する。</p> <p>マップ生成起動時のリターンコードとエラーメッセージを基に定義内容を見直し、マップ生成時のエラーによる、マップの誤りがあった場合は修正する。</p> <p>外字フォントを正しく作成しているか確認する。</p> <p>論理マップ長エリアに論理マップの長さを正しく格納しているか、論理マップと物理マップが対応しているか確認し、正しい論理マップの長さを格納する。</p> <p>出力論理マップ中に前の論理データが残っていないか AP を見直し、出力論理マップをクリアする。</p> <p>論理マップに格納したデータの文字コードが誤っていないか AP を見直し。</p> <p>物理マップと論理マップを作成した日付が一致しているか確認し、不一致の場合はマップ生成、および AP のコンパイルを直す。 物理マップと論理マップの不一致が起きないように、物理マップを一括生成したり、環境間で移動させた場合などには、ユティリティのマップコンペア機能を利用して、漏れないかチェックしておく。マップコンペア機能の詳細についてはマニュアル「XMAP3 開発・実行ガイド」を参照のこと。</p> <p>XMAP3 に正しい論理マップデータを渡しているか AP を見直し、物理マップと論理マップの対応を確認する。</p>

## 9. チューニングとトラブルの対処方法

現象	対処方法
	先頭 1 文字, または文字列の途中がデータ有無コードになっていないか確認し, その場合は, 出力したいデータの前のデータ有無コードを削除する。
	マップ展開方式と, 日立 COBOL のコンパイラオプションの指定が一致しているか確認し, 不一致の場合は, 指定を合わせてマップを生成し, AP のコンパイルをし直す。 マップ展開方式は, ドローセットアップやユティリティのマップ生成で指定する。
表示はされるがすぐに消える。	物理マップに対応した入力論理マップの長さを正しく設定する。
画面のサイズが小さくなる。	表示・印刷セットアップでフォントサイズを設定する。 フォント構成ファイルで, 3D 表示に使うフォントサイズを大きく設定し直す。 表示・印刷セットアップで, 「2D 表示」に設定し直す。

### 9.2.4 応答性能の対処方法

応答性能での問題が発生した場合の対処方法を, 現象別に次の表で説明します。該当する現象と推定原因から, 対処方法に従って対応してください。

なお, 次の表に示す対処方法は, ユーザの責任で行ってください。

表 9-6 応答性能での問題と対処方法

現象	推定原因	対処方法
同等レベルの CPU で動作が遅いものがある。	実メモリが少ない。	メモリを増設する。
	スワップファイルがあるドライブと XMAP などのあるドライブが同一ドライブになっている。	物理的に別なドライブがある場合 (1 台のハードディスクを二つのドライブに分けたものではなく, ハードディスク自体が複数台ある場合), 実行ファイルのあるドライブではないドライブにスワップファイルを設定すると, 動作が速くなることもある。
	ディスクアクセスが 16 ビットになっている (Windows 95, Windows 98 または Windows Me の場合)。	何らかの理由でディスクアクセスが 32 ビットアクセスではなく, 16 ビットアクセスになっている。「コントロールパネル」の「システム」を参照して原因を取り除く。 通常は Windows 95, Windows 98 または Windows Me に対応していない古いドライバを使用していることや, ハードウェアに適さないドライバを使用していることが原因となっている。

現象	推定原因	対処方法
	ディスク圧縮ソフトを使用している。	XMAP3 のファイルやマップなどを圧縮されていないドライブに置く。
	ほかの AP が同時に動作している。	不要な AP は起動しないようにする。
	1 個のオブジェクトの表示形態だけを「全面書換」にしたり、マッピングオプションに「マージ」を指定したりしている。	マッピングオプション、データ有無コードの設定を見直す。
	毎回画面をオープン/クローズしている。	オープンに関しては AP を複数の実行ファイル (.DLL または .EXE) に分割しているとき、AP 間でオープンを引き継ぐ指定をしているかを見直す。 日立 COBOL の場合、環境変数「CBLTERMSHAR=YES」を指定する。 クローズに関しては DISABLE 文の発行を見直す。
	ディスプレイドライバの設定による。	ディスプレイドライバの種別によって、表示色数、解像度の設定で表示速度が遅くなる場合がある。それぞれの値を変更して表示してみる。
画面表示性能が通常の場合と比べて遅くなる。	バックグラウンドで、ほかの AP が起動されている。 <ul style="list-style-type: none"> <li>LAN 構成で使用していた端末をスタンドアロン構成として使用している。</li> <li>システムエージェントなど、一定の間隔でデフラグやスキャンディスクを自動実行するプログラムが動作している。</li> </ul>	業務に影響しない AP はできるだけ起動しないようにする。
	サーバ側でスクリーンセーバが起動している。	C/S システム構成の場合、サーバ OS 上でスクリーンセーバが起動していると処理速度が遅くなるため、コントロールパネルでスクリーンセーバを「なし」にする。



# 付録

---

付録 A 属性情報一覧

---

付録 B XMAP3 で提供している定義パターン

---

付録 C 日立以外のコンパイラを使用するときの注意

---

付録 D 画面定義の制限値

---

付録 E XMAP3 のリターンコード・詳細コード

---

付録 F 用語解説

---

---

## 付録 A 属性情報一覧

### 付録 A.1 ウィンドウ制御情報一覧

定義画面ごとに設定する属性について説明します。この属性の設定は、画面属性ダイアログで設定します。

#### (1) 基準文字サイズ

レイアウト領域に適用する基準となる文字サイズです。文字サイズには、「大/中/小」があります。

#### (2) 表示形態

画面を表示するとき、直前に表示した画面に対して一部だけを書き換える（一部上書）か、全部を書き換える（全面書換）かを指定する属性です。この値は、ドローの画面属性ダイアログで値を設定します。また、ドローセットアップの「画面属性の動的変更」で変更したい「表示形態」の属性を指定した修飾名を用意しておくことで、AP から動的に表示形態を変更できます。なお、XMAP3 があらかじめ用意している修飾名も利用できます。表示形態には、次の 3 種類があります。表示形態の詳細については「1.5.3(2) 表示形態」を参照してください。

##### 一部上書

直前に表示した画面の、一部のオブジェクトだけを変更して表示します。全面書換に比べて表示時間を短縮できます。この場合、マッピングオプションの「マージ」か「論理マップだけ」とあわせて使用する必要があります。

##### 全面書換

表示中の画面を消去し、次の画面を全画面描画します。

##### 自動

一部上書と全面書換を XMAP3 に任せて、特に AP では意識しません。直前の画面と同じマップ名のときは「一部上書」を仮定します。

標準は「自動」です。

#### (3) Z 位置

Z 位置は「手前に表示」「標準表示」「後ろに表示」「一時非表示」から選べます。「一時非表示」は奥行き指定に関係なく、一時的にウィンドウを隠します。なお、二次ウィンドウの場合は、標準表示だけとなります。

##### 標準表示

表示するといちばん手前に表示されます。ほかの AP のウィンドウが表示されると、後ろへ移動されます。

##### 手前に表示

活性/不活性に関係なく、常に手前に表示されます。

**後ろに表示**

表示直後は活性状態で後ろに表示されます。一度不活性にしたあとで、ユーザが操作して活性にしたときは手前に表示されます。

**一時非表示**

ウィンドウを見えない状態にします。通常に表示するには、AP から動的変更をして「標準表示」などに切り替えます。一時非表示を利用すると、一時的にウィンドウをクローズしたような状態を作れます。

**(4) ウィンドウを表示した時にアラームを鳴らす**

ウィンドウを表示したとき、ディスプレイ端末のアラームを鳴らす場合に指定します。一般的にはエラー画面を表示するときを使用します。この属性は、ドローセットアップの「画面属性の動的変更」で表示形態の属性を設定した修飾名を利用することで、AP から動的に変更することもできます。

**(5) キーボードのロック状態を解除する**

画面を表示したら、キーボードから入力できる状態にするときに指定します。指定しないときは、キーボードのロック状態を直前の画面の属性を継続します。この属性は、ドローセットアップの「表示属性の動的変更」で表示形態の属性を設定した修飾名を利用することで、AP から動的に変更することもできます。

**(6) 入力・選択状態の扱い**

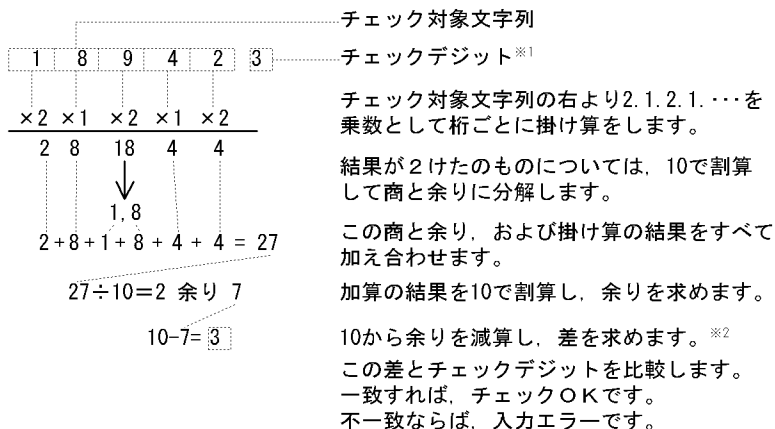
入力フィールド（入出力テキスト・フィールド）中に表示されているテキストデータの扱いに関する属性です。「表示形態」が「一部上書」のときの入力項目をどのような状態にするかを指定します。画面属性ダイアログで値を指定します。入力項目の扱いを AP 実行時に変更したい場合は、ドローセットアップの「表示属性の動的変更」の「キャラクタコントロール」タブで修飾名を指定しておきます。入力・選択状態の扱いについては、「2.3.1(1)(d) 入力・選択状態の扱い」または「3.3.1(1)(b) 入力フィールドの扱い」を参照してください。

**(7) モジュラスアルゴリズム**

入出力テキスト（または入出力フィールド）で指定するモジュラスチェックのアルゴリズム（チェック 10 またはチェック 11）を指定できます。モジュラスチェックを指定したフィールドからほかのフィールドへフォーカス（またはカーソル）が移動した時点でチェックされます。

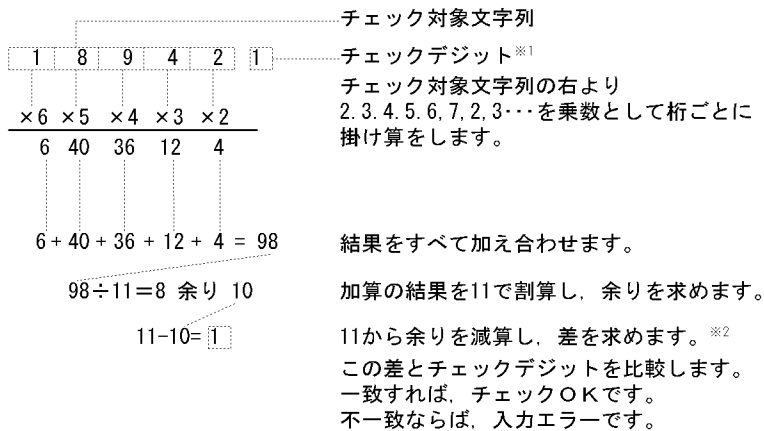
モジュラスチェックのアルゴリズムは、図 A-1、図 A-2 に示すとおり、フィールド最右端の数字がチェックデジットになります。フィールド内に数字以外の文字があると、チェックされますが結果は予測できません。したがって、この属性は、入力文字列が数字（データ型が数字）に指定しておく必要があります。また、項目長は 2 ~ 16 桁となります。これ以外の項目長の場合は、エラーとなります。

図 A-1 モジュラス 10 のアルゴリズム



注※1 チェック対象文字列がすべて0の場合は、チェックデジットを0とします。  
注※2 余りが0のとき（差が10のとき）は、チェックデジットを0とします。

図 A-2 モジュラス 11 のアルゴリズム



注※1 チェック対象文字列がすべて0の場合は、チェックデジットを0とします。  
注※2 余りが0のとき（差が11のとき）は、チェックデジットを0とします。  
余りが1のとき（差が10のとき）は、入力エラーです。

## (8) 入力単位

入出力は基本的に画面単位です。ただしオプションとして、フィールド単位およびイベント単位の入出力が選べます。入力単位には「画面」「フィールド」「イベント」「表示直後」があります。

### 画面

画面属性ダイアログの入力単位で「画面」を指定すると、AP から画面を表示すると



きに、入出力の制限（不活性）を指定しているオブジェクト以外は入力・選択可能な状態が表示されます。これらのオブジェクトを選択したり、データを入力したりしたあと、画面確定を設定しているメニューバー、プッシュボタンまたは画面確定キーを押すと、AP に制御が渡ります。なお、オブジェクトに「自動送信」を設定すると、そのオブジェクトからフォーカス・カーソルが離脱するタイミングで「送信」の画面確定が仮定されます。

#### フィールド

画面属性ダイアログの入力単位で「フィールド」を指定すると、AP から画面を表示するときに、AP で設定されているフォーカス（カーソル）があるオブジェクトだけが入力・選択可能になります。ただし、確定キー（送信キーや PF キー）、プッシュボタンおよびメニューバーは除かれます。

これらのオブジェクトを選択したり、データを入力したりしたあと、フォーカス（カーソル）が次のオブジェクトに移動するタイミングで AP に制御が渡ります。なお、各オブジェクトでの「自動送信」の指定は無効になります。

#### イベント

画面属性ダイアログの入力単位で「イベント」を指定すると、AP から画面を表示するときに、入出力の制限（不活性）を指定しているオブジェクト以外は入力・選択可能な状態が表示されます。なお、各オブジェクトでの「自動送信」の指定は無効になります。

これらのオブジェクトの一つを選択したり、データを入力したあと、フォーカス・カーソルが次のオブジェクトに移動するタイミングで AP に制御が渡ります。このとき、個々のオブジェクトを識別する通知コードが AP に渡ります。なお、イベントが発生したとき、AP に通知するかどうかは画面属性ダイアログで指定できます。イベントの定義に関しては「2.3.1(6) イベント通知コード」を参照してください。

#### 表示直後

画面属性ダイアログの入力単位で「表示直後」を指定すると、画面に対して何の操作がされなくても、一定の時間が過ぎると自動的に AP に制御を渡します。その場合のイベント通知コードは、送信キーを押したときと同じ値を返します。また、AP に制御が渡るまでの時間を設定せず、表示直後に AP に制御を渡すような設定もできます。なお、設定した時間が過ぎる前に、画面を確定する操作をした場合は、操作の方が優先されます。

画面には、入力必須属性を設定したオブジェクトを定義しないでください。入力必須属性を設定したオブジェクトを定義した画面を表示した場合、待機終了時にアプリケーションに対してエラーが通知されます。

スプラッシュ画面や、表示専用の掲示板的な画面を表示するときに利用できます。

スプラッシュ画面や掲示板的な画面については「1.4.1 GUI 画面で使用できる機能」を参照してください。

## 付録 A.2 キーエントリ・ラベル系オブジェクトの表示属性一覧（テキストとフィールド）

キーエントリ・ラベル系のオブジェクト（テキストとフィールド）は次のとおりです。

- GUI 画面
    - 固定テキスト，出力テキスト，入出力テキスト
  - CUI 画面およびフィールドボックス
    - 固定フィールド，出力フィールド，入出力フィールド
- この表示属性は，各種属性ダイアログで設定します。また，ここで説明する属性の中には，ドロースेटアップの「表示属性の動的変更」の「テキスト・フィールド・ポップアップ・コンボボックス」で設定した修飾名を用意しておくことで，AP から動的に変更することもできます。

キーエントリ・ラベル系のオブジェクト（テキストとフィールド）のオブジェクトの表示属性を次の表に示します。

表 A-1 キーエントリ・ラベル系のオブジェクト（テキストとフィールド）の表示属性

表示属性	固定		出力		入出力 <sup>1</sup>	
	テキスト	フィールド <sup>2</sup>	テキスト	フィールド <sup>2</sup>	テキスト	フィールド <sup>2</sup>
使用目的	×	×				
詳細目的 <sup>3 4</sup>	×	×	×	×		
空白入力	×	×	×	×		
遷移条件 <sup>4</sup>	×	×	×	×		
表示方法						
背景色	( 枠付き だけ )	×	( 枠付きだ け )	×		×
枠高		×		×		×
文字色						
文字の拡大					×	×
文字サイズ		×		×		×
文字の書体		×		×		×
文字の強調		×		×		×
テキスト・フィールド 反復	×	×				
枠付き		×		×	×	×
反転表示する						
フォーカス（カーソル） 設定	×	×	×	×		

表示属性	固定		出力		入出力 <sup>1</sup>	
	テキスト	フィールド <sup>2</sup>	テキスト	フィールド <sup>2</sup>	テキスト	フィールド <sup>2</sup>
入力済み	×	×	×	×		
自動送信	×	×	×	×		
入力必須 <sup>4 5</sup>	×	×	×	×		
ワンタッチクリア	×	×	×	×	<sup>5</sup>	
改行入力 <sup>6</sup>	×	×	×	×	×	
テキスト			×	×	×	×
テキスト揃え			×	×	×	×
コメントの記述	×	×				
APからデータを表示する(初期値)	×	×	×	×		
下位項目	×	×				

## (凡例)

- : 属性を動的変更できる。
- : 属性をダイアログで指定できる。
- ×

×

## 注 1

入出力テキストには必ず枠が付けられます。

## 注 2

フィールドの、「文字サイズ」「文字の書体」「文字の強調」は、フィールドボックスダイアログで一括で指定できます。

## 注 3

CUI画面では使用できません。GUI画面およびGUI画面のフィールドボックス中で使用できます。

## 注 4

テスト表示時には無効になります。なお、詳細目的と遷移条件は、次の場合だけ無効になります。

詳細目的：使用目的が「数字」のときの「モジュラスチェック」  
 遷移条件：「手動スキップ・全桁必須」

## 注 5

「入力済み(入力しなくても表示データをAPに返す)」が指定されている場合は、無効になります。

## 注 6

フィールドボックス内の入出力フィールドで、使用目的が「カナ」、「英数」および「日本語」の場合に使用できます。

## 付録 A.3 キーエントリ・ラベル系オブジェクトの表示属性 一覧（日付テキスト・フィールドと時刻テキスト・ フィールド）

キーエントリ・ラベル系オブジェクト（日付および時刻テキスト・フィールド）は次のとおりです。

- GUI 画面  
出力日付テキスト，出力時刻テキスト，入出力日付テキスト，入出力時刻テキスト
- CUI 画面およびフィールドボックス  
出力日付フィールド，出力時刻フィールド，入出力日付フィールド，入出力時刻フィールド

キーエントリ・ラベル系オブジェクト（日付および時刻テキスト・フィールド）の表示属性を次の表に示します。

表 A-2 キーエントリ・ラベル系オブジェクト（日付テキスト・フィールドと時刻テキスト・フィールド）の表示属性

表示属性	出力		入出力	
	時刻	日付	時刻	日付
表示書式				
入力形式	×	×		
チェック範囲	×	×		
空白入力	×	×		
遷移条件 <sup>1</sup>	×	×		
表示方法				
背景色	( 枠付きだけ )	( 枠付きだけ )		
枠高				
文字色				
文字の拡大			×	×
文字サイズ				
文字の書体				
文字の強調				
枠付き			×	×
テキスト・フィールド反復				
反転表示する				
フォーカス設定	×	×		
入力済み	×	×		
自動送信	×	×		

表示属性	出力		入出力	
	時刻	日付	時刻	日付
入力必須 <sup>1</sup> <sup>2</sup>	×	×		
ワンタッチクリア	×	×		
AP からデータを表示する	×	×		
下位項目				

(凡例)

- : 属性を動的変更できる。
- : 属性を指定できる。
- ×

×

注 1

テスト表示時には無効になります。なお、遷移条件は、「手動スキップ・全桁必須」が指定されている場合だけ無効になります。

注 2

「入力済み（入力しなくても表示データを AP に返す）」が指定されている場合は、無効になります。

## 付録 A.4 選択エン트리系オブジェクトの表示属性一覧

選択エン트리系のオブジェクトは次のとおりです。

ポップアップ、コンボボックス（メニュー選択コンボボックス / キー入力コンボボックス）、スピンボックス

この表示属性は、各種属性ダイアログで設定します。また、ここで説明する属性の中には、ドロースेटアップの「表示属性の動的変更」の「テキスト・フィールド・ポップアップ・コンボボックス」で表示属性を設定した修飾名を利用することで、AP から動的に表示属性変更できるものもあります。

選択エン트리系のオブジェクトの表示属性を次の表に示します。

表 A-3 選択エン트리系のオブジェクトの表示属性

表示属性		ポップアップ	コンボボックス	スピンボックス
属性	スピン種別	×	×	
	使用目的	-		×
	詳細項目	×	(キー入力だけ)	×
	最大値 / 最小値	×	×	
	増分値	×	×	
	表示方法			

	表示属性	ポップ アップ	コンボ ボックス	スピン ボックス
	遷移条件 <sup>1</sup>			
	背景色			
	枠高			
	文字色			
	文字サイズ			
	文字の書体			
	文字の強調			
	反転表示する			
	フォーカス設定			
	自動送信			
	コメントの記述			
	入力/選択済み			
	入力/選択必須 <sup>1 2</sup>		×	×
	ワンタッチクリア	(手動だけ)	×	×
	ボックス反復			
	AP から初期値を変更する	×		×
	(メニューの表示属性) 背景色		×	×
	(メニューの表示属性) ラベル色		×	×
	(メニューの表示属性) タイトル色		×	×
	初期値 (通知コード)	×		×
	AP からデータを表示する	×	×	
	下位項目	×	×	
	選択したらメニューを閉じる		×	×
メ	使用目的		-	-
ニ	メニュー形式 <sup>3</sup>	(可変だけ)	×	-
ユ	文字色	×		-
	アクセスキーを指定する		×	-
	選択ラベルと通知コードを同じ値にする		×	-
	タイトルを付ける		×	-
	最初にメニューを開く <sup>3</sup>		×	-
	コメントの記述		×	-
	AP から初期値を変更する		×	-

表示属性	ポップアップ	コンボボックス	スピンボックス
メニューデータをファイルで指定する <sup>3</sup>	(可変だけ)	×	-
メニューの項目数		(可変だけ)	-
キー		×	-
ラベル			-
選択ラベル		×	-
通知コード			-
初期値 <sup>3</sup>	(可変だけ)	×	-

(凡例)

- : 属性を動的に変更できる。
- : 属性を指定できる。
- ×
- : 該当しない。

注 1

テスト表示時には無効になります。なお、遷移条件は、「手動スキップ・全桁必須」が指定されている場合だけ無効になります。

注 2

「選択済み（選択しなくても通知コードを AP に返す）」が指定されている場合は、無効になります。

注 3

ポップアップフィールドには指定できません。

## 付録 A.5 候補選択コントロール系オブジェクトの表示属性一覧

候補選択コントロール系のオブジェクトは次のとおりです。

ラジオボタンボックス、チェックボタンボックス、リストボックス、トグルフィールド

この表示属性は、各種属性ダイアログで設定します。また、ここで説明する属性の中には、ドロースタックアップの「表示属性の動的変更」の「テキスト・フィールド・ポップアップ・コンボボックス」および「ボタン・リスト・メニュー」で表示属性を設定した修飾名を利用することで、AP から動的に表示属性変更できるものもあります。

候補選択コントロール系のオブジェクトの表示属性を次の表に示します。

表 A-4 候補選択コントロール系オブジェクトの表示属性

	表示属性	ラジオ ボタン	チェック ボタン	リスト <sup>1</sup> ボックス	トグル <sup>1</sup> フィールド
ボ	トグル種別	×	×	×	
タ	トグル色	×	×	×	
ン	使用目的		(可変)		-
ボ	背景色(枠付きだけ)				×
ッ	枠付き				×
ク	(垂直・水平)スクロールバーを付ける	×	×		×
ス	文字色	(可変だけ)	(可変だけ)		×
	文字サイズ				×
	文字の書体				×
	文字の強調				×
	ボタンを横に並べる			×	×
	フォーカス(カーソル)設定				×
	自動送信		×	(単一選択だけ)	
	自動スキップ(次のボックスへ)		<sup>2</sup>	<sup>2</sup>	×
	選択必須 <sup>3</sup> <sup>4</sup>		×	(単一選択だけ)	×
	フィールド反復	×	×	×	
ボ	使用目的	-	(固定)	-	
タ	表示方法	<sup>5</sup>	<sup>5</sup>	×	
ン	フォーカス(カーソル)設定				
	選択済みにする	<sup>5</sup>	<sup>5</sup>		
	不活性(選択できない状態にする)	<sup>5</sup>	<sup>5</sup>		
	ラベル色	<sup>5</sup>	<sup>5</sup>	×	×
	ラベル	(固定だけ)	(固定だけ)		×
	通知コード	(固定だけ)	(固定だけ)		
	自動スキップ(次のボタン/リスト項目/フィールドに)	×	<sup>5</sup>	(複数選択だけ)	



## (凡例)

- : 属性を動的変更できる。
- : 属性を指定できる。
- × : 属性を指定できない。
- : 該当しない。

## 注 1

リストボックスおよびトグルフィールドの表示属性は、一つのダイアログで指定します。

## 注 2

ボックス内の最後のチェックボタンやリスト項目を選択すると次のボックスにスキップします。

## 注 3

「選択済みにする」が指定されている場合は、無効になります。

## 注 4

テスト表示時には無効になります。

## 注 5

可変ボタンの場合、ボタン単位での指定はできません。

## 付録 A.6 グループコントロール系オブジェクトの表示属性一覧

グループコントロールオブジェクトは次のとおりです。

グループボックス、フィールドボックス、フレーム

この表示属性は、各種属性ダイアログで設定します。グループコントロールオブジェクトの表示属性を次の表に示します。

表 A-5 グループコントロールオブジェクトの表示属性

表示属性	グループボックス	フィールドボックス	フレーム
タイトルを付ける		×	×
枠付き			×
(垂直/水平)スクロールバーを付ける	×		×
背景色		×	×
スクロール禁止領域の背景色	×		×
スクロール領域の背景色	×		×
フォーカス設定	×		×
けい線の色	×		×
文字サイズ	×		×
文字の書体	×		×
文字の強調	×		×
フレームの反復	×	×	

(凡例)

- : 属性を指定できる。
- x : 属性を指定できない。

## 付録 A.7 グラフィックコントロール系オブジェクトの表示属性一覧

グラフィックコントロール系オブジェクトは次のとおりです。

セパレータ, けい線, 塗りつぶしフィールド, メッセージアイコン, グラフィック (固定 / 出力)

この表示属性は, 各種属性ダイアログで設定します。グラフィックコントロール系オブジェクトの表示属性を次の表に示します。

表 A-6 グラフィックコントロール系オブジェクトの表示属性

表示属性	セパレータ	フィールドボックス内けい線	塗りつぶしフィールド	メッセージアイコン	固定グラフィック	出力グラフィック
線の位置			x	x	x	x
線の種類			x	x	x	x
線の太さ	x		x	x	x	x
種別	x	x	x		x	x
ボックスサイズに合わせて表示する	x	x	x	x		
枠付き	x	x		x		
スクロールバーを付ける	x	x	x	x	x	
グラフィックデータの渡し方	x	x	x	x	x	
色	x	x		x	x	x

(凡例)

- : 属性を指定できる。
- x : 属性を指定できない。

## 付録 A.8 コマンドコントロール系オブジェクトの表示属性一覧

画面を確定するオブジェクトは次のとおりです。

プッシュボタン, メニューバー (プルダウンメニュー, カスケードを含む)

この表示属性は、各種属性ダイアログで設定します。また、ここで説明する属性の中には、ドロセットアップの「表示属性の動的変更」の「ボタン・リスト・メニュー」で表示属性を設定した修飾名を用意しておくことで、AP から動的に変更できるものもあります。

画面を確定するオブジェクトの表示属性を次の表に示します。

表 A-7 画面を確定するオブジェクトの表示属性

表示属性	プッシュボタン	プッシュボタンボックス	メニューバー <sup>1</sup>
枠付き	×		×
ボタンを横に並べる	×		×
フォーカス対象外	×		×
背景色	×		×
ボタン高	×		×
文字サイズ	×		×
文字の書体	×		×
文字の強調	×		×
ラベルの種類		×	×
テキスト		×	×
文字色		×	×
テキスト揃え		×	×
選択時の動作		×	
表示方法		×	×
通知コード		×	
フォーカス設定		×	×
不活性（選択できない状態にする）		×	
アクセスキーを使用する		×	×
プルダウン / カスケードメニューを設定する	×	×	
二次ウィンドウを閉じる	<sup>2</sup>	×	
メニューの下にセパレータ線を引く	×	×	
最後のテキストを右端に表示する	×	×	
アクセスキー		×	
制御項目データ名	×	×	
データ名		×	×
ショートカットキー	×	×	
ボックス名称	×		×

(凡例)

: 属性を動的変更できる。

: 属性を指定できる。

x : 属性を指定できない。

注 1

プルダウンメニューおよびカスケードメニューも含まれます。

注 2

二次ウィンドウの定義時だけ指定できます。

## 付録 A.9 表示属性の解説

画面定義で使用できる表示属性の解説を示します。

### (1) 英字

AP から初期値を変更する

AP から初期値データを変更して表示します。この場合、出力論理マップに初期値出力用の領域が展開されます。

AP からデータを表示する

入出力テキスト・フィールドに、あらかじめ AP から初期値データを表示します。

次の場合には、この指定を「オフ」にしてください。

- キーボードからの入力だけに使う (AP からデータを表示しない) 場合
- 「使用目的」が「パスワード」や「MCR」の場合

AP からテキストを変更する

AP からプッシュボタンのラベルテキストを変更できます。

この場合、ラベルテキスト項目が論理マップに展開されます。このラベルテキスト項目にテキストデータを設定することで、AP から動的にラベルテキストを変更できます。

AP へ通知するコードの指定

メニュー項目、またはプッシュボタンに設定するイベント通知コードの値を選びます。イベント通知コードは次の 2 種に分類できます。

- ファンクションキー、送信キー、割込キー、スクリーンキーと共用  
標準値では順に PF01 ~ PF84, ENTR, BREK, SCRN
- メニューバーのメニュー項目とプッシュボタンとで共用  
標準値では A061 ~ A071

なお、イベント通知コードの値や、AP が受け取る項目は、ドローセットアップでの設定に従います。

### (2) ア行

アクセスキー

メニューバー、ポップアップメニュー、プッシュボタンボックスで、候補の選択操作

をするためのキーです。

これらのオブジェクトにフォーカスが位置づいているとき、アクセスキーを押すことで該当するメニュー項目やボタンを選択できます。また、[ Alt ]+ アクセスキーを押すことでメニューバーのメニュー項目を選択（プルダウンメニューを表示）できます。

アクセスキーを使用する

選択候補（メニューやボタン）を選ぶ手段として、マウスやカーソルキーのほかにアクセスキーを使えるようにします。

アクセスキーに数字（0～9）を指定しておくと、テンキーで候補選択の操作ができます。

### （3）力行

カーソル設定（初期カーソルを本フィールドに設定する）

AP からカーソル位置を設定しないで画面を表示したとき（例えば、論理マップのカーソル制御項目に値を設定していないなど）、このボックスにカーソルを位置づけます。

どのオブジェクトにも初期カーソルを設定していない場合は、縦位置 - 横位置の順に、位置が最小で選択できるオブジェクトにカーソルを仮定します。

下位項目

オブジェクトの入出力となる一つのデータ項目を階層化して、複数の項目に細分できます。この場合、AP は細分化された下位項目を使ってデータを参照・格納できます。下位項目は、データ型が「文字（XX）」の場合に指定できます。

空白入力

選べる項目は、オブジェクトの種類や「詳細目的」に応じて異なります。なお、入力した全角スペースは、表示・印刷セットアップの「日本語入力」タブの「全角スペースコードの扱い」に応じて処理（全角のままか、半角2個に変換）されます。

- 禁止は、空白の入力はできません。
- 全桁（半角）は、半角空白だけ入力できます。未入力ではない「データ無し」を明示的に表す場合に使います。
- 一部（半角）は、半角空白が混在する文字列を入力できます。半角空白だけの入力はありません。
- 一部（全角）は、全角空白が混在する文字列を入力できます。全角空白だけの入力はありません。
- 一部（全・半角）は、全角空白、半角空白が混在する文字列を入力できます。全角・半角空白だけの入力はありません。
- 一部 & 全桁（半角）は、半角の空白が混在する文字列を入力できます。半角空白だけの入力もできます。
- 一部 & 全桁（全角）は、全角の空白が混在する文字列を入力できます。全角空白だけの入力もできます。
- 一部 & 全桁（全・半角）は、全角・半角の空白を自由に入力できます。

#### グラフィックデータの渡し方

- ファイル名指定  
ファイル名（拡張子を含む）だけを AP が設定する標準的な方法です。グラフィックデータのファイルのパス名は、表示・印刷セットアップで変更することもできます。また、AP からファイル名の代わりに「\* CLIP」と設定すると、クリップボードのデータを出力できます。
- フルパス  
ファイルをフルパスで設定する方法です。ユーザが指定するパス名の長さに応じて数値を選んでください。

#### けい線の色

フィールドボックス内に表示するけい線の色をボックス単位で指定できます。

### (4) サ行

#### 最後のテキストを右端に表示する

メニューバーの最後のメニュー項目をメニューバー上の右端に表示します。

#### 最初にメニューを開く

手動ポップアップにフォーカスが位置づいたときに、自動的にポップアップメニューを表示するよう指定します。

#### 最大値 / 最小値

スピンボックスで表示できる最大値と最小値を指定します。

#### 自動スキップ（選択したら次の項目にフォーカスを移す）

ボックス内でボタンまたはリスト項目を選択すると、ボックス内の次のボタンまたはリスト項目へ自動的にフォーカスが遷移します。

#### 自動スキップ（選択したら次のボタン / ボックス / フィールドにフォーカスを移す）

- ボタン  
ボックス内でボタンまたはリスト項目を選択すると、ボックス内の次のボタンまたはリスト項目へ自動的にフォーカスが遷移します。
- ボックス  
ボックス内での選択操作が完了すると、フォーカス対象の次のボックスへ自動的にフォーカスが遷移します。  
複数選択できるオブジェクト（複数選択リストボックス、チェックボタンボックス）の場合は、末尾の候補が選択されるとフォーカスが遷移します。
- フィールド  
オブジェクトでの選択操作が完了すると、カーソル対象の次のフィールドへ自動的にカーソルが遷移します。

#### 自動送信（入力（または選択）直後に AP へ制御を渡す）

テキストの入力や候補（メニュー項目やボタンなど）の選択動作ごとに、AP にイベントを通知します。

入力 / 選択後にほかのオブジェクトへフォーカスを移動すると、自動的に AP に対してイベントが通知されます。このとき返すイベント通知コードは、送信キーのもの

同じです。

なお、表示・印刷セットアップのカーソルの設定で、イベント発生時の自動送信を抑制できます。

イベント通知コードの値はドロースettingsで変更できます。動作はドロースettingsで変更できます。

#### 種別

メッセージアイコンの種別を指定します。メッセージアイコンの種別には、「クエスチョン」「ワーニング」「インフォメーション」「エラー」があります。

#### 使用目的（ポップアップ）

ポップアップで使用する表示属性で、「手動」にするか「自動」にするかを指定します。

##### • POP-（自動 - 英数 / 日本語）

メニュー項目の選択で入力できます。このオブジェクトにフォーカスが位置づくると自動的にメニューが表示されます。

##### • POP-（手動 - 数字 / カナ / 英数 / 日本語）

キーボードからの文字入力と、メニュー項目の選択の2とおりでデータを入力できます。このオブジェクトにフォーカスやカーソルが位置づいたときに、プッシュボタンやメニューバーのメニュー項目（「選択時の動作」に「ポップアップ表示」を指定したもの）を選ぶと、メニュー項目が表示されます。

#### 使用目的と詳細目的（GUI画面の場合）

使用目的および詳細目的はデータの使い道を指定することで、使用できるデータ型を限定し、設定します。使用目的および詳細目的は、「付録 A.9 表示属性の解説」の表 A-8 および表 A-9 を参照してください。

#### ショートカットキー

ドロースettingsでは、ショートカットキーの ID (01 ~ 16 の数値) を指定します。

実行時には、ドロースettingsで指定した ID に対応する 1 字の半角英字がショートカットキーになります。このときの ID と英字の対応は、表示・印刷セットアップの設定に従います。

#### 初期値（ファイル）

「メニューデータをファイルで指定する」を指定した場合、メニューデータファイル名の初期値を指定します。メニューデータファイルを読み込むポップアップをテスト表示する場合には、メニューデータファイルを用意し、「初期値」を指定しておく必要があります。

#### 垂直 / 水平スクロールバーを付ける

スクロール領域にスクロールバーを付けます。スクロールバーは、ボックスからはみ出したプレーン領域の部分を表示するために使います。

ボックスの右側（垂直）、下側（水平）、またはその両方にスクロールバーを付けられます。

#### スクロールバーを付ける

グラフィックデータがボックスのサイズを超えると、超えた部分を表示するための

スクロールバーを付けます。

スクロール領域の色 / スクロール禁止領域の色 / 背景色

オブジェクトや領域の背景の色を指定します。オブジェクトの背景色をウィンドウまたはグループボックスと同じにする場合は、「透明」を選びます。

レイアウト領域上のオブジェクトは XMAP3 の標準の色合い (RGB 値) で表示されます。実行時の色合いは、表示・印刷セットアップの設定に従います。

スピン種別

スピンボックスで扱うデータ型は「数値」です。

- 数値

主に数量や件数などを表す -999999999 ~ 999999999 のデータを扱えます。増分値は整数または小数 (小数点以下 2 桁まで) を指定できます。

制御項目データ名

データ項目名 (またはデータ名) は、データの入出力に関する各項目の領域の名称です。

データ名のうち、修飾名 表示属性の変更を AP から指示するための名前 を格納する領域の名称を制御項目データ名といいます。

遷移条件

オブジェクトでデータ入力や候補選択をしたあとのフォーカス・カーソルの動きについて指定します。

- 手動スキップ・右寄せ

入力データが桁数に満たない場合、タブキーやマウスで次の入力 / 選択できる領域にフォーカス / カーソルを移動すると、入力データが右寄せされます。

- 手動スキップ・全桁必須

1 文字でも入力した場合、桁数いっぱいに入力しないと入力エラーになり、タブキーやマウスでフォーカス / カーソルの移動ができなくなります。

- 手動スキップ

タブキーやマウスでフォーカス / カーソルを移動します。なお、桁数を超えるデータを入力しようするとエラーになります。

- 自動スキップ

桁数いっぱいまで文字を入力すると、フォーカス / カーソルが次の入力 / 選択できる領域に自動的に移動します。これに対して、手動スキップの場合はタブキーを押すことでフォーカス / カーソルが次の入力 / 選択できる領域に移動します。

選択時の動作

候補 (メニュー項目やボタン) が選択されたときの動作を指定します。

- 「AP へ通知する」は、AP イベントを通知します。このときの動作は、対応させるイベント通知コードの動作 (確定・中断など) に従います。
- 「ポップアップ表示」は、使用目的が「POP・手動」のポップアップのメニュー項目を開きます。なお、メニュー項目が開くのは、あらかじめフォーカス・カーソルが位置づけてあるポップアップです。
- 「コピーと貼り付け」は、クリップボードを介したデータのコピーや貼り付けをします。メニューバーでだけ指定できます。



選択済み（選択しなくても表示データを AP に返す）

オブジェクトを転送状態にします。この場合、入力論理マップのデータ項目に必ず（操作者が変更するかどうかに関係なく）次のデータ/コードを返します。ただし、選択データがない場合は、データ消去通知文字が入力論理マップのデータ項目に戻りません。

- 「選択済み」では、候補（ボタンなど）に対応する通知コード。

選択済みにする（チェックボタン、ラジオボタンだけ）

画面を表示した時点で、ボタンをすでに選択された状態にします。仮定値を表示する場合などに使います。

選択必須（選択しない時はエラーにする）

このオブジェクトでの候補選択の操作を省略できないようにします。選択していない場合は、画面確定時（AP へ制御を返す直前）にエラーとなり、AP イベントは通知されません。

この属性は、不活性（選択できない）状態にしないなど、ほかの表示属性の指定との対応に注意してください。なお、この指定は「選択済み（選択しなくても表示データを AP に返す）」が指定されている場合は無効になります。

選択ラベルと通知コードを同じ値にする

「通知コード」の指定と同じ値を「選択ラベル」にも適用します。これは AP に返す通知コードを確定項目として表示することを示します。この場合、「選択ラベル」の指定は不要です。

この指定をしない場合は、必ず「選択ラベル」を指定します。

線の位置

線の位置を指定します。線の位置には、「上」「中央」「下」があります。

線の種類

線の種類を指定します。線の種類には、「実線」と「破線」があります。

線の太さ

線の太さを指定します。線の太さには、「中」「太」「極太」があります。

増分値

スピンボックスのスタッカを押すたびに、値が幾つずつ増減するかを指定します。

## （5）タ行

タイトルを付ける（グループボックス）

グループボックスにタイトルを付けます。タイトルは、「グループボックスの横サイズ - 2 ます」以内の長さで指定できます。

タイトルを付ける（ポップアップ）

指定するタイトルをグループボックスに付けます。タイトルとして表示する文字列を「ラベル長」の長さ以内で指定します。「ラベル長」の長さを超えた場合は、タイトル長に合わせて「ラベル長」の長さが変わります。

#### チェック範囲

チェック範囲を「部分」か「全体」かで指定します。

- 「部分」

項目ごとのチェックです。チェック内容は次のとおりです。

<日付>

「YYYY」: 1 ~ 9999

「YY」 : 0 ~ 99

「MM」 : 1 ~ 12

「DD」 : 1 ~ 31

<時刻>

「HH」: 0 ~ 23

「MM」: 0 ~ 59

「SS」: 0 ~ 59

- 「全体」

カレンダーに従ったチェックです。例えば、うるう年の2月の日付や、各月の日付などのチェックをします。

#### 通知コード

候補（ボタンやリスト項目など）の選択によってデータ入力をするオブジェクトで、各候補に割り当てるコードです。

ユーザが選んだ候補に対応する通知コードが、AP が受け取る項目に格納されます。

#### 通知コード（トグルフィールド）

トグルフィールドがオン状態のときに AP に返す通知コードを指定します。

この通知コードは、「AP が受け取る項目」で定義するデータ項目に格納されます。

ここで定義するデータ値の長さやデータ型は、「AP が受け取る項目」の「長さ」と「データ型」に合わせてください。

#### データ長を変更する

使用目的が日本語の論理項目に関して、データ長を変更するかどうかを指定します。指定しない場合は、標準設定されます（デフォルト状態）。

#### テキスト（プッシュボタン）

ボタンのラベルに表示する文字列を指定します。

#### テキスト揃え

テキスト・フィールドやラベル中の文字列のそろえ方を指定します。

- 「中央」は、文字列を中央に寄せて表示します。
- 「左」は、文字列を左側に寄せて表示します。
- 「右」は、文字列を右側に寄せて表示します。
- 「均等」は、文字列を均等に配置して表示します。ただし、オブジェクトの長さによって均等に配置できない場合は、文字列は中央に寄せて表示されます。

ただし、ユーザ AP でラベルテキスト項目名に変更内容を指定した場合は、「テキスト揃え」は「左」になります。

### テキスト・フィールド反復

テキスト・フィールドは、反復定義で反復できます。反復定義は、一つのオブジェクトを縦または横方向に繰り返します。表形式で同じ属性のオブジェクトを並べる場合などに使います。

反復定義を使うと、同じ属性のオブジェクトを一つずつ配置するよりも効率良く定義できます。また、論理マップは配列（日立 COBOL では OCCURS 句）で展開されるので、AP のロジックも反復指定で実現できます。

### 動的変更（AP から表示属性を変更する）

AP が画面を表示するとき、表示属性を動的に変更できます。表示属性とは、テキスト・フィールドの文字色や、ボタンの活性 / 不活性などを指定する属性です。この場合、動的変更情報格納エリア（制御項目）が論理マップに展開されます。この制御項目に修飾名を設定することで、AP から動的に属性を変更できます。また、データ有無コードを設定すると、ドローで定義した属性が有効になります。修飾名と表示属性との対応は、ドローセットアップで設定します。

### トグル種別

トグルフィールドをチェックする印を指定します。

トグル種別は次のとおりです。

チェック：「 か 」

正誤：「 か ×」

はい / いいえ：「Y か N」

### トグル色

オブジェクトの文字の色を指定します。

レイアウト領域上のオブジェクトは XMAP3 の標準の色合い（RGB 値）で表示されます。実行時の色合いは、表示・印刷セットアップの設定に従います。

## （6）ナ行

### 長さ（アクセスキー）

アクセスキーの長さ（桁数）を指定します。アクセスキーの桁数は 1 ~ 4 桁で指定します。指定した長さより短いアクセスキーを入力しようとするとエラーになります。なお、メニュー項目をファイルで指定する場合、ドローでの長さの指定はできません。ポップアップメニューファイルのエディタで作成しておきます。

### 二次ウィンドウを閉じる

二次ウィンドウ上で AP に制御を返す操作（メニュー項目やボタンの選択）をしたとき、そのウィンドウを閉じます。この場合、別の二次ウィンドウを表示したり、一次ウィンドウへフォーカス（対話権）を戻すことになります。

この項目は、「通知コード」に指定したイベント通知コードの動作が「AP 通知（終了）」または「AP 通知（取消）」の場合に選択されます。

### 入力済み（入力しなくても表示データを AP に返す）

オブジェクトを転送状態にします。この場合、入力論理マップのデータ項目に必ず（操作者が変更するかどうかに関係なく）次のデータ / コードを返します。

ただし、表示データがない場合は、データ消去通知文字が入力論理マップのデータ項目に代入されます。

- 「入力済み」では、オブジェクト内の表示データ（文字列など）。ただし、ポップアップやコンボボックスの場合は表示データに対応する通知コード。

入力必須（入力しない時はエラーにする）

このオブジェクトでのデータ入力の操作を省略できないようにします。入力していない場合は、画面確定時（AP へ制御を返す直前）にエラーとなり、AP へイベントは通知されません。

この属性は、不活性（入力できない）状態にしないなど、ほかの表示属性の指定との対応に注意してください。なお、この指定は「入力済み（入力しなくても表示データを AP に返す）」が指定されている場合は無効になります。

## （7）八行

背景色

オブジェクトや領域の背景の色を指定します。オブジェクトの背景色をウィンドウまたはグループボックスと同じにする場合は、「透明」を選びます。

なお、「枠付き」を指定できるオブジェクトで、枠を付けない場合は「透明」で固定になります。

レイアウト領域上のオブジェクトは XMAP3 の標準の色合い（RGB 値）で表示されます。実行時の色合いは、表示・印刷セットアップの設定に従います。

次に各オブジェクトと背景色の関係を示します。

オブジェクトの種類	背景色の設定	結果
ボックス系オブジェクト (枠付きのテキスト・フィールド系オブジェクトも含む)	透明	画面の背景色が反映される
	色付き	設定した色が反映される
ボタン系オブジェクトおよびスクロールバー	-	常にグレー

（凡例）

- : 該当しない。

反転表示する

背景色と文字色を反転して表示します。項目を強調して表示する場合などに使用します。

ビットマップ

ボタンに表示するビットマップまたはビットマップとテキストの組み合わせを指定します。活性用と不活性用を指定します。

表示書式（入出力テキスト・フィールド）

テキスト・フィールド上での入力文字の表示のしかたを、次の項目から選びます。「使用目的」が「金額」または「数値」の場合に指定できます。

コンマ付加は、3 桁ごとにコンマ（,）を付けて表示します。「¥ 付加」「\* 付加」との

組み合わせもあります。

¥ 付加は、先頭に金額記号 (¥) を付けます。

ゼロサプレスは、前方のゼロ (0) を表示しません。

\* 付加は、前方のゼロ (0), ヌル, スペースをすべてアスタリスク (\*) に置き換えて表示します。

ゼロ付加は、前方のゼロ (0), ヌル, スペースをすべてゼロに置き換えて表示します。

表示書式 (日付・時刻)

日付や時刻を表示・印刷する形式を指定します。「年月日」や「時分秒」を表す単位の表記を選べます。また、一部の単位だけ表示・印刷の対象にする指定もできます。日付・時刻の表示書式の詳細は「付録 A.9 表示属性の解説」の表 A-10 および表 A-11 を参照してください。

表示方法 (キーエントリ系, 選択エントリ系)

オブジェクトや文字列の表示のしかたを指定します。指定できる種類は、オブジェクトの種類や使用目的などの指定に応じて異なります。なお、指定はレイアウト上の表示には反映されません。

- 「入力可能・標準表示」は、オブジェクトや文字を通常の状態を表示します。入力または選択できます。
- 「入力可能・文字だけ非表示」は、オブジェクトの外観は表示しますが文字は表示しません。入力または選択できます。
- 「入力不可・標準表示」は、テキストボックスまたはフィールドは表示されますが、入力または選択はできません。
- 「入力不可・グレースアウト表示」は、オブジェクトや文字を不活性状態のグレーで表示します。入力または選択はできません。
- 「入力不可・文字だけ非表示」は、オブジェクトの外観は表示しますが文字は表示しません。入力または選択はできません。
- 「入力不可・全体非表示」は、文字を含む全体を表示しません。入力または選択はできません。

データ入力用のオブジェクトで、入力できる状態にする場合は、「入力可能」と付くものを指定してください。なお、出力テキスト・フィールドでは「入力可能 (不可)」は表示されません。また、すべてのフィールドで、「グレースアウト表示」および「全体表示」の指定はできません。

表示方法 (候補選択コントロール系)

オブジェクトや文字列の表示のしかたを指定します。指定できる種類は、オブジェクトの種類や使用目的などの指定に応じて異なります。なお、指定はレイアウト上の表示には反映されません。

- 「選択可能・標準表示」は、オブジェクトや文字を見える状態を表示します。
- 「選択可能・文字だけ非表示」は、オブジェクトは表示しますが文字は表示しません。
- 「選択不可・標準表示」は、テキストボックスまたはフィールドは表示されますが選択できません。
- 「選択不可・グレースアウト表示」は、オブジェクトや文字を不活性状態のグレーで表

示します。選択はできません。

- 「選択不可・文字だけ非表示」は、オブジェクトは表示しますが文字は表示しません。選択はできません。
- 「入力不可・全体非表示」は、文字を含む全体を表示しません。

選択できる状態にする場合は、「選択可能」と付くものを指定してください。

表示方法（プッシュボタン）

オブジェクトや文字列の表示のしかたを指定します。指定できる種類は、オブジェクトの種類や使用目的などに応じて異なります。なお、指定はレイアウト領域上には反映されません。

- 「標準表示」は、オブジェクトや文字を見える状態で表示します。
- 「全体非表示」は、文字を含むオブジェクト全体を表示しません。

フォーカス設定（初期フォーカスを本ボックスに設定する）

AP からフォーカス位置を設定しないで画面を表示したとき（例えば、論理マップのフォーカス制御項目に値を設定していないなど）、このボックスにフォーカスを位置づけます。

どのオブジェクトにも初期フォーカスを設定していない場合は、縦位置 - 横位置の順に、位置が最小で選択可能なオブジェクトにフォーカスを仮定します。

フォーカス設定（フォーカスを本ボタン（またはリスト項目）に位置づける）

ボタンボックス、リストボックスにフォーカスが移ったときに、ボックス中で最初にフォーカスを位置づける候補（ボタンまたはリスト項目）にします。

なお、ボックス内のどの候補にも初期フォーカスを設定しない場合は、行数、列数の順に位置が最小で、活性状態の候補にフォーカスが仮定されます。

フォーカス対象外（フォーカス遷移の対象としない）

ボタンボックスを、タブキーによるフォーカス遷移の対象外にします。この場合、マウス操作でだけフォーカスを位置づけられます。

不活性（選択できない）状態にする

候補（ボタン/メニュー項目）を不活性にして、選択できない状態にします。選べる候補を制限したい場合に使います。

活性化する場合は、「動的変更」を指定し、AP から制御項目に修飾名を設定します。

プルダウン/カスケードメニューを設定する

メニューバー上のメニュー項目に、下位のメニューを設定します。メニューバー上のメニュー項目の下位項目として、プルダウンメニューを指定できます。プルダウンメニューの下位項目としてカスケードメニューを指定できます。

なお、それぞれのメニューを指定した場合、「選択時の動作」および「AP へ通知するコード」は、最下位のメニューにだけ設定できます。

フレームの反復

フレームを反復する回数、および間隔を指定できます。

ボックスサイズに合わせて表示する

ボックスの大きさに合わせてグラフィックデータを拡大・縮小して表示します。

グラフィックデータがビットマップ (.BMP)、および JPEG (.JPG または .JPEG) で、この指定が「オフ」の場合は、ボックスの中心に原寸大で配置されます。メタファイル (.WMF または .EMF) の場合は必ず「オン」になります。

#### ボタン高

プッシュボタンの縦幅を指定します。ボックスのサイズを超えない範囲で変更できません。

- 「2 ます」は約 2 文字分の高さです。ボタンを文字とほぼ同じ高さに合わせるときに使用します。
- 「2.5 ます」は約 2.5 文字分の高さです。ボタンラベルをほかのボックスの文字と同じ大きさにするときを使用します。
- 「3.5 ます」は約 3.5 文字分の高さです。ボタンを強調したり、アイコンを表示するときに使用します。
- 「4.5 ます」は約 4.5 文字分の高さです。ボタンを強調したり、アイコンを表示するときに使用します。

#### ボタンを横に並べる

ボタンボックス内で、複数のボタンを順に横方向に配置します。この場合、フォーカスも横方向（左から右、上から下の順）に遷移します。

#### ボックス名称

プッシュボタンボックスの名前を指定します。

この名前は、AP から初期フォーカスをこのボックスに設定する場合に、AP が指定する名前になります。

## (8) マ行

#### メニュー形式

ポップアップメニューの表示形式を指定します。メニュー形式は、「メニューデータをファイルで指定する」が指定されているときだけ、指定できます。

- 標準 (メニュー): メニュー項目を従来のメニュー形式で表示します。階層構造の下位メニュー項目を表示するときはカスケードメニューとして表示します。
- リスト - ポップアップ: メニュー項目をリストとしてポップアップ表示します。階層構造の下位メニュー項目を表示するときは、第 2 ~ 第 8 レベルのリストとして表示します。
- リスト - ダイアログ: メニュー項目を、独立したダイアログのリストとして表示します。階層構造の下位メニュー項目を表示するときは、第 2 ~ 第 8 レベルのリストとして表示します。

#### メニューデータをファイルで指定する

ポップアップメニューを、ポップアップメニューファイルから読み込んで表示する場合、「メニューデータをファイルで指定する」を指定します。ポップアップメニューファイルとは、ポップアップテキストに表示するメニューデータを格納したファイルで、あらかじめ用意しておきます。

#### メニューの値

ポップアップメニューのラベルの値，および通知コードを指定します。

#### メニューの下にセパレータ線を引く

プルダウンメニュー，カスケードメニューのメニュー項目の下にセパレータ線を引き，メニュー項目間を区切ります。

#### 文字サイズ

「大」は，基準文字サイズより大きい文字にします。

「標準」は，基準文字サイズと同じサイズにします。

「小」は，基準文字サイズより小さい文字にします。

「基準文字サイズ」は，画面属性ダイアログで指定します。

#### 文字サイズ（フィールドボックス）

プレーンの基準となる文字サイズを指定します。この指定は，プレーンのます目の基準になります。ただし，レイアウトの表示上は，文字の大きさは変わりません。

#### 文字色

オブジェクトの文字の色を指定します。

レイアウト領域上のオブジェクトは XMAP3 の標準の色合い（RGB 値）で表示されません。実行時の色合いは，表示・印刷セットアップの設定に従います。

#### 文字の拡大

固定または出力テキスト・フィールドの文字の拡大には次の種類があります。

「標準」は，文字を標準のサイズで表示します。

「平体」は，文字の横幅を 2 倍に拡大して表示します。

「長体」は，文字の縦幅を 2 倍に拡大して表示します。

「倍角」は，文字を縦横 2 倍に拡大して表示します。

#### 文字の強調

文字を「斜体」や「太字」にできます。

文字の強調は，各文字サイズや書体に適用できます。

なお，斜体にすると領域のサイズ（長さ）が 1 字分大きくなります。また，フィールドボックス内で文字を斜体にするとレイアウト上で最後の文字が正しく表示されないことがあります，実際の表示には影響しません。

#### 文字の書体

「標準」，「明朝」，「ゴシック」の 3 種類があります。

標準フォントは，表示・印刷セットアップで変更できます。

### (9) ラ行

#### ラベル色

オブジェクトの文字の色を指定します。

レイアウト領域上のオブジェクトは XMAP3 の標準の色合い（RGB 値）で表示されません。実行時の色合いは，表示・印刷セットアップの設定に従います。



### ラベルの種類

ボタンに付けるラベルの種類を指定します。テキストとビットマップの両方を表示する指定もできます。選択したラベルの種類に応じて「テキスト」や「ビットマップ」を指定してください。

## (10)ワ行

### 枠高

ボックスの枠のサイズ（縦幅）を変更できます。「2ます」、「2.5ます」および「3ます」があります。

枠高が「2ます」のオブジェクト同士は上下に隣接して配置できます。

### 枠付き（テキストに枠を付ける）

テキストの周囲の枠線を明示します。

「オフ」から「オン」にする場合、あらかじめオブジェクトの周囲に必要な間隔を空けて配置しておく必要があります。

### 枠付き（ボックスに枠を付ける）

ボックスの周囲の枠線を明示します。

「オフ」から「オン」にする場合、あらかじめオブジェクトの周囲に必要な間隔を空けて配置しておく必要があります。

### ワンタッチクリア（選択状態にする）

画面の表示時に、このオブジェクトに表示される出力データをすべて選択した状態にします。

ワンタッチで表示データを削除できる状態になります。表示データを必ず書き換える場合などに使います。

使用目的と、使用目的を指定できるオブジェクトの一覧を次の表に示します。

表中のオブジェクト名称は、次のとおりです。

- 出力・入出力テキスト（フィールド）：出力・入出力
- メニュー選択コンボボックス：メニュー選択コンボ
- キー入力コンボボックス：キー入力コンボ
- ラジオボタン：ラジオ
- チェックボタン：チェック
- リストボックス：リスト

表 A-8 使用目的

使用目的	オブジェクト	説明
数字	<ul style="list-style-type: none"> <li>出力・入出力</li> <li>キー入力コンボ</li> <li>リスト<sup>1</sup></li> <li>ラジオ<sup>1</sup></li> <li>チェック<sup>1</sup></li> <li>トグル<sup>1</sup></li> </ul>	数字コードデータの入力 必要に応じて「遷移条件」の「全桁必須」属性と組み合わせ 合わせて使用する。 COBOLの PICTURE 句に従った入力チェックをする。 モジュラスチェックをする。
POP (手動 数字)	<ul style="list-style-type: none"> <li>ポップアップ</li> </ul>	
金額 (GUI 画面)	<ul style="list-style-type: none"> <li>出力・入出力</li> </ul>	右寄せで表示する。また、先頭に「¥」を付けたり、3 桁 ごとに「,」を付けたり、小数点の入力を許したりする。 入力データ:「0 ~ 9」「+」「-」
数値 (GUI 画面)	<ul style="list-style-type: none"> <li>出力・入出力</li> </ul>	右寄せで表示する。オプションを使用して、負の値や小 数点の入力を許したり、チェックしたりできる。 入力データ:「0 ~ 9」「+」「-」
カナ	<ul style="list-style-type: none"> <li>出力・入出力</li> <li>キー入力コンボ</li> </ul>	カナ文字入力のため、日本語入力システムをオンにする。 カナだけの入力に加え、オプションを使用して数字や英 文字を入力できる。
POP (手動 カナ)	<ul style="list-style-type: none"> <li>ポップアップ</li> </ul>	
英数	<ul style="list-style-type: none"> <li>出力・入出力</li> <li>キー入力コンボ</li> <li>メニュー選択コ ンボ</li> <li>リスト</li> <li>ラジオ</li> <li>チェック</li> <li>トグル</li> </ul>	半角文字の入力用。オプションを使用して、英数、数字、 カナなどの入力を制限できる。
POP (手動 英数)	<ul style="list-style-type: none"> <li>ポップアップ</li> </ul>	
POP (自動 英数)		
日本語	<ul style="list-style-type: none"> <li>出力・入出力</li> <li>キー入力コンボ</li> <li>メニュー選択コ ンボ</li> <li>リスト<sup>2</sup></li> <li>ラジオ<sup>2</sup></li> <li>チェック<sup>2</sup></li> <li>トグル<sup>2</sup></li> </ul>	漢字の入力用。日本語入力システムをオンにする。オブ ションを使用して、漢字だけの入力か、漢字と英数の混 在入力かをチェックできる。
POP (手動 日本 語)	<ul style="list-style-type: none"> <li>ポップアップ</li> </ul>	
POP (自動 日本 語)		

使用目的	オブジェクト	説明
パスワード	• 出力・入出力	半角文字の入力用。入力した文字は表示されない。GUI画面では、オプションを使用して入力文字の代わりに「*」を表示するか、何も表示しないかを指定できる。
MCR	• 出力・入出力	キーボードではなく ID カード (MCR) での入力ができる。オプションを使用して、入力された文字を表示することもできる。

## 注 1

入力できるデータ型は「99999」「その他のピクチャ」です。

## 注 2

入力できるデータ型は「文字」「漢字」です。

表 A-9 詳細目的とデータ型

使用目的	詳細目的	データ型 <sup>1</sup>		入力できる文字
		COBOL		
		入力データ型	出力データ型	
数字 <sup>2</sup>	数字・ 数字記号 <sup>2</sup>	文字 (XX) 2	文字 (XX)	0 ~ 9 + - * . , ¥
		99999 <sup>2</sup>	99999	
		99999	ZZZZ9	
		その他のピクチャ	その他のピクチャ	
	モジュラス	文字 (XX)	文字 (XX)	0 ~ 9 最終桁がモジュラスチェック用数字
		99999	99999	
		99999	ZZZZ9	
		その他のピクチャ	その他のピクチャ	
	数字	文字 (XX)	文字 (XX)	0 ~ 9
		99999	99999	
金額	数字	文字 (XX)	文字 (XX)	0 ~ 9
		99999	99999	
	小数点付き	999V9	999.9	0 ~ 9 .(1個だけ)
		99V99	99.99	
		9V999	9.999	
		V9999	.9999	

使用目的	詳細目的	データ型 <sup>1</sup>		入力できる文字
		COBOL		
		入力データ型	出力データ型	
	符号付き	S99999	-99999	0 ~ 9 + -
	符号・小数点 付き	S999V9	-999.9	0 ~ 9 + -
		S99V99	-99.99	.(1個だけ)
		S9V999	-9.999	
SV9999	-.9999			
数値	数字	文字 (XX)	文字 (XX)	0 ~ 9
		99999	99999	
	小数点付き	999V9	999.9	0 ~ 9
		99V99	99.99	.(1個だけ)
		9V999	9.999	
		V9999	.9999	
	符号付き	S99999	-99999	0 ~ 9 + -
	符号・小数点 付き	S999V9	-999.9	0 ~ 9 + -
		S99V99	-99.99	.(1個だけ)
		S9V999	-9.999	
		SV9999	-.9999	
	数字・数字記号	文字 (XX)	文字 (XX)	0 ~ 9 + - * . , ¥
	カナ <sup>2</sup>	カナ・半角 <sub>2</sub>	文字 (XX)	文字 (XX)
カナ		カナ：ア～ンとカナ小文字		
カナ・英大・数		英大：A ~ Z		
カナ・英大・数・マイナス		数：0 ~ 9 マイナス：-		
カナ・英大・数・長音		長音：カナ長音記号		
カナ・英大・数・マイナス・長音				
カナ・数				
カナ・数・マイナス				

使用目的	詳細目的	データ型 <sup>1</sup>		入力できる文字
		COBOL		
		入力データ型	出力データ型	
	カナ・数・長音			
	カナ・数・マイナス・長音			
英数 <sup>2</sup>	半角 <sup>2</sup>	文字 (XX)	文字 (XX)	半角：すべての半角文字
	英大・カナ・数・記号			英大：A ~ Z 英小：a ~ z
	英大			カナ：ア～ンとカナ小文字
	アスタリスク・英大			数：0 ~ 9 マイナス：-
	英大・英小			アスタリスク：*
	英大・数・マイナス			コンマ：, 記号：# @ ¥ / などの英記号
	英大・数			
	英大・英小・コンマ			
	数・マイナス			
日本語 <sup>2</sup>	混在 <sup>2</sup> (全角・半角)	文字 (XX)	文字 (XX)	全角と半角
	漢字 <sup>2</sup>	文字 (XX)	文字 (XX)	全角だけ
		漢字 (NN)	漢字 (NN)	
		漢字 (XX) <sub>3</sub>	漢字 (XX) <sub>3</sub>	
パスワード	半角	文字 (XX)	文字 (XX)	半角：すべての半角文字
	英大			英大：A ~ Z
	英大・英小			英小：a ~ z
	英大・数			数：0 ~ 9
MCR	半角	文字 (XX)	出力なし	半角：すべての半角文字
POP (手動 - 数字)	数字・数字記号	文字 (XX)	文字 (XX)	0 ~ 9 + - * . , ¥
POP (手動 - カナ)	カナ・半角	文字 (XX)	文字 (XX)	カナ：ア～ンとカナ小文字 半角：すべての半角文字

使用目的	詳細目的	データ型 <sup>1</sup>		入力できる文字
		COBOL		
		入力データ型	出力データ型	
POP (手動 - 英数) (自動 - 英数)	半角	文字 (XX)	文字 (XX)	半角:すべての半角文字
POP (手動 - 日本語) (自動 - 日本語)	混在 (全角・半角)	文字 (XX)	文字 (XX)	全角と半角
	漢字	文字 (XX)	文字 (XX)	全角だけ
		漢字 (NN)	漢字 (NN)	

## 注 1

C 言語の場合は、データ型「char」だけ指定できます。

## 注 2

キー入力コンボボックスで指定できる使用目的、詳細目的およびデータ型です。

## 注 3

CUI 画面の入出力フィールドの場合にだけ指定できます。

## 使用目的

使用目的は、出力、入出力フィールドで入出力するデータを何にするのかなどを指定する属性です。使用目的には、次の種類があります。

## • 数字

半角の文字を入出力できます。入力できる文字はデータ型などに応じて異なり、指定した文字以外を入力するとエラーになります。データ型と遷移条件によって、次のようになります。

## 1. データ型が文字 (XX) 以外 (数字または数字編集) の場合

「0 ~ 9」「.」「,」「+」「-」「\*」「¥」「スペース」が入力できます。さらに、指定したピクチャパターンとのフォーマットチェックが行われます。

## 2. データ型が文字 (XX) で、遷移条件が右寄せ表示以外の場合

「0 ~ 9」「.」「,」「+」「-」「\*」「/」「(」「)」「<」「>」「=」「?」「:」「;」が使用できます。

## 3. データ型が文字 (XX) で、遷移条件が右寄せ表示の場合

「0 ~ 9」「.」「+」「-」が使用できます。

## • 金額 (GUI 画面の場合)

半角の文字を入出力できます。入力データの数字部分に、3桁ごとにコンマ (,) が付けられます。

## • 数値 (GUI 画面の場合)

半角の文字を入出力できます。データは右寄せで表示されます。件数や数量を入出力する場合に使用します。

- カナ  
カナおよび半角文字を入出力できます。日本語入力システムを使用してカナ入力モードになります。ただし、日本語入力モードを変更すると、カナ以外の半角文字（スペース、英数、記号）も入力できます。全角文字を入力すると、ドロースタックアップで設定してあるエラー通知文字を AP に返します。標準値は (FF)<sub>16</sub> です。
- 英数  
半角文字だけの入出力ができます。半角文字（スペース、英大文字、英小文字、数字、カナ、記号）が使用できます。全角文字を入力すると入力エラーになります。
- 日本語  
全角文字、全角と半角の混在文字の入力ができます。日本語入力システムを使用してカナ入力モードになります。データ型が「漢字 (NN)」または「漢字 (XX)」の場合、全角文字だけが入力できます。半角文字（半角スペースを除く）を入力すると、入力時にエラーになります。データ型が「文字 (XX)」の場合、全角と半角、および全角と半角の混在した文字を入力できます。
- パスワード (GUI 画面の場合)  
入力できる対象をパスワードにします。この場合、入力した文字を「\*」で置き換えて表示させるなど、表示方法を設定できます。
- MCR  
入力できる対象を、MCR (ID カードリーダー) からのデータにします。

表 A-10 表示書式と入力形式・格納形式 (日付)

日付ダイアログの表示書式 (表示書式は実際の日付に関係なく固定です)	入力形式 / 格納形式
1997 / 8 / 7	YYYYMMDD
1997 / 08 / 07	
1997- 8- 7	
1997-08-07	
1997. 8. 7	
1997.08.07	
1997 8 7	
19970807	
97 / 8 / 7	YYMMDD
97 / 08 / 07	
97- 8- 7	
97-08-07	
97. 8. 7	
97.08.07	
97 8 7	
970807	

日付ダイアログの表示書式 (表示書式は実際の日付に関係なく固定です)	入力形式 / 格納形式	
平成 9 年 8 月 7 日	GYMMDD	
平成 09 年 08 月 07 日		
平 9 年 8 月 7 日		
平 09 年 08 月 07 日		
年月 平成 9 年 8 月	GYMM	
年月 平成 09 年 08 月		
年月 平 9 年 8 月		
年月 平 09 年 08 月		
年月 1997 / 8	YYYYMM	
年月 1997 / 08		
年月 1997- 8		
年月 1997-08		
年月 1997. 8		
年月 1997.08		
年月 1997 8		
年月 199708		
年月 97 / 8		YYMM
年月 97 / 08		
年月 97- 8		
年月 97-08		
年月 97. 8		
年月 97.08		
年月 97 8		
年月 9708		
月日 8 / 7	MMDD	
月日 08 / 07		
月日 8- 7		
月日 08-07		
月日 8. 7		
月日 08.07		
月日 8 7		
月日 0807		



日付ダイアログの表示書式 (表示書式は実際の日付に関係なく固定です)	入力形式 / 格納形式
年 平 9 年	GYY
年 平 09 年	
年 9	YYYY
年 0009	
年 9	YY
年 09	
月 8	MM
月 08	
日 7	DD
日 07	

表 A-11 表示書式と入力形式・格納形式 (時刻)

時刻ダイアログの表示書式 (表示書式は実際の時刻に関係なく固定です)	入力形式 / 格納形式
21:08:07	HHMMSS
21: 8: 7	
21.08.07	
21. 8. 7	
PM09:08:07	HHMMSS (出力時刻だけ)
PM 9: 8: 7	
PM09.08.07	
PM 9. 8. 7	
09:08:07PM	
9: 8: 7PM	
09.08.07PM	
9. 8. 7PM	
21 時 08 分 07 秒	HHMMSS
21 時 8 分 7 秒	
210807	
21 8 7	
時分 21:08	HHMM
時分 21: 8	
時分 21.08	
時分 21. 8	

時刻ダイアログの表示書式 (表示書式は実際の時刻に関係なく固定です)	入力形式 / 格納形式
時分 PM09:08	HHMM (出力時刻だけ)
時分 PM 9: 8	
時分 PM09.08	
時分 PM 9. 8	
時分 09:08PM	
時分 9: 8PM	
時分 09.08PM	
時分 9. 8PM	
時分 21 時 08 分	HHMM
時分 21 時 8 分	
時分 2108	
時分 21 8	
分秒 08:07	MMSS
分秒 8: 7	
分秒 08.07	
分秒 8. 7	
分秒 08 分 07 秒	
分秒 8 分 7 秒	
分秒 0807	
分秒 8 7	
時 09 時	HH
時 9 時	
時 09	
時 9	
分 08 分	MM
分 8 分	
分 08	
分 8	
秒 07 秒	SS
秒 7 秒	
秒 07	
秒 7	

## 付録 A.10 修飾名と変更属性の標準値

画面の表示属性を AP から動的に変更する場合、ドロースettingsの表示属性の動的変更ダイアログで定義した修飾名を使用します。表示属性の動的変更ダイアログは定義するオブジェクトの系統別に、次のように分けられています。これらの系統と個々のオブジェクトの対応については、「2.4 GUI画面を構成するオブジェクト」を参照してください。

- キャラクタコントロール
- 候補選択コントロール
- コマンドコントロール
- ウィンドウ属性
- 位置属性
- 確定キー属性
- データ名

ここでは、XMAP3 が標準に提供する修飾名について説明します。

### (1) キャラクタコントロールの表示属性の標準値 (GUI / CUI 共通)

キャラクタコントロール (入出力テキスト・フィールド・ポップアップ・コンボボックス、ポップアップテキスト・フィールド) の表示属性の標準値を表 A-12、表 A-13 および表 A-14 に示します。これらの変更属性は、使用目的別に設定値を変更する必要があります。

また、使用目的が出力専用は、設定値が異なります。出力専用の際の表示属性の標準値について表 A-15 に示します。

表 A-12 キャラクタコントロールの表示属性の標準値 (数字系・日本語系・カナ系・英数字系キーエントリおよび日本語系選択エントリ)(1 / 3)

動的変更テーブルの データ名	修飾名	表示方法・ 入力可否属性	表示方法・ 非表示属性	遷移条件	詳細目的 <sup>2</sup> (モジュール チェック)
XMAP-IN-ATTR1	ER	入力可能	標準表示 <sup>1</sup>	変更なし	変更なし
XMAP-IN-ATTR2	RV	入力可能	標準表示 <sup>1</sup>	変更なし	変更なし
XMAP-IN-ATTR3	PT	入力不可	標準表示 <sup>1</sup>	変更なし	指定できない
XMAP-IN-ATTR4	NP	入力可能	標準表示 <sup>1</sup>	変更なし	変更なし
XMAP-IN-BK	BK <sup>3</sup>	変更なし	変更なし	指定できない	指定できない
XMAP-IN-GR	GR <sup>3</sup>	変更なし	変更なし	指定できない	指定できない
XMAP-IN-BL	BL <sup>3</sup>	変更なし	変更なし	指定できない	指定できない
XMAP-IN-LR	LR <sup>3</sup>	変更なし	変更なし	指定できない	指定できない
XMAP-IN-LB	LB <sup>3</sup>	変更なし	変更なし	指定できない	指定できない

動的変更テーブルの データ名	修飾名	表示方法・ 入力可否属性	表示方法・ 非表示属性	遷移条件	詳細目的 <sup>2</sup> (モジュラス チェック)
XMAP-IN-LG	LG <sup>3</sup>	変更なし	変更なし	指定できない	指定できない
XMAP-IN-LY	LY <sup>3</sup>	変更なし	変更なし	指定できない	指定できない
XMAP-IN-IV	IV <sup>3</sup>	入力不可	全体非表示	指定できない	指定できない
XMAP-IN-GO	GO <sup>3</sup>	入力不可	グレーアウト	指定できない	指定できない

注 1

非表示にしません。

注 2

「詳細目的」は適用オブジェクトグループが「数字系キーエントリ」の場合だけ有効です。

注 3

定義対象が CUI 画面の場合、これらの修飾名は「設定する修飾名」として選択されています。選択した場合、修飾名の属性はすべて「変更なし」となります。

表 A-13 キャラクタコントロールの表示属性の標準値(数字系・日本語系・カナ系・英数系キーエントリおよび日本語系選択エントリ)(2 / 3)

動的変更テーブルの データ名	修飾名	反転表示	背景色 <sup>1</sup>	文字色	入力済み
XMAP-IN-ATTR1	ER	変更なし	変更なし	赤	変更なし
XMAP-IN-ATTR2	RV	反転する	変更なし	黒	変更なし
XMAP-IN-ATTR3	PT	変更なし	変更なし	黒	指定できない
XMAP-IN-ATTR4	NP	変更なし	変更なし	黒	変更なし
XMAP-IN-BK	BK <sup>2</sup>	変更なし	変更なし	黒	指定できない
XMAP-IN-GR	GR <sup>2</sup>	変更なし	変更なし	緑	指定できない
XMAP-IN-BL	BL <sup>2</sup>	変更なし	変更なし	青	指定できない
XMAP-IN-LR	LR <sup>2</sup>	変更なし	ライトレッド	変更なし	指定できない
XMAP-IN-LB	LB <sup>2</sup>	変更なし	ライトブルー	変更なし	指定できない
XMAP-IN-LG	LG <sup>2</sup>	変更なし	ライトグリーン	変更なし	指定できない
XMAP-IN-LY	LY <sup>2</sup>	変更なし	ライトイエロー	変更なし	指定できない
XMAP-IN-IV	IV <sup>2</sup>	変更なし	変更なし	変更なし	指定できない
XMAP-IN-GO	GO <sup>2</sup>	変更なし	変更なし	変更なし	指定できない

## 注 1

CUI 画面では指定できません。

## 注 2

定義対象が CUI 画面の場合、これらの修飾名は「設定する修飾名」として選択されていません。選択した場合、修飾名の属性はすべて「変更なし」となります。

表 A-14 キャラクタコントロールの表示属性の標準値（数字系・日本語系・カナ系・英数系キーエントリおよび日本語系選択エントリ）(3 / 3)

動的変更テーブルのデータ名	修飾名	自動送信	入力必須	ワンタッチクリア <sup>1</sup>
XMAP-IN-ATTR1	ER	変更なし	変更なし	変更なし
XMAP-IN-ATTR2	RV	変更なし	変更なし	変更なし
XMAP-IN-ATTR3	PT	指定できない	指定できない	指定できない
XMAP-IN-ATTR4	NP	変更なし	変更なし	変更なし
XMAP-IN-BK	BK <sup>2</sup>	指定できない	指定できない	指定できない
XMAP-IN-GR	GR <sup>2</sup>			
XMAP-IN-BL	BL <sup>2</sup>			
XMAP-IN-LR	LR <sup>2</sup>			
XMAP-IN-LB	LB <sup>2</sup>			
XMAP-IN-LG	LG <sup>2</sup>			
XMAP-IN-LY	LY <sup>2</sup>			
XMAP-IN-IV	IV <sup>2</sup>			
XMAP-IN-GO	GO <sup>2</sup>			

## 注 1

CUI 画面では指定できません。

## 注 2

定義対象が CUI 画面の場合、これらの修飾名は「設定する修飾名」として選択されていません。選択した場合、修飾名の属性はすべて「変更なし」となります。

表 A-15 キャラクタコントロールの表示属性の標準値（ラベル）

動的変更テーブルのデータ名	修飾名	表示方法・非表示属性	反転表示	文字色
XMAP-OUT-ATTR1	RE	標準表示 <sup>1</sup>	変更なし	赤
XMAP-OUT-ATTR2	GR	標準表示 <sup>1</sup>	変更なし	緑
XMAP-OUT-ATTR3	BL	標準表示 <sup>1</sup>	変更なし	青
XMAP-OUT-ATTR4	YE	標準表示 <sup>1</sup>	変更なし	黄

動的変更テーブルのデータ名	修飾名	表示方法・ 非表示属性	反転表示	文字色
XMAP-OUT-ATTR5	CY	標準表示 <sup>1</sup>	変更なし	空
XMAP-OUT-ATTR6	MA	標準表示 <sup>1</sup>	変更なし	紫
XMAP-OUT-ATTR7	WH	標準表示 <sup>1</sup>	変更なし	白
XMAP-OUT-BK	BK <sup>2</sup>	変更なし	変更なし	黒
XMAP-OUT-DR	DR <sup>2</sup>	変更なし	変更なし	ダークレッド
XMAP-OUT-DB	DB <sup>2</sup>	変更なし	変更なし	ダークブルー
XMAP-OUT-DG	DG <sup>2</sup>	変更なし	変更なし	ダークグリーン
XMAP-OUT-DY	DY <sup>2</sup>	変更なし	変更なし	ダークイエロー
XMAP-OUT-IV	IV <sup>2</sup>	全体非表示	変更なし	変更なし
XMAP-OUT-GO	GO <sup>2</sup>	グレーアウト表示	変更なし	変更なし

注 1

非表示にしません。

注 2

定義対象が CUI 画面の場合、これらの修飾名は「設定する修飾名」として選択されていません。選択した場合、修飾名の属性はすべて「変更なし」となります。

## (2) 候補選択コントロールの表示属性の標準値 (GUI)

ラジオボタン、チェックボタン、リスト、トグルフィールドの表示属性の標準値を次の表に示します。

表 A-16 候補選択コントロールの表示属性の標準値

動的変更テーブル のデータ名	修飾名	表示方法 (リスト項目以外)	不活性 (選択できない状態にする)	フォーカス設定 (フォーカスをボタン/リストに位置づける)	選択済み にする
XMAP-BUTTON-SEL1	PT	標準表示	不活性	設定しない	選択前
XMAP-BUTTON-SEL2	ON	標準表示	活性	設定しない	選択済み
XMAP-BUTTON-SEL3	OF	標準表示	活性	設定しない	選択前
XMAP-BUTTON-SEL4	ST	標準表示	活性	設定する	選択前
XMAP-BUTTON-SEL5	IV	全体非表示	指定できない	設定しない	選択前

注

トグルフィールド以外です。

### (3) コマンドコントロールの表示属性の標準値 (GUI)

プッシュボタン、メニューバーの表示属性の標準値を次の表に示します。

表 A-17 コマンドコントロールの表示属性の標準値

動的変更テーブル のデータ名	修飾名	表示方法 (プッシュボタン だけ)	不活性 (選択できない 状態にする)	フォーカス設定 (フォーカスをボタンに 位置づける)
XMAP-BUTTON-SEL1	PT	標準表示	不活性	設定しない
XMAP-BUTTON-SEL2	ON	標準表示	活性	設定しない
XMAP-BUTTON-SEL3	OF	標準表示	活性	設定しない
XMAP-BUTTON-SEL4	ST	標準表示	活性	設定する
XMAP-BUTTON-SEL5	IV	全体非表示	指定できない	設定する

### (4) ウィンドウ属性の表示属性の標準値 (GUI / CUI 共通)

プッシュボタン、メニューバーの表示属性の標準値を次の表に示します。

表 A-18 ウィンドウ属性の表示属性の標準値

動的変更テーブル のデータ名	修飾名	表示形態	入力・選択状態 の扱い	キーボードのロッ ク状態を解除する	ウィンドウを表 示した時にアラ ームを鳴らす
XMAP-CNTRL1	WR	一部上書	状態を維持	解除する	鳴らさない
XMAP-CNTRL2	EW	全面書換	初期状態	解除する	鳴らさない
XMAP-CNTRL3	CL	一部上書	初期状態	解除する	鳴らさない
XMAP-CNTRL4	BL	一部上書	状態を維持	解除する	鳴らす

### (5) 位置属性の表示属性の標準値 (GUI)

表 A-19 位置属性の表示属性の標準値

動的変更テーブルのデータ名	修飾名	XY 位置	Z 位置
XMAP-WINDOW-RT	RT	右上	標準表示
XMAP-WINDOW-RD	RD	右下	標準表示
XMAP-WINDOW-LT	LT	左上	標準表示
XMAP-WINDOW-LD	LD	左下	標準表示
XMAP-WINDOW-IV	IV	変更なし	一時非表示
XMAP-WINDOW-FR	FR	中央	手前に表示
XMAP-WINDOW-VC	VC	中央	標準表示

(6) 確定キー属性の表示属性の標準値 (GUI)

表 A-20 確定キー属性の表示属性の標準値

動的変更テーブルのデータ名	修飾名	キー押下無効にするキー
XMAP-EVENT-AL	AL	なし
XMAP-EVENT-PC	PC	PF13 ~ PF84, PA1 ~ PA3, スクリーン消去

(7) データ名の表示属性の標準値 (GUI / CUI 共通)

表 A-21 データ名の表示属性の標準値

属性種別	データ名	論理マップ中でのマップ名
ウィンドウ属性の動的変更	マップ名 - CNTRL	マップ名 - CNTRLO
位置属性の動的変更	マップ名 - WINDOW	マップ名 - WINDOWW
確定キー属性の動的変更のデータ名	マップ名 - INC	マップ名 - INCO



## 付録 B XMAP3 で提供している定義パターン

XMAP3 では、用途に応じた画面サイズ、レイアウトのパターンを用意しています。画面のサイズを統一するために使うことをお勧めします。

定義パターンは、新規作成ダイアログの「パターンの選択」で指定できます。

### 付録 B.1 GUI 画面の定義パターン

GUI 画面（一次ウィンドウ）の定義パターンを表 B-1 に、GUI 画面（二次ウィンドウ）の定義パターンを表 B-2 に示します。

表 B-1 GUI 画面（一次ウィンドウ）の定義パターン

画面の種類	項番	定義パターン	サイズ（行×列）
3D	1	中汎用ウィンドウ（640 × 480 用）	47 × 91
	2	大汎用ウィンドウ（1,024 × 768 用）	79 × 145
	3	表（出力）	48 × 91
	4	表（入出力）	48 × 91
	5	一段ボタンメニュー	48 × 91
	6	二段ボタンメニュー	48 × 91
	7	一段ラジオボタンメニュー	48 × 91
	8	二段ラジオボタンメニュー	48 × 91
	9	ログインウィンドウ	30 × 57
2D	1	中汎用ウィンドウ（640 × 480 用）(2D)	48 × 80
	2	大汎用ウィンドウ（1,024 × 768 用）(2D)	53 × 85
	3	表（出力）(2D)	48 × 80
	4	表（入出力）(2D)	48 × 80
	5	一段ボタンメニュー（2D）	48 × 80
	6	二段ボタンメニュー（2D）	48 × 80
	7	一段ラジオボタンメニュー（2D）	48 × 80
	8	二段ラジオボタンメニュー（2D）	48 × 80
	9	ログインウィンドウ（2D）	30 × 50
フリー			48 × 80

注

この表の行数は半行単位です。

表 B-2 GUI 画面 (二次ウィンドウ) の定義パターン

画面の種類	項番	定義パターン名	サイズ (行×列)
3D	1	中ダイアログ ボタン下	38 × 68
	2	中ダイアログ ボタン右	38 × 68
	3	小ダイアログ ボタン下	28 × 46
	4	小ダイアログ ボタン右	28 × 46
	5	大ダイアログ ボタン下	48 × 91
	6	大ダイアログ ボタン右	48 × 91
	7	インフォメーションダイアログ	18 × 60
	8	クエスチョンダイアログ	18 × 60
	9	ワーニングダイアログ	18 × 60
	10	エラーダイアログ	18 × 60
	11	一段リストボックス (単一選択)	30 × 34
	12	一段リストボックス (複数選択)	30 × 34
	13	二段リストボックス (単一選択)	48 × 34
	14	二段リストボックス (複数選択)	48 × 34
2D	1	中ダイアログ ボタン下 (2D)	38 × 60
	2	中ダイアログ ボタン右 (2D)	38 × 60
	3	小ダイアログ ボタン下 (2D)	28 × 40
	4	小ダイアログ ボタン右 (2D)	28 × 40
	5	大ダイアログ ボタン下 (2D)	48 × 80
	6	大ダイアログ ボタン右 (2D)	46 × 80
	7	インフォメーションダイアログ (2D)	18 × 53
	8	クエスチョンダイアログ (2D)	18 × 53
	9	ワーニングダイアログ (2D)	18 × 53
	10	エラーダイアログ (2D)	18 × 53
	11	一段リストボックス (単一選択) (2D)	30 × 30
	12	一段リストボックス (複数選択) (2D)	30 × 30
	13	二段リストボックス (単一選択) (2D)	48 × 30
	14	二段リストボックス (複数選択) (2D)	48 × 30
フリー			48 × 80

## 注

この表の行数は半行単位です。

## 付録 B.2 CUI 画面の定義パターン

CUI 画面の定義パターンを次の表に示します。

表 B-3 CUI 画面の定義パターン

項番	定義パターン	サイズ (行×列)
1	汎用ウィンドウ	24 × 80
2	表	24 × 80
3	ファンクションメニュー	24 × 80
4	セレクションメニュー	24 × 80
5	ログインウィンドウ	24 × 80
6	フリー	24 × 80

注

この表の行数は半行単位です。

## 付録 B.3 XMAP3 で提供しているプログラムのパターン

### (1) 画面の AP パターン

XMAP3 が提供している AP パターンのうち、画面の AP のコーディングに利用するものについて説明します。

(a) メニュー画面

AP パターン名 (適用言語)	内容	使用ケース	対応する定義 サンプル
BTMENU01 (COBOL) (C 言語)	<p>ログイン画面からメニュー画面を起動させます。プッシュボタンで選択する多目的メニューです。ボタン、ファンクションキー、またはテンキー選択に対応して AP を起動します。</p> <ul style="list-style-type: none"> <li>ログイン画面は必要に応じて削除してください。</li> <li>実行ファイル(.EXE)を CALL 文で呼び出しています。内部ルーチン用に変更することもできます。</li> </ul>	<ul style="list-style-type: none"> <li>すべての業務をまとめた初期メニューとしての使用を想定しています。</li> <li>内部ルーチン用に変更して、サブメニューとしても使用できます。</li> </ul>	<p>ログインウィンドウ (GPASW1), 一段ボタンメニュー (BMENU1) との併用を想定しています。</p>
SLMENU01 (COBOL) (C 言語)	<p>ログイン画面からメニュー画面を起動します。ラジオボタン、またはテンキーで選択する初期メニューです。選択した番号に対応して AP を起動します。</p> <ul style="list-style-type: none"> <li>ログイン画面は必要に応じて削除してください。</li> <li>実行ファイル(.EXE)を CALL 文で呼び出しています。内部ルーチン用に変更することもできます。</li> </ul>	<ul style="list-style-type: none"> <li>すべての業務をまとめた初期メニューとしての使用を想定しています。</li> <li>内部ルーチン用に変更して、サブメニューとしても使用できます。</li> </ul>	<p>ログインウィンドウ (GPASW1), 一段ラジオボタンメニュー (SMENU1) との併用を想定しています。</p>

## (b) 画面入出力

AP パターン名 (適用言語)	内容	使用ケース	対応する定義 サンプル
GENDSP01 (COBOL) (C 言語)	汎用画面入出力プログラムです。一つの画面に対応した画面入出力処理をします。 <ul style="list-style-type: none"> <li>1画面だけの表示を想定しています。追加画面は、AP 部品 NXTDSP01 を追加します。</li> <li>入力後のチェック結果によって、AP 部品を随時挿入します。</li> <li>COBOL 用では、メインプログラムへは戻らないことを想定しています。メインプログラムへ戻す場合には、「STOP RUN」を変更して使用します。</li> </ul>	<ul style="list-style-type: none"> <li>メニュープログラムを使用したメインメニューから呼ばれることを想定しています。</li> </ul>	一般の画面に適用できます。
GENDSP02 (COBOL)	汎用画面入出力プログラムです。一つの画面に対応した画面入出力処理を、CALL インタフェースを利用して行います。 <ul style="list-style-type: none"> <li>1画面だけの表示を想定しています。追加画面は、同じく CALL インタフェースを使用している GENDSP03 を追加します。</li> <li>入力後のチェック結果によって、AP 部品を随時挿入します。</li> <li>メインプログラムへは戻らないことを想定しています。メインプログラムへ戻す場合には、「STOP RUN」を変更して使用します。</li> </ul>	<ul style="list-style-type: none"> <li>メニュープログラムを使用したメインメニューから呼ばれることを想定しています。</li> <li>COBOL の CALL インタフェースを使用して画面入出力を行うケースを想定しています。</li> </ul>	一般の画面に適用できます。
GENDSP03 (COBOL)	GENDSP02 から呼び出されるサブプログラムです。	<ul style="list-style-type: none"> <li>GENDSP02 との組み合わせを想定しています。</li> </ul>	一般の画面に適用できます。
GENFLD01 (COBOL) (C 言語)	<ul style="list-style-type: none"> <li>フィールド単位アクセス用入出力プログラムです。フォーカス・カーソルが位置づいているフィールドだけ入力できます。</li> <li>AP では、入力データをチェックし、対応する処理を行います。</li> <li>チェックで OK であれば、次に入力したいフィールドにフォーカス・カーソルを位置づけて画面を再表示します。</li> </ul>	<ul style="list-style-type: none"> <li>フィールド単位に AP に返して、入力データをチェックすることを想定しています。</li> <li>COBOL の「WINDOW SECTION」で実現していた業務を GUI に移行する際の AP を想定しています。</li> </ul>	フィールド単位画面 (FIE1GC または FIE1GH) の使用を想定していません。

AP パターン名 (適用言語)	内容	使用ケース	対応する定義 サンプル
GENEVN01 (COBOL) (C 言語)	<p>イベント単位アクセス用入出力プログラムです。画面上のどこからでも入力でき、AP にイベントを返します。</p> <ul style="list-style-type: none"> <li>AP では、イベント通知コードの領域に、入力したフィールドのフォーカス・カーソル定数が返ってきます。</li> <li>イベントごとの処理を行うセクションを設けて、各フィールドの処理を行います。</li> </ul>	<ul style="list-style-type: none"> <li>複数のトグルフィールドから一つだけを選択して処理を行うことを想定しています。</li> </ul>	<p>イベント単位画面 (EVE1GC または EVE1GH) の使用を想定しています。</p>
DSPPR01 (COBOL) (C 言語)	<p>GENDSP01 と GENREP01 とを合わせた汎用画面入出力・帳票印刷プログラムです。画面入出力を行ったあと、帳票を出力します。</p>	<ul style="list-style-type: none"> <li>メニュープログラムを使用したメインメニューから呼ばれることを想定しています。</li> </ul>	<p>一般の画面・帳票に適用できます。</p>

## (2) 画面の AP 部品

標準提供 AP パターンのうち、画面の AP のコーディングに利用する AP 部品を示します。

AP パターン名 (適用言語)	内容	使用ケース	対応する定義サンプル
NXTDSP01 (COBOL) (C 言語)	<p>画面入出力処理の追加コーディングです。</p>	<p>GENDSP01, DSPPR01 への挿入を想定しています。</p>	<p>一般の画面に適用できます。</p>
ATRCRS01 (COBOL) (C 言語)	<p>特定の入力 / 入出力テキストまたはフィールドの文字色を変更して、フォーカスまたはカーソルを該当項目に設定します。該当項目以外は、もとの画面の状態のままとします。</p>	<p>GENDSP01, DSPPR01, BTMENU01, SLMENU01, NXTDSP01 への挿入を想定しています。</p>	<p>一般の画面に適用できます。</p>
CLRINP01 (COBOL) (C 言語)	<p>表示画面の入力項目だけをヌルクリアして初期状態にします。データエントリ画面での利用を想定しています。</p>	<p>GENDSP01, DSPPR01, NXTDSP01 への挿入を想定しています。</p>	<p>一般の画面に適用できます。</p>
FLDHLP01 (COBOL) (C 言語)	<p>ファンクションキーやブッシュボタンが押されたときに、カーソルのある項目に対応するヘルプ画面を表示します。フィールドごとのヘルプ表示を想定しています。</p>	<p>GENDSP01, DSPPR01, BTMENU01, SLMENU01, NXTDSP01 への挿入を想定しています。</p>	<p>小ダイアログ、中ダイアログ (HLP01) との併用を想定しています。</p>

AP パターン名 (適用言語)	内容	使用ケース	対応する定義サンプル
MODATR01 (COBOL) (C 言語)	<p>特定の入力 / 入出力 / 出力テキストまたはフィールドの文字色を変更します。</p> <ul style="list-style-type: none"> <li>入力チェック後のエラー表示用として使用します。</li> <li>表示データ代入処理とあわせて使用することで、初めから文字色を変更したい場合にも応用できます。</li> </ul>	<p>GENDSP01 , DSPPRT01 , BTMENU01 , SLMENU01 , NXTDSP01 , GENFLD01 への挿入を想定しています。</p>	<p>一般の画面に適用できます。</p>
PATWRT01 (COBOL) (C 言語)	<p>表示されている画面に対して、ある 1 項目だけを書き換えます。ほかの項目の状態はそのままとします。</p> <ul style="list-style-type: none"> <li>エラーメッセージなどをメッセージフィールドに表示する場合に使用します。</li> <li>1 項目だけでなく、複数項目またはすべての項目を書き換えるように変更することで、固定部分をそのままにして可変部分をすべて書き換えて表示する場合にも応用できます。</li> </ul>	<p>GENDSP01 , DSPPRT01 , BTMENU01 , SLMENU01 , NXTDSP01 , GENFLD01 , GENEVN01 への挿入を想定しています。</p>	<p>一般の画面に適用できます。</p>
SETCRS01 (COBOL) (C 言語)	<p>表示されている画面に対して、フォーカスやカーソル位置を設定し直します。ほかの項目の値は変更しません。</p>	<p>GENDSP01 , DSPPRT01 , BTMENU01 , SLMENU01 , NXTDSP01 への挿入を想定しています。</p>	<p>一般の画面に適用できます。</p>
BTPROT01 (COBOL) (C 言語)	<p>表示されている画面に対して、ボタンを不活性状態に変更します。</p>	<p>BTMENU01 , SLMENU01 , GENDSP01 , NXTDSP01 への挿入を想定しています。</p>	<p>一般の画面に適用できます。</p>

## 付録 C 日立以外のコンパイラを使用するときの注意

日立 COBOL 以外の製品を利用して、XMAP3 が用意する CALL インタフェースを使う場合の注意事項を説明します。

### 付録 C.1 他社 COBOL との仕様比較

日立 COBOL と他社 COBOL との仕様比較を次の表に示します。サンプルコーディングなどを変更する際に、参考にしてください。

なお、このマニュアルは日立 COBOL を前提に書かれていますので、必要に応じて、置き換えてお読みください。

表 C-1 日立 COBOL と他社 COBOL との仕様比較

機能		日立 COBOL	Merant Micro Focus COBOL	富士通 COBOL85
CALL インタフェース		CALL 関数名 USING ...	CALL special_name 関数名 USING ...	CALL 関数名 USING ...
USING		<ul style="list-style-type: none"> <li>一意名</li> <li>BY VALUE 実数 / 一意名</li> </ul>	<ul style="list-style-type: none"> <li>一意名</li> <li>BY VALUE 実数 / 一意名</li> </ul>	<ul style="list-style-type: none"> <li>一意名</li> <li>BY VALUE 一意名</li> </ul>
領域定義	日本語定数	N 日本語		
	16 進数	X16 進定数		
	数値設定	COMP, COMP-5		
	数値設定 (C 互換)	S9(4) COMP-5 (short) S9(8) COMP-5 (long)		
その他の言語仕様	予約語の位置	8 カラム以降	8 カラム以降	12 カラム以降
	PERFORM 文	PERFORM 一意名 : 一意名 SECTION (有効範囲は END まで) : 一意名 -END.	PERFORM 一意名 : 一意名 SECTION (有効範囲は END まで) : 一意名 -END.	PERFORM 一意名 : 一意名 SECTION (有効範囲は次の SECTION まで)

### 付録 C.2 C 言語を使用するときの注意

C 言語を使用して、XMAP3 で画面入出力の AP を開発および実行する場合のサポート状況について次の表に示します。



表 C-2 Windows 95 , Windows 98 , および Windows NT 4.0 の場合

C 言語製品名 (会社名)	サポート状況
Visual C++ 4.X (マイクロソフト社)	
Visual C++ 5.X (マイクロソフト社)	
Visual C++ 6.X (マイクロソフト社)	
Visual C++ .NET (マイクロソフト社)	

(凡例)

: 使用できる。

: 実行だけできる。

表 C-3 Windows Me , Windows 2000 , Windows XP , Windows Server 2003 , および Windows Server 2003 x64 の場合

C 言語製品名 (会社名)	サポート状況
Visual C++ 4.X (マイクロソフト社)	
Visual C++ 5.X (マイクロソフト社)	
Visual C++ 6.X (マイクロソフト社)	
Visual C++ .NET (マイクロソフト社)	

(凡例)

: 使用できる。

: 実行だけできる。

## 付録 C.3 Visual Basic , Delphi , Access を使用するときの注意

Visual Basic , Delphi または Access を使用して , XMAP3 で画面入出力の AP を開発および実行する場合に , XMAP3 でサポートしているバージョンを次に示します。

### Visual Basic

Visual Basic Version 6.0 以降

### Delphi

- Windows 95 , および Windows NT 4.0 の場合 : Delphi 3.1
- Windows 98 , Windows Me , および Windows 2000 の場合 : Delphi 5

### Access

- Windows XP の場合 : Access 97 以降
- Windows 95 , Windows 98 , Windows Me , Windows NT 4.0 , および Windows 2000 の場合 : Access For Windows 95 Version 7.00 以降

## 付録 D 画面定義の制限値

### 付録 D.1 オブジェクトの個数およびサイズ

オブジェクトの個数やサイズの制限値は、論理上の値です。性能上の制約や、マップサイズの範囲をあわせて考慮してください。

GUI画面で定義したオブジェクトの個数は、ドローの[表示] [オブジェクト数]を選択して、種別ごとに表示して確認できます。オブジェクト数を表示する操作の詳細については、ヘルプを参照してください。

表 D-1 画面定義の制限値

オブジェクト		制限値
マップサイズ	物理マップサイズ	32KB
	論理マップサイズ	
定義パターンの登録 部品の登録		定義対象の種別ごとに 32 個。 ただし、レイアウト部品は一次ウィンドウと二次ウィンドウの合計で 32 個。
ウィンドウ	個数	一次ウィンドウは 1 個 / 定義。 二次ウィンドウは 3 個 / 一次ウィンドウ。
	サイズ	画面の解像度に従う。 GUI画面の最大は、レイアウト領域のサイズで縦 96 ます × 横 160 ます (48 行 × 160 列)。 CUI画面は 24 行 × 80 列固定。
	タイトル	78 桁 (定義時)。表示上の文字数は画面サイズに従う。
	壁紙	ファイル (.bmp) のサイズで 31,000 バイト。
メニューバーのメニュー項目	個数	24 項目 / メニューバー。 メニュー項目の合計は 96 項目 / 定義。
	サイズ	ラベル長 64 桁。
プルダウンメニューの項目	個数	24 項目 / メニューバーの項目。 メニュー項目の合計は 96 項目 / 定義。
	サイズ	ラベル長 64 桁。
カスケードメニューの項目	個数	24 項目 / プルダウンメニューの項目。 メニュー項目の合計は 96 項目 / 定義。
	サイズ	ラベル長 64 桁。
コメントの記述	画面属性	画面名 : 40 文字 (半角) <sup>1</sup> 。 コメント : 56 文字 (半角) × 4 行 <sup>2</sup> 。
	イベント通知コード	文字数 : 40 文字 (半角) <sup>2</sup> 。
	オブジェクト	文字数 : 40 文字 (半角)。

オブジェクト		制限値	
ボックス系オブジェクト		個数：各種の総和で 255 項目 / 定義。	
テキスト	テキスト全般	サイズ：レイアウト領域に収まる範囲 <sup>3</sup> 。	
	固定テキスト	レイアウト領域に収まる範囲。 ただし、文字サイズが標準文字サイズより小さい場合、160 文字（半角）以内。	
	使用目的が「MCR」の入出力テキスト・フィールド	個数：1 個 / 定義。 文字数：カードの文字列は 64 文字（半角）以内。	
プッシュボタン	ボタン	個数	各種別に 64 個 / 定義。
		サイズ	ウィンドウサイズに従う。
	ボタン	個数	24 ボタン / ボタンボックス。
		サイズ	ラベル長 64 桁。
	ボタンに貼り付けるグラフィック	サイズ	31,000 バイト。
ラジオボタン	ボックス	サイズ	ウィンドウサイズに従う。
チェックボタン	ボタン	サイズ	ラベル長 64 桁 <sup>3</sup> 。
リストボックス	ボックス	サイズ	プレーンサイズ以下でレイアウト領域に収まる範囲。
		プレーン	個数
		サイズ	ラベル長 255 桁 / リストボックス。 項目数 254 個 / リストボックス。 ただし、組み合わせの制限はマップサイズの制限に従う。
フィールドボックス	ボックス	個数	4 ボックス / 定義。
		サイズ	プレーンサイズ以下でレイアウト領域に収まる範囲。
	プレーン	個数	1 プレーン / ボックス。
		サイズ	127 行 × 255 桁。
グループボックス	サイズ	レイアウト領域に収まる範囲。	
	タイトル	長さは「グループボックスの横サイズ - 2 ます」、または 62 文字（半角）以内。	
ポップアップ	ボックス	サイズ：レイアウト領域に収まる範囲。	
	メニュー項目（ファイルを使用しない場合）	個数：2 ~ 128 項目 / 1 メニュー（プレーン）。 画面上の表示行数：レイアウト領域に収まる範囲。 サイズ：ラベル長、選択ラベル長、通知コード長とも 64 桁 <sup>3</sup> 以内。 アクセスキー：は半角 1 ~ 4 文字の英大文字（A ~ Z）または数字（0 ~ 9）。	

オブジェクト		制限値
	メニュー項目（ポップアップメニューファイル使用）	32,000 項目 / ファイル。 256 項目 / 1 階層。 サイズ：ラベル長，選択ラベル長，通知コード長とも 64 桁 <sup>3</sup> 以内。 アクセスキー：は半角 1 ~ 4 文字の英大文字（A ~ Z）または数字（0 ~ 9）。
	メニューの階層	8 階層。
コンボボックス <sup>4</sup>	ボックス	66 桁以内。
	メニュー項目	個数：2 ~ 128 項目 / 1 メニュー（プレーン）。 画面上の表示行数：レイアウト領域に収まる範囲。 サイズ：ラベル長，通知コード長とも 64 桁 <sup>3</sup> 以内。
セパレータ	個数	2,000 本（推奨）。
	サイズ	レイアウト領域に収まる範囲。
アイコン	個数	1 個 / 定義。
	サイズ	変更できない。
固定グラフィック 出力グラフィック	個数	固定 / 出力それぞれ 100 個（推奨）。
	サイズ	固定グラフィック：31,000 バイト / オブジェクト。 出力グラフィック：100KB / オブジェクト（推奨）。
フィールドボックス内		各オブジェクトは，プレーン領域に収まるサイズで配置できる。フレーム，塗りつぶしフィールド，けい線はほかのオブジェクトに重ねて配置できる。スクロール禁止領域には，入力 / 選択系のオブジェクトは配置できない。また，スクロール禁止領域とスクロール領域とにわたる配置もできない。
フィールド系オブジェクト		個数：総和で 2,000 個 / フィールドボックス（推奨）。 また，CUI 画面では 2,048 個 / 定義（推奨）。
固定フィールド	サイズ	GUI 画面の最大は，半角で 2,048 文字以内。 CUI 画面の最大は，レイアウト領域に収まる範囲。
出力フィールド 入出力フィールド	サイズ	GUI 画面の最大は，半角で 13,679 桁 <sup>5</sup> 以内（ただし，初期値を指定する場合は，2,048 桁以内）。 CUI 画面の最大は，レイアウト領域に収まる範囲。
予約フィールド	サイズ	255 桁以内。
ポップアップフィールド	個数	個数：2 ~ 128 項目 / 1 メニュー（プレーン）。 画面上の表示行数：レイアウト領域に収まる範囲。
	サイズ	ラベル長，通知コード長とも 64 桁以内 <sup>3</sup> 。
フレーム	個数	制限なし。
けい線，矩形	個数	2,000 本（推奨）。

- 注 1  
テスト支援での制限があります。詳細はマニュアル「XMAP3 開発・実行ガイド」を参照してください。
- 注 2  
ドキュメンテーション支援での制限があります。詳細はマニュアル「XMAP3 開発・実行ガイド」を参照してください。
- 注 3  
データ型に数字編集を指定した場合、入力 は 18 桁、出力 は 30 桁以内になります。
- 注 4  
大量の項目を持つコンボボックスおよびキー入力コンボボックスを、画面に大量に定義すると、生成されるウィンドウ数が Windows の上限値を超えてしまい、画面を表示する際に AP にエラーが通知されることがあります。この場合は、定義するコンボボックスの個数またはメニュー項目数を減らすようにしてください。
- 注 5  
使用目的に「英数」を指定した場合の値です。指定できる値は、定義条件に応じて異なります。

## 付録 D.2 使用できるグラフィックデータ

### (1) グラフィックオブジェクトとデータ形式の関係

オブジェクトごとに、使用できるデータ形式を示します。

オブジェクト	ビットマップファイル (.BMP)	メタファイル (.WMF)	拡張メタファイル (.EMF)	JPEG ファイル <sup>2</sup> (.JPG および .JPEG)	備考
壁紙		x	x	x	カラー /
固定グラフィック	1				モノクロ対応
出力グラフィック					
ボタン		x	x	x	

(凡例)

- : 使用できる。
- x : 使用できない。

- 注 1  
固定グラフィックにビットマップファイルを使用する場合、Windows 形式だけを使用できません。
- 注 2  
使用できる JPEG 画像の形式は JFIF 準拠 ベースライン形式 (基本 DCT 方式の JFIF01-01, および JFIF01-02) だけです。その他の形式 (プログレッシブ方式など) は使用できません。

## (2) グラフィックの色

画面で使用するグラフィックデータの色は次のとおりです。ただし、ディスプレイがそれぞれの色数を表現できることが前提となります。

- 2色 (1ビットモノクロ)
- 16色 (4ビット)
- 256色 (8ビット)
- 約6万色 (16ビット HighColor)
- 約1,600万色 (24ビット TrueColor)

使用する色が多くなると、グラフィックデータのサイズも大きくなります。通常は256色で運用することをお勧めします。ただし、JPEG画像については1,600万色のグラフィックデータとして扱われますのでJPEG画像を使用する場合はディスプレイを6万色以上に設定してください。

また、ディスプレイで表示できる色が256色の場合、瞬間的に色化け（ほかのAPの壁紙の色が異なる色に変わる）が発生することがあります。そのため、色はWindowsの基本20色とXMAP3の文字色、背景色に指定した色の範囲で作成することをお勧めします。ただし、表示するディスプレイが6万色以上であれば、色化けは発生しません。

なお、色数が少ないディスプレイに、ディスプレイより多い色数を使用したグラフィックデータを表示しても、エラーにはなりません。ただし、表示できる色数が少ないので、意図した内容が表示できません。

ディスプレイで表示できる色と作成したグラフィックデータの色数の関係を示します。

ディスプレイで表示 できる色数	作成したグラフィックデータの色数				
	2	16	256	3万 / 6万	1,600万色
2		1	×	×	×
16			×	×	×
256				×	×
3万 / 6万					2
1,600万色					

### (凡例)

- : 表示できる。
- △ : ある程度表示できる。
- ×

注 1  
一般的にモノクロのディスプレイが階調表示します。

注 2  
表示は劣化しますが、見た目上では劣化の判別ができない程度です。

## 付録 E XMAP3 のリターンコード・詳細コード

リターンコード、詳細コードとエラー内容を次の表に示します。16進表示のコードは、日立 COBOL のコンソール画面に表示されるエラーコードに対応しています。日立 COBOL のテストデバッグなどで、メモリ中のリターンコードを直接参照する場合は、1バイト目と2バイト目が逆になりますので注意してください。表中で使用する記号の意味を次に示します。

(S) システムの処理

(P) プログラムの処理

なお、次の表のリターンコード詳細に示すエラーコードが、20480 ~ 20516 (16進数では (5000)<sub>16</sub> ~ (5024)<sub>16</sub>) の場合、すでにオープン状態であっても強制的にクローズ要求されたものとして処理します。

クローズ処理によって、表示されている画面が消えたり、その後の要求が 24577 エラー (16進数では (6001)<sub>16</sub> エラー) となったりする場合がありますので、エラー原因を解決して再度アプリケーションを起動してください。

リターンコードと詳細コードを次の表に示します。

表 E-1 リターンコードと詳細コード

詳細コード	内容	リターンコード
1036 (040C) <sub>16</sub>	定義している予約項目名が、OpenTP1 で提供している予約項目名と一致しない。 (S) 処理を続行する。 (P) 予約項目に定義している予約項目名が OpenTP1 で提供している予約項目名と一致することを確認して、物理マップを再作成後、アプリケーションを再実行する。	4
1044 (0414) <sub>16</sub>	定義されたイベント通知コードと対応しない確定キーが入力された。 (S) 処理を続行する。 (P) 入力された確定キーに対応するイベント通知コードを定義し、物理マップを再作成して AP を再実行する。	4
1060 (0424) <sub>16</sub>	カーソル位置が不正である。AP で設定するカーソル位置の値が画面上に存在しない。 (S) 処理を続行する。 (P) 画面範囲内のカーソル位置を設定するように AP を修正し、再コンパイル後、再実行する。位置を設定しないで領域をクリアする場合には (00) <sub>16</sub> を設定する。	4

詳細コード	内容	リターンコード
3072 (0C00) <sub>16</sub>	物理マップの内容が破壊されている。 原因として次のことが考えられる。 <ul style="list-style-type: none"> <li>• デバイス種別とマップが一致していない。例えば、ディスプレイを表示するときに AP で指定したマップがプリンタ用である。</li> <li>• マップ情報が破壊されている。</li> </ul> (S) 処理を終了する。 (P) 物理マップを再作成して、アプリケーションを再実行する。	8
3104 (0C20) <sub>16</sub>	実行環境の XMAP3 より上位のバージョンで作成した物理マップが使用された。 (S) 処理を終了する。 (P) 実行環境の XMAP3 のバージョンを、物理マップを作成した XMAP3 と同じバージョンにして、AP を再実行する。	8
4100 (1004) <sub>16</sub>	何らかの原因でデータ化けが発生したことにより、物理マップ中の入力物理マップ名情報が破壊されている。 (S) 処理を終了する。 (P) 物理マップを再作成して、アプリケーションを再実行する。メッセージとマップ名を正しく対応づける。	8
4104 (1008) <sub>16</sub>	何らかの原因でデータ化けが発生したことにより、物理マップ中の作成管理情報が破壊されている。 (S) 処理を終了する。 (P) 物理マップを再作成して、アプリケーションを再実行する。	8
5120 (1400) <sub>16</sub>	指定した物理マップが見つからない。または、AP で指定した仮想端末と指定した物理マップのデバイス種別が一致しない。 (S) 処理を終了する。 (P) <ul style="list-style-type: none"> <li>• 表示・印刷セットアップの「アプリケーション 1」タブの「マップ」で指定しているフォルダ、または AP を実行したカレントフォルダに、指定した物理マップ（拡張子は pmp）があることを確認し、AP を再実行する。</li> <li>• AP で指定した仮想端末名（例えば DSP001）に対して、物理マップのデバイス種別（例えばデバイス ID が ND/NC）が合っているか確認し、AP を修正後、再実行する。</li> </ul>	8
5124 (1404) <sub>16</sub>	物理マップのロード処理中に入出力エラーが発生した。 (S) 処理を終了する。 (P) 物理マップを再作成して、アプリケーションを再実行する。	8



詳細コード	内容	リターンコード
16388 (4004) <sub>16</sub>	<p>マッピング処理プログラムの実行環境が不正なため、エラーが発生した。原因としては、次のことが考えられる。</p> <ul style="list-style-type: none"> <li>・ インストールが正しくされていない。</li> <li>・ XMAP3 外字機能がインストールされている環境で、コード変換 SDK、コード変換ランタイム、日立コード変換・Server Runtime または日立コード変換・Client Runtime がインストールされていない。</li> </ul> <p>(S) 処理を終了する。 (P)</p> <ul style="list-style-type: none"> <li>・ 再インストールを行う。</li> <li>・ 実行時に必要な PP のインストールをする。</li> </ul>	8
16400 (4010) <sub>16</sub>	<p>何らかの原因でデータ化けが発生したことにより、物理マップ中のサイズ管理情報が破損している。</p> <p>(S) 処理を終了する。 (P) 物理マップを再作成して、AP を再実行する。</p>	8
16428 (402C) <sub>16</sub>	<p>同時に動作しているほかのアプリケーションなどの関係、またはリソースの物理的不足により、マッピング処理の実行時に必要なメモリが不足した。</p> <p>(S) 処理を終了する。 (P)</p> <ul style="list-style-type: none"> <li>・ ほかのアプリケーションを終了するか、PC を再起動する。</li> <li>・ PC 上のリソースを見直し、メモリの増設などを行う。</li> </ul>	8
16436 (4034) <sub>16</sub>	<p>仮想端末と物理マップのデバイス種別が一致していない。原因としては、次のことが考えられる。</p> <ul style="list-style-type: none"> <li>・ 画面用の仮想端末に帳票用の物理マップを指定した。</li> <li>・ 物理マップのマップ名称をエクスプローラなどで変更した。</li> <li>・ 物理マップが壊れている。</li> <li>・ XMAP3 インストールフォルダ下の ETC フォルダの X3MWCONF ファイルがない、または空ファイルになっている。</li> </ul> <p>(S) 処理を終了する。 (P)</p> <ul style="list-style-type: none"> <li>・ AP で指定している仮想端末と物理マップのデバイス種別が一致することを確認し、再コンパイル・再実行する。</li> <li>・ 物理マップを再作成して、AP を再実行する。</li> <li>・ X3MWCONF ファイルがないまたは空ファイルの場合は、アンインストール後に再インストールする。</li> </ul>	8
20480 (5000) <sub>16</sub>	<p>XMAP3 の定義ファイルの内容と C/S システム環境での通信関係の設定が不正である。</p> <p>(P) OS でのネットワーク構成との不整合が生じたことによる障害が考えられる。サーバ、全クライアントおよび OS をすべて終了し再起動する。</p>	8
20482 (5002) <sub>16</sub>	<p>二次ウィンドウが表示されていない状態で二次ウィンドウに対して入力要求を行ったためにエラーが発生した。AP 内での処理シーケンスが不正になっていることが考えられる。</p> <p>(P) 画面の送受信のシーケンスを見直す。</p>	8

詳細コード	内容	リターンコード
20484 (5004) <sub>16</sub>	<p>同時に動作しているほかのアプリケーションなどの関係、またはリソースの物理的不足により、表示サービスの実行時に必要なメモリが不足した。</p> <p>(P)</p> <ul style="list-style-type: none"> <li>• ほかのアプリケーションを終了するか、PC を再起動する。</li> <li>• PC 上のリソースを見直し、メモリの増設などを行う。</li> </ul>	8
20485 (5005) <sub>16</sub>	<p>C/S システム環境での通信処理で、回復不能エラーが発生した。同時に動作しているほかのアプリケーションなどの関係で、PC 上のリソース（主にメモリ）が不足したことが考えられる。また、LAN の回線上で障害、データ化けが発生したことが考えられる。</p> <p>(P) ほかのアプリケーションを終了するか、PC を再起動する。</p>	8
20487 (5007) <sub>16</sub>	<p>何らかの原因でデータ化けが発生したことにより、物理マップ中のバージョン情報が破壊されている。</p> <p>(P) 物理マップを再作成して、アプリケーションを再実行する。</p>	8
20488 (5008) <sub>16</sub>	<p>メインウィンドウ（一次ウィンドウ）が表示されていない状態で、サブウィンドウ（二次ウィンドウ）を表示すると発生する。一般的には AP 内での処理が不正になっていることが考えられる。</p> <p>(P) 表示するマップ名および画面の表示処理シーケンスを見直す。</p>	8
20489 (5009) <sub>16</sub>	<p>XMAP3 内部処理で何らかの異常が発生したことにより、表示サービスのプログラムがダウンした。</p> <p>(P) システム的な要因が原因である（システムリソースの異常・ほかのアプリケーションでの異常）ことが考えられる。PC を再起動・再実行する。</p>	8
20490 (500A) <sub>16</sub>	<p>画面属性の入力単位に「表示直後」を定義した画面で、「入力必須」または「選択必須」を定義したオブジェクトが存在する。</p> <p>(P) 画面属性およびオブジェクトに指定している属性を見直し、物理マップを再作成して、アプリケーションを再実行する。</p>	8
20493 (500D) <sub>16</sub>	<p>XMAP3 内部処理で何らかの異常が発生したため、サーバ内のプログラムに論理矛盾が起きた。</p> <p>(P) システム的な要因が原因である（システムリソースの異常・ほかのアプリケーションでの異常）ことが考えられる。PC を再起動・再実行する。</p>	8
20508 (501C) <sub>16</sub>	<p>TCP/IP による通信処理で何らかの障害が発生した。または、インストールの失敗により、表示に必要なモジュールが格納されていない。</p> <p>(P)</p> <ul style="list-style-type: none"> <li>• 1 回も動作していない場合は、TCP/IP 関連の設定環境を見直す。動作中に発生した場合は、相手 WS / PC の状態を確認し、AP を再実行する。</li> <li>• 既存のモジュールをアンインストールし、再度インストールを行う。</li> </ul>	8

詳細コード	内容	リターンコード
20514 (5022) <sub>16</sub>	<p>表示サービスが利用できない。</p> <ul style="list-style-type: none"> <li>• 出力する仮想端末の PC が起動していないため、XMAP3 が動作していない。</li> <li>• 出力する仮想端末関連の定義が不正なため、XMAP3 の表示・印刷サービスが起動されていない。</li> <li>• 仮想端末名ファイルと、サービス名ファイルの定義情報が不一致。</li> <li>• サービス名ファイルと、表示・印刷セットアップの定義情報が不一致。</li> </ul> <p>(P)</p> <ul style="list-style-type: none"> <li>• 出力先の PC を起動する。</li> <li>• C/S セットアップを実行し、表示・印刷セットアップで AP から指定している仮想端末名のサービス名称を一致させてから、AP を再実行する。</li> </ul>	8
20515 (5023) <sub>16</sub>	<p>C/S システム環境の場合、表示サービスが存在する環境の XMAP3 と、AP が動作している環境の XMAP3 のバージョンが不整合である。</p> <p>(P) それぞれインストールされている XMAP3 のバージョンを確認し、整合性のあるバージョンになるようにインストールし直す。基本的に XMAP3 間では同一バージョンを利用する。WS との C/S システム構成の場合、前提となる WS 側のバージョンを調査しておく。</p>	8
20516 (5024) <sub>16</sub>	<p>XMAP3 の出力処理中に Windows または XMAP3 内部で回復不能なエラーが発生した。</p> <p>(P) システム的な要因が原因（システムリソース不足、プリントマネージャの異常検知）であることが考えられる。PC を再起動後、再実行する。</p>	8
24576 (6000) <sub>16</sub>	<p>jsvwdrv 関数を実行時、指定したパラメタが不正である。または TRANSCEIVE 要求、RECEIVE 要求、jsvwdrv 関数実行時に誤った論理マップ名を指定しているか、リトルエンディアン/ビッグエンディアンが誤っている物理マップを指定している。</p> <p>(S) 処理を終了する。</p> <p>(P) AP 中で指定したパラメタが正しいかを見直し、再コンパイル後、再実行する。</p>	8
24577 (6001) <sub>16</sub>	<p>jsvwdrv 関数の発行順序が不正である（OPEN 要求を行わずにほかの要求を行った、または SEND 要求を行わずに RECEIVE 要求を行った）。</p> <p>(S) 処理を終了する。</p> <p>(P) AP 中の jsvwdrv 関数を実行している処理で発行順序が正しいかを見直し、再コンパイル後、再実行する。</p>	8

詳細コード	内容	リターンコード
24578 (6002) <sub>16</sub>	<p>仮想端末名ファイルの内容に下記の不正がある。</p> <ul style="list-style-type: none"> <li>• AP で設定している仮想端末名称が定義されていない。</li> <li>• 1 行の終わりに改行が設定されていない。</li> <li>• 1 行が 511 バイトを超えている。</li> <li>• ファイルサイズが 32,767 バイトを超えている。</li> <li>• 下記の各パラメタが形式外 仮想端末名：8 文字以内，先頭は英字，英数字を使用 デバイス：指定できる文字は固定 サーバホスト名称：32 文字以内 サーバサービス番号：5 文字以内，数字を使用， 1,024 ~ 65,535 の範囲 サービス名称：40 文字以内，半角文字を使用 環境定義ファイル名称：64 文字以内</li> </ul> <p>(S) 処理を終了する。 (P) AP で設定した仮想端末名を見直す，または仮想端末定義ファイルの設定内容を見直し，再実行する。</p>	8
24579 (6003) <sub>16</sub>	<p>ファイルシステム容量不足や，何らかの原因により，ログファイルのアクセス中にエラーが発生した。</p> <p>(S) 処理を終了する。 (P)</p> <ul style="list-style-type: none"> <li>• ほかのアプリケーションを終了するか，PC を再起動する。</li> <li>• ログファイルを出力するために指定したフォルダが書き込み可能であるかを確認する。</li> </ul>	8
24580 (6004) <sub>16</sub>	<p>仮想端末名ファイルのオープン処理，または物理マップロード中に必要なメモリが不足した。</p> <p>(S) 処理を終了する。 (P) ほかのアプリケーションを終了するか，PC を再起動する。</p>	8
24581 (6005) <sub>16</sub>	<p>仮想端末名ファイルをアクセス中に入出力エラーが発生した。</p> <p>(S) 処理を終了する。 (P)</p> <ul style="list-style-type: none"> <li>• 仮想端末名ファイルが読めるかを確認し，再実行する。</li> <li>• 読めない場合は，アンインストール後に再インストールする。</li> </ul>	8
24584 (6008) <sub>16</sub>	<p>COBOL の CALL 文を使用して XMAP3 の画面をアクセスする場合，または C 言語でアクセスする場合，共通インタフェーステーブル中に指定した入力論理マップの大きさが，実際の論理マップの大きさより小さい。</p> <p>(S) 処理を終了する。 (P) AP の処理を見直し，再コンパイル後，再実行する。</p>	8

---

## 付録 F 用語解説

### (英字)

---

#### AP (Application Program)

画面・帳票に入出力する業務プログラムです。

#### API (Application Programming Interface)

画面・帳票にデータを入出力するときの AP と XMAP3 のプログラムインタフェースです。

#### AP が受け取る項目

入力論理マップのデータ項目で、AP が受け取る次のデータを格納します。

- ・キーボードから画面上に入力されたデータ
- ・画面上で選択された候補 (メニューやボタン) に対応する通知コード

#### AP が渡す項目

出力論理マップのデータ項目で、AP が渡すデータ (AP 実行時に毎回変更する内容) を格納します。

#### AP 環境ファイル (X3MWDRV)

マッピングライブラリ実行時の各種情報を取得するための設定ファイルです。このファイルの内容は、表示・印刷セットアップやロギング支援のプロパティで設定します。

ログの取得、物理マップのロードパス、物理マップの常駐サイズを指定します。デバッグやチューニングのときに必要に応じて設定します。

場所と名前 : XMAP3 インストール先 ¥ETC¥X3MWDRV

#### AP パターン・AP 部品

AP の定型的な型として XMAP3 が提供しているパターンで、COBOL 用と C 言語用があります。

AP パターンは、プログラムの標準的な骨組みです。AP 部品は、処理の手続きです。

これらを組み合わせると、AP を効率良く作成でき、定型的なコーディングの統一も図れます。

格納場所 : XMAP3 インストール先 ¥PATTERNS¥COBOL (または C)

#### C/S システム (Client / Server System)

サーバとクライアントで処理を分散する業務形態です。また、一つの処理をネットワーク上のサーバとクライアントで分担する機器構成モデル、または処理方式です。

#### C/S セットアップ

XMAP3 の C/S システム環境に関する設定をするための機能です。

C/S セットアップは、サーバとなる PC (アプリケーションがある PC) で設定する必要があります。

クライアント PC でセットアップを実行しても、設定内容は無効になります。

#### CD 項

XMAP3 で画面の入出力をするために COBOL プログラム中にコーディングする論理端末定義ブロックです。物理マップ名称や端末名称などを指定します。

COBOL 言語での通信記述項といいます。

## CSV ファイル ( Comma Separated Values )

データの区切りをコンマ ( , ) や改行で表すテキストデータを格納するファイルです。表計算ソフトで入力して、ユーザ独自の形式に編集できます。

XMAP3 では、ドキュメンテーション支援で各種情報を CSV ファイルに出力できます。

## CUI 画面

キーボードからの文字入力を中心の画面です。従来のメインフレーム型オンライン業務か、MS-DOS ベースの業務で使用していた画面です。

XMAP3 では、画面の入出力を物理マップと論理マップを使って実現しています。

## EUC ( Extended UNIX Code )

ターゲットが「AIX」、「Linux」、「Solaris」の場合に使用する文字コードです。

## GUI 画面

キーボードからの入力のほかに、マウスによる操作ができる画面です。Windows の「ボタン」「スクロール」「プルダウン・カスケード」「ポップアップ」などが使用できます。

XMAP3 では、画面の入出力を物理マップと論理マップを使って実現しています。

## MAPPING MODE 句 ( マッピングオプション ( マッピングモード ) )

COBOL の通信記述項にマッピングオプションを指定する領域です。次に示すマッピングオプションに対応しています。マッピングオプションを参照してください。

マージ : 空白または 0

論理マップだけ : 2

物理マップだけ : 3

## OLTP ( Online Transaction Processing )

オンライントランザクション処理のことで、OpenTP1 で実現します。OpenTP1 を使用すると、クライアントアプリケーション型の OLTP システムを構築できます。

## RECEIVE

画面からデータを入力するときに日立 COBOL 用のプログラムの中にコーディングする命令語です。

## SEND

画面を出力するときに日立 COBOL 用のプログラムの中にコーディングする命令語です。

## TRANSCIVE

画面を出力してデータを AP に入力するときに日立 COBOL 用のプログラムの中にコーディングする命令語です。

## XMAP3 インストール先

インストール時にユーザが任意に指定できます。特に指定しない場合は、次のフォルダ ( ディレクトリ ) にインストールされます。

- Windows 95, Windows 98, Windows Me, Windows NT 4.0, Windows 2000, Windows XP, および Windows Server 2003 の場合

Windows フォルダがあるドライブ ¥Program Files¥HITACHI¥XMAP3

- Windows Server 2003 x64 の場合  
Windows フォルダがあるドライブ ¥Program Files(x86)¥HITACHI¥XMAP3

## (ア行)

---

### アクセスキー

メニューバー、ポップアップメニュー（ポップアップメニューファイルを使用する場合も含める）、およびプッシュボタンボックスで、候補の選択操作をするためのキーです。

これらのオブジェクトにフォーカスが位置しているとき、アクセスキーを押すことで該当するメニュー項目やボタンを選択できます。また、[Alt]+アクセスキーを押すことでメニューバーのメニュー項目を選択（プルダウンメニューを表示）できます。

### 一次ウィンドウ

メインとなる画面で、何も表示されていない状態のときに表示します。業務を選択するメニューや、各業務の主画面として使用します。

一つの AP から同時に複数表示されることはなく、すでに表示している一次ウィンドウの消去後に次の一次ウィンドウが表示されます。

### イベント通知コード

コマンドコントロールオブジェクト（プッシュボタン、メニューバー）や確定キー（PF キーなど）に割り当てるコードです。オブジェクトやキーを操作するとイベントが発生し、通知コードに対応づけた動作（AP 通知など）が実行されます。これを「INC 定数」といいます。また、入力単位がイベントのとき通知される「イベント定数」もあります。

イベント通知コードの値はドローセットアップで変更できます。動作はドローで変更できます。

### ウィンドウ

ユーザ作成画面です。タイトルバー、メニューバー、レイアウト領域、オペレータインジケータで構成されます。

### ウィンドウ種別

XMAP3 で定義する画面には、一次ウィンドウと二次ウィンドウがあります。

一次ウィンドウは、メインとなる画面で、何も表示されていない状態のときに表示します。業務を選択するメニューや、各業務の主画面として使用します。原則として、一つの AP から同時に複数表示されることはなく、すでに表示されている一次ウィンドウを消去してから次の一次ウィンドウが表示されます。

二次ウィンドウは、一次ウィンドウを表示したまま新しく表示する画面です。データ入力の入力補助、メッセージ、ヘルプなどに使用します。一次ウィンドウに重ねて複数（最大 3 個）の二次ウィンドウを表示できます。このときの操作対象は、最後に表示した二次ウィンドウだけになります。

### 埋字

入力（または出力）されたデータが、AP が受け取る項目（または AP が渡す項目）の長さより短いとき、残りの領域を埋める文字です。右側と左側のどちらに埋字を格納するかは、桁寄せの指定に従います。ドローで作成したオブジェクトには、ドローセットアップで指定した埋字が仮定されます。必要に応じて、ドローのダイアログで変更できます。なお、埋字に指定できるのは半角文字だけです。埋字を「スペース」で指定した場合、漢字専用で使うオブジェクトでも 1 文字に付き半角スペース二つが埋字されます。

## エラー通知文字

XMAP3 が入力データにエラーを検出した場合に、AP が受け取るデータです。

使用目的が「カナ」で詳細目的が「カナ・半角」(詳細目的は GUI 画面だけ)のエントリ系オブジェクトに対して、全角文字を入力したときに、入力データのエラーとなります。

また、属性の「空白入力」が「一部 & 全桁(半角)」または「一部(半角)」のエントリ系オブジェクトに対して、全角の空白を入力した場合も入力データのエラーとなります。ただし、表示・印刷セットアップの「日本語入力」タブで、全角スペースを半角スペース 2 個に変換する指定をしている場合は該当しません。

## オペレータインジケータ

XMAP3 がキーボード状態や送信状態などのメッセージを画面に表示するための領域です。ユーザプログラムからの操作はできません。

オペレータインジケータの有無はドロウで定義します。表示位置などは表示・印刷セットアップで設定できます。

## (カ行)

---

### カーソル・フォーカス制御

カーソル・フォーカス制御は、ドロウセットアップで設定します。

カーソル・フォーカス制御は、同時に制御するか別々に制御するかを選択できます。

### カーソル制御

カーソル制御は、ドロウセットアップのカーソル・フォーカス制御で設定します。カーソル制御には、論理カーソルと行列(2進)カーソルの2種類があります。

論理カーソルは、カーソル位置をカーソル定数で制御します。カーソル定数は XMAP3 が生成するため、画面のオブジェクト(フィールド)の位置を変更しても AP を変更する必要はありません。

### カーソル定数

入力可能なフィールドにカーソルを位置づけるための定数です。

### 下位項目

オブジェクトの入出力となる一つのデータ項目を階層化して、複数の項目に細分できます。この場合、AP は細分化された下位項目を使ってデータを参照・格納できます。

下位項目は、データ型が「文字(XX)」の場合に指定できます。

### 隠しフィールド

実際の画面には表示されないフィールドで、入力固定項目として使います。

入力固定項目とは、入力論理マップに必ず固定の値を返すデータ項目で、論理マップ可変部の最初のデータ項目として展開されます。

### 仮想画面

画面ごとのデータを AP とやり取りするために、XMAP3 が持っている仮想的な画面です。

AP から返されたデータは、仮想画面に展開されてから、実際のウィンドウに表示されます。ユーザの入力データも、仮想画面を介して AP に返されます。



## 仮想端末名ファイル

AP がプログラム中で使用する仮想端末名、デバイス、サービス名などの設定ファイルです。このファイルの内容は、C/S セットアップの「C/S 構成」で設定します。

なお、スタンドアロン環境でプリンタを 1 台だけ使用している場合は、特に設定する必要はありません。

場所と名前：XMAP3 インストール先 ¥ETC¥X3MWHOST

## 壁紙

指定するグラフィックをレイアウト領域の各オブジェクトの背後に表示します。壁紙のグラフィックには、ビットマップファイル (BMP) が使えます。レイアウト領域より小さい壁紙はタイル状に並べられて表示されます。レイアウト領域より大きい壁紙は、はみ出し部分 (下側・右側) は表示されません。なお、壁紙はレイアウト領域には表示されません。確認には、テスト表示を利用してください。

## 画面属性

入力項目の扱いや表示形態などの画面全体に関する属性です。GUI 画面では画面属性ダイアログで属性を指定します。また、画面属性は AP 実行時に動的に変更することもできます。AP から動的に変更するには、ドローセットアップの「画面属性の動的変更」で変更したい属性を定義します。ここで指定した修飾名を制御項目に代入することで、指定した属性に変更できます。

## キーエントリ

キーボードから文字列を入力するための各種オブジェクトの総称です。

入出力テキスト・フィールド、入出力日付/時刻テキスト・フィールドが該当します。

## 基準文字サイズ

GUI 画面の作業領域全体の基準となる文字サイズです。CUI 画面では変更できません。この値は、画面属性ダイアログの「基準文字サイズ」で設定します。

画面のレイアウト領域でオブジェクトを配置するときの単位となるグリッド (ます目) は、このサイズが基準になります。文字の縦幅の 1/2 がます目の縦幅、半角文字の幅がます目の横幅になります。

各オブジェクトでは、この基準文字サイズを基に「標準」「大」「小」の文字サイズ指定ができます。

## 空白入力

データ項目に半角の空白、全角の空白、またはそれらの混合の挿入を許すかどうかを指定します。

## グラフィック

グラフィックデータを出力するオブジェクトです。データには、ビットマップ (.BMP)、メタファイル (.WMF)、拡張メタファイル (.EMF)、JPEG ファイル (.JPG、または .JPEG) が使えます。出力するデータをドローで定義する「固定グラフィック」と、AP から指定する「出力グラフィック」があります。

固定グラフィックは配置する前にあらかじめグラフィックファイルを用意しておきます。

出力グラフィックは領域だけを配置します。グラフィックデータは、AP からファイル名またはクリップボード経由で渡します。

## グラフィックコントロール

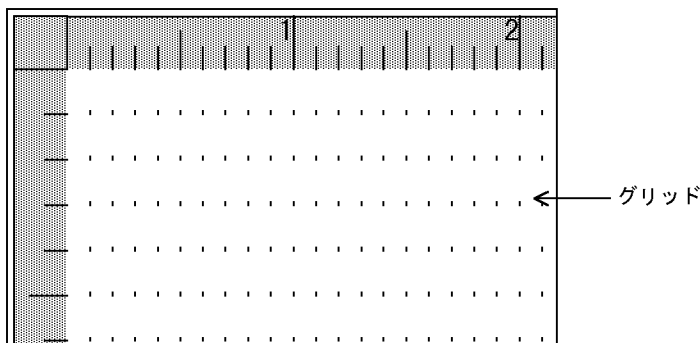
画面内のオブジェクトを整理・強調するための各種オブジェクトの総称です。

グラフィック、セパレータ、けい線、矩形、メッセージアイコン、塗りつぶしフィールドが該当します。

## グリッド

ドローのレイアウト領域上の座標を示す格子状の線や点で、オブジェクト配置位置となります。格子一つ分の単位を「ます」といいます。

画面定義でのますのサイズは、画面属性で指定する基準文字サイズに従います。行方向は文字サイズの半分、列方向は半角文字の幅がそれぞれますの縦幅 / 横幅となります。



## グループコントロール

中にほかのオブジェクトを配置できる各種オブジェクトの総称です。

グループボックス、フィールドボックス、フレームが該当します。

## グループボックス

画面上のオブジェクト（セパレータを除く）を囲んでグループ化するオブジェクトです。

必要に応じて、グループボックス単位にフォーカス遷移の順序を設定できます。

## 桁

使用する文字数を半角換算で数えた単位です。

## 桁寄せ

次の場合に、データを右や左に寄せてやり取りすることです。

- ・入力データが、AP が受け取る項目の長さより短い
- ・出力データが、AP が渡す項目の長さより短い

## 後退キー（Back Space）

表示・印刷セットアップで [ BackSpace ] キーの指定に対応して動作が次のように異なります。

- ・直前の項目 : テキスト・フィールドの先頭にカーソルを位置づける。テキスト・フィールドの先頭では、前の位置にあるテキスト・フィールドにフォーカスが移動する。
- ・1文字削除 : 直前の文字を削除し、間を詰めます。
- ・ヌル置換 : 直前の文字をヌル (00)<sub>16</sub> で置き換えます。

## 候補選択コントロール

画面に表示された各候補から対象を選ぶための各種オブジェクトの総称です。

リストボックス、ラジオボタン、チェックボタン、トグルフィールドが該当します。

## 項目

AP と XMAP3 でやり取りする論理マップ中の各要素を項目といいます。画面や帳票に表示するユーザデータを扱う項目をデータ項目といい、表示属性を変更する目的で使用する項目を制御項目とい

います。データ項目は、データ型によって、次のように呼びます (COBOL の場合)

- ・文字項目 : データ型が文字 (XX) のときのデータ項目
- ・数字項目 : データ型が数字 (99) のときのデータ項目
- ・数字編集項目 : データ型が数字編集 (Z, 9, ¥ などの組み合わせ) のときのデータ項目
- ・漢字項目 : データ型が日本語 (NN) のときのデータ項目

### 固定テキスト・フィールド

固定の文字列を出力するオブジェクトです。文字列はドローで定義します。全角文字を使える「日本語」と、半角文字だけを使える「英数」があります。

### コマンドコントロール

選択操作で、イベントを発行して処理を実行するための各種オブジェクトの総称です。プッシュボタンとメニューバーが該当します。

### コントロールメニュー

GUI 画面で、タイトルバー左端のアイコン (またはボタン) から表示される Windows のシステムメニューです。

XMAP3 では、コントロールメニューに [ 閉じる ] メニューを使用するかどうかを表示・印刷セットアップで設定できます。

### コンボボックス

候補 (メニュー項目) を選択したり、データを直接、キー入力したりして、入力するデータを指定するオブジェクトです。

必要な場合だけ入力するオプションを選択するときなどに使います。

メニューは付属しているボタンを選ぶと表示されます。メニュー項目をドローで定義する「固定コンボボックス」と、AP から指定する「可変コンボボックス」があります。

なお、ドローでは [ コンボボックス ] を使用するとメニュー選択するコンボボックスを定義でき、[ キー入力コンボボックス ] を使用すると、キー入力するコンボボックスが定義できます。このマニュアルではこれらを区別するため、メニュー選択するコンボボックスを「メニュー選択コンボボックス」、キー入力するコンボボックスを「キー入力コンボボックス」と記述し、総称として「コンボボックス」と記述しています。

## ( サ行 )

---

### サービス名ファイル ( X3PHOST )

サーバに関連するサービス名やサービス種別などの設定ファイルです。このファイルの内容は、C/S セットアップの「C/S 構成」で設定します。C/S セットアップで設定できない情報は、直接このファイルを設定してください。

この情報はサーバ側だけに設定します。スタンドアロン環境の場合は、このファイルの設定は不要です。

場所と名前 : XMAP3 インストール先 ¥ETC¥X3PHOST

### 再定義名

定義済みの論理マップを別定義として使用する際の名称です。半角で 30 字以内で定義します。

入力論理マップ用と出力論理マップ用とで別の名称にすることもできます。

## 削除キー (Delete)

入力ができるテキスト・フィールドのデータを 1 文字ずつ削除するためのキーです。

## 時刻 (入出力, 出力)

時分秒を表示できます。時分秒は、時刻ダイアログ (出力 / 入出力) での指定します。「時」「分」「秒」の区切りには何を使うか、直前のゼロを表示するかしないかなど、どのような形式で表示するかは時刻ダイアログの選択肢で選べます。このような定義をしておくこと、入力した数字データは、定義した形式になって表示されます。

## 修飾名

表示属性の変更を AP から指示するための定数です。次の手順で利用します。

1. ドローセットアップで、修飾名と表示属性の対応を定義する。
2. ドローで、オブジェクトや画面の属性に「動的変更」を指定する。  
制御項目 (修飾名を格納する領域) が論理マップに展開されます。
3. AP で、制御項目に修飾名を格納する。  
修飾名に設定された表示属性に従って、画面に表示 (または帳票に印刷) されます。

## 出力テキスト・フィールド

AP から指定する可変の文字データを出力するオブジェクトです。

出力する文字に応じて、種類 (使用目的) を選べます。

## ショートカットキー

メニューバーのメニュー項目の選択操作をするためのキーです。

フォーカス・カーソルの位置に関係なく、[Ctrl] + 半角英字 1 字を押すことで該当するメニュー項目を選択できます。

なお、メニューバーのメニュー項目、またはカスケードメニューを持つプルダウンメニューにはショートカットキーを定義できません。

## 初期クリア文字

AP 実行時に、画面からの入力データを AP が受け取る前に、XMAP3 が入力論理マップの各項目をあらかじめクリアしておく文字です。使用目的が「漢字」の場合でも、半角スペースが入ります。

なお、AP が受け取るデータの優先順位は次のとおりです。

1. キー入力データ / 選択した通知コード (正常操作)
2. 入力済み / 選択済みするとき、表示されているデータ / 通知コード
3. エラー通知文字
4. 初期値 / 初期通知コード
5. 初期クリア文字

## 初期値

初期値には、入力と出力によって初期値の内容が次のように変わります。

### 《出力》

出力論理マップの各項目にあらかじめ設定しておく値です。AP 実行時に、出力データの代わりにデータ有無コードを設定した場合は初期値が有効になります。

### 《入力》

AP 実行時に、画面からの入力データを AP が受け取る前に、入力論理マップの各項目にあらかじめ設定しておく値です。なお、AP が受け取るデータの優先順位は次のとおりです。

1. キー入力データ / 選択した通知コード

2. 入力済み / 選択済みのとき, 表示されているデータ / 通知コード
3. エラー通知文字
4. 初期値
5. 初期クリア文字

### 数字編集項目

指定したピクチャ句に従って, 数字を編集して入出力します。ピクチャ句は, 次の文字で指定します。

入力: SV9

出力: \* + - ¥9 . , Z /

### スクロール領域

ウィンドウやボックス内で, 表示範囲を移動できる領域です。表示範囲の移動はスクロールバーで操作します。

これに対し, 表示範囲の移動対象外の領域を「スクロール禁止領域」といい, ウィンドウやボックス内に常に表示されます。

### スピンボックス

入出力テキストの右端に, テキストに表示されている数字の増減をコントロールするステップが付いているボックスです。増分値を指定すると, 指定した間隔で表示されている数字が増減します。

### スプラッシュ画面

メニューバー, タイトルバーおよびオペレーティングエータを表示しない画面です。壁紙などのグラフィックや表示直後に次の画面と切り替わる定義と併用して, AP を実行する前にその AP や製品を効果的に紹介するのに利用できます。

### セットアップ情報の反映

ユティリティの機能の一つです。すでに作成したマップ定義ファイルや書式定義ファイルに対して, ドローセットアップの情報を反映します。ドローで画面を定義したあとに, ドローセットアップで動的変更などの設定を変更した場合に使います。

### セパレータ

画面上の各オブジェクトの間を区切るときに使う GUI 画面用の線オブジェクトです。縦線と横線が使えます。

### 選択エントリ

選択操作で文字列を入力するための各種オブジェクトの総称です。コンボボックス, ポップアップ, スピンボックスが該当します。

### 挿入キー (Insert)

テキスト・フィールド上の文字列の間に, 文字を挿入するための半角 1 文字分を空けます。挿入した空きに文字を書き込まないときは, AP が受け取るデータ上で空きが詰められます。

動作の詳細については, 表示・印刷セットアップで設定できます。

## (タ行)

### チェックボタン

入力するデータを候補（ボタン）の選択によって指定するオブジェクトです。グループ内の複数のボタンから複数の選択ができます。ラベル長や選択肢の数が少ないときに使います。

ボタンラベルをドローで定義する「固定チェックボタン」と、AP から指定する「可変チェックボタン」があります。

チェックボタンのグループを表すボックスを「チェックボタンボックス」といいます。

### 通信記述項（CD 項）

XMAP3 で画面の入出力をするために COBOL プログラム中にコーディングする論理端未定義ブロックです。物理マップ名称や端末名称などを指定します。

（画面の例）

GD DSP FOR I-O WS		
MAP NAME	IS 画面マップ名	: 物理マップを格納する領域
SYMBOLIC TERMINAL	IS 画面端末名	: 仮想端末名を格納する領域
MAPPING MODE	IS マッピングモード	: マッピングオプションを格納する領域
STATUS KEY	IS 画面-RC.	: 画面の入出力が正しく実行されたかを判定する領域

### 通知コード（データ入力用）

候補（ボタンやリスト項目など）の選択によってデータ入力をするオブジェクトで、各候補に割り当てるコードです。

ユーザが選んだ候補に対応する通知コードが、AP が受け取る項目に格納されます。

### 定義パターン

ドローで画面を新規作成するときに、用途に応じて指定するパターンです。標準的なレイアウト領域のサイズやパターンなどがあらかじめ設定されていて、ドローでのレイアウト定義の基本フォーマットとなります。

XMAP3 が標準提供している定義パターンのほかに、ユーザが独自に作成・登録することもできます。

### データ有無コード

AP（論理マップのテキスト・フィールドのデータ名の領域）にデータが設定されていないことを示す 1 バイトの文字コードです。(00)<sub>16</sub> ~ (FF)<sub>16</sub> の間で任意で、通常は (1F)<sub>16</sub> を仮定します。

コードを変更する場合は、使用する言語に応じて次のとおりになります。

COBOL の場合

1. SEND / RECEIVE のとき、通信記述項の DATA ABSENCE CODE 句
2. CALL のとき、オープン要求のインタフェース

C 言語の場合

1. jsvwadrv 関数のオープン要求インタフェース

### データ型

COBOL のピクチャ句で指定するデータの型です。文字型や数字型があります。C の場合は、char だけです。ドローで作成したオブジェクトには、ドローセットアップで指定したデータ型が仮定さ

れます。必要に応じて、ドロワーで変更できます。

データ型は、文字用、数字用、数字編集用に分かれます。使用できるデータ型は次のとおりです。

#### 《入力用のデータ型》

- 文字用 : 文字 (XX), 漢字 (NN), 漢字 (XX)
- 数字用 : 99999
- 数字編集用 : S99999, 999V9 など

#### 《出力用のデータ型》

- 文字用 : 文字 (XX), 漢字 (NN), 漢字 (XX)
- 数字用 : 99999
- 数字編集用 : Z, 9, ¥などを組み合わせた各種数字編集文字列

また、登録されていないデータ型を指定したい場合、「自由な設定」ボタンで表示されるダイアログで任意に設定できます。

### データキー (Ctrl + End)

画面上のすべての入力テキスト (または入力フィールド) の文字を消去するためのキーです。データクリアキーともいいます。

### データ消去通知文字

AP 実行時に、画面からフィールドキーで入出力テキスト・フィールドのデータを消去したり、(00)<sub>16</sub> のデータを受信したときに AP が受け取るデータです。したがって、AP が受け取ったデータがデータ消去通知文字であれば、画面操作でデータ消去されたと判断できます。

初期クリア文字や初期値とデータ消去通知とを分けておくことで、入力操作がなかったのか、データ消去されたのかを区別できます。

ただし、「入力済み」属性のオブジェクトで、画面確定時に入力データや表示データがない場合は、データ消去されていなくてもデータ消去通知文字が返ります。

なお、使用目的が「漢字」の場合でも、半角スペースが入ります。

### データ長

論理マップの長さ (バイト数) です。通常は、桁を基に各ターゲットに応じた値が自動的に計算されます。ユーザ任意の値を設定できるターゲットもあります。

### データ名, 制御項目データ名

データ項目名 (またはデータ名) は、データの入出力に関する各項目の領域の名称です。データ名のうち、修飾名 (表示属性の変更を AP から指示するための定数) を格納する領域の名称を制御項目データ名といいます。

### トグルフィールド

「YES / NO」などの印を付けるためのオブジェクトです。表の各行末に配置して、「オン / オフ」や「済 / 未」を表す場合などに使います。

### ドロワー

画面を定義する機能 (エディタ) です。画面のレイアウトや各オブジェクトの属性を定義します。また、ドロワーからテスト表示や論理マップ表示などの機能も利用できます。

### ドロワーセットアップ

画面の定義に関する標準の値を設定するための機能です。

XMAP3 では、セットアップする項目の標準値を用意していて、この標準値を変更する場合にだけ

設定します。このセットアップの内容はマップの形式に関係していますので、画面を作成する前にセットアップを済ませておく必要があります。

## (ナ行)

---

### 二次ウィンドウ

データ入力補助、メッセージ、ヘルプなどのために、一次ウィンドウのデータを表示したまま新しく表示する画面です。一次ウィンドウに重ねて表示され、このとき操作できるのは二次ウィンドウだけとなります。

### 日本語入力に関するセットアップ

日本語入力機能の使用に関する設定です。例えば、使用目的が日本語の場合、日本語入力状態にするかどうかになります。この設定は、表示・印刷セットアップの「日本語入力」の次の項目で設定します。

日本語入力の自動制御 / 日本語入力方式 / 自動起動するオブジェクト種別  
/ 日本語入力対象外オブジェクト / 日本語入力変換位置

### 入出力テキスト・フィールド

キーボードなどから文字データを入力するオブジェクトです。また、AP から指定する文字データを表示できます。

初期表示値（出力データ）を示し、書き換えが必要な場合だけ入力する使い方ができます。  
入出力するデータの意味に応じて、種類（使用目的）を選べます。

### 入力単位（画面 / フィールド / イベント / 表示直後）

画面を確定させ、AP へ制御を渡すタイミングの種類です。通常は、画面の最後までデータを入力したり選択したあとで、確定の機能を持つキーやボタンを押して、AP へ制御を渡します（画面）。

「フィールド」では、フォーカスが位置づいているオブジェクトでデータの入力、選択を行い、そのオブジェクトからフォーカスが離脱した時点で AP へ制御が渡ります。「イベント」では、画面上のすべてのオブジェクトを対象にして、何かイベントが起これば AP に制御が渡ります。「表示直後」では、一定の待機時間が過ぎると AP に制御が渡されます。

### 塗りつぶしフィールド

フィールドボックス内の領域に色を付けるオブジェクトです。ほかのオブジェクトを重ねて配置できます。1 行置きに色のパターンを付けて行を見やすくする場合などに使います。

## (ハ行)

---

### 反復定義

一つのオブジェクトを縦または横方向に繰り返す定義で、表形式で同じ属性のオブジェクトを並べる場合などに使います。

反復定義を使うと、同じ属性のオブジェクトを一つずつ配置するよりも効率良く定義できます。また、論理マップは配列（日立 COBOL では OCCURS 句）で展開されるので、AP のロジックも反復指定で実現できます。



## ビッグエンディアン

バイナリ形式のデータを扱う際、メモリ上に左から右側へ1バイトずつ格納する形式です。

(例)

(000A)<sub>16</sub> を2バイトのメモリに格納した場合は、(000A)<sub>16</sub> になります。

## 日付(入出力, 出力)

年月日を表示できます。年月日は、日付ダイアログ(出力/入出力)で指定します。「年」を表すのに西暦にするか和暦にするか、「年」「月」「日」の区切りには何を使うか、直前のゼロを表示するかないかなど、どのような形式で表示するかは日付ダイアログの選択肢で選べます。このような定義をしておくこと、入力した数字データは、定義した形式になって表示されます。

## 表示・印刷環境ファイル

画面表示の設定ファイルです。このファイルの内容は、表示・印刷セットアップで設定します。なお、このファイルの設定項目には、直接ファイルをエディタで開いて編集する(セットアップで設定できない)項目もあります。表示・印刷セットアップで設定できる項目については、セットアップ情報が優先されます。

場所と名前 : XMAP3 インストール先 ¥ETC¥X3PCONF

## 表示・印刷セットアップ

XMAP3 が表示・印刷する画面・帳票の環境を必要に応じて設定するための機能です。

ユーザ画面の外観や操作キーの割り当て、プリンタの印刷モードなどの各種設定があります。

## フィールドキー(End)

テキスト・フィールドで、カーソル以降の文字を削除します(標準の場合)。

割り当てるキーや動作は表示・印刷セットアップで変更できます。

## フィールドボックス

多量のデータをまとめて入出力するためのボックスオブジェクトです。

次のような場合に適しています。

- ・ けい線を使って、表形式でオブジェクトを並べる
- ・ CUI 画面を基本にして一部だけを GUI 化(ファンクションキーによる選択部分だけのボタン化など)
- ・ 文章データなど、複数行にわたる文字データの表示

## フォーカス

GUI 画面の中で、あるオブジェクトを操作対象にすることをフォーカスを位置づけるといいます。

## フォーカス制御

フォーカス位置の情報を制御するための制御項目です。AP では、フォーカスを設定したいオブジェクトに対応するフォーカス定数を、制御項目に格納して渡します。また、画面上でフォーカスが位置づいているオブジェクトに対応したフォーカス定数を、制御項目で受け取ります。なお、フォーカス定数は、XMAP3 が生成します。このため、画面のオブジェクトの位置を変更しても、AP を変更する必要はありません。

## フォーカス定数

入力や選択が可能なボックスにフォーカスを位置づけるための定数です。

## 不活性（選択できない状態にする）

候補（ボタンやメニュー）を選択できない状態にします。表示方法と組み合わせで、選べる候補を制限したい場合に使います。AP 実行中に選択可能な状態に戻したい場合は、該当するダイアログで「AP から表示属性を変更する」を指定し、AP から制御項目に活性を指定した修飾名（ドローセットアップで指定）を指定することで実現できます。

## プッシュボタン

イベント（実行する処理）を候補（ボタン）の選択によって指定するオブジェクトです。イベントに対応する処理として「AP 通知」や「ポップアップ表示」などを指定できます。確定キーの代わりとしても使えます。

ボタンラベルはドローで定義し、AP 実行時に変更できます。ボタンラベルには、文字列のほかにグラフィックを表示することもできます。

プッシュボタンのグループを表すボックスを「プッシュボタンボックス」といいます。

## 物理画面

実際のディスプレイ画面です。各ウィンドウは、最終的にこの物理画面に表示されます。

## 物理マップ

画面のオブジェクトの位置などを格納した情報ファイルです。ファイル名はマップ名にデバイス ID を付加した名前です。拡張子は .PMP です。画面の定義終了時に、論理マップとともに生成されます。ディスプレイやプリンタの入出力データを論理マップデータに変換したり、論理マップデータを入力出力データに変換したりするときに XMAP3 が参照します。

## フレーム

複数のオブジェクトの組み合わせを縦方向または横方向に反復して並べるためのオブジェクトです。テキスト・フィールドなどを組み合わせて表形式で並べる場合に使います。論理マップには配列として展開されます。

## プレーン

フィールドボックス、リストボックス、ボタンボックスが持つ領域で、ボックス内に表示するフィールドやボタンなどを配置するための平面です。

プレーンの属性の文字サイズは、プレーン内で共通の配置単位や文字サイズとなり、中のすべてのオブジェクトに適用されます。

フィールドボックスとリストボックスでは、ボックスよりも大きいプレーンを指定できます。画面上では、ボックスサイズの範囲だけが表示され、範囲外の部分はスクロールをして表示します。

## ポップアップ

入力するデータを候補（メニュー項目）の選択によって指定するオブジェクトです。また、キーボードからの入力を併用する使い方もできます。

メニュー項目をドローで定義する「固定ポップアップ」と、AP から指定する「可変ポップアップ」があります。

## ポップアップメニューファイル

XMAP3 では、ポップアップテキストに表示するメニューデータを AP 実行時にファイルとして渡せます。このファイルをポップアップメニューファイルといいます。ポップアップメニューファイルは、あらかじめ作成して用意しておく必要があります。

ポップアップメニューファイルは、メニュー項目が大量にある場合や、メニューを大分類、小分類にして表示したい場合に使用します。また、ホスト連携機能を利用する場合、PC 側にポップアップ

メニューファイルを持つことで、ホストと PC 間で転送するデータ量を減らせます。ポップアップメニューファイルは、ユティリティの「ポップアップメニューエディタ」で編集できます。

## (マ行)

### マッピングオプション (マッピングモード)

「マージ」、「論理マップだけ」、「物理マップだけ」の三つの指定があります。この指定は、COBOL の SEND, RECEIVE, TRANSCEIVE 文を使用したときに指定する MAPPING MODE 句と、COBOL の CALL 文および C 言語の時に指定するマッピングインタフェース領域があります。

マージは、論理マップと物理マップをマージして、同一画面をすべて書き換えるときに指定します。ただし、同じ画面に対し、2 回目以降の表示では固定部分 (タイトルやけい線) は再描画しません。論理マップだけは、一般的に 2 回目以降の表示で、入力した状態をそのままにして部分書き換えをするときに指定します。

物理マップだけは、メニュー表示など、物理マップだけで初期表示するときに指定します。

### マップ生成

ユティリティの機能の一つです。マップ定義ファイルから、論理マップファイルと物理マップファイルを生成します。また、書式定義ファイルから書式イメージファイルと行制御データファイルを生成します。

### マップ定義ファイル

ドロで定義した画面や帳票の定義情報であるソースマップを格納したファイルで、拡張子は .IMP です。

画面・帳票の定義終了時に、論理マップとともに生成されます。物理マップと論理マップは、このマップ定義ファイルを基に生成されます。

### マップ展開方式

マップ生成時に出力される論理マップの数値領域の展開形を指定します。展開形式には、次に示す 2 種類があります。

- リトルエンディアン用マップを展開  
Windows, Linux の場合で、スタンドアロンまたは C/S システム構成のときに使用できる展開方式で論理マップを生成します。
- ビッグエンディアン用マップを展開  
C/S (Windows, Linux 以外の WS で共用) 構成の場合に使用できる展開方式で論理マップを生成します。

### メッセージアイコン

操作者に対して何らかの注意を促すときに表示します。メッセージアイコンは、基本的にメッセージと一緒に出力することをお勧めします。アイコンには、インフォメーション、ワーニング、クエスチョン、エラーの 4 種類があります。

### メニュー形式

可変ポップアップテキストでポップアップメニューファイルを使用する場合、表示するポップアップメニューの形式を選択できます。メニュー形式には、標準のメニュー形式とリスト形式があり、リスト形式にはさらに「ポップアップ型」と「ダイアログ型」があります。

## メニューバー

イベント（実行する処理）を候補（ボタン）の選択によって指定するオブジェクトです。イベントに対応する処理として「AP 通知」や「ポップアップ表示」などを指定できます。確定キーの代わりとしても使えます。

メニューの内容はドロワーで定義し、AP からの指定はできません。

プルダウンメニューやカスケードメニューを使って階層化でき、処理内容に応じたグループ分けができます。

## モジュラスチェック

「使用目的」が「数字」の場合に、モジュラスチェックをするかどうかを指定できます。指定した場合、画面の属性ダイアログで指定するモジュラスアルゴリズム（チェック 10 または 11）に従って、入力文字列（数字）がチェックされます。チェック条件に合った場合は AP イベントは返しません。合わない場合はエラーになり、テキスト・フィールドからカーソル・フォーカスが移動できなくなります。

### ・チェック 10

次のチェックをし、一致する（合格）かを判定します。

1. 入力したデータの最後の 1 バイト（チェックデジット）・入力したデータのバイト数（入力したデータの最後の 1 バイトを除く）を取り出す。
2. 入力したデータのバイト数の右側より 2, 1, 2, 1, ... を乗数とし桁ごとに掛け算をする。
3. 掛け算の結果が 2 桁となったものについては 10 で割り算し、商とあまりに分解する。
4. 「2」「3」での掛け算と商およびあまりの結果をすべて加え合わせる。
5. 加算の結果を 10 で割り算し、あまりを求める。
6. 10 からあまりを減算し、差を求め、この差を入力したデータの最後の 1 バイトと比較する。

### ・チェック 11

次のチェックをし、一致する（合格）かを判定します。

1. 入力したデータのバイト数（入力したデータの最後の 1 バイトを除く）を取り出す。
2. 入力したデータのバイト数の右側より 2, 3, 4, 5, 6, 7, 2, 3, 4, 5, 6, 7, ... の順に乗数を割り当て桁ごとに掛け算をする。
3. 結果をすべて加え合わせる。
4. 和を 11 で割り算し、あまりを求める。
5. 11 からあまりを減算し、差を求める。この差を入力したデータの最後の 1 バイトと比較する。

## (ヤ行)

---

### 予約フィールド・テキスト

OLTP サーバ構成の環境で、OpenTP1 が提供する予約項目を出力するオブジェクトです。

日付や時刻、論理端末名称などのデータを、あらかじめ論理マップに設定することなく AP の実行時、動的に出力できます。

## (ラ行)

---

### ラジオボタン

入力するデータを候補（ボタン）の選択によって指定するオブジェクトです。グループ内の複数のボタンから一つだけ選択できます。ラベル長や選択肢の数が少ないとき、また、「YES / NO」など

の単純な選択をするときに使います。  
 ボタンラベルをドロワーで定義する「固定ラジオボタン」と、AP から指定する「可変ラジオボタン」  
 があります。  
 ラジオボタンのグループを表すボックスを「ラジオボタンボックス」といいます。

## ラベル

画面に文字列を表示するための各種オブジェクトの総称です。  
 固定テキスト・フィールド、出力テキスト・フィールド、出力日付/時刻テキスト・フィールドが  
 該当します。

## リストボックス

入力するデータを候補（リスト項目）の選択によって指定するオブジェクトです。リストの内容は  
 AP から指定します。ラベル長や選択肢が多いときに使います。必要に応じて、スクロールバーを付  
 けられます。  
 リストから一つだけ項目を選べる「単一選択リストボックス」と、複数の項目を選べる「複数選択  
 リストボックス」があります。

## リトルエンディアン

バイナリ形式のデータを扱う際、メモリ上に右から左側へ 1 バイトずつ格納する形式です。  
 (000A)<sub>16</sub> を 2 バイトのメモリに格納した場合は、(0A00)<sub>16</sub> になります。

## レイアウト領域

画面定義時に、各オブジェクトを配置する領域です。  
 画面では、タイトルバー、メニューバー、オペレーティングデータを除く領域です。

## 論理マップ

画面の可変情報を設定する出力論理マップと、画面から入力されたデータなどを設定する入力論理  
 マップがあります。ファイルの拡張子は、COBOL の場合は .CBL、C 言語の場合は .H です。画  
 面・帳票の定義終了時に、登録集原文として物理マップとともに生成されます。  
 COBOL で AP を作成する場合は、COPY 文で論理マップを取り込みます。AP 実行時には、AP に  
 取り込まれた論理マップのデータ領域を介して、データがやり取りされます。

## (ワ行)

---

### 割込キー ( Break ), ( Ctrl ) + ( Pause )

送信要求の発生を AP に通知するキーです。  
 割込のイベントを通知するキーです。



---

# 索引

## A

---

Access 747  
ANK コード 178  
AP 759  
API 759  
AP インタフェース 66, 158  
AP インタフェースの概要 492  
AP が受け取る項目 759  
AP からウィンドウ位置属性を変更する 67  
AP からウィンドウ属性を変更する 66  
AP から確定キー属性を変更する 67  
AP からのフィールド属性の変更 586  
AP が渡す項目 759  
AP 環境ファイル 759  
AP 間でオープンを引き継ぐ 518  
AP 間でオープンを引き継ぐ場合 494  
AP 作成時のポイント 683  
AP サンプルのコンパイルと実行手順 605  
AP にイベントが渡るタイミング 25  
AP のコーディング方法 491  
AP パターン 558  
AP パターン・AP 部品 759  
AP 部品 559  
AP 分割時の注意 495  
AP への通知 162  
atof 534  
atol 534

## C

---

C/S システム 759  
C/S セットアップ 759  
CALL 文による方法 508  
CALL 文の発行順序 509  
CALL 命令 510  
CBLTERMSHAR=YES 494  
CD 項 759  
COBOL 開発マネージャ 516, 517  
COBOL 開発マネージャでの XMAP3 の利用  
方法 520

COBOL での画面表示命令 550  
COBOL の画面入出力命令 502  
CSV ファイル 760  
CUI 画面 2, 760  
CUI 画面属性の設定 155  
CUI 画面で使用できる機能 10  
CUI 画面でのフィールド単位のデータ入力  
572  
CUI 画面の機能内容 11  
CUI 画面の構成 150  
CUI 画面の作成 149  
CUI 画面の設計 150  
CUI 画面のレイアウト手順 152  
CUI 画面を構成するオブジェクト 161  
C 言語 746  
C 言語固有のコーディング例 533  
C 言語での画面表示命令 554  
C 言語での表示方法 543  
C 言語による画面の入出力 554  
C 言語の画面入出力命令 524

## D

---

Delphi 747  
Delphi での表示方法 544  
Delphi でのファイルの取り込み 545  
DISABLE 文 506  
DLL の呼び出し 543, 544, 545

## E

---

EUC 760

## G

---

GUI 画面 2, 6, 760  
GUI 画面設計でこんなこともできます 51  
GUI 画面設計のポイント 53  
GUI 画面属性の設定 60  
GUI 画面でのイベント単位のデータ入力  
577

GUI 画面でのフィールド単位のデータ入力  
574

GUI 画面の機能内容 9

GUI 画面の構成 38

GUI 画面の作成 37

GUI 画面の設計 38

GUI 画面のレイアウト手順 56

GUI 画面を構成するオブジェクト 70

GUI 画面を設計するときの検討課題 53

## J

---

jsvwadrv 関数 526

jsvwadrv 関数の発行順序 525

JSVWATBL.CBL 502

## M

---

MAPPING MODE 句 507, 760

mcftalcle 12

mcftpx 12

MCR 129, 377

MCR 入力のマッピング規則 265

memcmp 534

memcpy 534

memset 533

## O

---

OLE2 オートメーションのサービス機能 478

OLTP 760

OLTP サーバ構成での画面表示 550

## P

---

PA1 キー 181

PA2 キー 181

PA3 キー 181

PF1 ~ PF12 キー 181

PF13 ~ PF24 キー 181

PF25 ~ PF36 キー 181

PF37 ~ PF48 キー 181

PF49 ~ PF60 キー 181

PF61 ~ PF72 キー 181

PF73 ~ PF84 キー 181

## R

---

RECEIVE 760

RECEIVE 文 506

## S

---

SEND 760

SEND 文 505

sizeof 演算子 533

sprintf 535

STATUS KEY 句のリターンコード 507

## T

---

TRANSCEIVE 760

TRANSCEIVE 文 506

## V

---

Visual Basic 747

Visual Basic および Access での表示方法  
543

Visual Basic および Access でのファイルの  
取り込み 545

## W

---

Windows 専用の汎用入出力関数 536

## X

---

X3MODTBL.CBL 502

X3PHOST 765

XMAP-COM 510

XMAP-REQ 511

XMAP\_COM アドレス 526

XMAP\_REQ アドレス 527

XMAP3 インストール先 760

XMAP3 関連ファイルの関係 29

XMAP3 共通定義 12

XMAP3 実行支援 492

XMAP3 実行支援の AP インタフェース 492

XMAP3 で作成できる画面 2

XMAP3 の画面設計の概要 3

XmapDrvClose 関数 538



XmapDrvCreateOpen 関数 537  
 XmapDrvCreate 関数 540  
 XmapDrvGetError 関数 540  
 XmapDrvOpen 関数 541  
 XmapDrvReceive 関数 539  
 XmapDrvSend 関数 538  
 XmapDrvSetDataCode 関数 542  
 XmapDrvSetMapOption 関数 541  
 XmapDrvTransceive 関数 539  
 XY 位置 47, 60

## Z

Z 位置 47, 61

## あ

アクセスキー 761  
 アクセスキーを使用した固定ポップアップの  
 メニュー選択 412  
 アクセスキーを使用する 464  
 アスタリスク 179

## い

一次ウィンドウ 49, 71, 761  
 一次ウィンドウから二次ウィンドウの表示  
 50  
 一時非表示 49, 61  
 位置属性の表示属性の標準値 737  
 一部上書 15  
 イベント 499  
 イベント単位 20, 22  
 イベント通知コード 64, 251, 498, 761  
 イベント通知コードダイアログ 342  
 イベント通知コードのコメント 158  
 イベント通知コードのチェック 498  
 イベント定数の論理マップ生成規則 333  
 イベントとイベント通知コード 27  
 インタフェーステーブルの取り込み方法 529  
 インタフェース領域 502  
 インタフェース領域 (JSVWATBL.H) 524  
 インタフェース領域の取り込み方法 513  
 インフォメーション 124

## う

ウィンドウ 9, 40, 60, 155, 761  
 ウィンドウ位置 341  
 ウィンドウ各部の名称 150  
 ウィンドウ種別 761  
 ウィンドウ種別の ID 74  
 ウィンドウ属性の表示属性の標準値 737  
 ウィンドウのサイズ変更を抑止する 69  
 ウィンドウの表示形態 340  
 ウィンドウ表示 47  
 ウィンドウ表示位置と重なり 47  
 ウィンドウ表示位置の制御情報 329  
 ウィンドウ表示の制御情報 246  
 ウィンドウを表示したときにアラームを鳴ら  
 す 63, 156  
 後ろに表示 48, 61  
 埋字 350, 358, 761  
 埋字と桁寄せに関するマッピング規則  
 262, 265

## え

英大文字 178  
 英小文字 178  
 エスケープキー 185, 200  
 エラー 124  
 エラー通知 499  
 エラー通知文字 68, 159, 762  
 エラー内容 753

## お

凹型 122  
 応答性能での問題と対処方法 692  
 オープンインタフェース領域の形式  
 (XMAP-OPN, COBOL) 512  
 オープンインタフェース領域の形式  
 (XMAP\_OPN, C 言語) 527  
 オープン要求 514, 530  
 オブジェクト同士の位置関係 32  
 オブジェクトの遷移条件 135, 171  
 オプション設定要求 515, 532  
 オペレータインジケータ 41, 150, 762  
 オペレータインジケータを付ける 68

## か

---

カーソル 133, 170  
カーソル・フォーカス制御 762  
カーソル位置の通知 500  
カーソル移動キー 183, 187  
カーソル制御 762  
カーソル制御項目とフィールドボックスの関連づけ 138  
カーソル定数 762  
カーソルの遷移 163  
下位項目 77, 163, 366, 762  
下位項目の論理マップ生成規則とマッピング規則 268  
下位項目を指定したときの扱い 236  
解像度 43  
隠しフィールド 68, 159, 487, 762  
隠しフィールドの論理マップ生成規則とマッピング規則 266  
各制御キーの機能 187  
確定キー属性の表示属性の標準値 738  
確定キーの制御情報 267  
カスケードメニュー 73  
仮想画面 39, 762  
仮想端末の自動割当て 502, 524  
仮想端末名 28  
仮想端末名格納エリア 504  
仮想端末名ファイル 763  
カナ 178  
壁紙 52, 763  
壁紙を設定する 63  
可変項目の論理マップ生成規則とマッピング規則 252  
可変コンボボックス 89  
可変チェックボタン 106, 443  
可変ポップアップ 92  
可変ラジオボタン 105, 437  
画面 498  
画面確定キー 181, 202  
画面確定時の動き 76  
画面確定のタイミング 24, 498  
画面機能の概要 1  
画面サイズ 9, 43, 151  
画面設計の流れ 4

画面操作 12  
画面属性 763  
画面属性の詳細 60, 155  
画面属性の動的変更 69, 160  
画面単位 20, 21, 610  
画面定義とコーディングの関連 339  
画面定義の制限値 748  
画面に対するコメントの記述 63, 157  
画面の設計のポイント 53  
画面の定義サンプルと AP パターンの組み合わせ (COBOL) 559  
画面の定義サンプルと AP パターンの組み合わせ (C 言語) 562  
画面の表示フォント 51, 151  
画面の表示様式 46  
画面バッファ 12  
画面表示 12  
画面表示と帳票印刷 600, 602  
画面表示命令 (COBOL) 503  
漢字 179  
漢字コード 179

## き

---

キーエントリ 763  
キーで選択するボタン 128  
キーでのデータ入力と編集 203  
キー入力コンボボックス 89, 206  
キー入力コンボボックス (可変) 430  
キー入力コンボボックス (固定) 434  
キーの機能 187  
キーボードのカスタマイズ 185  
キーボードのロック状態を解除する 63, 156  
記号 179  
基準文字サイズ 31, 69, 763  
機能の割り当て 185  
キャラクタコントロールの表示属性の標準値 733  
旧バージョン提供ファイルのコンパイル時のトラブル 689  
行単位 629  
行単位のデータ入力 581  
共通インタフェース領域の形式 (XMAP-COM, COBOL) 511

共通インタフェース領域の形式  
(XMAP\_COM, C 言語) 526  
金額入力 205

## く

---

空白 (全角, 半角) 179  
空白 (全角だけ) 179  
空白 (半角) 179  
空白入力 76, 763  
クエスチョン 124  
矩形 123, 170  
グラフィック 763  
グラフィックコントロール 763  
グラフィックファイルのフォーマット 120  
グラフィック利用画面 8  
クリック 209  
グリッド 31, 151, 764  
グループ間のフォーカス順設定 135  
グループコントロール 764  
グループボックス 107, 764  
クローズ要求 516, 531

## け

---

掲示板 52  
掲示板の表示 597  
けい線 122, 169  
けい線の線種 122  
桁 764  
桁寄せ 764  
桁寄せ向き 350, 358

## こ

---

後退キー 189, 764  
後退キー (バックスペースキー) 183  
候補選択コントロール 764  
候補選択コントロールの表示属性の標準値  
736  
項目 764  
コードエラーの通知 77, 163  
コードエラーを検出したときのマッピング規  
則 264  
固定グラフィック 117

固定コンボボックス 89  
固定チェックボタン 106, 446  
固定テキスト・フィールド 765  
固定テキストボックス 83  
固定フィールド 111, 164  
固定ポップアップ 92  
固定ラジオボタン 105, 440  
コピー&ペースト 214  
コピー&ペースト操作 214  
コピー・ペースト操作キー 200  
コピー操作キー 184  
コピーの対象 214  
コマンドコントロール 765  
コマンドコントロールの表示属性の標準値  
737  
コントロール+復改 193  
コントロールメニュー 40, 150, 765  
コンパイラ 746  
コンパイル時の主なエラー 687  
コンパイル時のポイント 518  
コンボボックス 87, 765  
コンボボックスのフォーカス移動 210  
コンボボックスの論理マップ生成規則とマッ  
ピング規則 296  
コンマ 178

## さ

---

サービス名ファイル 765  
再定義名 68, 159, 765  
削除キー 184, 197, 766

## し

---

次画面 66, 140, 157, 172  
次画面ダイアログ 344  
時刻 80, 85  
時刻 (入出力, 出力) 766  
資産の登録 520  
実行時の主なエラー 688  
実行時のポイント 519, 685  
実線 122, 123  
始点キー 183, 190  
自動 15

自動スキップ 76, 135, 163, 171  
 自動スキップ (選択したら次のボックス・ボタン・項目にフォーカスを移す) 135  
 自動的にポップアップメニューを表示する 384, 406  
 シフト+復改 192  
 修飾名 18, 69, 766  
 自由な位置 61  
 出力カーソル制御 247  
 出力グラフィック 118, 477  
 出力グラフィックの論理マップ生成規則とマッピング規則 318  
 出力時刻テキスト 352  
 出力データ 112  
 出力テキスト 346  
 出力テキスト・フィールド 766  
 出力テキストボックス 84  
 出力日付テキスト 352  
 出力フィールド 111, 165  
 出力フィールドの再表示 589  
 出力要求 515, 531  
 出力論理マップ 17  
 手動スキップ 76, 135, 163, 171  
 手動スキップ・全桁必須 76, 135, 163, 171  
 手動スキップ・右寄せ 76, 135, 163, 171  
 手動でポップアップメニューを表示する 387, 409  
 手動表示 387  
 手動ポップアップ 389  
 手動ポップアップの設定 411  
 種別 62  
 詳細コード 753  
 詳細目的 76, 364  
 詳細目的とデータ型 366  
 使用目的 75, 163  
 使用目的 / 詳細目的とデータ型 233  
 ショートカットキー 766  
 初期カーソル 171  
 初期クリア文字 67, 159, 766  
 初期値 362, 766  
 初期フォーカス・カーソル 134  
 初期フォーカスを本ボックスに設定する 110

## す

---

数字 178  
 数字から数値に変換 546  
 数字編集項目 348, 767  
 数値から数字に変換 546  
 数値入力 204  
 ズームイン, ズームアウト 52  
 スクリーン消去キー 181  
 スクロールキー 183, 194  
 スクロール禁止領域 109  
 スクロールバーの付加 109  
 スクロール量 211  
 スクロール領域 767  
 スピンボックス 100, 767  
 スピンボックスの論理マップ生成規則とマッピング規則 324  
 スプラッシュ画面 6, 51, 566, 767  
 スペースキー 182, 185, 200  
 スペースコード 180  
 スペースバー 200

## せ

---

制御キー 180  
 制御項目データ名 769  
 制御項目の論理マップ生成規則とマッピング規則 259  
 セットアップ情報の反映 767  
 セバレータ 121, 767  
 セバレータの位置 122  
 全角文字 179  
 全画面表示 46  
 選択エントリ 767  
 選択キー 185  
 選択済みの状態 19  
 選択ラベルと通知コードを別々にする 394, 415  
 先頭位置以外からのフォーカス順序の設定 137  
 全面書換 15

## そ

---

送信 (実行) キー 181, 202

挿入キー 184, 195, 767  
 増分値 100  
 属性情報一覧 696

## た

---

ダイアログの表示 594  
 待機時間 66  
 タイトルバー 41, 71, 150, 161  
 タイトルバーを付ける 68  
 タブキー 183, 191  
 単一選択リストボックス 102

## ち

---

チェックボタン 768  
 チェックボタンボックス 105  
 チェックボタンボックスの論理マップ生成規則とマッピング規則 289  
 中央 61  
 チューニングとトラブルの対処方法 681  
 長音 178

## つ

---

通信記述項 (CD 項) 768  
 通知コード (データ入力用) 768

## て

---

定義サンプル 558  
 定義時の性能を向上させるためのポイント 682  
 定義パターン 58, 153, 768  
 定義パターン, 部品とは 58  
 定義パターンの登録 58, 154  
 提供している定義パターン 739  
 定数部 225  
 定数部の論理マップ生成規則 331  
 データ有無コード 18, 361, 768  
 データ有無コード格納エリア 505  
 データ型 768  
 データキー 184, 199, 769  
 データキーを押したときのマッピング規則 263

データ消去通知文字 67, 159, 769  
 データ消去の通知 77, 164, 499  
 データ長 769  
 データ入力キー 178  
 データ未入力の通知 77, 164, 499  
 データ名 769  
 データ名とコメントの変更 145  
 データ名に関するコメントの編集 175  
 データ名の検索と置換 145  
 データ名の順序をソートする 144  
 データ名の表示属性の標準値 738  
 データ名の編集 143  
 データ名編集 143  
 テスト表示 147, 176  
 手前に表示 48, 61  
 デリミタ 82  
 点線 122, 123  
 転送状態 19  
 転送モード 19

## と

---

動的変更 18  
 動的変更テーブル 336, 502  
 動的変更テーブル (X3MODTBL) 519  
 動的変更テーブル (X3MODTBL.H) 524  
 登録集原文 519  
 トグル種別 106  
 トグルフィールド 106, 206, 418, 769  
 トグルフィールドの論理マップ生成規則とマッピング規則 322  
 凸型 122  
 ドライバ依存 34  
 取消キー 185, 203  
 ドロー 769  
 ドローセットアップ 769

## に

---

二次ウィンドウ 49, 71, 74, 482, 770  
 二次ウィンドウの位置 63  
 二次ウィンドウの表示に関するマッピング規則 314  
 二重線 123

日本語入力に関するセットアップ 770  
 入出力時刻テキスト 368  
 入出力テキスト 355  
 入出力テキスト・フィールド 770  
 入出力テキストボックス 75  
 入出力日付テキスト 368  
 入出力フィールド 112, 162  
 入力・選択状態の扱い 61  
 入力カーソル制御 249  
 入力数字編集項目の論理マップ生成規則とマッピング規則 258  
 入力済み 19  
 入力操作がなかったときのマッピング規則 264  
 入力単位 66, 770  
 入力データチェックの通知 77  
 入力データのチェック 163  
 入力できる半角文字 178  
 入力バイト数格納項目 78, 164  
 入力バイト数格納項目の論理マップ生成規則とマッピング規則 260  
 入力フィールドの扱い 156  
 入力フィールドのクリア 584  
 入力要求 516, 532  
 入力論理マップ 17

## ぬ

---

塗りつぶし 125  
 塗りつぶしフィールド 770

## は

---

ハードウェア依存 34  
 背景色 62, 151  
 パスワード 375  
 破線 122, 123  
 パターン 58, 153  
 バックタブキー 183, 192  
 半角 178  
 反復定義 770  
 汎用 AP インタフェース 536

## ひ

---

左上 / 右下 61  
 左寄せ配置の指定 548  
 ビッグエンディアン 771  
 日付 80, 85  
 日付 (入出力, 出力) 771  
 日付・時刻項目の論理マップ生成規則とマッピング規則 319  
 日付テキスト・フィールドと時刻テキスト・フィールド (出力) 85  
 日付テキスト・フィールドと時刻テキスト・フィールド (入出力) 80  
 日付入力 205  
 非転送状態 19  
 描画方式 12  
 表形式の一覧表示 587  
 表けい線の繰り返し 124, 170  
 表けい線の枠 124, 170  
 表示・印刷環境ファイル 771  
 表示・印刷セットアップ 28, 771  
 表示形態 15, 60, 155  
 表示仕様 71, 161  
 表示直後 (強制送信) 24  
 表示に関する対処方法 690  
 標準のフォーカス順序 137  
 標準パターン 558  
 標準パターンを使用した AP の作成 557  
 標準パターンを使用した機能別の作成例 (COBOL) 564  
 標準表示 48, 61

## ふ

---

ファイルの取り込み 545  
 ファンクションキー 481  
 フィールド 498  
 フィールドキー 184, 197, 771  
 フィールドキーを押したときのマッピング規則 262  
 フィールド単位 20, 21, 636  
 フィールドボックス 108, 380, 771  
 フィールドボックス内のオブジェクト 111  
 フィールドボックスの属性 109

フォーカス 133, 771  
 フォーカス・カーソル位置のチェック 499  
 フォーカス・カーソル位置の論理マップ生成規則とマッピング規則 307  
 フォーカス・カーソル移動キー 183, 187  
 フォーカス・カーソルの移動順序 134  
 フォーカス位置の設定 137  
 フォーカス位置の通知 (GUI 画面の場合) 500  
 フォーカス順序 107  
 フォーカス順序設定の解除 138  
 フォーカス順序の対象となる条件 136  
 フォーカス制御 771  
 フォーカス設定の動的変更 138  
 フォーカス対象外 135, 461  
 フォーカス定数 771  
 フォーカス定数の論理マップ生成規則 334  
 フォーカスとカーソル 133  
 フォーカスの位置づけ 486  
 フォーカスの遷移 76  
 不活性 772  
 複数ウィンドウ表示後に一次ウィンドウ表示 50  
 複数ウィンドウ表示後に二次ウィンドウ表示 51  
 複数選択リストボックス 102  
 復改キー 183, 191  
 プッシュボタン 125, 459, 772  
 プッシュボタンの表示の動的変更 467, 471  
 プッシュボタンのラベルテキストを変更する 472  
 プッシュボタンボックス (コマンドボタン) 125  
 プッシュボタンボックスの論理マップ生成規則とマッピング規則 282  
 プッシュボタンを不活性状態で表示する 468  
 物理画面 39, 772  
 物理画面, 仮想画面, ウィンドウおよびプレーンの関係 39  
 物理マップ 13, 772  
 物理マップだけ 13, 14  
 物理マップ名称格納エリア 504  
 部品 58, 153

部品の登録 58, 154  
 部品の利用 142, 174  
 ブルダウンメニュー 73  
 フレーム 420, 772  
 フレーム (横 / 縦繰り返し) 115, 167  
 フレームに配置できるオブジェクト 115, 167  
 フレームの操作 115, 167  
 フレームを使用したときの扱い 237  
 プレーン 772  
 プレーン領域 42  
 プログラムサンプルのコンパイルと実行手順 (COBOL) 606  
 プログラムサンプルのコンパイルと実行手順 (C 言語) 608  
 プログラムサンプルのファイル名一覧 (COBOL) 605  
 プログラムサンプルのファイル名一覧 (C 言語) 607  
 プログラムテンプレート 561

---

## へ

ペースト操作キー 184  
 ペーストの対象 216  
 ヘルプ画面の表示 592  
 ベンダ定義文字 179

---

## ほ

ボックス間のフォーカス移動 209  
 ボックス枠の付加 109  
 ポップアップ 91, 772  
 ポップアップテキスト 91  
 ポップアップフィールド 114  
 ポップアップメニューの論理マップ生成規則とマッピング規則 270  
 ポップアップメニューファイル 772

---

## ま

マージ 13, 14  
 マイナス 178  
 マイナスタブキー 183, 194  
 マウス 208

マウス操作 208  
 マウスとキー操作 177  
 ますの高さ 42  
 ます目とオブジェクト 32  
 マッピングインタフェース領域の形式  
 (XMAP-MDO, COBOL) 512  
 マッピングインタフェース領域の形式  
 (XMAP\_MDO, C 言語) 528  
 マッピングオプション  
 13, 231, 505, 507, 773  
 マッピングオプションと表示形態の組み合わ  
 せ 16  
 マッピング方式 13  
 マップ生成 773  
 マップ定義ファイル 773  
 マップ展開方式 773  
 マップ名の決定 57, 153

## み

---

右上 / 右下 61  
 右寄せ配置の指定 548  
 未選択 19  
 未入力 19

## む

---

無通知コード割り当てキー 181

## め

---

命令 492  
 メッセージアイコン 124, 773  
 メニュー画面からの次画面表示 569  
 メニュー画面の表示 564  
 メニュー形式 773  
 メニュー選択コンボボックス 87  
 メニュー選択コンボボックス (可変) 424  
 メニュー選択コンボボックス (固定) 427  
 メニューバー 41, 72, 73, 774  
 メニューバーの論理マップ生成規則 281  
 メニューバーを使用した処理の選択 474  
 メニューバーを付ける 68

## も

---

文字色 372  
 文字入力キー 182  
 文字編集 DLL 545  
 文字編集キー 184, 195  
 モジュラスアルゴリズム 63, 157  
 モジュラスチェック 774

## ゆ

---

有効となるキー 178

## よ

---

要求インタフェース領域の形式 (XMAP-  
 REQ, COBOL) 511  
 要求インタフェース領域の形式  
 (XMAP\_REQ, C 言語) 527  
 用語と ANK コードの関係 178  
 予約項目名 140, 172  
 予約テキスト・フィールド 139, 337  
 予約フィールド 171  
 予約フィールド・テキスト 774

## ら

---

ラジオボタン 774  
 ラジオボタンボックス 104  
 ラジオボタンボックスの論理マップ生成規則  
 とマッピング規則 285  
 ラベル 775  
 ラベルテキストを動的変更するときの注意  
 128

## り

---

リストボックス 102, 452, 775  
 リストボックス内のフォーカス移動 210  
 リストボックスの論理マップ生成規則とマッ  
 ping規則 292  
 リターンコード 753  
 リターンコード格納エリア 505  
 リターン情報 528  
 リトルエンディアン 775  
 リンク (MAKE) 時の主なエラー 688



リンケージ時のポイント 519

## れ

---

レイアウトパターン 558  
レイアウト領域 41, 150, 775  
レイアウト領域のグリッド 42  
レイアウト領域の設定に関する用語 31  
レイアウト領域の背景色 49  
例題プログラム 605

## ろ

---

論理端末定義 12  
論理データ 17  
論理マップ 13, 17, 775  
論理マップ可変部 224  
論理マップ固定部 224  
論理マップ生成規則で使う用語 225  
論理マップ生成規則とマッピング規則 223  
論理マップだけ 13, 14  
論理マップの構成 224  
論理マップの再定義 543, 544  
論理マップの取り込み方法 (COBOL) 502  
論理マップの取り込み方法 (C 言語) 525

## わ

---

ワーニング 124  
枠高 75  
割込キー 181, 775  
和暦 86  
ワンタッチクリア 206



# 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。

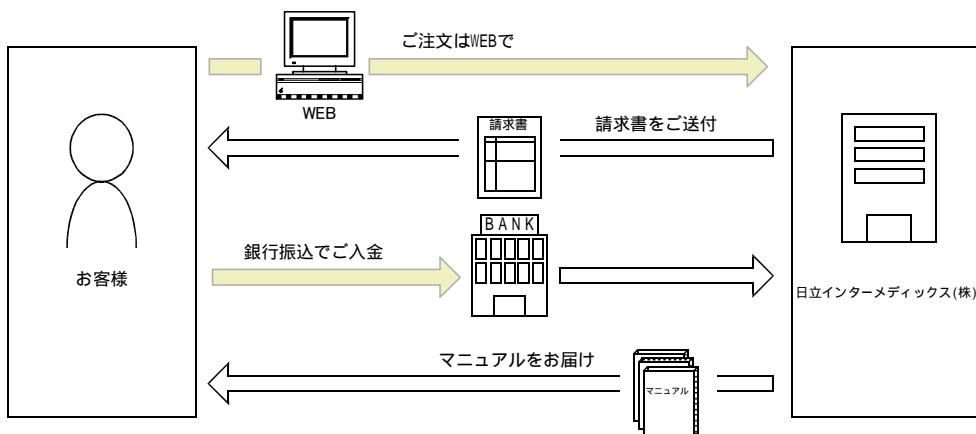
マニュアル一覧	日立コンピュータ製品マニュアルを製品カテゴリ、マニュアル名称、資料番号のいずれかから検索できます。
CD-ROMマニュアル情報	複数マニュアルを格納したCD-ROMマニュアルを提供しています。どの製品に対応したCD-ROMマニュアルがあるか、を参照できます。
マニュアルのご購入	日立インターメディックス(株)の「日立コンピュータ製品マニュアルサイト」からお申し込みできます。 (詳細は「3. マニュアルのご注文」を参照してください。)
Web提供マニュアル一覧	インターネットで参照できるマニュアルの一覧を提供しています。 (詳細は「2. インターネットからのマニュアル参照」を参照してください。)
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](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。